Segmentación y detección de objetos en imágenes y vídeo mediante inteligencia computacional

UNIVERSIDAD DE MÁLAGA

TESIS DOCTORAL

Miguel Ángel Molina Cabello

Departamento de Lenguajes y Ciencias de la Computación
Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga

Junio 2018

UNIVERSIDAD
DE MÁLAGA

AUTOR: Miguel Ángel Molina Cabello

http://orcid.org/0000-0002-8929-6017

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga

Segmentación y detección de objetos en imágenes y vídeo mediante inteligencia computacional

Memoria que presenta para optar al título de Doctor por la Universidad de Málaga

Miguel Ángel Molina Cabello

Dirigida por los Doctores

Ezequiel López Rubio y Rafael Marcos Luque Baena

Departamento de Lenguajes y Ciencias de la Computación
Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga

Junio 2018

UNIVERSIDAD DE MÁLAGA

Departamento de Lenguajes y Ciencias de la Computación
Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga

El Dr. D. Ezequiel López Rubio, Catedrático de Universidad perteneciente al área de Ciencia de la Computación e Inteligencia Artificial de la E.T.S. de Ingeniería Informática de la Universidad de Málaga, y el Dr. D. Rafael Marcos Luque Baena, Profesor Contratado Doctor perteneciente al área de Lenguajes y Sistemas Informáticos de la E.T.S. de Ingeniería Informática de la Universidad de Málaga,

Certifican que,

D. Miguel Ángel Molina Cabello, Ingeniero en Informática, ha realizado en el Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga, bajo su dirección, el trabajo de investigación correspondiente a su Tesis Doctoral titulada:

Segmentación y detección de objetos en imágenes y vídeo mediante inteligencia computacional

Revisado el presente trabajo, estimamos que puede ser presentado al tribunal que ha de juzgarlo. Y para que conste a efectos de lo establecido en la legislación vigente, autorizamos la presentación de este trabajo en la Universidad de Málaga.

Málaga, Julio de 2018

Fdo.: Dr. Ezequiel López Rubio Fdo.: Dr. Rafael Marcos Luque Baena

A Nuria,
por todo su apoyo durante este camino

Agradecimientos

Si he visto más lejos es porque estoy
sentado sobre los hombros de gigantes.

Isaac Newton

Estos agradecimientos deben empezar por Ezequiel y Rafa, las personas que me han guiado y ayudado en todo momento con su labor y consejos. Cualquier cosa positiva que diga de Ezequiel va a quedarse corta, tanto en el aspecto profesional como en el personal. Recuerdo que al poco de empezar esta andadura, Javi (su hermano) me comentó que "todo el mundo debería hacer un doctorado con Ezequiel". Tras completar este camino, me uno al sentimiento que encierra esta frase. De Rafa poco más puedo añadir; resaltarle mi gratitud por la propuesta que me hizo en su día para comenzar este proyecto cuando aún estaba inmerso en la empresa privada, hecho que añadió una nueva etapa común a las ya anteriormente disfrutadas en nuestras vidas. Como prueba de su compromiso hacia el trabajo desarrollado basta con citar las tutorías de los fines de semana en su domicilio particular o las tutorías por teléfono durante los viajes a/desde Extremadura.

También tengo que agradecer a Enrique su trabajo y enseñanzas desde el primer momento, así como la ayuda prestada por todos los compañeros de laboratorio y personas con las que he colaborado: Javi, Juanmi, Esteban, Karl, Jesús, Jorge, Julio, Héctor, Subi, Paco, Fran, Nico, Manuel Jesús y María Jesús.

En este sentido, me gustaría hacer una mención especial a Pepe. Él fue el que me hizo descubrir el mundo del procesamiento de imágenes y la visión por computador durante mi etapa estudiantil. Estoy seguro de que esta tesis ha sido posible por el amor e interés con el que Pepe transmitía esos conocimientos. El nacimiento y crecimiento de este grupo de investigación no habría sido posible sin él.

El tiempo que he estado en Leicester no hubiera sido lo mismo sin David. Desde el primer momento se interesó por cualquier aspecto que incidiera en mi estancia en su grupo de investigación, tanto en el ámbito personal como en el académico. También agradecer a Lipika su atención y a Jesús por hacer más llevaderas aquellas fechas, así como a Phuong y a Luong por facilitarme

la integración en el laboratorio.

Quiero mostrar mi gratitud a la Universidad de Málaga, a la Junta de Andalucía y al Gobierno de España por darme la oportunidad de desempeñar aquello que hacía muchos años era un sueño y hoy día es una realidad, disfrutando de mi trabajo y haciendo lo que más me gusta. De igual forma, agradecer a la Universidad de De Montfort en Leicester que me haya acogido como uno más. A nVidia quiero agradecerle la donación de varios recursos con los que he podido desarrollar más eficientemente mi labor.

Si hoy día he llegado a este punto ha sido gracias al esfuerzo, sacrificio y educación de mis padres. Ellos me han inculcado todos los valores necesarios para poder afrontar cualquier reto, siendo un constante ejemplo de referencia. También quiero dar las gracias a mi hermano por su apoyo emocional; la atención de una persona familiar con estudios afines propicia una mayor empatía. Y al resto de mi familia y amigos, por darme aliento durante este tiempo.

Para finalizar, quiero agradecer a Nuria todo su apoyo. Si hay una persona que ha sufrido esta tesis tanto o más que el que escribe estas líneas es ella. Por acompañarme en todo momento en este viaje, por disfrutar de las alegrías que ha conllevado su realización, por aumentar mi estima en los malos momentos, por haberme prestado parte de tu tiempo para culminar este proyecto... por todo ello, gracias.

Gracias a todos.

Resumen (Abstract)

Una imagen vale más que mil palabras.
(A picture is worth a thousand words.)
Anónimo (Anonymous)

0.1. Resumen

La presente tesis trata sobre el procesamiento y análisis de imágenes y video mediante sistemas informáticos.

Primeramente se hace una introducción, donde se especifica el contexto, los objetivos y la metodología utilizados. Tras esto se muestran los antecedentes, señalando los fundamentos de la videovigilancia, área sobre la que se van a aplicar los modelos desarrollados, las dificultades existentes para modelar el fondo de las escenas de los videos y diversos algoritmos con la finalidad de detectar los objetos que aparecen en primer plano, seguido de las principales características del aprendizaje profundo, transporte inteligente y sistemas con cámara PTZ, exponiendo finalmente como se evalúan los métodos y distintos conjuntos de datos que se utilizan para tal fin.

A continuación se muestran tres partes, cada una con diferentes apartados. La primera parte se basa en los estudios desarrollados que tratan sobre la fase de segmentación. En esta parte se explican diferentes modelos desarrollados cuyo objetivo es la detección de objetos en primer plano mediante el uso de hardware genérico o hardware especifico, como lo es el de bajo coste, la segmentación de objetos en un ámbito especifico, como lo es la detección de glóbulos rojos en muestras de sangre, o el estudio de cómo influye la reducción del tamaño de las imágenes al rendimiento de los algoritmos basados a nivel de pixel.

La segunda parte de la tesis describe los trabajos en los que se hace uso de una cámara PTZ. Así, en esta parte se presentan dos trabajos con esta particularidad. El primero de ellos hace un seguimiento del objeto más anómalo del escenario, siendo el propio sistema el que decide cuáles son los objetos anómalos y cuáles no, decisión que se basa en un modelo entrenado previamente con un video en el que no aparecen objetos anómalos. Por su parte, el segundo trabajo muestra un sistema que indica a la cámara los

movimientos a realizar en función de la salida producida por un modelo de fondo no paramétrico y mejorada con un gas neuronal creciente.

La tercera parte es relativa a los estudios desarrollados que tienen relación con el transporte inteligente, como es la clasificación de los vehículos que aparecen en secuencias de tráfico. En uno de estos trabajos se aplican técnicas tradicionales como la segmentación y posterior extracción de rasgos o características; el segundo hace uso de una fase de segmentación y de varias redes convolucionales en lugar de la extracción de características, complementado con un estudio del redimensionado de las imágenes para proveerlas en el formato necesario a cada red; y el tercero utiliza un modelo de red que detecta y clasifica objetos, tras lo que se realiza una estimación de la contaminación generada por lo vehículos.

Por último, se exponen las conclusiones generales que se han obtenido tras la realización de la presente tesis y se comentan unas posibles líneas futuras de investigación que la amplíen y complementen.

0.2. Abstract

This thesis is about image and video processing and analysis by computer systems. First of all, an introduction is carried out, where the context, the goals and the employed methodology are specified. After that, the previous works are shown, by emphasizing the video surveillance fundamentals, which is the topic where the developed models are going to be applied. Then, it is exhibited the difficulties to model the background of the scenes of the videos and several algorithms with the purpose to detect the foreground objects. In addition, the main features of the deep learning technique, intelligent transport and PTZ camera systems are described. The evaluation of the methods and different datasets which are used for that objective are also detailed.

Then, three parts follow, each one with different sections. The first part is based on the developed research works which are about the segmentation step. In this part different developed models whose aim is the detection of foreground objects by employing generic or specific hardware, like the low cost one, the object segmentation in an specific field, like the detection of red blood cells in blood images, or the study about how the image downsampling affects to the performance of the pixel-level algorithms are explained.

The second part of the thesis describes the works where a PTZ camera is used. In this part, two works with this particularity are presented. The first one carries out the tracking of the most anomalous object in the scenario, where the system rules which ones are anomalous and which ones are non-anomalous. The decision is based on a previous trained model with a video where anomalous objects do not appear. On the other side, the second work exhibits a system that indicates the movements to be done for the camera

according to the produced output by a non-parametric model, enhanced with a growing neural gas.

The third part is about the research related to intelligent transport, like the classification of the vehicles which appear in traffic sequences. One of these works applies traditional techniques like the segmentation and the subsequent features extraction. The second one employs a segmentation step and several convolutional neural networks instead of the feature extraction, complemented by a study of the image resize in order to provide to each net with the required format. And the third one uses a model that detects and classifies objects, and then an estimation of the pollution generated by the vehicles is done.

Finally, the general conclusions obtained after the fulfilment of the presented thesis are described and possible future research lines that extend and complement it are sketched.

Índice

Índice de figuras

Índice de Tablas

Capítulo 1

Introducción

Soy de los que piensan que la ciencia tiene una gran belleza. Un científico en su laboratorio no es sólo un técnico: también es un niño colocado ante fenómenos naturales que lo impresionan como un cuento de hadas.

Marie Curie

Resumen: Este capítulo presenta una breve introducción a esta tesis, exponiendo el contexto y la motivación en los que se desarrolla, así como los objetivos fijados y la metodología empleada. Por último, se ofrece la estructura utilizada esta tesis.

1.1. Contexto

El desarrollo de las tecnologías (tanto de dispositivos como de herramientas) y la fácil accesibilidad por parte de la sociedad a dicha tecnología ha propiciado la existencia de multitud de datos. En este sentido, el uso de las cámaras de vídeo ha ido creciendo en los últimos años, tanto para uso particular como profesional. Por otro lado, cada vez es mayor el número de videos e imágenes que circulan por internet, existiendo un gran número de plataformas en la red que almacenan numerosos datos de este tipo. Por ejemplo, Youtube es actualmente la plataforma de video más importante; por su parte, Facebook, que es la red social más popular, es una plataforma en la que primordialmente predominan las imágenes.

Actualmente, la mayor parte de las cámaras que se utilizan para este tipo de tareas son cámaras 2D, que recogen la secuencia de vídeo y posteriormente se procesa imagen por imagen para detectar objetos y reconocer

Figura 1.1: Ejemplo de dispositivos con los que captar imágenes y videos. De izquierda a derecha pueden observarse una cámara digital, un teléfono móvil, unas cámaras de vigilancia y unas cámaras PTZ, respectivamente.

el movimiento de los mismos para poder determinar qué actividad están realizando. Las cámaras que se utilizan pueden ser aparatos destinados únicamente a tal fin, como por ejemplo las cámaras digitales que se usan a nivel personal o las cámaras de videovigilancia ubicadas en distintos puntos de la red de carreteras; o bien cámaras integradas en otros dispositivos con diversas funcionalidades, como es el caso de los teléfonos móviles. También existe una alternativa mucho más novedosa como puede ser el uso de las denominadas cámaras *PTZ* (Pan-Tilt-Zoom), que son aquellas cuyo objetivo puede realizar movimientos horizontales (pan) y verticales (tilt), y cambiar la distancia focal (zoom), con las que se puede hacer un seguimiento de los objetos presentes en el escenario. Otro tipo de cámaras son las cámaras en 3D, que constan de dos objetivos, a modo de ojo derecho e izquierdo. Cada objetivo recoge su secuencia de vídeo de manera independiente, de forma que este par de secuencias de imágenes se puede utilizar para procesarlas al igual que las cámaras 2D.

En la figura 1.1 se muestran algunas cámaras y dispositivos que permiten la captura de imágenes y la grabación de videos, tales como una cámara digital, un teléfono móvil, dos cámaras de videovigilancia estática y dos cámaras PTZ.

Indistintamente del tipo de cámara utilizada, hay que tener en cuenta que se genera un gran volumen de datos, ya sea por el alto número de fotogramas por segundo, la resolución de las imágenes, o incluso el uso de múltiples canales, como es el caso de las cámaras multiespectrales.

Debido al volumen de la cantidad de datos de imágenes y video que se manejan, conocido como *datos visuales masivos* (*visual big data*) se hace necesario realizar un procesamiento de imágenes ya que revisar dicha información de forma exhaustiva y a cargo de una persona es inviable debido a este gran volumen de información. Es por ello por lo que sería interesante poder procesarlo automáticamente porque sería deseable contar con aplicaciones rápidas, donde se necesita una respuesta instantánea debido a que los usuarios son interactivos, realizándose los servicios en tiempo real, y manejándose numerosas imágenes de gran tamaño.

Con estas aplicaciones se podrían reconocer las actividades que forman

parte de la escena que capta la imagen y así realizar otras labores a partir de dicho procesamiento, máxime si dicho procesamiento se desarrolla en tiempo real. Centrándonos en la videovigilancia, sería deseable que los ordenadores pudieran reconocer personas, hacerles un seguimiento, comprender su comportamiento, detectar movimientos sospechosos o situaciones peligrosas, objetos perdidos, generar señales de aviso o alarma...

Además, sería interesante dar respuesta a lo que se conoce internacionalmente como *consultas basadas en el contenido* (*Content based retrieval*), es decir, poder contar con un buscador inteligente que permitiera la recuperación en grandes bases de datos de imágenes y vídeos basada en su contenido. Con este buscador se podrían satisfacer búsquedas como, por ejemplo: "encontrar todos los videos en los que aparezca un gato".

En el caso de la videovigilancia, existen multitud de propuestas que solucionan parcialmente partes del problema, pero aún no se le ha logrado dar respuesta de una manera global, además de que surgen nuevos y más complejos desafíos.

1.2. Objetivos

El principal objetivo que se ha planteado en esta tesis es el de desarrollar nuevos modelos y algoritmos que permitan detectar con mayor calidad objetos de primer plano en las secuencias de imágenes y video para mejorar los sistemas de videovigilancia. Para ello es imprescindible estudiar y analizar el modelado del fondo de la escena y la posterior segmentación de los objetos en primer plano que aparezcan en dicha escena. La segmentación es la primera etapa de la teledetección y de la calidad del resultado producido dependerán las siguientes etapas. Sin embargo, los escenarios y condiciones en los que se puede implantar un sistema de videovigilancia son muy diversos y pueden llegar a presentar condiciones especiales. Es por ello por lo que se pretende diseñar métodos que se adapte a las diferentes necesidades que pueden surgir, desarrollando nuevos algoritmos de detección de objetos alternativos a los ya existentes, tanto algoritmos de tipo genérico como específicos atendiendo a requisitos de hardware o escenarios concretos.

Dentro de los sistemas de videovigilancia, la mayor parte de ellos son tradicionales y cuentan con una cámara estacionaria, que se encuentra fija en el escenario en el que se desarrolla la acción. Sin embargo, los sistemas de videovigilancia que están formados por cámaras PTZ también suponen un número representativo del total. Así, se tratarán especialmente este tipo de sistemas. Este hecho viene además motivado por la dificultad extra que entrañan, ya que el objetivo no se centra únicamente en la detección de los objetos de primer plano, sino también en generar automáticamente y proporcionar a la cámara aquellos comandos necesarios para que pueda realizar un movimiento de forma que pueda practicar la cobertura del escenario más

idónea posible.

Por otro lado, se desea aplicar estas técnicas de segmentación a campos concretos. En este caso se ha optado por tratar cuestiones relativas al transporte inteligente, de manera que se pueda avanzar en sus prestaciones con el uso de sistemas de videovigilancia. El objetivo que se ha planteado es el de detectar y clasificar los vehículos que aparecen en una carretera para, posteriormente, poder llevar a cabo las acciones que se estimen oportunas.

Para cumplir estos objetivos generales, se han abordado otros objetivos más específicos como los siguientes:

- Desarrollar métodos de detección de objetos que puedan ejecutarse en dispositivos hardware de bajo coste.
- Desarrollar métodos de detección de objetos de primer plano con la singularidad de que están diseñados para situaciones donde el modelado de fondo sea complejo, como aquellos fondos que son dinámicos.
- Estudiar la aplicación de técnicas para poder reutilizar algoritmos que no ofrecen un rendimiento adecuado con las actuales condiciones que ofrecen las cámaras. Por ejemplo, el mayor tamaño de las imágenes influye negativamente en el tiempo de cómputo de su análisis.
- Estudiar el reconocimiento de objetos que puedan considerarse como anomalías, particularmente los que sean capturados por una cámara PTZ para poder practicarles un seguimiento.
- Generar diferentes alternativas que permitan la detección y clasificación de vehículos en videos de tráfico.
- Aplicar métodos de detección y clasificación de vehículos para estimar la velocidad de aquellos que se detecten y la contaminación que producen.

1.3. Metodología

Los algoritmos de detección de fondo y objetos, además de resolver el problema para el que están diseñados, han de cumplir varios requisitos a la hora de ofrecer su solución. La complejidad de estos métodos ha de ser baja para poder solventar los problemas anteriormente comentados en un tiempo de cómputo razonable, sobre todo si se trata de un sistema en tiempo real. Por tanto, la calidad del método estará determinada por la fiabilidad de la solución y por el tiempo que consume en calcularla.

En este sentido, se utilizan los siguientes principios metodológicos para el desarrollo de nuestra tesis:

- **Método científico**. Este método es la base de nuestra investigación. Todo proyecto que siga este procedimiento estará sujeto a los principios de reproducibilidad (poder repetir el experimento) y refutabilidad (poder demostrar que la hipótesis es falsa). Se caracteriza por seguir unas etapas bien definidas: Observación, Inducción, Hipótesis, Experimentación, Demostración o refutación de la hipótesis, y Conclusiones o tesis científica.

- **Metodología iterativa e incremental**. El desarrollo del proyecto se realiza partiendo de una base a la que se le añadirán funcionalidades cada cierto tiempo teniendo en cuenta las necesidades más prioritarias en dicho momento, sin que esta adición afecte de manera negativa al resto del proyecto ya implementado.

- **Metodología de implementación**. Se debe usar un estilo de programación adecuado para que el programa obtenido sea modificable, modular, extensible, sencillo, documentado, etc.

- **Criterios de evaluación**. Uso de medidas de rendimiento, calculadas entre la imagen con el resultado ideal del sistema de detección del fondo (también denominado *Ground Truth Mask*) y la imagen segmentada por nuestro algoritmo.

- **Comparación con otros modelos**. Estudiar las diferencias de los resultados producidos por nuestro método con respecto a otros del estado del arte basándonos en las medidas de rendimiento establecidas anteriormente.

Para la implementación de estos nuevos modelos se hace uso de diversas técnicas como pueden ser las redes autoorganizadas, los modelos estadísticos o la aproximación estocástica. Las redes neuronales, computacionalmente hablando, son un sistema de procesamiento de la información cuyo mecanismo de funcionamiento está inspirado en lo que actualmente se conoce del sistema neurológico biológico. Por su parte, los modelos estadísticos tratan de capturar las características fundamentales de las imágenes y vídeos de entrada, ignorando la información irrelevante o errónea. Con respecto a la aproximación estocástica, se trata de un conjunto de algoritmos cuyo objetivo es estimar los valores de distintos parámetros de modelos estadísticos en línea ("online"), es decir, aprendiendo de los datos conforme se van generando.

1.4. Estructura de la tesis

La estructura de esta tesis se compone de un primer bloque donde se exponen los antecedentes, otros tres bloques compuestos por seis, tres y dos

capítulos, respectivamente, donde se exponen los trabajos y estudios desarrollados y un último bloque con las conclusiones y los trabajos futuros.

El primer bloque está formado por el Capítulo 2 muestra los antecedentes, que está constituido por las consideraciones necesarias para abordar el estudio de los sistemas de videovigilancia y la detección de objetos en primer plano. En la Sección 2.1 se comentan los fundamentos de la videovigilancia, en qué consiste dicho campo y por qué se estudia. En la Sección 2.2 se indica la problemática del modelado del fondo de la escena de una secuencia de video, mientras que en la Sección 2.3 se detallan diversas propuestas de la literatura cuyo objetivo es modelar el fondo. En la Sección 2.4 se introduce el concepto de aprendizaje profundo (deep learning), que ha supuesto un punto de inflexión en el análisis de imágenes. La aplicación de sistemas de videovigilancia a un área concreta es mostrada en la Sección 2.5, en concreto al transporte inteligente. Por su parte, la Sección 2.6 se centra en los sistemas de videovigilancia que hacen uso de una cámara PTZ, analizando las dificultades añadidas que ello conlleva. En la Sección 2.7 se especifica la forma en la que se evalúan los diversos métodos y algoritmos para poder establecer una comparativa entre ellos y poder determinar el rendimiento de cada uno, ofreciéndose tanto las métricas más usuales como los conjuntos de datos que sirven como test.

El siguiente bloque está formado por los capítulos 3-8, en los que la temática de estos seis capítulos se centra en la segmentación de imágenes o secuencias de video. Este bloque es el que más se ha tratado a lo largo de la presente tesis ya que la fase de segmentación de un sistema de videovigilancia es la primera etapa y de su resultado dependen las siguientes.

En el Capítulo 3 se desarrolla un método de detección de objetos en primer plano basado en aprendizaje competitivo mediante mapas autoorganizados. En la Sección 3.1 se hace una introducción de la importancia de los métodos de detección de objetos de primer plano y un recorrido por algunas de las diferentes propuestas, especialmente aquellas que están basadas en el uso de redes neuronales. En la Sección 3.3 se especifica el modelo desarrollado, mientras que en la Sección 3.4 se presenta cómo dicho modelo puede ser aplicado al modelado del fondo de una escena y detectar objetos de primer plano. En la Sección 3.5 se muestran los diversos experimentos realizados explicando los métodos comparadores, las secuencias de test y los resultados obtenidos tanto de forma cualitativa como cuantitativa. En la Sección 3.6 se discuten diversas características de la propuesta. Por último, en la Sección 3.7 se ofrecen las conclusiones de este capítulo.

El Capítulo 4 presenta un método de detección de primer plano para videovigilancia en el que se hace uso de lógica difusa basada en la estimación de los estados de iluminación de los píxeles. En la Sección 4.1 se detalla la necesidad de los métodos de detección de objetos y se presentan algunos ejemplos. La metodología del sistema desarrollado es mostrada en la Sección

4.2, diferenciando entre el modelo de detección utilizado como referencia y el modelo de lógica difusa que mejora la salida producida por el modelo de referencia basándose en una estimación de la iluminación del fotograma. En la Sección 4.3 se exhiben los experimentos realizados, señalando los métodos competidores, las secuencias de video utilizadas y los diferentes resultados obtenidos. Finalmente, las conclusiones de este capítulo se encuentran en la Sección 4.4.

En el Capítulo 5 se ha desarrollado un sensor inteligente de detección de movimiento basado en procesamiento de video usando mapas autoorganizados. En la Sección 5.1 se hace una introducción de los métodos de detección de objetos tradicionales y la necesidad de adaptar y/o crear nuevos métodos adaptados a dispositivos hardware de bajo coste como son los microcontroladores. En la Sección 5.2 se describe brevemente el sistema microcontrolador y la propuesta desarrollada. Posteriormente, en la Sección 5.3 se introduce el modelo SOM para detectar el movimiento diseñado para adaptarse a las necesidades de computación de los microcontroladores. La Sección 5.4 se explican los detalles de la aplicación implementada. Los resultados experimentales obtenidos se muestran en la Sección 5.5, mientras que la Sección 5.6 describe las conclusiones de este capítulo.

El Capítulo 6 trata sobre arquitecturas de redes neuronales superficiales y profundas para la clasificación de células sanguíneas usando la Transformada del Círculo de Hough. En la Sección 6.1 se describe la utilidad de la detección de las células, en particular de los glóbulos rojos, en imágenes médicas, así como diversos trabajos que hacen uso de la Transformada del Círculo de Hough o de redes neuronales para el análisis de imágenes de este tipo. Tras esto, la metodología de la propuesta es expuesta en la Sección 6.2, estableciendo cómo se detectan glóbulos mediante la Transformada del Círculo de Hough y cómo se clasifican en rojos o de otro tipo gracias a una red neuronal. En la Sección 6.3 se observan los diferentes experimentos que se han llevado a cabo, detallando las imágenes de test y los resultados obtenidos. Por último, la Sección 6.4 señala las conclusiones obtenidas en este capítulo.

En el Capítulo 7 se muestra un nuevo modelo de gas neuronal autoorganizado basado en las Divergencias de Bregman (GHBNG) y su aplicación a la detección de anomalías en secuencias de video. En la Sección 7.1 se hace una introducción a los mapas autoorganizados, el gas neuronal creciente y otros modelos más evolucionados, así como la aplicación de redes neuronales a la videovigilancia. La Sección 7.2 describe el nuevo modelo GHBNG, mientras que la Sección 7.3 presenta varios experimentos que demuestran la capacidad autoorganizativa del modelo, además de su aplicación concreta a la detección de objetos anómalos en videos. Finalmente, las conclusiones se encuentran en la Sección 7.4.

El Capítulo 8 y último de este bloque, describe un estudio de la reducción

del tamaño del fotograma para la detección de primer plano en secuencias de video y la propuesta de un marco de trabajo para los métodos de detección de objetos, sobre todo los basados a nivel de píxel, mediante el uso de una compresión del fotograma. La Sección 8.1 describe la problemática existente con los métodos de detección de objetos de primer plano basados a nivel de píxel y su aplicación en videos cuyo tamaño de fotograma es relativamente grande. La Sección 8.2 expone la metodología de la propuesta, especificando los procesos de compresión y descompresión. La Sección 8.3 muestra los resultados experimentales, mientras que la Sección 8.4 presenta algunas conclusiones del trabajo.

El tercer bloque está constituido por los capítulos 9 y 10, que tratan sobre sistemas de videovigilancia que hacen uso de una cámara PTZ.

El Capítulo 9 hace referencia a un sistema de detección de objetos anómalos mediante búsqueda activa con cámaras PTZ. La Sección 9.1 describe el objetivo de la búsqueda de anomalías en los sistemas de videovigilancia y la propuesta de un nuevo sistema no parámetrico basado en la distribución de Dirichlet para determinar qué objetos de una escena son anómalos y cuáles no, mientras que para la detección de objetos se utiliza una red neuronal de propuesta de regiones. La Sección 9.2 presenta la descripción del sistema de detección de anomalías propuesto. Por su parte, la Sección 9.3 presenta los resultados experimentales llevados a cabo. Tras esto, las características más importantes de la propuesta son discutidas en la Sección 9.4. La finalización de este capítulo está formada por las conclusiones de la Sección 9.4.

El Capítulo 10 presenta un controlador neuronal para cámaras PTZ basado en detección de primer plano mediante un modelo no panorámico. En la Sección 10.1 detalla las características de un sistema de videovigilancia compuesto por una cámara PTZ, así como varias propuestas de la literatura. La arquitectura del sistema desarrollado se especifica en la Sección 10.2, detallando cómo se detecta el primer plano, cómo un modelo de red neuronal decide la región de interés y cómo se transmiten los comandos adecuados a la cámara para que cubra dicha región. La Sección 10.3 muestra los experimentos realizados y los resultados obtenidos. Y por último, en la Sección 10.4 se comentan las conclusiones de este capítulo.

El cuarto bloque consta de un total de tres capítulos, del 11 al 13, y se compone de los trabajos desarrollados que se alojan en el área del transporte inteligente.

El Capítulo 11 trata sobre la detección de tipos de vehículos mediante grupos de redes neuronales convolucionales empleando imágenes con superresolución. La Sección 11.1 muestra diversas consideraciones sobre los sistemas de videovigilancia, la importancia de la detección y seguimiento de objetos, concretamente en los videos de tráfico, y la aplicación del aprendizaje profundo al análisis de imágenes. La Sección 11.2 presenta la arquitectura del sistema de la propuesta, mientras que la Sección 11.3 establece cómo se com-

porta el marco de trabajo de la clasificación de vehículos y cómo se aplican para tal fin la superresolución y las redes neuronales convolucionales. La Sección 11.4 ofrece los resultados experimentales sobre un conjunto de videos de tráfico públicos. Por último, la Sección 11.5 finaliza el capítulo con las conclusiones obtenidas.

En el Capítulo 12 se realiza un sistema de clasificación de vehículos en entornos de tráfico mediante gas neuronal creciente. Una introducción acerca de los sistemas de videovigilancia aplicados a secuencias de tráfico y el objetivo de clasificar los vehículos en categorías atendiendo a su tamaño es comentada en la Sección 12.1. El modelo del sistema propuesto se expone en la Sección 12.2. A continuación, en la Sección 12.3 se describen los experimentos realizados y los resultados obtenidos. Finalmente, la Sección 12.4 recoge las conclusiones de este capítulo.

El Capítulo 13 trata sobre un sistema de estimación de la contaminación en carreteras mediante cámaras estáticas y redes neuronales. En la Sección 13.1 se expone una introducción sobre los sistemas de gestión de tráfico y cómo la contaminación puede afectar al tránsito de vehículos. La Sección 13.2 presenta la arquitectura de la propuesta, donde cada subsección describe sus diferentes partes al detalle: detección y seguimiento de vehículos, estimación de la velocidad y estimación de la contaminación. La Sección 13.3 muestra los resultados experimentales obtenidas, mientras que la Sección 13.4 resume las conclusiones.

Finalmente, el quinto y último bloque está compuesto por el Capítulo 14, en el que se detallan las conclusiones generales obtenidas tras la realización de la presente tesis y se describen las líneas de trabajo futuras.

Capítulo 2

Antecedentes

Necesitamos conocer el pasado para enfrentarnos al presente y prever el futuro.

Paul Johnson

Resumen: Este capítulo explica los conceptos, teoría y trabajos en los que se basa la tesis. Primero se explican los fundamentos de la videovigilancia, en concreto de la fase de segmentación y las dificultades existentes para modelar el fondo de una escena y la detectección de objetos en primer plano, así como varios ejemplos de algoritmos que cumplen este cometido. Posteriormente, se ofrece una visión general de cómo el análisis de imágenes puede interaccionar con temáticas como el aprendizaje profundo, el transporte inteligente o las cámaras PTZ. Por último, se señala cómo evaluar el rendimiento de los métodos y varios conjuntos de datos que se utilizan para tal fin.

2.1. Fundamentos de la videovigilancia

El principal problema que nos encontramos a la hora de abordar el problema de la videovigilancia automática es el procesamiento computacional de las imágenes, donde se hace necesario dotar de inteligencia a los ordenadores para que puedan analizar automáticamente las imágenes deseadas. Este procesamiento de teledetección se puede descomponer en tres fases: una primera etapa de detección de los objetos en movimiento (en primer plano), una segunda fase que controla el seguimiento de dichos objetos y una tercera y última acción que consiste en el análisis del comportamiento.

Todo este proceso automático es el mayor inconveniente que existe actualmente. En este punto, nuestro cometido se centra en la primera fase, donde

Figura 2.1: Ejemplo de segmentación producida por un algoritmo de detección de objetos y fondo. La primera imagen se corresponde con el fondo del escenario de un video, la segunda imagen es un fotograma del video, la tercera imagen es el resultado ideal de la segmentación del fotograma, mientras que la cuarta imagen es la segmentación real de dicho fotograma obtenida tras aplicar un algoritmo de detección de objetos de primer plano.

es crítico detectar el fondo y los objetos que aparecen en ellas, utilizando información temporal de la secuencia de vídeo capturada.

Existen muy diversos algoritmos para resolver el problema planteado pero no existe "el algoritmo perfecto" que funcione correctamente en todos los casos. Esto se debe a que no es trivial dar con un algoritmo que pueda gestionar las situaciones inesperadas que puedan aparecer y obtener unos resultados fiables. Esto se debe a que influyen múltiples factores adversos, como modificaciones en la iluminación (la escena en general o alguna zona en particular), aparición de sombras en primer plano como consecuencia de la presencia de objetos en segundo plano iluminados desde el fondo, presencia de ruido debido a una mala calidad de imagen o bien movimiento repetitivo de objetos fijos (ramas de árboles que se agitan por la acción del viento), etc. Todos estos aspectos, como son la eliminación de ruido, deben ser tenidos en cuenta por el método de detección de objetos y fondo de la imagen.

Dos ejemplos reales de la aplicación de algoritmos de detección del fondo y de objetos en primer plano en una imagen pueden observarse en las figuras 2.1 y 2.2.

En la figura 2.1, la primera imagen se corresponde con el fondo de la escena (paisaje con carretera); en la segunda imagen aparece un objeto (autobús) dentro de la escena; la tercera imagen se corresponde con la segmentación ideal, mientras que la cuarta imagen contiene la segmentación obtenida con un algoritmo de detección de objetos. Como se puede comprobar, existen zonas como las ventanas o las ruedas del autobús que pertenecen al objeto y que, sin embargo, el algoritmo empleado no ha reconocido como pertenecientes al mismo. Estos errores han podido ser debido a un color similar al del fondo de la escena. Por otro lado, vemos que existe cierto ruido como consecuencia de las sombras existentes en la escena junto con el cambio de posición de algunos elementos, como las ramas de los árboles, motivados supuestamente por el cambio de la posición del sol con respecto a la escena y por la acción del viento.

Figura 2.2: Ejemplo de segmentación producida por tres algoritmos distintos de detección de primer plano. En la fila superior se puede observar el fondo del escenario de un video, un fotograma del video y la segmentación ideal de dicho fotograma, respectivamente. En la fila inferior, cada imagen se corresponde con la segmentación producida para el fotograma mostrado en la fila superior por diferentes algoritmos de detección de objetos de primer plano.

Por su parte, en la figura 2.2 la primera imagen de arriba muestra el escenario en el que se desarrolla el video. A continuación, en la segunda imagen de arriba se observa un fotograma del video con un objeto que se desea segmentar, y en la tercera imagen de arriba vemos la segmentación ideal de dicho fotograma. Finalmente, las tres imágenes de abajo se corresponden con el resultado producido por tres algoritmos distintos de detección de objetos de primer plano. Como se puede comprobar, el resultado de cada algoritmo es distinto, presentando distinta calidad de resultado.

2.2. Modelado del fondo

Como se ha comentado anteriormente, la primera etapa que se realiza en el proceso de teledetección en un sistema de videovigilancia es la segmentación, fase en la que se detectan los objetos que forman parte del primer plano de la escena y se separan del fondo del escenario. El resultado final de un sistema de este tipo depende en gran medida del resultado obtenido en esta primera etapa.

Para realizar esta detección de los objetos se hace necesario que el algoritmo aprenda el fondo del escenario, es decir, que lo modele. Dependiendo del escenario esta tarea puede resultar más o menos difícil debido a los numerosos inconvenientes que afectan al desarrollo de esta tarea. Algunos de los más importantes son los siguientes:

- Sombras. La aparición de sombras constituye uno de los principales problemas a la hora de modelar el fondo ya que en multitud de ocasiones los algoritmos las detectarán como pertenecientes a primer plano.

- Iluminación de la escena. Al cambiar la iluminación del escenario, el fondo que modelan los algoritmos contiene unas tonalidades diferentes del actual escenario, lo que afectará a que dichas partes con distinta tonalidad sean reconocidas como de primer plano. Además, un cambio de iluminación puede provocar sombras.

- Camuflaje. Este error se produce cuando el fondo y el primer plano presentan unas características similares, de forma que los algoritmos no son capaces de detectar el primer plano.

- Fondos dinámicos. Son aquellos en los que se produce un movimiento repetitivo de uno de sus elementos, como por ejemplo el agua de un río o el movimiento de las ramas de un árbol.

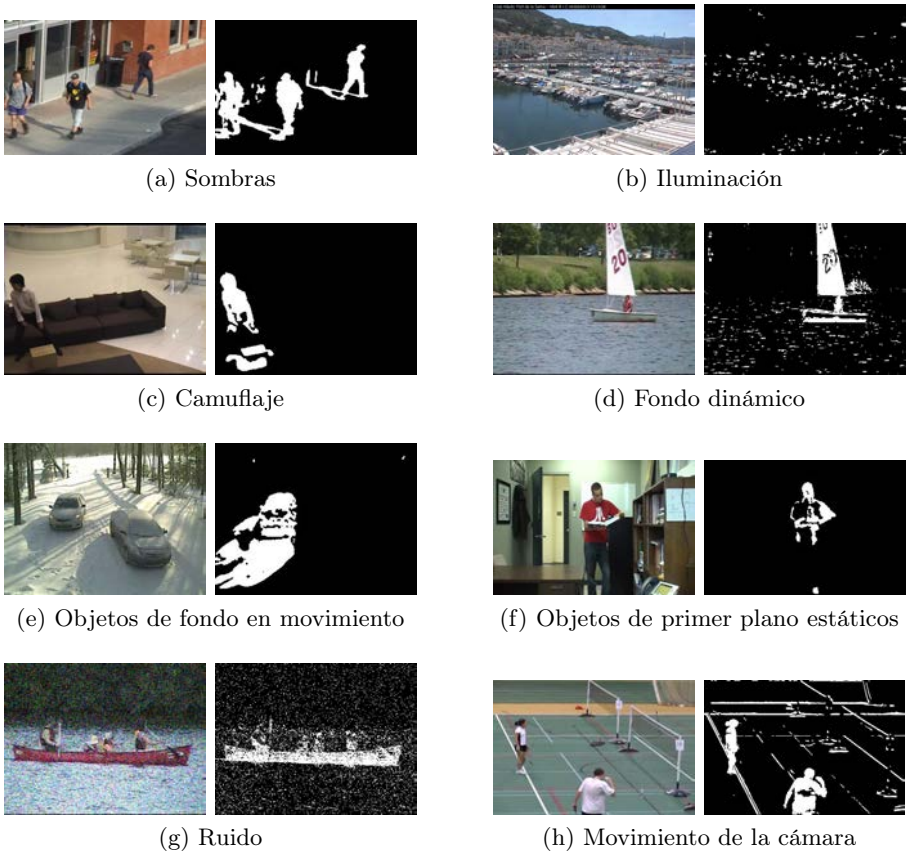
- Objetos de fondo en movimiento. En este caso se trata de un movimiento puntual de un objeto que se considera del fondo. Por ejemplo, en el caso de un coche que se encontraba aparcado y se va, numerosos algoritmos consideran que el hueco que deja pertenece al primer plano.

- Objetos de primer plano estáticos. En otras ocasiones sucede que un objeto de primer plano se queda quieto durante bastante tiempo, con lo que el algoritmo puede aprender que dicho elemento pertenece al fondo y, por tanto, no considerarlo como objeto de primer plano.

- Ruido. El video puede contener ruido como consecuencia de un proceso de grabado defectuoso o un dispositivo de mala calidad.

- Movimiento de la cámara. Una cámara estática puede sufrir vibraciones momentáneas debido a, por ejemplo, la acción puntual del viento, produciendo un movimiento de la escena y el consiguiente error en el modelado del fondo.

En la figura 2.3 se muestra un ejemplo de cada una de estas situaciones problemáticas. En la subfigura (a) se puede observar como el algoritmo detecta las sombras provocadas por las personas que se encuentran caminando por la acera como elementos de primer plano; en la subfigura (b) se ve como el cambio de iluminación del escenario no es reconocido por el algoritmo y éste cree que dicho cambio es primer plano; en la subfigura (c) parte de las piernas de la persona se muestran como no pertenecientes a primer plano debido a que su color es muy parecido a la parte baja del sofá; en la (d) se muestra como primer plano el movimiento del agua; en (e) el vehículo de la izquierda se encontraba aparcado y ha comenzado a desplazarse, lo que motiva al algoritmo a que considere como primer plano el sitio que ocupaba; en la subfigura (f) la persona lleva un tiempo considerable en la misma posición, hecho que implica que el algoritmo considere que la persona pertenece al fondo; en (g) el video ha sido grabado con ruido (en este caso concreto se le



(a) Sombras (b) Iluminación

(c) Camuflaje (d) Fondo dinámico

(e) Objetos de fondo en movimiento (f) Objetos de primer plano estáticos

(g) Ruido (h) Movimiento de la cámara

Figura 2.3: Inconvenientes que dificultan el modelado del fondo. Cada subfigura muestra una dificultad que se puede encontrar al modelar el fondo de un escenario. Para cada subfigura, a la izquierda se muestra un fotograma de un video y a la derecha el resultado producido por un algoritmo de detección de objetos en primer plano.

ha añadido un ruido gaussiano de forma artificial), haciendo que el resultado del algoritmo tenga excesivo ruido; por último, en (h) la cámara estática que estaba grabando la escena ha sufrido una vibración, produciendo que el algoritmo considere como primer plano ese desplazamiento de la imagen. Todos estos resultados producidos por los algoritmos son no deseados y los métodos que se desarrollen deben incorporar mecanismos para anularlos o bien mitigarlos lo máximo posible.

2.3. Métodos de detección de objetos en primer plano

En el caso de las cámaras estáticas, existen numerosos algoritmos para modelar el fondo y detectar los objetos de primer plano. La naturaleza de estos algoritmos es muy diferente aunque su fin sea el mismo. Los hay basados en modelos gaussianos, en distribuciones de mixtura utilizando gaussianas como componentes, no paramétricos basados en funciones de núcleo, en redes neuronales, en lógica difusa, etc. Los algoritmos también pueden estar basados a nivel de píxel, donde se modela el fondo para cada píxel del fotograma o bien para una región o bloque de píxeles. Otro aspecto a considerar son las características o rasgos que se extraen de la imagen. A continuación se ilustran algunos ejemplos de algoritmos:

- Una distribución gaussiana. El método *Pfinder* (Wren et al. 1997, *WrenGA*) se considera uno de los primeros métodos de detección de primer plano. En su modalidad para detectar los objetos de primer plano utiliza una única distribución gaussiana para modelar el fondo.
- Distribuciones de mixtura de gaussianas. Como se utiliza un mayor número de componentes se puede hacer frente a problemas más complejos que en el caso de utilizar una solamente, como, por ejemplo, fondos dinámicos. Un algoritmo de este tipo es el que se puede encontrar en Zivkovic 2004 (*ZivkovicGMM*).
- No paramétricos basados en funciones de núcleo (*kernel density estimation*). Al usar un modelo no paramétrico no se hace necesario realizar una búsqueda de los valores óptimos de los distintos paramétros que utiliza. En Elgammal et al. 2000 (*ElgammalKDE*) se emplea esta estrategia.
- Redes neuronales. Un tipo de redes neuronales muy usual de los que se sirven algunos métodos son los mapas autoorganizados (*Self-Organizing Maps* o *SOM*). Por ejemplo, el método *SOBS* (Maddalena y Petrosino, 2008) es de este tipo. También se pueden encontrar modelos más complejos como el algoritmo *FSOM* (López-Rubio et al., 2011a), que emplean mapas autoorganizados probabilísticos.
- Lógica difusa. En El Baf et al. 2008 (*FuzzyElBaf*) se hace uso de esta técnica para modelar el fondo, utilizando reglas para definir el grado de semejanza entre el fondo y el fotograma.
- Rasgos. Generalmente se utiliza el espacio de color RGB que se corresponde con los canales rojo, verde y azul del fotograma de entrada, pero existen otros rasgos básicos que pueden aportar más información, como el RGB normalizado o el espacio de color HSV, o bien más

complejos, como los de tipo Haar. El método *MFBM* (López-Rubio y López-Rubio, 2015) hace uso de combinaciones de diferentes rasgos para obtener la configuración óptima de valores de los parámetros que utiliza.

2.3.1. Biblioteca BGS

En multitud de ocasiones, la investigación se ve lastrada por la falta de código o ficheros ejecutables de los métodos que se encuentran publicados en la literatura. Sin embargo, también es cierto que existen trabajos con los que se mitiga esta carencia. Éste es el caso de la biblioteca *BGS* (Sobral, 2013), que es una librería de código abierto cuyo objetivo es proporcionar el código de diversos métodos de detección de objetos en primer plano que se encuentran ya publicados y que puede descargarse gratuitamente de su página web[1]. Se ha implementado en C++ y hace uso de la librería OpenCV.

2.4. Aprendizaje profundo

La importancia del *Aprendizaje profundo* (*Deep Learning*) radica en que es un tipo de redes neuronales que están siendo muy utilizadas en la actualidad debido a que se obtienen resultados significativamente mejores que con las redes tradicionales y están suponiendo una revolución en la manera en la que se trataban todos los problemas relacionados con el análisis de imágenes. Ello se ha debido a la mayor capacidad computacional de los ordenadores actuales y, sobre todo, a la gran innovación en las tarjetas gráficas y su *unidad de procesamiento gráfico* (*Graphics Processing Unit* o *GPU*) que llevan incorporadas.

Una arquitectura de aprendizaje profundo es una red neuronal multicapa con numerosas capas ocultas. Muchas de estas capas calculan funciones no lineales, que, en el caso del análisis de imágenes hacen que el sistema sea sensible a pequeños detalles e insensible a grandes variaciones como la iluminación, fondo o alrededores de los objetos. Este tipo de redes no solo es aplicable a análisis de imágenes, sino que también se ha utilizado con éxito en otros campos como el reconocimiento automático del habla.

La arquitectura más popular en el análisis de imágenes es la que se corresponde con las redes neuronales convolucionales (*Convolutional Neural Networks* o *CNNs*), que utilizan varias capas convolucionales combinadas con otros tipos de capas. La idea general es que las capas convolucionales son capaces de extraer características complejas de la imagen al mismo tiempo que reducen la dimensionalidad de los datos tratados.

Dentro de las CNNs hay varios tipos de arquitecturas, implementando distinto número y tipo de capas, tipología de la salida, etc. Dos de los tipos

[1] `https://github.com/andrewssobral/bgslibrary`

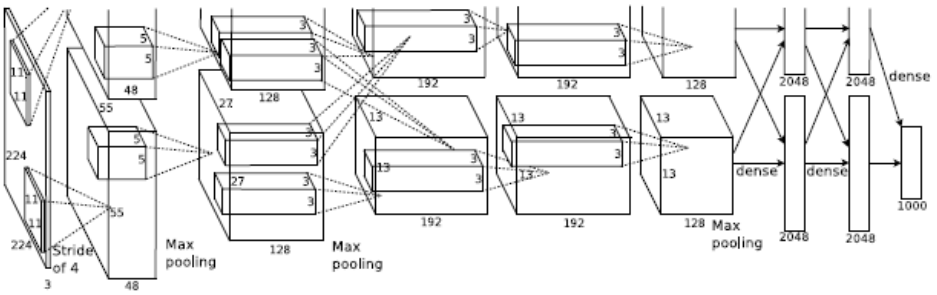


Figura 2.4: Arquitectura de la CNN AlexNet. Imagen extraída de Krizhevsky et al. 2012.

más utilizados son la red AlexNet y la red Faster R-CNN.

La red AlexNet (Krizhevsky et al., 2012) fue desarrollada para clasificar el conjunto de datos *ImageNet* y se probó con el conjunto de datos *CIFAR-10*[2], que está compuesto por 60,000 imágenes de color distribuidas en 10 clases. Tras entrenar la red, se consiguió una tasa de error de tan solo el 15.3 %, lo cual era muy bajo en comparación con las propuestas existentes hasta ese momento. La figura 2.4 muestra la arquitectura de la red Alexnet. Básicamente, el cometido de esta red es proporcionarle una imagen como entrada y produce como salida su clasificación, es decir, la clase más probable para dicha imagen.

Por su parte, el modelo Faster R-CNN (Ren et al., 2017) es más complejo ya que su cometido es el de detectar y clasificar objetos. Para ello predice límites de objetos y puntuaciones de objetividad (probabilidades) en cada posición, proporcionando el área de interés mediante el rectángulo mínimo que encierra al objeto (*bounding box*) y las probabilidades de pertenencia a cada clase. En la figura 2.5 se muestra un ejemplo de uso de esta red. Tal y como se aprecia, el sistema es capaz de detectar objetos y clasificarlos.

2.5. Transporte inteligente

Una vez se ha estudiado la primera etapa de los sistemas de videovigilancia, se puede proceder a las siguientes fases que componen este tipo de sistema. Así, tras producirse el resultado de la detección de los objetos en primer plano se obtiene qué partes del fotograma son primer plano y cuáles se corresponden con el fondo, con lo que el siguiente paso sería determinar las acciones que pueden llevarse a cabo.

En esta tesis se ha deseado estudiar la aplicación de métodos de detección de objetos en primer plano al transporte inteligente. Este concepto hace referencia al desarrollo tecnológico que permite una mejor gestión del transporte

[2] `https://www.cs.toronto.edu/~kriz/cifar.html`

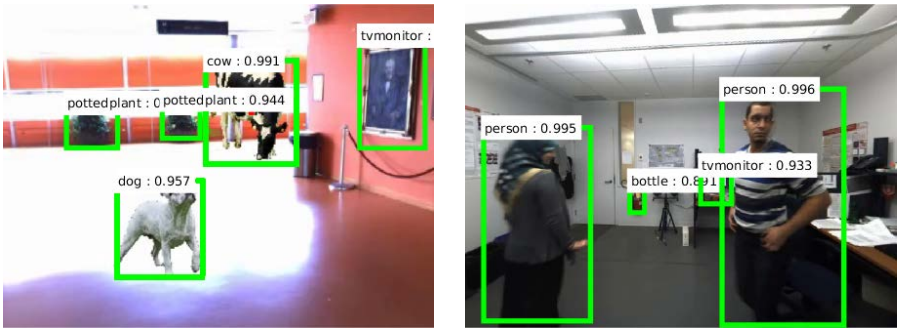


Figura 2.5: Ejemplo de salida producida por el modelo Faster R-CNN. El modelo entrenado reconoce clases como persona, botella, monitor de televisión, planta de maceta, vaca o perro. En las imágenes se observa en verde la región rectangular mínima (bounding box) de los objetos detectados y en un cuadro de texto junto a cada uno de ellos se muestra la clase más probable predicha por el sistema y la probabilidad de pertenencia a dicha clase.

y nace como consecuencia del alto nivel de tráfico que soportan numerosas vías que pertenecen a la red de carreteras, para poder aportar soluciones que reduzcan el tiempo de viaje, el consumo de combustible o la contaminación que se produce. En el caso de la videovigilancia, numerosas vías poseen sistemas de cámaras que permiten monitorizar el estado del tráfico en puntos concretos. Por tanto, sería deseable desarrollar aplicaciones que pudieran utilizar la información facilitada por las cámaras para tomar las acciones que se consideren oportunas. Por ejemplo, si se detecta que hay un alto volumen de tráfico pesado en una determinada zona que está provocando atascos y los consiguientes retrasos, sería posible redirigir el tráfico en un punto previo con el objetivo de impedir esa congestión del tráfico.

Para la implementación de nuevos avances en este campo puede basarse en trabajos previos que pueden encontrarse en la literatura. Uno de ellos es (Luque-Baena et al., 2015b), en el que se da respuesta a la monitorización de vehículos en videos de tráfico, generando información muy útil del mismo como la trayectoria seguida por cada vehículo.

2.6. Cámara PTZ

Todo lo comentado hasta ahora se basa en videos capturados por una cámara estática. En esta tesis se ha propuesto estudiar diversos aspectos relacionados con las cámaras PTZ. Este tipo de cámaras añade complejidad a los sistemas de videovigilancia debido a que la cámara puede ser controlada por el propio programa, realizando los movimientos que considere oportunos en cada instante de tiempo. Además, por el efecto de esos movimientos el área del escenario observado es mayor que el área del fotograma.

Para ilustrar dicha complejidad, se considera mostrar el problema de la segmentación de objetos en primer plano y modelado de fondo. En este caso para dar respuesta existen diversas propuestas a dicha problemática. Una primera consiste en ir modelando todo el fondo y actualizar solo aquella parte que se corresponde con la parte de fondo recogida por el fotograma. Otra propuesta consiste en ir modelando únicamente el fondo correspondiente al fotograma. En la primera propuesta se tiene el inconveniente de localizar la parte de fondo que corresponde con el fondo recogido en el fotograma. Por su parte, el inconveniente de la segunda propuesta es que el modelo comete numerosos errores en la zona de los bordes que corresponden al último movimiento realizado por la cámara, es decir, si la cámara ha realizado un movimiento vertical hacia la derecha, la parte correspondiente al borde derecho del modelo de fondo será muy sensible a errores, ya que esta parte no estaba modelada.

Otro problema que conlleva este tipo de sistemas es que la realización de pruebas y experimentos en tiempo real puede ser complicada. En este caso hay que tener en cuenta que en un instante dado, el siguiente fotograma dependerá de los movimientos que ejecute la cámara. Por tanto, no se podría medir el rendimiento del sistema. Sin embargo, es posible utilizar entornos simulados con los que poder considerar las distintas opciones que surjan. Un ejemplo de simulador de este tipo es la librería *Virtual PTZ* (Chen et al., 2015). Esta biblioteca simula una cámara PTZ y su posible comportamiento en un escenario gracias a un video capturado en dicho escenario mediante una cámara de 360 grados. De esta forma, el escenario puede ser almacenado en un video panorámico. Tras esto, se considera la cámara PTZ situada en el lugar donde estaba situada la cámara de 360 grados y, realizando una serie de transformaciones matemáticas oportunas en la parte de fotograma que queda observada por la cámara PTZ, se obtiene el fotograma observado.

La librería Virtual PTZ se ha escrito en C++ haciendo uso de la librería OpenCV y es de código libre, estando accesible en su página web[3].

2.7. Evaluación del rendimiento

El hecho de que existan multitud de algoritmos y métodos enfocados a solventar un problema dado, hace que se necesite un mecanismo de evaluación que nos permita poder compararlos entre ellos y poder decidir si uno es mejor que otro de una manera totalmente objetiva. Es por ello por lo que para lograr tal fin se hace necesaria la utilización de lo que se denomina *máscara de verdad* (*ground truth mask* o GT), que se corresponde con el resultado ideal que debería producir un algoritmo. De esta forma, se puede comparar este resultado ideal con el resultado que produce un algoritmo y, mediante esta comparación, establecer la bondad de dicho algoritmo.

[3] `https://bitbucket.org/pierre_luc_st_charles/virtualptz_standalone`

Para llevar a cabo esta comparación de forma objetiva, se hace uso de una serie de medidas de rendimiento y de diversos conjuntos de datos de test que permiten valorar cuantitativamente el resultado producido por un algoritmo.

2.7.1. Medidas de rendimiento

Existen multitud de medidas, y cada una de ellas mide un aspecto concreto. Algunas de las medias más representativas son la *exactitud* (*accuracy* o *Acc*), la *exactitud espacial* (*spatial accuracy* o *S*) o la *F-medida* (*F-measure* o *F-m*). Estas medidas representan el porcentaje de aciertos del sistema y ofrecen una idea general de la idoneidad de un método, proporcionando un valor numérico en el rango $[0, 1]$ donde 0 es el peor resultado y 1 es el mejor posible.

Otras medidas que se utilizan son la exhaustividad (*recall* o *RC*), la precisión (*precision* o *PR*), la especificidad (*specificity* o *SP*), la tasa de falsos negativos (*false negative rate* o *FNR*), o la proposición de fallo (*fall-out* o *false positive rate* o *FPR*). Algunas de ellas se deben comparar entre sí por pares para valorar de manera global un método. Por ejemplo, se suele utilizar la precisión contra el recall.

Para el cálculo de estas medidas se hace uso de los verdaderos positivos (TP), verdaderos negativos (TN), falsos positivos (FP) y falsos negativos (FN). Los TP se corresponden con aquellos píxeles pertenecientes al primer plano que el método detecta como tales, mientras que los TN son aquellos píxeles pertenecientes al fondo del escenario que el método así reconoce. Por su parte, los FP son píxeles que el método detecta como primer plano pero que en realidad son del fondo, mientras que los FN son aquellos píxeles que el método considera que pertenecen al fondo pero que en realidad son del primer plano.

Siguiendo con el primero de los ejemplos utilizados anteriormente, el de la segmentación del autobús en una carretera, vamos a ilustrar estos conceptos de forma visual. Así, en la figura 2.1, las zonas pertenecientes al objeto detectadas correctamente son TP, mientras que las zonas detectadas correctamente como fondo de la escena son TN. Por su parte, el ruido existente en la zona de los árboles son FP, mientras que las ruedas no detectadas son FN.

Por su parte, si se analiza en detalle el resultado producido por cada algoritmo en el segundo ejemplo mostrado con anterioridad, en la figura 2.2 se observa como la primera imagen muestra un bajo número de FN y un elevado número de FP, mientras que la segunda ofrece una alta cantidad de FN y una cifra pequeña de FP. Por su parte, la tercera imagen muestra un resultado con un número bajo tanto de FP como de FN, por lo que se consideraría como el mejor resultado de los tres analizados. Lo ideal es que

el resultado producido por un algoritmo ofrezca un número bajo tanto de FP como de FN pero, llegado el caso de que no se pueda disminuir ambos indicadores, es preferible tener un resultado con mayor número de FP que de FN. Esto se debe a que los FP son más fáciles de tratar y corregir mediante técnicas de postprocesado de la imagen.

Dicho esto, las definiciones de cada una de las medidas comentadas son las siguientes:

$$RC = \frac{TP}{TP + FN} \qquad PR = \frac{TP}{TP + FP} \qquad SP = \frac{TN}{FP + TN} \tag{2.1}$$

$$FNR = \frac{FN}{TP + FN} \qquad FPR = \frac{FP}{FP + TN} \tag{2.2}$$

$$S = \frac{TP}{TP + FN + FP} \qquad Acc = \frac{TP + TN}{TP + FP + FN + TN} \tag{2.3}$$

$$\text{F-measure} = 2 * \frac{PR * RC}{PR + RC} \tag{2.4}$$

2.7.2. Conjuntos de datos de test

Al igual que ya se han mostrado cuáles son las medidas utilizadas para establecer la bondad de un método a la hora de compararlo con el rendimiento de otros, ahora se va a ver los conjuntos de datos de test que se utilizan como punto de referencia para dicha comparativa. En el campo de la videovigilancia existen multitud de secuencias de imágenes y video de diversa índole, las cuales se encuentran disponibles al público, incluso incluyendo su ground truth. A continuación se citan varios de estos conjuntos de datos (dataset).

2.7.2.1. Changedetection.net

Existen diversos conjuntos de datos conformados por videos de diversa índole para estudiar el rendimiento de los métodos de segmentación. Uno de ellos es la base de datos de videos *ChangeDetection.net*[4] (Goyette et al., 2012), que en su versión de 2014 ofrece videos organizados en 11 categorías con entre 4 y 6 videos por cada categoría. Los videos presentan un gran número de situaciones reales, con escenas de interior o exterior, personas andando, vehículos circulando, etc, además de inconvenientes que suelen afectar al rendimiento de los algoritmos de segmentación como cambios repentinos de iluminación, sombras o efectos de camuflaje. Las categorías en las que se encuentra dividido sirven para conocer el rendimiento de un algoritmo en unas condiciones ambientales concretas. Por ejemplo, la categoría *Baseline*

[4] `http://changedetection.net/`

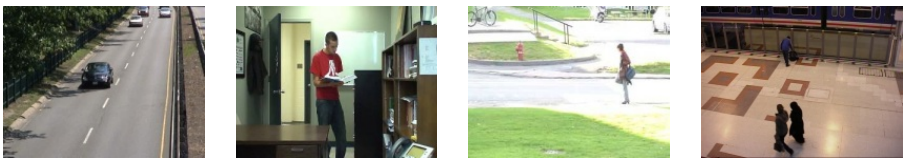


Figura 2.6: Videos pertenecientes a la categoría *Baseline* del conjunto de datos *ChangeDetection.net*. De izquierda a derecha se muestran fotogramas de los videos *Highway*, *Office*, *Pedestrians* y *PETS2006*, respectivamente.

(Básica) contiene videos simples que deberían ser fácilmente gestionados por los algoritmos; por su parte, los videos de la categoría *Dynamic Background* (Fondo Dinámico) contienen elementos en movimiento que se consideran del fondo de la escena, tales como un río, el mar o una fuente. Los videos que forman la categoría Baseline pueden observarse en la figura 2.6. Además, este conjunto de datos ofrece un código para evaluar un método y un listado con el rendimiento obtenido por varios algoritmos de la literatura.

2.7.2.2. US-Highway

Además de los conjuntos de datos genéricos, también existen otros específicos para un problema concreto. Así, por ejemplo, para estudiar el problema del transporte se puede hacer uso del conjunto de datos *US Highway 101 (US 101)*[5]. Este conjunto se compone de varios videos tomados en una autovía con 8 cámaras sincronizadas que cubren un área de unos 640 metros de la misma ya que se encuentran situadas en la parte superior de un edificio adyacente, grabando todos los vehículos que pasan por dicha carretera. Además, se incluye la información de la trayectoria de cada vehículo. Otro conjunto de datos del mismo tipo es el *Lankershim Boulevard Dataset*[6]

2.7.2.3. Virtual PTZ

La evaluación del rendimiento de un algoritmo para videos de cámara PTZ es compleja porque dependiendo del movimiento de la cámara el fotograma que se obtiene corresponderá a una parte del escenario u otra. Para poder evaluar este tipo de algoritmos existen conjuntos de datos específicos, como el que incorpora la librería *Virtual PTZ* y que puede encontrarse en su página web[7]. Estos videos son grabados mediante una cámara panorámica de forma que todo el escenario es captado por dicha cámara. Luego, gracias a la propia librería, se simula el funcionamiento de una cámara PTZ situada en el mismo punto donde se encontraba la cámara panorámica que grabó

[5] `https://www.fhwa.dot.gov/publications/research/operations/07030/`
[6] `https://www.fhwa.dot.gov/publications/research/operations/07029/`
[7] `https://drive.google.com/file/d/0B55Ba7lWTLh4TFIxbHduU0hEb1U`

Figura 2.7: Video *scenario3* perteneciente al conjunto de datos *Virtual PTZ*.

dicho video. En la figura 2.7 puede verse un fotograma del video *scenario3*, el cual pertenece a este conjunto de datos. Tal y como se observa, la imagen es panorámica y puede observarse el escenario prácticamente por completo (tan solo se pierde parte de la zona inferior y superior).

2.7.2.4. ASH Image Bank

Actualmente, muchas de las imágenes que se obtienen en los laboratorios mediante microscopios son digitalizadas y procesadas por ordenador para hacer más fácil y rápido el proceso de análisis de las imágenes. Además de otros campos, las imágenes de muestras de sangre pueden aportar información muy valiosa de la salud de un paciente. Por ejemplo, en hematología (rama de la medicina que realiza el estudio de la sangre) se utiliza el número de células sanguíneas en la muestra de sangre. Así, el porcentaje ocupado por los glóbulos rojos en relación con el total de sangre es conocido como hematocrito. Este valor es muy importante para la detección de varias enfermedades. Un incremento o reducción apreciable de hematocrito puede ser un indicativo de anemia. En muchos casos, estas enfermedades no son graves pero pueden ser causadas por otros problemas. Por tanto, la detección temprana puede ser vital.

Ejemplos de este tipo de conjunto de datos son el *ASH Image Bank*[8], donde las imágenes que lo componen muestran varios tipos de células sanguíneas (sobre todo glóbulos rojos). Además, las imágenes muestran una amplia variedad de escalas, diferentes grados de iluminación y células solapadas, por lo que se pueden estudiar diversas condiciones. En la figura 2.8 se pueden observar varias imágenes pertenecientes a dicho conjunto de datos.

[8] `http://imagebank.hematology.org`

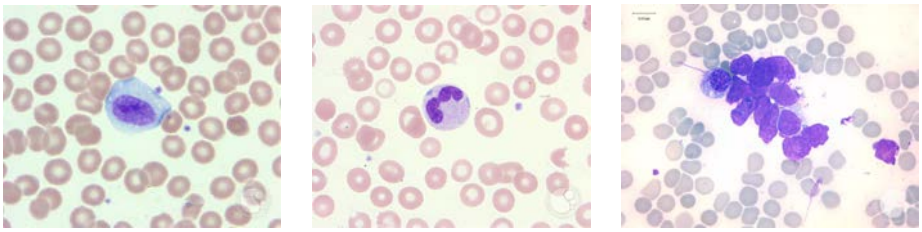


Figura 2.8: Imágenes de muestras sanguíneas pertenecientes al conjunto de datos *ASH Image Bank*.

Otro conjunto de datos de este tipo es el *Acute Lymphoblastic Leukemia Image Database for Image Processing (ALL-IDB)* [9]

2.7.2.5. ImageNet

ImageNet[10] es un conjunto de datos de imágenes que se propone con motivo del *ImageNet Large Scale Visual Recognition Challenge (ILSVRC)*, ofreciendo más de diez millones de imágenes clasificadas en más de veinte mil clases ambiguas, aunque para el reto se utiliza una lista de mil categorías no ambiguas y un conjunto de entrenamiento de más de un millón de imágenes. Este reto se celebra desde 2010, pero año a año se ha visto un salto cualitativo en el rendimiento de los sistemas gracias a la irrupción del aprendizaje profundo.

2.7.2.6. VOC Datasets

Estos conjuntos de datos fueron originados en el *The Pascal Visual Object Classes (VOC) Challenge*[11]. El objetivo de estos retos era reconocer objetos en escenarios reales. Para este propósito, veinte diferentes clases de objetos eran seleccionadas (por ejemplo, persona, pájaro, gato, perro o coche) y conjuntos de entrenamiento/validación y test eran creados. Los conjuntos de datos más utilizados son los de 2007 y 2012. El primero tiene 9963 imágenes con 24640 objetos etiquetados, mientras que en el segundo hay 1153 imágenes con 27450 objetos etiquetados. Ambos se encuentran divididos en dos conjuntos con el 50 % de las imágenes para entrenamiento/validación y 50 % para test. Además de las imágenes, se adjuntan el *bounding box* (rectángulo de menor área que encapsula a un objeto) y la clase de cada objeto que aparece en cada imagen.

[9] `http://crema.di.unimi.it/~fscotti/all/`
[10] `http://www.image-net.org/`
[11] `http://host.robots.ox.ac.uk/pascal/VOC/index.html`

Parte I

Segmentación

Esta primera parte de la tesis presenta los trabajos desarrollados en torno a la segmentación de imágenes. Así, en esta parte se engloban trabajos de distinta índole como nuevos algoritmos para ejecutarlos en condiciones hardware generales o bien específicas, como los dispositivos de bajo coste; estudios sobre cómo reducir el tamaño de las actuales imágenes digitales, las cuales son excesivamente grandes para que puedan utilizarse con ellas los algoritmos tradicionales; o la segmentación de imágenes de un campo concreto, como es el caso de la segmentación de glóbulos rojos en muestras sanguíneas.

Capítulo 3

Detección de primer plano mediante aprendizaje competitivo para distribuciones de entrada variantes

La mayoría de las ideas fundamentales
de la ciencia son esencialmente sencillas,
y por regla general pueden ser expresadas
en un lenguaje comprensible para todos.

Albert Einstein

Resumen: Uno de los retos más importantes en las aplicaciones de visión por computador es el modelado del fondo, especialmente cuando es dinámico y la distribución de entrada puede no ser estacionaria, es decir, la distribución de los datos de entrada podría cambiar con el tiempo (por ejemplo, cambios de iluminación, árboles en movimiento, agua, etc.). En este capítulo se propone una red neuronal de aprendizaje no supervisado que puede superar los progresivos cambios en la distribución de entrada. Ello se basa en un mecanismo de aprendizaje dual que gestiona los cambios de la distribución de entrada separadamente del grupo de detección. La propuesta es adecuada para escenas donde el fondo varía despacio. El rendimiento del método es probado frente a varios detectores de primer plano del estado del arte, tanto cuantitativamente como cualitativamente, con resultados favorables.

3.1. Introducción

En las aplicaciones de visión por computador, como la videovigilancia o el análisis del movimiento humano, la capacidad de extraer los objetos de interés de una secuencia de video es una tarea preliminar crucial.

La videovigilancia es una tecnología clave para la seguridad pública (por ejemplo, en las redes de transporte, centros de las ciudades, escuelas y hospitales), gestión eficiente de las redes de transporte y servicios públicos (por ejemplo, semáforos, cruces de carreteras), reconocimiento de patrones del comportamiento humano, etc (Maddalena y Petrosino, 2008). Los fotogramas normalmente son capturados de una escena usando una cámara estática que comprime la información del video. En este escenario, detectar los objetos entrantes es una etapa esencial en el análisis de la escena, ya que un gran número de características pueden ser extraídas del primer plano para ejecutar tareas de clasificación y reconocimiento en siguientes fases. Un supuesto usual es que los fotogramas de una escena sin objetos de interés muestran un comportamiento regular, que puede ser descrito con un modelo de fondo estadístico. Una propuesta típica para discriminar los objetos de interés con respecto del fondo es la detección mediante un proceso de sustracción del fondo. La idea de este proceso consiste en eliminar la actual imagen del modelo de fondo de referencia. Un píxel de la actual imagen es considerado fondo cuando su valor observado (color) está bien representado por el modelo. En otro caso, el píxel es considerado primer plano.

Sin embargo, el modelo de fondo es un problema en aplicaciones con circunstancias difíciles, como cambios de iluminación, árboles en movimiento, agua, expositores de video, ventiladores giratorios, sombras, camuflaje (los objetos de primer plano son similares al fondo), cambios ocasionales del verdadero fondo (por ejemplo eliminando un objeto de la escena), tráfico, etc. Estos problemas no pueden ser resueltos por los modelos simples de fondos estáticos. Existen muchas soluciones que gestionan los problemas de fondo dinámico mencionados anteriormente. Pero el coste computacional de muchos de ellos es caro, lo que obstaculiza su uso en aplicaciones de tiempo real. Incluso el caso cuando un análisis visual es usado para evaluar después los eventos detectados, los datos acumulados pueden llegar a ser muchos para un proceso a posteriori. Por tatno, muchos sistemas necesitan de un procesado en tiempo real para mantener el rito del flujo de datos del video de entrada.

Se han desarrollado redes neuronales que ofrecen soluciones en el campo del modelado de fondo (Maddalena y Petrosino, 2008; López-Rubio et al., 2011a). En este capítulo, se propone un modelo neuronal que es un caso especial de los mapas autoorganizados. Está especialmente diseñado para gestionar las distribuciones de entrada no estables con una estrategia de aprendizaje en dos fases, donde la neurona ganadora ejecuta una más inten-

sa adaptación al patrón de entrada que las neuronas restantes, que aprenden lentamente para seguir los cambios de la distribución de entrada. El algoritmo del modelo de aprendizaje propuesto está diseñado para optimizar una función de coste que considera estos dos objetivos explícitamente. Por tanto, la contribución original de la propuesta es doble:

- Se propone un modelo de red neuronal de aprendizaje no supervisado, que es específicamente diseñado para gestionar las necesidades de las distribuiciones de entrada variables.
- El modelo de aprendizaje propuesto es aplicado al problema del modelado del fondo, que es uno de los problemas clave en el desarrollo de los actuales sistemas de videovigilancia.

La propuesta difiere sustancialmente de otros modelos de fondo previos basados en mapas autoorganizados (López-Rubio et al., 2011a). En particular, el problema de las neuronas que dejan de representar alguna entrada, es decir, el problema de las neuronas muertas, no está resuelto por una topología de mapa autoorganizado estándar. Al contrario, un nuevo término es añadido a la función de coste para mover un poco todas las neuronas al centro de la distribución de entrada. Por tanto, todas las neuronas siguen la distribución de entrada tal y como varía a lo largo del tiempo, mientras que a la red neuronal se le impone una topología no fija. Esto también difiere de otros modelos de fondo de tipo ART (Adaptive Resonance Theory o teoría de resonancia adaptativa, Luque et al. 2010) de dos maneras. Por un lado, la nueva propuesta no necesita un parámetro de vigilancia para evitar las neuronas muertas. Por otro lado, en la nueva propuesta todas las neuronas son utilizadas para modelar el fondo de la escena, lo que provoca un mejor uso de ellas y elimina la necesidad de un procedimiento para decidir qué neuronas están asociadas al fondo.

3.2. Trabajo previo

Hay innumerables métodos de modelado de fondo y varios estudios pueden encontrarse en la literatura (Bouwmans, 2014b; Sobral y Vacavant, 2014).

De acuerdo a estos estudios, hay tres tipos de métodos:

- Modelos básicos, donde el fondo es modelado usando la media, la mediana o un histograma a lo largo del tiempo (Zheng et al., 2006).
- Modelos estadísticos, donde la distribución del fondo es modelada utilizando una gaussiana (Wren et al., 1997) o una mixtura de gaussianas (Stauffer y Grimson, 1999) o una estimación de la densidad del núcleo (Elgammal et al., 2000) entre otros.

- Modelos de estimación. En este caso, el fondo es estimado utilizando un filtro, por ejemplo Wiener (Toyama et al., 1999), Kalman o Tchebychev, y los píxeles que se desvían significativamente de su valor estimado son considerados primer plano.

Actualmente el modelado de fondo es una prolífica área de investigación con numerosas propuestas para gestionar el problema de la detección del movimiento en secuencias de video incorporando diversas estrategias y complejas características (Kim y Kim, 2016; Yeum y Dyke, 2015; Ortega-Zamorano et al., 2016; Lacabex et al., 2016) para mejorar las propuestas tradicionales. La aplicación de redes neuronales al campo del aprendizaje automático no es nuevo y hay numerosos trabajos que han contribuido en numerosas disciplinas dentro de este área (Paris et al., 2015; Adeli y Hung, 1995, 1993; Hung y Adeli, 1993, 1994; Adeli y Hung, 1994; Martinez et al., 2017).

Especialmente, en el campo del modelado de fondo se han propuesto diferentes tipos de redes neuronales, de aprendizaje competitivo y mapas autoorganizados a funciones de base radial o perceptrones multicapa. El aprendizaje competitivo estándar, que está muy relacionado con la propuesta de este capítulo, puede ser usado para propósitos de cuantificación de vectores, que significa que puede encontrar libros de códigos que representan un conjunto de vectores de entrada gracias a un conjunto reducido de prototipos. Esto es tradicionalmente empleado para obtener representaciones de datos visuales (Yang et al., 2014b). Esta idea está también propuesta en otros trabajos de modelado de fondo basado en libros de códigos (Kim et al., 2005) con la diferencia que en este modelo todas los prototipos están asociados con entidades (neuronas) que aprenden continuamente con diferentes intensidades para evitar neuronas muertas, es decir, neuronas que no representan ningún grupo de la distribución de entrada. El aprendizaje competitivo tiene que lidiar con las distribuciones de entrada que no son estables a lo largo del tiempo. Es decir, hay aplicaciones prácticas donde la distribución de entrada varía durante el funcionamiento del sistema (Gurubel et al., 2014). Este es el caso de las aplicaciones de visión por computador mencionadas anteriormente.

Varios modelos neuronales de aprendizaje competitivo han sido propuestos en la literatura. Una propuesta neuronal de tipo ART (Luque et al., 2008a, 2010) puede ser utilizada para gestionar las escenas con fondo dinámico, donde las neuronas menos activadas están asociadas con el primer plano y las más activas se corresponden con el fondo. La principal diferencia con el aprendizaje competitivo tradicional es que cada neurona incorpora un ratio de vigilancia para su activación. Otras alternativas consideran el vecindario de las neuronas como conjuntos difusos para ofrecer más flexibilidad al modelo neuronal (Luque Baena et al., 2008), o adaptan el vecindario de cada neurona al avance del aprendizaje (Palomo y López-Rubio, 2016). Además, se pueden considerar otro tipo de medidas de divergencia como la distancia

euclídea (López-Rubio et al., 2014).

Los mapas autoorganizados también son considerados como una solución alternativa para modelar el fondo y varios trabajos basados en este tipo de aprendizaje no supervisado pueden encontrarse en la literatura. En particular, varios de ellos modelan cada píxel mediante un mapa autoorganizado (Maddalena y Petrosino, 2008; López-Rubio et al., 2011a; Zhao et al., 2015; Singh et al., 2010), por lo que mantienen una topología entre las neuronas que considera que cada prototipo puede aprender de acuerdo a la distancia con respecto a la neurona ganadora. La topología debería ser fijada a priori durante la fase de inicialización; una topología de malla es normalmente empleada. Esto puede ser visto como una de las principales restricciones de este tipo de modelos, ya que los datos podrían estar distribuidos de una manera que no se parece a la topología especificada con anterioridad. Esta es la principal diferencia con los métodos de aprendizaje competitivo donde se considera la no relación entre neuronas para ofrecer mayor flexibilidad al modelo. Otras propuestas neuronales basadas en las redes de Hopfield (Luque et al., 2008b; Subudhi et al., 2015) son aplicadas al modelado del fondo, pero solo mejoran la máscara básica que contiene los objetos de primer plano generados por un modelo previo.

Las redes neuronales convolucionales son aplicadas a la detección de objetos en secuencias de video, desde el reconocimiento de masas (Zhang et al., 2015; Chen et al., 2016b), reconocimiento de objetos y acciones (Reddy y Shah, 2013) o la sustracción de fondo (Braham y Droogenbroeck, 2016). Sin embargo, todas estas propuestas requieren una fase de entrenamiento para construir el modelo, mientras que la propuesta de este capítulo surge de una metodología no supervisada donde no hay datos de entrenamiento y no se necesitan restricciones sujetas al tipo de objetos a ser detectados. Etiquetar manualmente un montón de objetos para generar muestras de entrenamiento (Braham y Droogenbroeck 2016 requiere etiquetar la mitad de cada secuencia) solo es viable para experimentos individuales pero no para la detección de movimiento en alguna de los cientos de miles de cámaras IP de videovigilancia instaladas en las ciudades, edificios o carreteras. También se ha propuesto un etiquetado automático del conjunto de datos por otros algoritmos de sustracción de fondo (Bianco et al., 2017), aunque en este caso los resultados no mejoran las tradicionales técnicas propuestas en la literatura (Braham y Droogenbroeck, 2016). Dadas estas consideraciones, los algoritmos de aprendizaje profundo mencionados anteriormente son adecuados para posteriores etapas de los sistemas de videovigilancia, es decir, aquellos dedicados al análisis de comportamiento, que están fuera del objetivo de este capítulo.

3.3. Modelo

En esta sección se propone una red neuronal que es un caso especial de los mapas autoorganizados. Es adecuada para aprender las distribuciones de entrada no estacionaria. La mayor dificultad para adaptar una red neuronal competitiva a este tipo de problema es ese, incluso si una neurona ha posicionado correctamente su prototipo al centro de un grupo, el grupo podría desaparecer a medida que el tiempo pasa. Por tanto, es posible que la neurona deje de representar cualquier muestra de entrada actual, que la conduciría a su muerte, es decir, su prototipo no ganaría la competición nunca más, por lo que no aprendería. Para solventar este problema, se propone que todos los prototipos de neuronas se muevan un poco hacia el actual centro de la distribución de entrada general. De esta manera todos los prototipos permanecen cerca de las actuales muestras en el espacio de entrada, que les ofrece una mejor oportunidad de representar un grupo. Esta estrategia está implementada añadiendo un término a la función de coste del estándar k-medias, por lo que hay una pequeña penalización para la distancia euclídea al cuadrado desde cada prototipo a la actual muestra de entrada. La razón es que el término extra asegura que incluso en el caso de que una neurona no gane la competición durante un largo tiempo, ella se moverá hacia el centro de la distribución de entrada donde hay más posibilidades de encontrar muestras para representar.

Sea D la dimensión del espacio de entrada, es decir, las muestras de entrada son $\mathbf{x} \in \mathbb{R}^D$. Se ha adoptado el espacio de color RGB, por tanto, en este caso $D = 3$. Como se hace en el aprendizaje competitivo tradicional, se consideran N neuronas con $N > 1$, donde cada neurona i almacena un vector prototipo $\mathbf{w}_i \in \mathbb{R}^D$, con $i \in \{1, ..., N\}$. Se considera la siguiente función de coste:

$$\mathcal{E} = \frac{1}{2} \sum_{j=1}^{M} \left\| \mathbf{x}_j - \mathbf{w}_{Winner(\mathbf{x}_j)} \right\|^2 + \frac{\lambda}{2} \sum_{j=1}^{M} \sum_{i=1}^{N} \|\mathbf{x}_j - \mathbf{w}_i\|^2 \tag{3.1}$$

donde M es el número de muestras de entrada, $\|\cdot\|$ se considera la norma euclídea, λ es un parámetro de ponderación y $Winner\,(\mathbf{x}_j)$ es el índice de la neurona ganadora, es decir, la neurona cuyo prototipo está más cercano de la muestra de entrada $\mathbf{x}_j$:

$$0 \leq \lambda \ll 1 \tag{3.2}$$

$$Winner\,(\mathbf{x}) = \arg \min_{i \in \{1,...,N\}} \|\mathbf{x} - \mathbf{w}_i\| \tag{3.3}$$

Es bien sabido que el aprendizaje competitivo estándar está garantizado para alcanzar un mínimo local del Error Cuadrático Medio (Mean Squared Error Ueda y Nakano 1994), que es el primer término de la ecuación (3.1). Esto avisa a las estrategias para escapar de estos locales mínimos. El segundo término propuesto en la ecuación (3.1) cumple este rol, ya que arrastra un poco a todos los prototipos desde sus respectivos centros de grupos, que son asociados con el mínimo local del Error Cuadrático Medio, al centro de las distribuciones de entrada. Este esquema de aprendizaje dual basado en dos términos de coste con diferente objetivos facilita la búsqueda de configuraciones de grupo alternativas.

Las redes neuronales de aprendizaje competitivo pueden ser utilizadas para el particionado de grupos (Menéndez et al., 2014), es decir, cada muestra pertenece exactamente a un grupo. En este caso el número de neuronas está fijado para conseguir el funcionamiento en tiempo real, en contra de los métodos de agrupamiento que permiten división o agrupamiento de grupos (Chira et al., 2014). El grupo de las muestras de entrada asociado con la neurona i es definido como sigue:

$$C_i = \left\{ \mathbf{x}_j \in \mathbb{R}^D \mid i = Winner\left(\mathbf{x}_j\right) \right\} \tag{3.4}$$

De (3.4) la función de coste (3.1) puede ser reescrita como sigue:

$$\mathcal{E} = \frac{1}{2} \sum_{j=1}^{M} \sum_{i=1}^{N} \mathbb{I}\left(\mathbf{x}_j \in C_i\right) \left\|\mathbf{x}_j - \mathbf{w}_i\right\|^2 + \frac{\lambda}{2} \sum_{j=1}^{M} \sum_{i=1}^{N} \left\|\mathbf{x}_j - \mathbf{w}_i\right\|^2 \tag{3.5}$$

donde $\mathbb{I}$ es la función indicadora:

$$\mathbb{I}\left(\textit{condición}\right) = \begin{cases} 1 & \text{si } \textit{condición} \text{ es verdadero} \\ 0 & \text{si } \textit{condición} \text{ es falso} \end{cases} \tag{3.6}$$

Nótese que el primer término de (3.1) es la función de coste estándar del aprendizaje competitivo, que tiene la intención de minimizar la media de las distancias al cuadrado del prototipo de cada neurona con respecto de las muestras de entrada para que esa neurona sea la ganadora. Por tanto, intenta encontrar un libro de códigos adecuado para la cuantificación de vectores. Por otro lado, el segundo término de (3.1) trata de minimizar la media de las distancias al cuadrado de cada prototipo con respecto a todas las muestras de entrada. Por tanto, se ocupa mover los prototipos en $\mathbb{R}^D$ a la media de la distribución de entrada. De esta forma, en caso de que la distribución de entrada se mueva a lo largo del tiempo, el segundo término de (3.1) hace que todos los prototipos sigan la distribución de entrada en $\mathbb{R}^D$. Este

movimiento debería ser lento con respecto al ajuste de los prototipos para encontrar un buen libro de códigos, de ahí la condición (3.2). Debe señalarse que el segundo término de (3.1) tiene la misma complejidad computacional que el primero. Esto es porque ambos implican el cálculo de las distancias euclídeas desde todas las muestras de entrada a todos los prototipos de las neuronas. También es interesante resaltar que para el aprendizaje en línea se tiene $M = 1$ para cada paso de tiempo (solo una muestra de entrada, es decir, que la carga computacional es reducida.

Una vez que la función de coste $\mathcal{E}$ está definida, su gradiente puede ser calculado. De (3.5) se tiene:

$$\frac{\partial \mathcal{E}}{\partial \mathbf{w}_i} = -\sum_{j=1}^{M} \mathbb{I}\left(\mathbf{x}_j \in C_i\right)\left(\mathbf{x}_j - \mathbf{w}_i\right) - \lambda \sum_{j=1}^{M} \left(\mathbf{x}_j - \mathbf{w}_i\right) \tag{3.7}$$

Para el aprendizaje por lotes, un punto crítico de $\mathcal{E}$ puede ser obtenido resolviendo esta ecuación:

$$\frac{\partial \mathcal{E}}{\partial \mathbf{w}_i} = 0 \tag{3.8}$$

De esta forma, una regla de actualización del aprendizaje por lotes para los prototipos puede ser encontrado. Sea $\operatorname{card}\left(C_i\right)$ el cardinal del i-ésimo grupo, es decir, el número de muestras de entrenamiento que pertenecen a C_i. Luego:

$$\sum_{j=1}^{M} \mathbb{I}\left(\mathbf{x}_j \in C_i\right)\left(\mathbf{x}_j - \mathbf{w}_i\right) + \lambda \sum_{j=1}^{M} \left(\mathbf{x}_j - \mathbf{w}_i\right) = 0 \tag{3.9}$$

$$\sum_{j=1}^{M} \mathbb{I}\left(\mathbf{x}_j \in C_i\right)\mathbf{x}_j + \lambda \sum_{j=1}^{M} \mathbf{x}_j = \left(\operatorname{card}\left(C_i\right) + \lambda M\right)\mathbf{w}_i \tag{3.10}$$

$$\mathbf{w}_i = \frac{1}{\operatorname{card}\left(C_i\right) + \lambda M}\left(\sum_{j=1}^{M} \mathbb{I}\left(\mathbf{x}_j \in C_i\right)\mathbf{x}_j + \lambda \sum_{j=1}^{M} \mathbf{x}_j\right) \tag{3.11}$$

El algoritmo de aprendizaje por lotes viene dado por:

1. Inicializar cada prototipo $\mathbf{w}_i$ a la muestra de entrenamiento dibujada uniformemente aleatoria desde el conjunto de entrenamiento.
2. Calcular los grupos C_i asociados con cada neurona por (3.4).
3. Actualizar los prototipos aplicando (3.11).
4. Si el número máximo de iteraciones se ha alcanzado o se ha logrado la convergencia, entonces para. En otro caso, ir al paso 2.

La complejidad computacional del algoritmo por lotes es $O(MND)$, es decir, lineal en el tamaño del conjunto de datos M, el número de neuronas N y la dimensión de la entrada D.

Para aplicar el modelo propuesto al modelo del fondo, se debe desarrollar una versión de aprendizaje en línea, ya que los datos de entrada vienen de los fotogramas de video capturados en tiempo real. Además, la complejidad computacional del algoritmo debe ser muy pequeña, ya que se debe ejecutar una optimización separada para cada píxel del fotograma. El método de optimización elegido por muchas aplicaciones de aprendizaje en línea del estado del arte es el gradiente estocástico descendiente, que debe ejecutarse en tiempo real debido a su extremedamente baja complejidad computacional (Chen et al., 2016a; Chakrabartty et al., 2013). La aplicación del método del gradiente estocástico descendiente al modelo conlleva a la siguiente regla de actualización de los prototipos:

$$\Delta \mathbf{w}_i = -\eta \frac{\partial \mathcal{E}}{\partial \mathbf{w}_i} =$$

$$\eta \left(\sum_{j=1}^{M} \mathbb{I}\left(\mathbf{x}_j \in C_i\right)\left(\mathbf{x}_j - \mathbf{w}_i\right) + \lambda \sum_{j=1}^{M} \left(\mathbf{x}_j - \mathbf{w}_i\right) \right) \tag{3.12}$$

donde η es una tasa de aprendizaje positiva.

Para el aprendizaje en línea se tiene únicamente una muestra de entrenamiento en cada tiempo, esto es, $M = 1$, por lo que:

$$\Delta \mathbf{w}_i = \eta \left(\mathbb{I}\left(\mathbf{x}\left(t\right) \in C_i\right)\left(\mathbf{x}\left(t\right) - \mathbf{w}_i\right) + \lambda \left(\mathbf{x}\left(t\right) - \mathbf{w}_i\right) \right) \tag{3.13}$$

$$\Delta \mathbf{w}_i = \eta \left(\mathbb{I}\left(\mathbf{x}\left(t\right) \in C_i\right) + \lambda \right)\left(\mathbf{x}\left(t\right) - \mathbf{w}_i\right) \tag{3.14}$$

Esto puede ser reescrito como:

$$\mathbf{w}_i\left(t+1\right) = \left(1 - \eta_i\left(t\right)\right)\mathbf{w}_i\left(t\right) + \eta_i\left(t\right)\mathbf{x}\left(t\right) \tag{3.15}$$

donde

$$\eta_i\left(t\right) = \begin{cases} \eta_{winner} & \text{if } i = Winner\left(\mathbf{x}\left(t\right)\right) \\ \eta_{loser} & \text{if } i \neq Winner\left(\mathbf{x}\left(t\right)\right) \end{cases} \tag{3.16}$$

$$\eta_{winner} = \eta\left(1 + \lambda\right) \tag{3.17}$$

$$\eta_{loser} = \eta\lambda \tag{3.18}$$

Nótese que se debe usar η_{winner} y η_{loser} como los parámetros del método en lugar de η y λ, donde se debe tener:

$$0 \leq \eta_{loser} \ll \eta_{winner} \leq 1 \tag{3.19}$$

El algoritmo de aprendizaje en línea es como sigue:

1. Inicializar cada prototipo $\mathbf{w}_i$ a una muestra de entrenamiento dibujada uniformemente aleatoria del conjunto de entrenamiento.

2. Dado una nueva muestra de entrenamiento $\mathbf{x}(t)$, calcular la neurona ganadora por (3.3).

3. Actualizar los prototipos por la aplicación de (3.15).

4. Si no hay más muestras de entrenamiento disponibles, entonces para. En otro caso, ir al paso 2.

La complejidad computacional de este algoritmo en línea es $O(ND)$, es decir, lineal con respecto al número de neuronas N y la dimensión de la entrada D.

Nótese que el modelo es un caso especial del algoritmo de mapas autoorganizados, donde el valor de la función de vecindad para la neurona ganadora es η_{winner} y un valor plano η_{loser} es utilizado para todas las demás neuronas en el mapa autoorganizado.

El modelo neuronal presentado anteriormente es llamado Aprendizaje Competitivo para Distribuciones de Entrada Variantes (Competitive Learning for Varying Input Distributions o CL-VID). Se puede utilizar para numerosas aplicaciones de aprendizaje no supervisado, pero en este trabajo se centra en el problema del modelado del fondo para los sistemas de visión por computador de videovigilancia. Las características de las aplicaciones de modelado de fondo son:

- El algoritmo de aprendizaje debe ser en línea, es decir, debería aceptar una muestra de entrada en cada paso de tiempo, que es el actual color del píxel observado.

- El algoritmo debe ser extremadamente rápido porque debe ejecutarse una vez por píxel para cada fotograma entrante en tiempo real.

- La distribución de entrada a ser aprendida no se espera que sea muy compleja, ya que consta del color observado para un solo píxel, por lo que la búsqueda del mínimo local debería ser apropiada.

La siguiente sección explica cómo emplear el modelo propuesto para este propósito.

3.4. Modelado del fondo

Ahora se aplica el modelo de red neuronal presentado en la sección anterior al problema de la detección de primer plano. Este problema consiste en separar el fondo y los objetos de primer plano que aparecen en una secuencia de video. Para este fin, un modelo de fondo puede ser aprendido en cada píxel. Un modelo de fondo tiene la intención de constituir las características de los colores del fondo observado en un píxel particular. Por lo tanto, deben ser actualizados cada vez que se toma un nuevo fotograma del video. Este es uno de los problemas del aprendizaje no supervisado, porque no hay un supervisor que proporcione información sobre qué píxeles corresponden con el fondo en un fotograma dado. Además, los colores del fondo pueden cambiar a lo largo del tiempo debido a cambios en las condiciones de iluminación, sombras, y otros eventos como el movimiento de los árboles, cambiando el tono general del fondo.

Para aprender el fondo de una escena, una red neuronal competitiva está asociada con cada píxel, es decir, aprende la información del color del fondo correspondiente a ese píxel. Luego es posible estimar si el píxel pertenece a primer plano evaluando cómo de bien el modelo de fondo representa el color del actual píxel observado. Esto se hace considerando el error de cuantificación $q_{n,\mathbf{r}} \in \mathbb{R}$ en el paso de tiempo n del píxel de las coordenadas del fotograma $\mathbf{r} \in \mathbb{Z}^2$, con un color observado actual $\mathbf{x}_{n,\mathbf{r}} \in \mathbb{R}^D$, representado por la red neuronal competitiva asociada con el píxel:

$$q_{n,\mathbf{r}} = \min_{i \in \{1,\ldots,N\}} \|\mathbf{x}_{n,\mathbf{r}} - \mathbf{w}_{i,n,\mathbf{r}}\| \tag{3.20}$$

donde $\mathbf{w}_{i,n,\mathbf{r}} \in \mathbb{R}^D$ se posiciona para el prototipo en el paso de tiempo n de la i-ésima neurona de la red neuronal competitiva asociada con el píxel con coordenadas $\mathbf{r}$. La imagen formada por los errores de cuantificación $q_{n,\mathbf{r}}$ es procesada con un filtro de la mediana de tamaño 5×5 píxeles para conseguir una imagen filtrada con elementos $\tilde{q}_{n,\mathbf{r}}$. Esto se hace para reducir el ruido en la imagen de cuantificación. Se declara que el píxel pertenece a un objeto de primer plano y si solo si $\tilde{q}_{n,\mathbf{r}}$ es superior a un umbral T:

$$\text{Píxel } \mathbf{r} \text{ pertenece a un objeto de primer plano} \Leftrightarrow \tilde{q}_{n,\mathbf{r}} > T \tag{3.21}$$

donde $T > 0$ es un parámetro configurable del sistema. La razón de este procedimiento es que los objetos de primer plano normalmente tienen un color que es diferente de los colores del fondo aprendidos por los prototipos.

La inicialización de la red neuronal competitiva correspondiente a cada píxel se ejecuta utilizando muestras de entrenamiento correspondientes a los colores observados en ese píxel durante los primeros 100 fotogramas de la secuencia de video.

Para una aplicación de detección de primer plano, a veces es aconsejable desactivar el proceso de aprendizaje para aquellos píxeles que son considerados de primer plano. De esta forma, los modelos de fondo de aquellos píxeles no están contaminados por el aprendizaje de aquellas muestras que no pertenecen al fondo. Para este fin, se define un parámetro L booleano del algoritmo que es configurado a *true* para indicar que el proceso de aprendizaje para los píxeles de primer plano es desactivado. Por tanto, para los experimentos de detección de primer plano mostrados en la Sección 3.5, la ecuación (3.15) es sustituida por la siguiente:

$$\mathbf{w}_i(t+1) =$$

$$\begin{cases} \mathbf{w}_i(t) & \text{si } L \wedge (q_{n,\mathbf{r}} > T) \\ (1-\eta_i(t))\,\mathbf{w}_i(t) + \eta_i(t)\,\mathbf{x}(t) & \text{en otro caso} \end{cases} \tag{3.22}$$

Bajo ciertas condiciones difíciles, es posible que los prototipos de un píxel permanezcan atascados en un conjunto de colores que corresponden con un objeto de primer plano previo. Esto puede ocurrir, por ejemplo, cuando la secuencia de video comienza con algunos objetos de primer plano y todos los prototipos de algunos píxeles son inicializados a colores de primer plano. Para prevenir este problema, se define un tiempo límite de Z fotogramas, que es un parámetro del algoritmo. Si un píxel ha sido declarado continuamente como primer plano para más de Z fotogramas, entonces todas sus neuronas son reinicializadas al color del píxel observado actualmente. Este procedimiento asegura que el modelo de fondo puede recuperarse de los problemas comentados anteriormente. El parámetro ha sido fijado a $Z = 1000$ en todos los experimentos, que ha ofrecido buenos resultados. En casos donde los objetos de primer plano permanecen quietos, un valor mayor de este parámetro Z podría producir un mejor rendimiento. Debe remarcarse que el valor del parámetro booleano L no tiene ningún efecto en este procedimiento. Es decir, la reinicialización del píxel se ejecuta independientemente del valor de L.

En términos de complejidad computacional, el método de modelado de fondo necesita $O(NDNumRowsNumCols)$ operaciones para cada fotograma de entrada, donde N es el número de neuronas de cada píxel, D es la dimensionalidad de la entrada ($D = 3$ en todos los experimentos), y el tamaño del fotograma del video es $NumRows \times NumCols$. Esto significa que es lineal con respecto al tamaño de las variables del problema.

3.5. Resultados experimentales

En esta sección se muestran los resultados de los experimentos computacionales que se han ejecutado. El rendimiento de la propuesta es compa-

rada contra varios métodos de segmentación de video del estado del arte. La Subsección 3.5.1 describe el software y el hardware que se han utilizado para implementar y ejecutar los métodos. Las secuencias de video que se han considerado se encuentran especificadas en 3.5.2. Después, la Subsección 3.5.3 especifica la variedad de parámetros configurables para las propuestas testeadas, mientras que los resultados obtenidos se muestran en términos cualitativos y cuantitativos en las Subsecciones 3.5.4 y 3.5.5, respectivamente. Por último, una comparativa usando las métricas y los métodos más similares de *ChangeDetection* se presentan en la Subsección 3.5.6.

3.5.1. Métodos comparadores

Con el objetivo de testear la idoneidad de la propuesta, se han considerado varios métodos bien conocidos de la literatura para las comparativas. La Tabla 3.1 muestra los rasgos clave que caracterizan a cada método competidor. El método más directo y estándar trata de modelar el fondo utilizando una distribución gaussiana (Wren et al., 1997), que es llamado *WrenGA*. También se consideran dos propuestas basadas en mixtura de gaussianas, llamadas *GrimsonGMM* (Stauffer y Grimson, 1999) y *ZivkovicGMM* (Zivkovic y van der Heijden, 2006). El algoritmo *ElgammalKDE* (Elgammal et al., 2000) es empleado ya que es uno de los métodos más citados basados en la estimación gaussiana de la densidad del núcleo.

También se han probado otros algoritmos no basados en distribuciones gaussianas. Uno de ellos es el método basado en redes neuronales artificiales llamado *MaddalenaSOBS* (Maddalena y Petrosino, 2008), que modela el fondo con una red neuronal autoorganizada. Otra propuesta es *OliverPCA* (Oliver et al., 2000), que construye el modelo del fondo aplicando análisis de componente principal. También se utilizan los métodos *ElBafFuzzy* (El Baf et al., 2008) (un método basado en las características del color y la textura) y *CucchiaraSakbot* (Cucchiara y Piccardi, 2003) (que modela el fondo con una mediana temporal de las intensidades del píxel). Para testear estos métodos se ha considerado su implementación de la versión 1.3.0 de la librería BGS.

Además, dos trabajos relacionados del grupo de investigación se han añadido para completar el estudio: el método *MFBM* (López-Rubio y López-Rubio, 2015), que está basado en la teoría de la aproximación estocástica, y el método *FSOM* (López-Rubio et al., 2011a), que modela la escena con una distribución uniforme para el primer plano y un modelo de mapa probabilístico autoorganizado para el fondo.

El método propuesto ha sido implementado en Matlab, con archivos MEX escritos en C++ para las partes que consumen más tiempo y código Matlab para el resto.

Una máscara de segmentación es obtenida para cada fotograma de la secuencia tras la aplicación del modelo de fondo. Esta máscara es binaria,

Nombre	Característica del modelo
WrenGA(Wren et al., 1997)	Una distribución gaussiana
GrimsonGMM(Stauffer y Grimson, 1999)	K distribuciones gaussianas
ZivkovicGMM(Zivkovic y van der Heijden, 2006)	Número no fijado de distribuciones gaussianas
ElgammalKDE(Elgammal et al., 2000)	Estimador no paramétrico de densidad del núcleo
MaddalenaSOBS(Maddalena y Petrosino, 2008)	Redes neuronales artificiales
OliverPCA(Oliver et al., 2000)	Análisis de componente principal
ElBafFuzzy(El Baf et al., 2008)	Características de color y textura sumado con la integral de Choquet
CucchiaraSakbot(Cucchiara y Piccardi, 2003)	Mediana temporal de las intensidades de los píxeles
MFBM(López-Rubio y López-Rubio, 2015)	Aproximación estocástica
FSOM(López-Rubio et al., 2011a)	Distribución uniforme y mapa probabilístico autoorganizado

Tabla 3.1: Resumen de las características clave utilizado por cada método.

donde los píxeles que pertenecen a primer plano están activados (valor 1) mientras que los píxeles pertenecientes al fondo no están activados (valor 0). Nótese que cada video ofrece un conjunto de datos de máscaras binarias ideales (llamadas máscaras de verdad, Ground Truth o GT), que es normalmente generado por una persona. Por tanto, es posible comparar las máscaras binarias obtenidas por los algoritmos con estas máscaras de verdad y obtener valores cuantitativos para medir el rendimiento de los métodos.

Varios métodos aplican una fase de postprocesado después del modelado del fondo para eliminar y limpiar la máscara segmentada de objetos espurios y píxeles aislados. Este paso mejora la salida del algoritmo e incrementa levemente la precisión de los resultados. Ya que el método FSOM incorpora postprocesado y el SOBS ejecuta un postprocesado implícito, esta estrategia ha sido añadida al resto de métodos para hacer la comparativa lo más justa posible. Los experimentos han sido ejecutados en un ordenador personal de 64 bits con un procesador i7 3,60 GHz de 8 núcleos, 32 GB RAM y hardware estándar. La implementación del método propuesto no necesita recursos de GPU, por lo que no requiere ningún hardware gráfico especial.

3.5.2. Secuencias

Se han utilizado varios videos para testear el rendimiento de los métodos comparadores. La variedad de secuencias cubre diferentes situaciones que suponen un desafío para el problema de la detección de primer plano, como

Método	Parámetros
MFBM	$\alpha = 0{,}01, F = [1\ 2\ 3]$
Propuesto CL-VID	$T = 0{,}01$, $N = 6$, $\eta_{winner} = 0{,}05$, $\eta_{loser} = 0{,}0001$, $L = true$
CucchiaraSakbot	$T = 25$, $F = 15$
ElBafFuzzy	$C = RGB$
ElgammalKDE	$T = 10^{-9}$, $S = 0{,}5$, $K = 70$
FSOM	$\alpha = 0{,}01$
GrimsonGMM	$T = 12$, $\alpha = 0{,}0025$, $K = 3$
MaddalenaSOBS	$s_1 = 75$, $s_0 = 245$ $\alpha_1 = 75$, $N = 100$
OliverPCA	$T = 250$, $N = 20$, $M = 10$
WrenGA	$T = 12$, $\alpha = 0{,}005$
ZivkovicGMM	$T = 30$, $\alpha = 0{,}001$, $K = 3$

Tabla 3.2: Configuraciones elegidas para cada método.

fondos dinámicos o cambios de iluminación. Todos los videos pueden encontrarse en los conjuntos de datos 2012 y 2014 de la página web de ChangeDetection.net[1], divididos en 6 y 11 categorías, respectivamente. Las categorías y las características básicas de cada una de ellas están descritas en su página web.

3.5.3. Selección de paramétros

Los valores elegidos de los parámetros han sido seleccionados de acuerdo a tres criterios: los valores recomendados en sus artículos originales, el valor por defecto que provee la librería BGS y valores adicionales que se han detectado en los experimentos que ofrecen buen rendimiento.

Para el método propuesto se ha seleccionado un rango de valores para los parámetros η_{winner}, η_{loser} y T. También se ha incluido el parámetro booleano L que indica si el proceso de aprendizaje para aquellos píxeles que son considerados pertenecientes a primer plano está activo. Además, se han probado diferentes valores para el número de neuronas con la mejor configuración para el resto de parámetros del método de acuerdo a la exactitud. Se ha hecho esto porque el número de posibles combinaciones de parámetros del algoritmo propuesto es bastante elevado. Finalmente, se han seleccionado aquellos valores que ofrecen un mejor rendimiento en algunas de las secuencias probadas.

La configuración elegida para cada método es mostrada en la Tabla 3.2.

[1] http://changedetection.net/

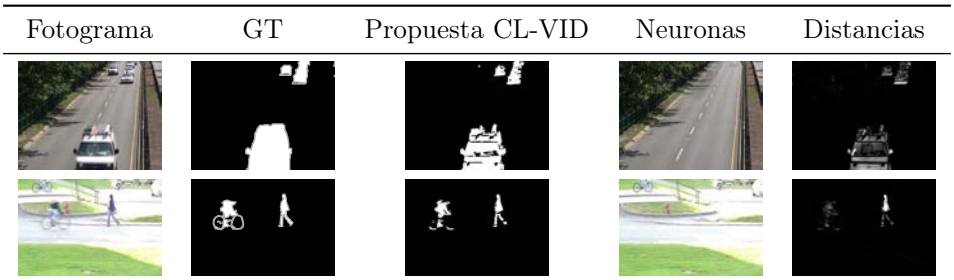


Figura 3.1: Representación gráfica del funcionamiento del método propuesto. Las columnas muestran, de izquierda a derecha: un fotograma de una secuencia, su máscara de verdad, la máscara resultante producida por el algoritmo (fue ejecutado con $N = 4$ neuronas), el valor mediano de los prototipos de las neuronas en ese momento y la distancia entre el píxel del fotograma y su neurona ganadora. La primera fila muestra el fotograma 1546 de Highway y la segunda fila se corresponde con el fotograma 473 de Pedestrians.

3.5.4. Resultados cualitativos

En esta subsección se muestran los resultados de una forma visual.

El funcionamiento de la propuesta es ilustrado en la Figura 3.1. En ella se muestra cómo el modelo aprende y cómo determina si un píxel pertenece al primer plano o al fondo. En la columna Neuronas se muestran los valores de los prototipos de las neuronas en un cierto momento. Estos prototipos, que representan los colores RGB, constituyen el modelo de fondo aprendido por la propuesta. Por otro lado, la columna Distancias representa la distancia entre el valor de cada píxel del fotograma y la neurona ganadora correspondiente a ese píxel. Estas distancias y el parámetro umbral se utilizan para determinar si un píxel pertenece al primer plano.

En la Figura 3.2 se pueden ver algunos resultados cualitativos que se han obtenido de nuestros experimentos. Cada columna se corresponde con un fotograma perteneciente a una secuencia probada. La primera fila es el fotograma original de un video, y la segunda fila se corresponde con la máscara de verdad de primer plano. Las siguientes filas son los resultados producidos por todos los métodos testeados.

Los métodos de detección de primer plano normalmente están afectados por problemas típicos. Uno de estos problemas es el cambio abrupto de iluminación en la escena. Este problema puede ser observado en los videos BusStation y PeopleInShade (categoría Shadow), columnas (d), (e) y (f) de la Figura 3.2.

Otro problema son los fondos dinámicos. Los métodos confunden los fondos que se mueven como parte del primer plano, como se puede ver en el video Canoe (categoría Dynamic background), Figura 3.2 columna (c).

Además, el problema de camuflaje también se puede observar en los re-

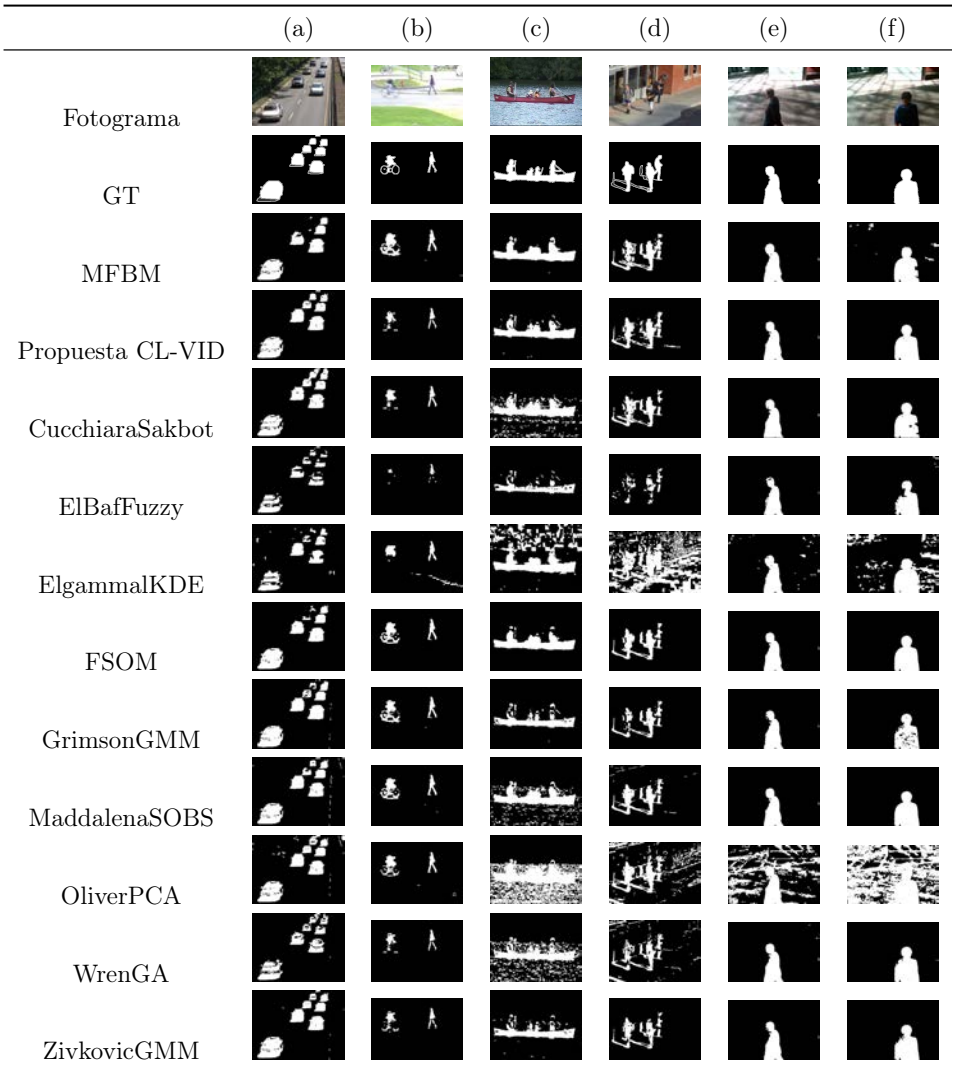


Figura 3.2: Resultados cualitativos para algunas escenas de referencia. De izquierda a derecha: fotograma 796 de Highway (a), fotograma 471 de Pedestrians (b), fotograma 960 de Canoe (c), fotograma 1011 de BusStation (d), y fotogramas 348 y 696 de PeopleInShade (e) y (f), respectivamente. La primera y segunda filas se corresponden con un fotograma original del video y la máscara de verdad. Las filas restantes son los resultados proporcionados por los métodos comparadores.

sultados. Este problema ocurre cuando el método no puede discriminar los píxeles cercanos del mismo color, de modo que algunos de ellos pertenecen al primer plano y otros al fondo, lo que provoca una segmentación errónea. Este problema se puede observar en todos los videos en mayor o menor medida pero el video Highway (categoría Baseline) es donde mejor se aprecia,

columna (a) en la Figura 3.2.

Por último, otra conocida dificultad ocurre cuando el método incorpora erróneamente sombras en el primer plano. Esto se conoce como el problema de las sombras proyectadas. Este problema es muy frecuente y puede ser fácilmente observado en el video BusStation, columna (d) de la Figura 3.2.

3.5.5. Resultados cuantitativos

Ahora se muestra una comparativa desde el punto de vista cuantitativo del método con otros métodos competidores del estado del arte.

Primero de todo debe señalarse que el problema del camuflaje puede producir dificultades en la medición de la exactitud de los métodos. Esto es porque hay píxeles que son difíciles de asignar al fondo o al primer plano, incluso para un humano. La máscara de verdad de todos los videos testeados tiene movimientos no sabidos, normalmente alrededor de los objetos en movimiento debido a la semi-transparencia y el desenfoque de movimiento. De acuerdo con esto, se han seguido las directrices proporcionadas por Change-Detection.net y no se han considerado estos píxeles para calcular el resultado del rendimiento. Los píxeles sombra de la máscara de verdad no son tenidos en cuenta.

Otro punto a resaltar es la selección de la medida exactitud. Ella se basa en cómo de bien un método detecta el primer plano y no el fondo. Esto es porque hay muchos fotogramas sin píxeles de primer plano. Esta medida se llama *exactitud espacial* (AC) y ha sido utilizada para evaluar otros algoritmos de detección de objetos de primer plano (Li et al., 2004; Toyama et al., 1999), y se define como sigue:

$$AC = \frac{\text{card}\,(A \cap B)}{\text{card}\,(A \cup B)} \tag{3.23}$$

donde *card* ofrece el número de elementos de un conjunto, A es el conjunto de todos los píxeles que pertenecen al primer plano (máscara de verdad), y B es el conjunto de todos los píxeles que son clasificados como primer plano por el método analizado. Además, se ha calculado otra medida de rendimiento muy popular, que es la F-medida, calculada como una combinación de la precisión y la exhaustividad. Nótese que cuanto mayor una de estas medidas mejor es el método.

El procedimiento de medición del rendimiento es como sigue. Para cada método se ha ejecutado para el video seleccionado, lo que produce un conjunto de imágenes binarias de salida, una imagen por cada fotograma del video. Luego se ha aplicado una función de postprocesado a cada imagen binaria. Tras esto, se calculan las medidas de rendimiento sobre las imágenes binarias postprocesadas, donde solo se consideran aquellos fotogramas cuya máscara de verdad presenta píxeles de primer plano. Las medidas de rendimiento son

calculadas considerando las sombras y los píxeles del contorno como píxeles del fondo. La medida de rendimiento general de un método para el video seleccionado será la media de todos los valores de rendimiento para los fotogramas considerados. Por último, se comparan los rendimientos de todos los métodos.

Se ha ejecutado el proceso descrito con el conjunto de datos 2012. La Tabla 3.3 muestra la exactitud mediana de los métodos testeados y sus categorías. Se puede observar que la propuesta es la que mejor rendimiento obtiene y la mejor en cinco categorías.

Por último, se ha ejecutado un test de significación estadística para comparar los dos mejores métodos de acuerdo a la Tabla 3.3 y algunos de los videos del conjunto de datos (Boats, Bungalows, WinterDriveway y Backdoor, que pertenecen a diferentes categorías). Primero se ha analizado si el rendimiento logrado por estos dos métodos para los fotogramas considerados son distribuidos de forma gaussiana. Los datos de rendimiento de cada método están compuestos por la exactitud de cada fotograma analizado de cada secuencia, que suma un total de 3414 fotogramas. El test de normalidad seleccionado ha sido el test de Lilliefors. El resultado de esta prueba es que ninguno de los conjuntos de datos procedentes de los dos métodos de mejor rendimiento se pueden suponer distribuidos por una gaussiana, con un nivel de significación del 5 %. Por lo tanto, se ha elegido el test no paramétrico de suma de rangos de Wilcoxon para comprobar la hipótesis nula de la igualdad de las medianas de los valores de rendimiento de cada método. El test rechazó la hipótesis nula de medianas iguales al 5 % de nivel de significación, con un p-valor de $3{,}9721 \times 10^{-30}$. Como consecuencia, el mejor método (la propuesta CL-VID) es juzgado de ser mejor que el segundo método (FSOM) con un alto nivel de significación estadístico.

Además, se ha estudiado el tiempo de ejecución por fotograma de cada método con varias secuencias. La propuesta no es el método más rápido pero en muchos casos obtiene uno de los mejores con respecto al resto de propuestas.

Otro aspecto importante de la propuesta es el parámetro del número de neuronas. Se ha estudiado este parámetro con la mejor configuración para cada video del primer conjunto de videos y es posible observar que el modelo funciona correctamente con un número reducido de neuronas. Por tanto, con 4 o 6 neuronas el modelo captura el fondo correctamente sin demasiada complejidad computacional.

3.5.6. Comparativa ChangeDetection

Por último, se ha comparado la propuesta con varios métodos de la web de ChangeDetection haciendo uso de las métricas que propone esta base de datos. Los métodos comparadores seleccionados han sido elegidos de acuerdo

Tabla 3.3: Resultados de exactitud de los métodos testeados para varias categorías de video. La primera columna especifica el algoritmo. La mediana de la exactitud obtenida para las categorías de video Baseline, Camera Jitter (CJ), Dynamic Background (DB), Intermittent Object Motion (IOM), Shadow y Thermal se muestran de la segunda a la séptima columna, respectivamente. La última columna muestra el valor mediano de las exactitudes medianas de las categorías. Los mejores resultados se resaltan en **negrita**.

Método	Baseline	CJ	DB	IOM	Shadow	Thermal	Total
MFBM	0,472	**0,559**	0,472	0,198	0,503	0,422	0,472
Propuesta CL-VID	**0,700**	0,434	**0,522**	**0,344**	**0,645**	**0,486**	**0,504**
CucchiaraSakbot	0,479	0,321	0,284	0,204	0,467	0,280	0,303
ElBafFuzzy	0,478	0,374	0,312	0,131	0,413	0,230	0,343
ElgammalKDE	0,555	0,447	0,207	0,267	0,414	0,477	0,430
FSOM	0,582	0,549	0,449	0,228	0,518	0,393	0,484
GrimsonGMM	0,442	0,441	0,401	0,223	0,438	0,281	0,420
MaddalenaSOBS	0,650	0,387	0,271	0,307	0,511	0,437	0,412
OliverPCA	0,454	0,195	0,089	0,270	0,405	0,454	0,337
WrenGA	0,452	0,349	0,270	0,178	0,422	0,276	0,313
ZivkovicGMM	0,504	0,417	0,375	0,231	0,474	0,330	0,396

a su similitud con el modelo propuesto. La comparativa emplea el conjunto de datos 2014 con respecto a la F-medida, y los resultados que se han obtenido pueden observarse en las Tablas 3.4 y 3.5. El ranking de acuerdo a los resultados obtenidos se muestra en las Tablas 3.6 y 3.7.

Se puede observar que el rendimiento logrado es similar a los otros métodos. La propuesta obtiene el mejor rendimiento en dos categorías y es la segunda en otras tres categorías. Se deben resaltar los resultados de la categoría Baseline, donde la propuesta obtiene el mejor rendimiento. También se debe indicar que el ranking medio no es el mejor poque es negativamente afectado por la categoría PTZ. Esto puede observarse en la columna (j) de la Tabla 3.6. Por tanto, esto afecta al rendimiento obtenido en la columna Ranking Medio. Sin embargo, el valor de la columna (j) no tiene ninguna influencia en la columna del Ranking Mediano de la propuesta.

3.6. Discusión

En esta sección se discuten varias características importantes de la propuesta.

Primero de todo, debe señalarse que el modelo propuesto podría ser obtenido como un caso especial del modelo de mapas autoorganizados. De manera similar, el aprendizaje competitivo estándar puede ser obtenido como un caso especial del modelo propuesto donde $\lambda = 0$ en la ecuación (3.1).

Método	(a)	(b)	(c)	(d)	(e)	(f)
Propuesta CL-VID	**0,937**	0,552	0,521	0,518	0,841	0,726
Distancia euclídea	0,872	0,508	0,487	0,489	0,679	0,631
GMM - Stauffer	0,825	0,633	0,597	0,521	0,737	0,662
GMM - Zivkovic	0,838	0,633	0,567	0,533	0,732	0,655
IUTIS-1	0,930	0,419	0,600	0,507	**0,849**	0,717
KDE - ElGammal	0,909	0,596	0,572	0,409	0,803	**0,742**
Distancia de Mahalanobis	0,464	0,180	0,336	0,229	0,335	0,138
SC_SOBS	0,933	**0,669**	0,705	**0,592**	0,779	0,692
SOBS_CF	0,930	0,652	**0,715**	0,581	0,772	0,714

Tabla 3.4: Resultados de la F-medida para los métodos testeados para diferentes categorías de videos del conjunto de datos 2014. La primera columna especifica el algoritmo comparado. La media de la F-medida obtenida para las categorías de video Baseline (a), Dynamic Background (b), Camera Jitter (c), Intermittent Object Motion (d), Shadow (e) y Thermal (f) se muestran de la segunda a la séptima columna, respectivamente. Los mejores resultados son resaltados en **negrita**.

Método	(g)	(h)	(i)	(j)	(k)	Total
Propuesta CL-VID	0,739	**0,584**	0,472	0,030	0,473	0,581
Distancia euclídea	0,670	0,502	0,386	0,040	0,414	0,516
GMM - Stauffer	0,738	0,537	0,410	**0,152**	0,466	0,571
GMM - Zivkovic	0,741	0,507	0,396	0,105	0,417	0,557
IUTIS-1	0,671	0,569	**0,477**	0,045	**0,583**	0,579
KDE - ElGammal	**0,757**	0,548	0,437	0,037	0,448	0,569
Distancia de Mahalanobis	0,221	0,080	0,137	0,037	0,336	0,227
SC_SOBS	0,662	0,546	0,450	0,041	0,488	**0,596**
SOBS_CF	0,637	0,515	0,448	0,037	0,470	0,588

Tabla 3.5: Resultados de la F-medida para los métodos testeados para diferentes categorías de videos del conjunto de datos 2014. La primera columna especifica el algoritmo comparado. La media de la F-medida obtenida para las categorías de video Bad Weather (g), Low Framerate (h), Night Videos (i), PTZ (j) y Turbulence (k) se muestran de la segunda a la sexta columna, respectivamente. Los mejores resultados son resaltados en **negrita**.

Método	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)
Propuesta CL-VID	1	6	7	5	2	2	3	1	2	9	3
Distancia euclídea	6	7	8	7	8	8	6	8	8	5	8
GMM - Stauffer	8	3	4	4	6	6	4	5	6	1	5
GMM - Zivkovic	7	4	6	3	7	7	2	7	7	2	7
IUTIS-1	3	8	3	6	1	3	5	2	1	3	1
KDE - ElGammal	5	5	5	8	3	1	1	3	5	6	6
Distancia de Mahalanobis	9	9	9	9	9	9	9	9	9	7	9
SC_SOBS	2	1	2	1	4	5	7	4	3	4	2
SOBS_CF	4	2	1	2	5	4	8	6	4	8	4

Tabla 3.6: Ranking de los resultados de la F-medida para los métodos testeados para diferentes categorías de videos del conjunto de datos 2014. La primera columna especifica el algoritmo comparado. El ranking del método de cada categoría se muestra en las columnas (a) a (k), donde cada columna representa: Baseline (a), Dynamic Background (b), Camera Jitter (c), Intermittent Object Motion (d), Shadow (e), Thermal (f), Bad Weather (g), Low Framerate (h), Night Videos (i), PTZ (j) y Turbulence (k). Los valores más bajos son mejores.

Método	Media	Rnk Medio	Mediana	Rnk Mediano
Propuesta CL-VID	3,73	3	3	1
Distancia euclídea	7,18	8	8	8
GMM - Stauffer	4,73	6	5	5
GMM - Zivkovic	5,36	7	7	7
IUTIS-1	3,27	2	3	1
KDE - ElGammal	4,36	4	5	5
Distancia de Mahalanobis	8,82	9	9	9
SC_SOBS	3,18	1	3	1
SOBS_CF	4,36	4	4	4

Tabla 3.7: Ranking de los resultados de la F-medida para los métodos testeados para diferentes categorías de videos del conjunto de datos 2014. La primera columna especifica el algoritmo comparado. La media y la mediana del ranking sobre las categorías de cada método se muestra en las columnas Media y Mediana, respectivamente. Por último, las columnas Rnk Medio y Rnk Mediano muestran el ranking de acuerdo a las columnas Media y Mediana, respectivamente. Los valores más bajos son mejores.

Sin embargo, los tres modelos son esencialmente diferentes, tal y como se argumenta a continuación.

La novedad clave de la propuesta es que los prototipos de todas las neuronas que no ganan son movidos en la misma magnitud hacia la actual muestra de entrada. Esto difiere del aprendizaje competitivo estándar, donde las neuronas que no ganan no aprenden, lo que a menudo produce neuronas muertas, esto es, neuronas que no representan ninguna parte significativa de la distribución de entrada. Por otro lado, la esencia del mapa autoorganizado es, como su nombre indica, la construcción de un mapa computacional donde las neuronas están cercanas entre sí en el mapa de cuadrícula tienen prototipos similares. Autoorganización significa que las interacciones locales gobernadas por una regla local pueden conducirse a un orden global. En el caso del mapa autoorganizado, la autoorganización es implementada como refuerzo de cooperación entre las neuronas cercanas en el mapa de cuadrícula, es decir, aparece un orden global del mapa. Por otro lado, la propuesta no muestra autoorganización en ningún sentido, ya que la tasa de aprendizaje es la misma para todas las neuronas no ganadoras. En otras palabras, el mapa autoorganizado no está destinado a tener la misma tasa de aprendizaje para todas las neuronas no ganadoras porque esto vencería su propósito, es decir, la autoorganización. Esto ha sido conocido desde el inicio de los mapas autoorganizados (Kohonen, 1990). Además, como se ha hecho explícito en la literatura, la función de vecindad nunca debe ser plana (Kohonen, 2013). Los mapas autoorganizados nunca se usan con dicha configuración de parámetros en la práctica, lo que significa que nada equivalente al modelo propuesto ha sido intentado anteriormente. La propuesta tiene un objetivo completamente diferente, nombrado como la recuperación automática de neuronas muertas. Resumiendo, podría decirse que los tres modelos tienen las siguientes características fundamentales:

- El aprendizaje competitivo estándar pretende la minimización del error de cuantificación, sin ninguna cooperación entre las neuronas. No intenta evitar las neuronas muertas.
- El modelo propuesto combina el error de cuantificación con la evitación de las neuronas muertas, para lograr libros de códigos más balanceados donde todas las neuronas representan una parte significativa de la distribución de entrada.
- El mapa autoorganizado está diseñado para construir mapas donde la conectividad en el mapa cuadriculado corresponde a la proximidad del prototipo en el espacio de entrada.

Las redes neuronales no supervisadas ya han sido utilizadas para modelar el fondo de secuencias de video. En particular, los mapas autoorganizados han sido propuestos previamente en la literatura (Maddalena y Petrosino,

2008; López-Rubio et al., 2011a). El aprendizaje competitivo estándar puede ser visto como un caso especial de mapa autoorganizado, donde la topología que conecta las neuronas ha sido eliminada. Esta ausencia de cooperación entre las neuronas no ayuda porque algunas de ellas podrían dejar de representar algunas muestras de la entrada, es decir, ellas podrían morir. Este hecho es particularmente apremiante en el contexto del modelado del fondo, ya que la distribución de entrada puede dejarse llevar por los fotogramas del video. Sin embargo, la propuesta parte de un aprendizaje competitivo estándar en el que todas las neuronas son arrastradas un poco hacia el actual centro de la distribución de entrada. En este sentido las neuronas no pueden permanecer lejos de los datos de entrada, lo que evita el problema de las neuronas muertas. Consecuentemente, la propuesta puede ser vista como una alternativa a los mapas autoorganizados ya que solventa el problema de las neuronas muertas de manera diferente, no imponiendo una topología fija a las neuronas, que pueden ayudar a mejorar la flexibilidad de la red para adaptarse a la distribución de entrada. También es interesante resaltar que los modelos de mapa autoorganizado que intentan mejorar la distribución de los prototipos como el ViSOM (Yin, 2002) no son factibles para el problema de la detección del primer plano debido a su excesiva complejidad computacional. Esto es provocado por dos factores: por un lado, las ecuaciones de actualización de los prototipos son más complejas; y por otro lado, el número de neuronas que requiere para explotar completamente las capacidades del modelo es muy alto.

El modelo propuesto muestra algunas similitudes con el método propuesto por Kim et al. 2005. En ese trabajo, los autores proponen construir un libro de códigos que representa la distribución del color en un píxel dado. Sin embargo, no incluyen ninguna provisión para paliar el problema de las neuronas muertas comentado anteriormente. Los prototipos que no han ganado la competición durante un largo tiempo simplemente son considerados como colores del primer plano e ignorados, lo que implica un desperdicio de la información que ellos llevan. Podría ser el caso de que solo un prototipo está vivo, mientras que todos los demás permanecen congelados en estados de color obsoletos, sin sentido. La estrategia de la propuesta está diseñada para mantener todas las neuronas vivas, es decir, su información es actualizada y útil todo el tiempo.

De acuerdo con los resultados obtenidos en la Sección 3.5, se pueden destacar algunas fortalezas y debilidades del método propuesto. El resultado obtenido más interesante es que el método propuesto tiene mejor rendimiento que el resto de métodos probados en los videos de la categoría Baseline, por lo que la propuesta gestiona correctamente situaciones sin problemas especiales. Además, el rendimiento en las secuencias de la categoría Shadow es un poco superior al resto de métodos desarrollados por el grupo de investigación, con un alto nivel de significación estadística. Por tanto, la propuesta gestiona

adecuadamente los problemas de sombras de los objetos de primer plano. La propuesta incluso logra un alto rendimiento en la categoría Thermal. En general, el método CL-VID consigue un rendimiento muy competitivo en general en todo tipo de escenas, información que puede ser observada en las Tablas 3.3, 3.4 y 3.5.

3.7. Conclusiones

Un nuevo modelo de aprendizaje no supervisado ha sido presentado, que es capaz de adaptarse a las distribuciones de entrada que varían a lo largo del tiempo. Ha sido integrado en un sistema de visión por computador para la detección de objetos de primer plano en secuencias de video. De acuerdo a los resultados cualitativos y cuantitativos ofrecidos, el modelo de aprendizaje propuesto ha sido satisfactoriamente aplicado a distribuciones no estables, que son comunes en las aplicaciones de visión por computador. En este sentido, un modelo neuronal con dos mecanismos de aprendizaje ha sido desarrollado, uno de ellos para seguir el flujo general de la distribución de entrada y el otro para ajustar las neuronas a grupos específicos de la entrada.

Se han mostrado resultados experimentales en detalle, utilizando diferentes conjuntos de videos y comparando con métodos del estado del arte desde un punto de vista cualitativo y cuantitativo. La propuesta ha obtenido buenos resultados en una comparativa con varios algoritmos del estado del arte para el problema del modelado del fondo, testeado sobre un conjunto de videos que pertenecen a uno de los más recientes y usados estándares de comparación público. En particular, los resultados han demostrado la efectividad y robustez de la propuesta de aprendizaje competitivo para gestionar situaciones sin problemas especiales y apariciones de sombras.

Capítulo 4

Detección de primer plano basada en la estimación de los estados de iluminación

Conforme la complejidad de un sistema aumenta, nuestra capacidad para ser precisos y construir instrucciones sobre su comportamiento disminuye hasta el umbral más allá del cual, la precisión y el significado son características excluyentes.

Lofti A. Zadeh

Resumen: En este capítulo se propone un modelo probabilístico basado en mapas autoorganizados combinado con un modelo de lógica difusa para la detección de objetos en primer plano en secuencias de video. El método consiste en dos partes diferenciadas. Primero, se aplica un modelo probabilístico basado en mapas autoorganizados, donde un conjunto de características de los píxeles es seleccionado. En esta etapa, se usa una distribución uniforme para representar el primer plano. Después, la salida de este modelo es aplicada a un subsistema de lógica difusa, que estima los estados de iluminación de los píxeles. Los resultados experimentales demuestran el buen rendimiento de la propuesta comparado con otros métodos competidores.

4.1. Introducción

La detección de objetos en primer plano es un problema clave en el diseño de sistemas de visión por computador. Los algoritmos que resuelven este

problema deben gestionar numerosas dificultades que surgen en los videos de la vida real. Estos inconvenientes incluyen cambios de iluminación, aparición de sombras en el primer plano como consecuencia de objetos luminosos en el fondo o el movimiento repetitivo de objetos en el fondo de la escena (olas del mar, ramas de árboles...), además de muchos otros.

En las aplicaciones de visión por computador, como videovigilancia o análisis del movimiento humano, la capacidad de extraer objetos de interés de una secuencia de video es una tarea preliminar crucial. Videovigilancia es una tecnología clave para la seguridad pública (Geronimo et al., 2010; Räty, 2010), gestión eficiente en redes de transporte y servicios públicos (Luque-Baena et al., 2015b; Kamijo et al., 2000; Cheng y Hsu, 2011), reconocimiento de patrones de comportamiento humano (Turaga et al., 2008; Oliver et al., 1999), etc.

Un supuesto típico de este tipo de aplicaciones es que los fotogramas de una escena sin objetos de interés muestran algunos comportamientos normales, que pueden ser bien descritos por un modelo estadístico del fondo. Una propuesta para discriminar objetos de interés del fondo es la detección mediante procesos de sustracción del fondo, que consiste en restar la imagen actual del modelo de fondo de referencia.

Sin embargo, el modelado del fondo supone un reto en las aplicaciones reales como consecuencia de los inconvenientes anteriormente comentados. Por tanto, estos problemas no pueden ser resueltos por simples y estáticos modelos de fondo. Por ejemplo, el fondo de una escena general puede ser modelado utilizando una distribución gaussiana (Wren et al., 1997) o con una red neuronal autoorganizada (Maddalena y Petrosino, 2008). Así, tal y como se ha visto en la Subsección 3.2 en el Capítulo 3, hay varias propuestas en la literatura que modelan el fondo de una secuencia de video, empleando diferentes técnicas como mixturas de gaussianas o redes neuronales probabilísticas.

Tradicionalmente, los métodos de detección del primer plano no generan una máscara de segmentación limpia y fiable, y normalmente requieren de técnicas de post-procesado que corrigen y mejoran la máscara inicial obtenida. El objetivo es filtrar los valores espurios que aparecen en la máscara de primer plano, basándose principalmente en la vecindad y similitud de los píxeles adyacentes. Los operadores morfológicos son considerados como la técnica más popular (Parks y Fels, 2008), aunque hay otras propuestas basadas en la aplicación de filtros temporales, que analizan la salida de cada región o bloque todo el tiempo, o en las redes de Hopfield (Luque et al., 2008b). La aplicación de sistemas basados en reglas difusas también han sido consideradas para el modelado del fondo en numerosas ocasiones, principalmente como una técnica que complementa otras estándar (por ejemplo una mixtura de gaussianas) y que tras su combinación, el resultado final mejora (Zhao et al., 2012; Baf et al., 2008).

En este capítulo se presenta un modelo basado en mapas autoorganizados probabilísticos que incorporan una adecuada variedad de características de los píxeles. Este mapa autoorganizado con rasgos para la detección de primer plano (*Featured Foreground Self Organizing Map* o *FFSOM*) está combinado con un sistema basado en reglas difusas que mejora la máscara de salida de primer plano considerando los cambios de iluminación de la escena global.

El resto del capítulo está estructurado como sigue. La metodología de la propuesta se describe en la Sección 4.2. Los resultados experimentaless se muestran en la Sección 4.3. Finalmente, las conclusiones se presentan en la Sección 4.4.

4.2. Metodología

El sistema de detección del primer plano propuesto (FFSOM) tiene dos partes. Primero se aplica un modelo base (Subsección 4.2.1), que está basado en mapas autoorganizados probabilísticos. Después, la salida del modelo base es combinada con otros datos y proporcionada al subsistema de lógica difusa (Subsección 4.2.2), que estima los estados de iluminación de los píxeles para mejorar el rendimiento de la detección de primer plano.

4.2.1. Modelo de referencia

El sistema de detección de primer plano primero calcula los valores de D características (o rasgos) de cada píxel de un fotograma entrante con tamaño $NumRows \times NumCols$ píxeles. El conjunto de características adecuadas que se han considerado se presentan en (López-Rubio y López-Rubio, 2015). Tras esto, el vector de características $\mathbf{t} \in \mathbb{R}^D$ para la posición del píxel $\mathbf{x} \in \{1, ..., NumRows\} \times \{1, ..., NumCols\}$ es proporcionado como muestra de entrada a un algoritmo de aprendizaje para adaptar los parámetros a una distribución de mixturas probabilísticas con dos componentes de mixturas ($Back$ para el fondo y $Fore$ para el primer plano):

$$p_{\mathbf{x}}(\mathbf{t}) = \pi_{Back,\mathbf{x}} p_{\mathbf{x}}(\mathbf{t} \mid Back) + \pi_{Fore,\mathbf{x}} p_{\mathbf{x}}(\mathbf{t} \mid Fore) \tag{4.1}$$

Los valores de primer plano del vector de características son modelados por una distribución uniforme sobre el espacio de todos los posibles vectores de características, por lo que un objeto entrante de primer plano puede ser representado igualmente bien:

$$p_{\mathbf{x}}(\mathbf{t} \mid Fore) = U(\mathbf{t}) \tag{4.2}$$

$$U(\mathbf{t}) = \begin{cases} 1/Vol(\mathcal{S}) & \text{iff } \mathbf{t} \in \mathcal{S} \\ 0 & \text{iff } \mathbf{t} \notin \mathcal{S} \end{cases} \tag{4.3}$$

donde $\mathcal{S}$ es el soporte de la distribución uniforme y $Vol\left(\mathcal{S}\right)$ es el volumen D-dimensional de $\mathcal{S}$. La distribución de los valores del fondo del vector de características es representado mediante medias de un mapa probabilístico autoorganizado:

$$p_{\mathbf{x}}\left(\mathbf{t} \mid Back\right) = \frac{1}{H}\sum_{i=1}^{H} p_{\mathbf{x}}\left(\mathbf{t} \mid i\right) \tag{4.4}$$

donde H es el número de componentes de mixtura (unidades) del mapa autoorganizado, y las probabilidades a priori o proporciones mezcladas son supuestas a ser iguales. Más detalles sobre el algoritmo de aprendizaje para la anterior mixtura definida son proporcionados en (López-Rubio et al., 2011a). La probabilidad bayesiana de que la muestra observada (valor del vector de características) $\mathbf{t}$ sea primer plano es dada por

$$R_{Fore,\mathbf{x}}\left(\mathbf{t}\right) = \frac{\pi_{Fore,\mathbf{x}} p_{\mathbf{x}}\left(\mathbf{t} \mid Fore\right)}{\pi_{Back,\mathbf{x}} p_{\mathbf{x}}\left(\mathbf{t} \mid Back\right) + \pi_{Fore,\mathbf{x}} p_{\mathbf{x}}\left(\mathbf{t} \mid Fore\right)} \tag{4.5}$$

Sin embargo, $R_{Fore,\mathbf{x}}\left(\mathbf{t}\right)$ es propenso a ruido debido a píxeles aislados que cambian sus características aleatoriamente. Las correlaciones de Pearson $\rho_{\mathbf{x},\mathbf{y}}$ permiten obtener una version con menos ruido de $R_{Fore,\mathbf{x}}\left(\mathbf{t}\right)$ combinado con la información de los 8-vecinos $\mathbf{y}$ de $\mathbf{x}$:

$$\widetilde{R}_{Fore,\mathbf{x}}\left(\mathbf{t}\right) = \mathrm{trunc}\left(\frac{1}{9}\sum_{\mathbf{y}\in Neigh(\mathbf{x})} \rho_{\mathbf{x},\mathbf{y}} R_{Fore,\mathbf{y}}\left(\mathbf{t}\right)\right) \tag{4.6}$$

donde $Neigh\left(\mathbf{x}\right)$ contiene el píxel $\mathbf{x}$ y sus 8-vecinos $\mathbf{y}$.

4.2.2. Modelo difuso

Aquí se propone un modelo de lógica difusa del estado de iluminación de cada píxel del actual fotograma del video. Hay tres variables de entrada: *Rugosidad* (*Rough*), *Diferencia* (*Difference*) y *Referencia* (*Baseline*); sus valores serán nombrados como $\alpha_i, \beta_i, \gamma_i \in [0, 1]$, respectivamente. La interpretación de las variables es la siguiente:

- *Rugosidad* indica si la transformación de los colores en los fotogramas previos a los colores del actual fotograma no es suave en la vecindad del píxel i. Si α_i es alto, entonces es improbable que un cambio de iluminación esté ocurriendo, ya que los cambios de iluminación producen cambios suaves en los colores del fondo y de los objetos del primer plano.

- *Diferencia* indica si el color del píxel i en el actual fotograma es muy diferente con respecto al fotograma anterior para el píxel i. Si β_i es

alto, entonces o un cambio de iluminación está ocurriendo o un objeto de primer plano está presente.

- *Referencia* indica si un modelo de fondo tomado como referencia (Sección 4.2.1) ha detectado un objeto de primer plano en el píxel i. En este sentido $\gamma_i \in \{0, 1\}$ indica la probabilidad o seguridad de que el píxel i sea considerado como primer plano.

Los valores de las variables de entrada *Rugosidad* y *Diferencia* son calculadas como se indica en (López-Rubio y López-Rubio, 2015). Las tres variables de entrada son transformadas al espacio difuso mediante unas funciones trapezoidales de pertenencia. Una transformación trapezoidal de pertenencia μ se define como sigue:

$$\mu\left(\delta, [a, b, c, d]\right) = max\left(min\left(\frac{\delta - a}{b - a}, 1, \frac{d - \delta}{d - c}\right), 0\right) \tag{4.7}$$

donde $\delta \in [0, 1]$, a es el límite inferior, d es el límite superior, b es el límite de soporte inferior, c es el límite de soporte superior, y $a \leq b \leq c \leq d$. Las funciones trapezoidales de pertenencia que se han definido tienen los siguientes parámetros, $[a, b, c, d]$:

- Muy bajo: $[0, 0, 0{,}15, 0{,}20]$.
- Bajo: $[0{,}15, 0{,}20, 0{,}30, 0{,}35]$.
- Medio: $[0{,}30, 0{,}35, 0{,}65, 0{,}70]$.
- Alto: $[0{,}65, 0{,}70, 0{,}80, 0{,}85]$.
- Muy alto: $[0{,}80, 0{,}85, 1, 1]$.

Las anteriores funciones de pertenencia son descritas en la Figura 4.1, que muestra el grado de pertenencia para la variable de entrada *Referencia*. Las otras variables de entrada (*Diferencia* y *Rugosidad*) tienen las mismas funciones trapezoidales de pertenencia.

Tras esto, las siguientes reglas difusas son aplicadas:

- IF *Referencia* es muy alto THEN el píxel pertenece al primer plano.
- IF *Referencia* es alto AND (NOT (*Diferencia* es bajo OR *Diferencia* es muy bajo)) AND (NOT (*Rugosidad* es bajo OR *Rugosidad* es muy bajo)) THEN el píxel pertenece al primer plano.
- IF (*Referencia* es bajo OR *Referencia* es muy bajo) AND (*Diferencia* es alto OR *Diferencia* es muy alto) AND (*Rugosidad* es alto OR *Rugosidad* es muy alto) THEN el píxel pertenece al primer plano.

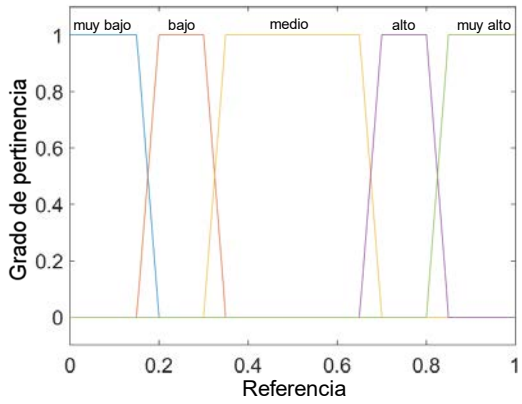


Figura 4.1: Grado de pertenencia de las entradas al sistema. La imagen muestra la pertenencia de una entrada (en este caso es Referencia (Baseline)).

Por último los valores de verdad difusos para el primer plano son transformados del espacio difuso a un número real en el intervalo $[0, 1]$, que es interpretado como la probabilidad de que un píxel pertenezca al primer plano. Nótese que si ninguna regla es satisfecha entonces el píxel es asociado al fondo.

Con todas estas consideraciones se ha definido un sistema de inferencia difusa de Mamdani y su método de transformación del espacio difuso para calcular los valores de salida es el centroide del área bajo el conjunto de salida difuso.

Después de la aplicación del modelo difuso, se aplica un umbral para hacer binaria la salida del modelo difuso. Después se aplica un postprocesado adicional para eliminar píxeles espúreos que pueden aparecer en la salida resultante. Consiste en rellenar huecos, es decir, píxeles de fondo aislados, y eliminar objetos de primer plano que tienen un área menor que un parámetro especificado previamente.

Como el método propuesto emplea un mapa autoorganizado con H unidades en cada píxel, su complejidad computacional es $O(H \cdot NumRows \cdot NumCols)$, donde $NumRows \times NumCols$ es el tamaño del fotograma entrante en píxeles.

4.3. Resultados experimentales

En esta sección se presenta el rendimiento de la detección de primer plano de la propuesta, además de una comparación con otros algoritmos del estado del arte.

Tabla 4.1: Conjunto de videos empleados para ejecutar los experimentos.

Nombre	Categoría	Fotograma	Tamaño
office	Baseline	2050	360x240
pedestrians	Baseline	1099	360x240
abandonedBox	Intermittent Object Motion	4500	432x288
parking	Intermittent Object Motion	2500	320x240
busStation	Shadow	1250	360x240
cubicle	Shadow	7400	352x240
tunnelExit_0_35fps	Low Framerate	4000	700x440
turnpike_0_5fps	Low Framerate	1500	320x240
busyBoulvard	Night Videos	2760	640x364
streetCornerAtNight	Night Videos	5200	595x245

4.3.1. Métodos

El método FFSOM está basado en el método de detección de objetos FSOM (López-Rubio et al., 2011a), que fue desarrollado previamente por el grupo de investigación y que está incluido en las comparativas. El código de este método puede ser descargado gratuitamente en su página web [1].

Además, se han seleccionado algunos métodos de referencia de tipo SOM de la literatura, notados como SOBS (Maddalena y Petrosino, 2008) y SC-SOBS (Maddalena y Petrosino, 2012), donde el segundo también utiliza un esquema difuso.

Por otro lado, se han seleccionado otros tipos de algoritmos para completar la comparativa. El primero es el algoritmo nombrado como GA (Wren et al., 1997), que modela cada píxel usando una gaussiana. Otro método gaussiano es el método GMM (Stauffer y Grimson, 1999), que usa dos mixturas de gaussianas. Además, se ha seleccionado el método PAWCS (St-Charles et al., 2015a), que está basado en palabras sin agrupamiento. Por último, el algoritmo SuBSENSE (St-Charles et al., 2015b) también se incluye en la comparativa, que se basa en un modelo estadístico no paramétrico basado en muestras.

Por tanto, los métodos se han organizado en dos grupos diferentes: métodos de tipo SOM y métodos de otro tipo. Las principales características de todos los métodos se muestran en la Tabla 4.2 y la implementación de estos métodos se ha tomado de la librería BGS versión 1.3.0, que se encuentra disponible en su página web [2].

Ya que el método FFSOM y el método FSOM incluyen un postprocesado

[1] http://www.lcc.uma.es/ %7Eezeqlr/fsom/fsom.html
[2] https://github.com/andrewssobral/bgslibrary

Tabla 4.2: Valores considerados de los parámetros para los métodos competidores, formando el conjunto de configuraciones experimentales.

Método	Parámetros
FFSOM	Características, $F = [1\ 19\ 20]$ Tamaño de paso, $\alpha = 0{,}01$ Número de neuronas, $N = 12$
FSOM (López-Rubio et al., 2011a)	Tamaño de paso, $\alpha = 0{,}01$ Número de neuronas, $N = 12$
SOBS (Maddalena y Petrosino, 2008)	Sensibilidad, $s_1 = 75$ Sensibilidad de entrenamiento, $s_0 = 245$ Tasa de aprendizaje, $\alpha_1 = 75$ Paso de entrenamiento, $N = 100$
SC-SOBS (Maddalena y Petrosino, 2012)	(raíz cuadrada de) Número de vectores de peso, $n = 3$ Umbral distancia e1 para la fase de entrenamiento, $e1 = 1{,}0$ Umbral distancia e2 para la fase de test, $e2 = 0{,}008$ Tasa de aprendizaje c1 para la fase de entrenamiento, $c1 = 1{,}0$ Tasa de aprendizaje c2 para la fase de test, $c2 = 0{,}05$
GA (Wren et al., 1997)	Umbral, $T = 12$ Tasa de aprendizaje, $\alpha = 0{,}005$
GMM (Stauffer y Grimson, 1999)	Umbral, $T = 12$ Tasa de aprendizaje, $\alpha = 0{,}0025$ Número de gaussianas, $K = 3$
PAWCS (St-Charles et al., 2015a)	Desplazamiento del umbral de distancia del descriptor absoluto, $ddto = 2$ Umbral de distancia de color mínimo absoluto, $mcdt = 20$ Palabras locales para construir submodelos de fondo, $mlw = 50$ Palabras globales para construir el modelo de fondo global, $mgw = 50$ Muestras para calcular la tasa de aprendizaje de medias móviles, $sma = 100$
SuBSENSE (St-Charles et al., 2015b)	Desplazamiento del umbral de distancia del descriptor absoluto, $ddto = 3$ Umbral de distancia de color mínimo absoluto, $mcdt = 30$ Muestras para construir el modelo de fondo, $bs = 50$ Muestras necesarias para considerar un píxel/bloque como fondo, $rbs = 2$ Muestras para calcular la tasa de aprendizaje de medias móviles, $sma = 100$

y el método SOBS tiene un postproceso implícito, se ha añadido un postprocesado a todos los métodos restantes para hacer la comparativa lo más justa posible.

Los experimentos que se muestran se han ejecutado en un ordenador personal de 64 bits con un procesador Intel i7 3,60 GHz de ocho núcleos, 32 GB RAM y hardware convencional. La implementación del método propuesto no utiliza recursos de GPU, por lo que no necesita un hardware gráfico específico.

Figura 4.2: Resultados cualitativos para algunas escenas utilizadas. De izquierda a derecha: fotograma 1040 de busStation (a), fotograma 2019 de office (b), fotograma 1783 de parking (c) y fotograma 644 de pedestrians (d), respectivamente. La primera y segunda filas corresponden al video original y al ground truth. Las filas siguientes son los resultados de los métodos comparadores.

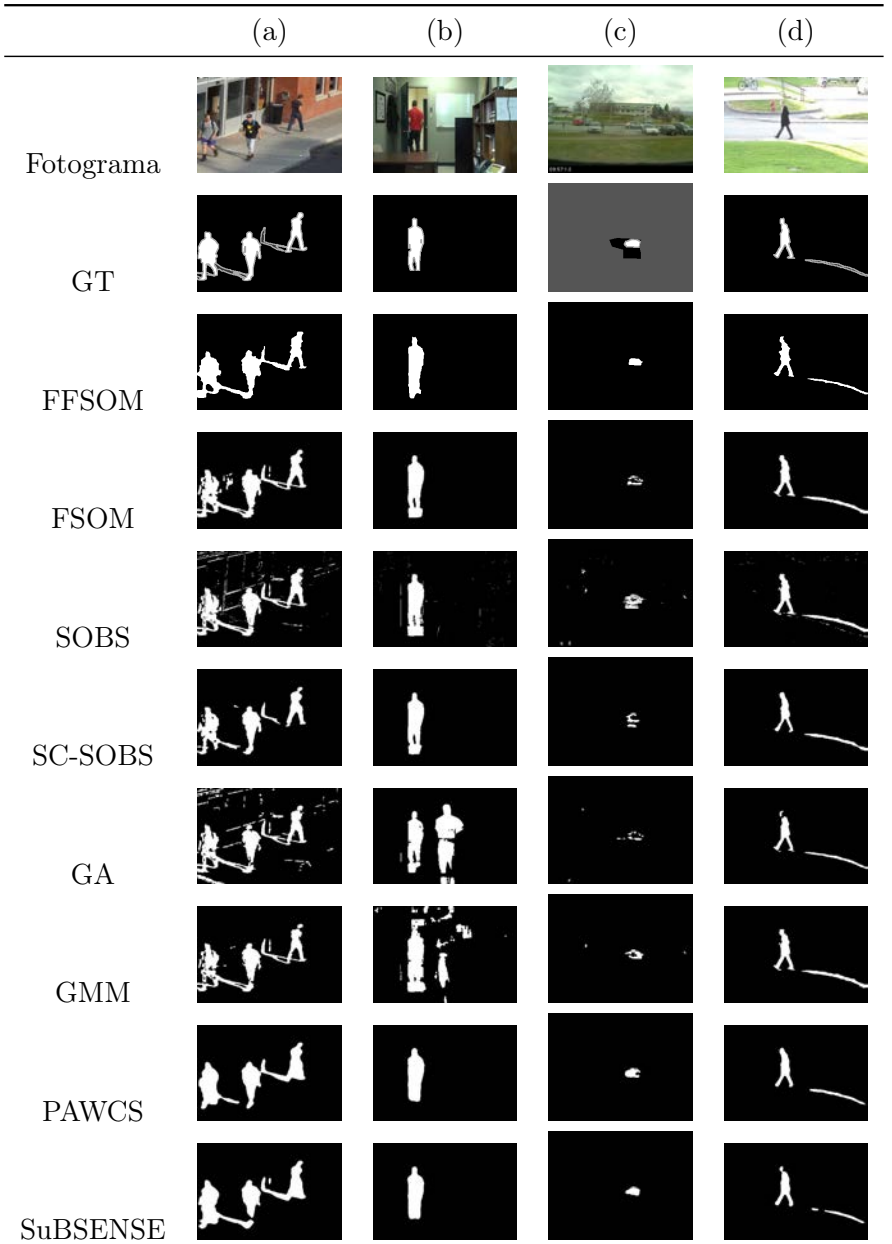


Tabla 4.3: Resultados de exactitud (el mayor es el mejor). Cada columna corresponde a un método y la fila indica el video. Cada celda muestra la media de la exactitud. Los mejores resultados de los métodos de tipo SOM están remarcados en **negrita** y los mejores resultados de los algoritmos de otro tipo están remarcados en *cursiva*.

	Tipo SOM				Otro tipo			
Secuencia	FFSOM	FSOM	SOBS	SC_SOBS	GA	GMM	PAWCS	SuBSENSE
office	0,608	0,542	0,685	**0,781**	0,356	0,290	0,722	*0,759*
pedestrians	0,514	0,504	0,431	**0,580**	*0,574*	0,483	0,537	*0,574*
abandonedBox	0,182	0,188	0,270	**0,308**	0,150	0,142	*0,606*	0,552
parking	**0,482**	0,387	0,299	0,238	0,203	0,313	*0,466*	0,264
busStation	**0,606**	0,573	0,427	0,542	0,397	0,452	*0,581*	0,555
cubicle	0,423	0,423	0,293	**0,471**	0,356	0,453	0,541	*0,597*
tunnelExit	0,339	0,317	0,311	**0,420**	0,200	0,287	0,427	*0,528*
turnpike	0,524	0,600	0,546	**0,637**	0,542	0,591	*0,686*	0,637
busyBoulvard	**0,253**	0,216	0,234	0,238	0,190	0,222	0,242	*0,258*
streetCornerAtNight	**0,261**	0,215	0,167	0,218	0,213	0,163	0,345	*0,356*
Mediana	**0,453**	0,405	0,305	0,446	0,285	0,301	0,539	*0,554*

4.3.2. Secuencias

Un conjunto de videos han sido seleccionados del conjunto de datos 2014 de la página web de ChangeDetection.net website[3]. Se han seleccionado dos videos de diferentes categorías de este conjunto de datos: Baseline (videos con condiciones ideales), Intermittent Object Motion (secuencias con objetos de fondo parándose y moviéndose), Shadow (videos con sombras), Low Framerate (videos grabados con tasa de fotogramas baja) y Night Videos (secuencias grabadas de noche). Los videos seleccionados se muestran en la Tabla 4.1.

4.3.3. Selección de parámetros

Se han definido un conjunto de valores fijos para los parámetros de los métodos para hacer la comparativa. Los valores configurados para cada método se han seleccionado de acuerdo a las recomendaciones de los autores y se muestran en la Tabla 4.2. De acuerdo a los valores de entrada, las tres características seleccionadas ($F = \{[1\ 19\ 20]\}$) consisten en el canal rojo, el canal azul usando el filtro de la mediana y el canal rojo normalizado, además de la aplicación de un filtro de la mediana para eliminar ruido de la imagen. Esta combinación se ha mostrado como la más adecuada en estudios previos (López-Rubio y López-Rubio, 2015).

[3] http://changedetection.net/

4.3.4. Resultados

Por un lado, desde un punto de vista cualitativo, los resultados producidos por los métodos comparadores son muy similares en la mayoría de los casos, como se muestra en la Figura 4.2. Por otro lado, hay otras secuencias cuyas imágenes segmentadas se obtienen con ruido y esto provoca un peor resultado cuantitativo. Además, hay otros problemas presentes como camuflaje (píxeles de primer plano y fondo son muy similares) o cambios repentinos de iluminación en la escena.

La bondad de cada método y la comparativa con los otros puede ser evaluada con diferentes medidas de rendimiento cuantitativo. Para llevar a cabo esta tarea se han considerado aquellos fotogramas cuyo ground truth presenta algún píxel de primer plano.

Una de las que se han seleccionado es la exactitud espacial (*spatial accuracy*), que se ha usado en comparativas de otras publicaciones (Li et al., 2004; Toyama et al., 1999). Además, la F-medida (F-measure) también se considera, que es una proporción entre la precisión (PR) y la exhaustividad (recall o RC). Estas medidas se definen de la siguiente manera:

$$AC = \frac{\text{card}\,(A \cap B)}{\text{card}\,(A \cup B)} \qquad F - measure = 2 * \frac{PR * RC}{PR + RC} \tag{4.8}$$

donde *card* es el número de elementos de un conjunto, A es el conjunto de todos los píxeles que pertenecen a primer plano, y B es el conjunto de todos los píxeles clasificados como primer plano por el método analizado.

La media de la exactitud, la F-medida, la precisión y la exhaustividad para la mejor configuración de cada video es mostrada en las Tablas 4.3, 4.4, 4.5. Se ha utilizado la mediana como medida media ya que es una medida representativa de la bondad de las propuestas porque ofrece una información más robusta que la media debido a que es menos sensible al ruido.

FFSOM presenta el mejor rendimiento en general de todos los métodos de tipo SOM que se han probado de acuerdo a la exactitud y la F-medida (Tablas 4.3 y 4.4). Además, presenta el mejor rendimiento en cuatro de los diez videos analizados y obtiene mejores resultados que el método FSOM. Por tanto, podemos considerar el método FFSOM como una mejora del algoritmo FSOM.

Otro punto a destacar es el alto rendimiento de la exhaustividad de la propuesta. Como puede ser observado en la Tabla 4.6, el FFSOM muestra el mejor rendimiento de los métodos de tipo SOM en seis de los diez videos analizados. Además, presenta el mejor rendimiento en tres de los videos seleccionados en la comparativa con los algoritmos de otro tipo (videos *pedestrians*, *turnpike* y *streetCornerAtNight*). Es un hecho que se puede considerar la exhaustividad más importante que la precision porque es más deseable tener falsos positivos que falsos negativos debido a que los falsos positivos son más fáciles de manejar que los falsos negativos. Por tanto, es más deseable tener

Tabla 4.4: Resultados de la F-medida (el mayor es mejor). Cada columna se corresponde con un método y las filas indican los videos. Cada celda muestra la media de la F-medida. Los mejores resultados de los métodos de tipo SOM están remarcados en **negrita** y los mejores resultados de los algoritmos de otro tipo están remarcados en *cursiva*.

	Tipo SOM				Otro tipo			
Secuencia	FFSOM	FSOM	SOBS	SC_SOBS	GA	GMM	PAWCS	SuBSENSE
office	0,746	0,693	0,809	**0,874**	0,509	0,431	0,837	*0,861*
pedestrians	0,671	0,662	0,595	**0,726**	*0,723*	0,643	0,689	0,722
abandonedBox	0,261	0,267	0,421	**0,468**	0,219	0,207	*0,743*	0,694
parking	**0,628**	0,546	0,449	0,370	0,321	0,455	*0,616*	0,372
busStation	**0,751**	0,725	0,594	0,695	0,563	0,601	*0,730*	0,710
cubicle	0,558	0,559	0,417	**0,605**	0,482	0,585	0,668	*0,727*
tunnelExit	0,473	0,450	0,452	**0,556**	0,296	0,395	0,570	*0,637*
turnpike	0,684	0,747	0,705	**0,775**	0,700	0,741	*0,812*	0,774
busyBoulvard	**0,387**	0,338	0,362	0,359	0,301	0,343	0,364	*0,383*
streetCornerAtNight	**0,392**	0,336	0,271	0,341	0,337	0,264	0,478	*0,497*
Mediana	**0,593**	0,553	0,451	0,580	0,410	0,443	0,679	*0,702*

Tabla 4.5: Resultados de la precisión (el mayor es el mejor). Cada columna se corresponde con un método y las filas indican los videos. Cada celda muestra la media de la precision. Los mejores resultados de los métodos de tipo SOM están remarcados en **negrita** y los mejores resultados de los algortimos de otro tipo están remarcados en *cursiva*.

	Tipo SOM				Otro tipo			
Secuencia	FFSOM	FSOM	SOBS	SC_SOBS	GA	GMM	PAWCS	SuBSENSE
office	0,798	0,702	0,702	**0,806**	0,598	0,659	0,780	*0,793*
pedestrians	0,515	0,505	0,435	**0,595**	0,600	0,486	0,559	*0,607*
abandonedBox	0,213	0,227	0,273	**0,313**	0,195	0,189	*0,612*	0,601
parking	**0,635**	0,528	0,359	0,423	0,473	0,458	*0,571*	0,491
busStation	**0,683**	0,639	0,461	0,654	0,441	*0,681*	0,649	0,590
cubicle	0,442	0,500	0,309	**0,522**	0,503	0,571	0,592	*0,628*
tunnelExit	0,490	**0,635**	0,430	0,506	*0,723*	0,629	0,590	0,541
turnpike	0,530	0,769	0,607	**0,802**	0,748	0,713	0,789	*0,816*
busyBoulvard	0,473	**0,505**	0,386	0,503	0,562	0,442	*0,577*	0,465
streetCornerAtNight	0,279	0,246	0,183	**0,285**	0,293	0,177	*0,655*	0,450
Median	0,503	**0,516**	0,408	0,514	0,532	0,529	*0,602*	0,596

un rendimiento alto en exhaustividad que un alto rendimiento en precisión.

4.4. Conclusiones

Se ha presentado un nuevo método de detección de objetos en primer plano, que es llamado FFSOM. Este sistema se divide en dos partes. Primero se aplica un modelo base, que está basado en mapas autoorganizados probabilísticos. Aquí se selecciona un conjunto apropiado de características

Tabla 4.6: Resultados de la exhaustividad (el mayor es el mejor). Cada columna se corresponde con un método y las filas indican los videos. Cada celda muestra la media de la exhaustividad. Los mejores resultados de los métodos de tipo SOM están remarcados en **negrita** y los mejores resultados de los algoritmos de otro tipo están remarcados en *cursiva*.

	Tipo SOM				Otro tipo			
Secuencia	FFSOM	FSOM	SOBS	SC_SOBS	GA	GMM	PAWCS	SuBSENSE
office	0,751	0,769	**0,968**	0,963	0,504	0,452	0,912	*0,949*
pedestrians	**0,991**	0,988	0,978	0,951	0,930	*0,980*	0,932	0,910
abandonedBox	0,402	0,396	**0,970**	0,964	0,286	0,283	*0,987*	0,894
parking	**0,754**	0,635	0,721	0,400	0,281	0,536	*0,792*	0,374
busStation	**0,862**	0,861	0,855	0,787	0,802	0,620	0,860	*0,900*
cubicle	**0,869**	0,731	0,842	0,776	0,541	0,669	0,815	*0,903*
tunnelExit	0,634	0,448	0,639	**0,728**	0,214	0,345	0,721	*0,943*
turnpike	**0,981**	0,732	0,844	0,758	0,663	0,775	*0,844*	0,750
busyBoulvard	0,456	0,352	**0,485**	0,410	0,274	0,415	0,367	*0,460*
streetCornerAtNight	**0,880**	0,735	0,786	0,626	0,581	*0,796*	0,447	0,667
Mediana	0,808	0,732	**0,843**	0,767	0,523	0,578	0,830	*0,897*

de píxel y el primer plano es modelado utilizando una distribución uniforme. Después, tres variables de entrada son proporcionadas a un modelo de lógica difusa: *Rugosidad*, que indica una posible iluminación debido a que están ocurriendo transformaciones de color bruscas; *Diferencia*, que mide la diferencia en el color del píxel que podría estar producida por un cambio de iluminación o de primer plano; y *Referencia*, que es la salida del modelo de detección de objetos tomado como base, y proporciona la estimación a priori de la probabilidad de que el píxel pertenezca al primer plano.

Cada regla difusa está diseñada para detectar una condición que indica que el píxel considerado pueda pertenecer al primer plano. Más de una regla puede activarse para el mismo píxel.

Los resultados han sido comparados con aquellos proporcionados por otros algoritmos relacionados, que han sido probados sobre diez secuencias de video pertenecientes a diferentes categorías. Los resultados obtenidos confirman el buen rendimiento del FFSOM. Este algoritmo proporciona una cantidad de falsos negativos particularmente baja, lo que indica que es indicado para escenas con pequeños objetos de primer plano que podrían ser no detectados si el número de falsos negativos fuera muy alto.

Capítulo 5

Sensor de detección de movimiento basado en mapas autoorganizados

La pregunta de si un computador puede pensar no es más interesante que la pregunta de si un submarino puede nadar.

Edsger Dijkstra

Resumen: Muchas de las actuales aplicaciones de la visión por computador están basadas en hardware caro y de alto rendimiento para salvar las necesidades de computación pesada por parte de los algoritmos empleados. Estas arquitecturas están limitadas en la práctica debido a las limitaciones financieras y técnicas. En este capítulo se utiliza una estrategia diferente, desarrollándose un sistema de visión por computador barato y fácil de montar para la detección de movimiento. Esto se consigue mediante tres pasos. En primer lugar, se emplea una plataforma hardware flexible y asequible. Segundo, el algoritmo de detección de movimiento está específicamente adaptado para ocasionar una carga computacional muy pequeña. Y tercero, se sigue un paradigma de programación de punto fijo para implementar el sistema para reducir aún más los requisitos computacionales. El sistema propuesto es comparado en los experimentos con los detectores de movimiento estándar para un amplio número de videos de referencia. Los resultados mostrados indican que la propuesta logra un rendimiento sustancialmente mejor, mientras que es asequible y fácil de instalar en la práctica.

5.1. Introducción

La detección de movimiento es el proceso de detectar un cambio en la posición de un objeto respecto de sus alrededores o un cambio en los alrededores respecto a un objeto. La detección de movimiento puede ser lograda mediante métodos mecánicos o electrónicos, pero lo normal es utilizar sensores electrónicos.

Los sensores de movimiento pueden ser activos o pasivos. Los sensores pasivos no emiten ninguna energía al medio y son los más usuales. Son sensibles a la temperatura de la piel de una persona a través de la radiación emitida del cuerpo negro a longitudes de onda del infrarrojo medio, en contraste con los objetos del fondo a temperatura ambiente. Por su parte, los sensores activos emiten una señal como luz, microondas o sonido al entorno y detectan algún cambio en el comportamiento de respuesta.

Actualmente se están introduciendo nuevas técnicas en los sistemas de detección de movimiento mediante uso de cámaras digitales capaces de grabar video. Es posible usar la salida de una cámara para detectar el movimiento mediante el uso de software. La detección de movimiento normalmente es ejecutada por algoritmos software de supervisión. Cuando el algoritmo detecta movimientos activa la cámara para comenzar a capturar el evento. Esto también se denomina detección de actividad. Un avanzado sistema de videovigilancia de detección de movimiento puede analizar el tipo de movimiento para ver si hace sonar una alarma (García et al., 2014; Gómez et al., 2015).

Los mapas autoorganizados (Self-Organizing Map o SOM) son un tipo de red neuronal artificial que es capaz de realizar un aprendizaje no supervisado (Kohonen, 1982). Desde su propuesta, las SOM han sido aplicadas al descubrimiento de conocimiento, minado de datos, detección de estructuras en datos de alta dimensionalidad y mapeo de estos datos en un espacio de representación bidimensional (Yin, 2008; Kohonen, 2013). Este mapeo conserva las relaciones entre los datos de entrada y preserva su topología. Por lo tanto, esta red neuronal ha tenido un amplio rango de aplicaciones durante décadas (Samuel Kaski y Kohonen, 1998; Oja et al., 2003). En particular, ha sido aplicada a varias áreas de la visión por computador, como la cuantificación de color (Dekker, 1994; Papamarkos, 1999; Xiao et al., 2012; Palomo y Domínguez, 2014), y la segmentación de imágenes (Bhandarkar et al., 1997; Dong y Xie, 2005; Maddalena y Petrosino, 2008; Lacerda y Mello, 2013).

El SOM está basado en un proceso de aprendizaje incremental (en línea), que tiene mejor habilidad para escapar del mínimo local que el aprendizaje por lotes (Bermejo y Cabestany, 2002) y consume menos tiempo computacional en los problemas de cuantificación del color (Chang et al., 2005). Además, ha sido empleado anteriormente para detectar objetos de primer plano en secuencias de video (Maddalena y Petrosino, 2008; López-Rubio et

al., 2011b), como ya se comentó en la Subsección 3.2. Sin embargo, estas propuestas necesitan un SOM por cada píxel del fotograma del video. Por lo tanto, un SOM debe ser entrenado y consultado para cada píxel y fotograma en tiempo real a medida que la secuencia de video avanza. Es por ello que no son adecuadas para su implementación en microcontroladores, que no tienen recursos computacionales para acometer esta complicada tarea.

Todos estos esquemas necesitan una gran cantidad de computación, lo que conlleva un importante reto en los sistemas de visión por computador (Casanova et al., 2013). Por esta razón ha sido necesario el uso de los PCs para implementar este tipo de procesos de aprendizaje, es decir, los sistemas resultantes son muy caros y complejos para producirlos a gran escala. En este capítulo se propone cambiar la estrategia para obtener detectores de movimiento más baratos y simples.

Las placas de microcontroladores son dispositivos hardware económicos, pequeños y flexibles. Son normalmente empleados en importantes tecnologías como sistemas embebidos (Marwedel, 2006; Mamdoohi et al., 2012), sistemas de tiempo real (Kopetz, 1997; Wang et al., 2010) y redes de sensores inalámbricas (Yick et al., 2008; Sengupta et al., 2013). Tienen una reducida cantidad de recursos hardware y limitada velocidad de cómputo, no permitiendo el uso en tareas complejas. Sin embargo, los recientes avances en la fuerza de cómputo de los microcontroladores y un cambio en su paradigma de programación permite la inclusión de esquemas de aprendizaje en los dispositivos (aprendizaje en el chip), adaptando su comportamiento dinámicamente de acuerdo a los datos detectados (Ortega-Zamorano et al., 2014b; Aleksendrić et al., 2012; Mahmoud et al., 2013).

Los microcontroladores son utilizados frecuentemente en sistemas de detección de movimiento debido a su consumo bajo de energía y su coste reducido. Las personas con desafíos cinéticos pueden beneficiarse de dispositivos de entrada basados en microcontroladores específicamente diseñados para ellos, que miden el movimiento en un avión en tiempo real (Papadimitriou et al., 2006). También se ha propuesto un prototipo de placa de circuito impreso (Printed Circuit Board o PCB) que integra un microcontrolador para estimar movimiento y proximidad (Dobrzynski et al., 2012). En este prototipo, ocho fotodiodos se utilizan como sensores de luz. La eficiencia de las plantas de energía solar puede ser mejorada con sistemas de baja energía que estiman el movimiento de las nubes (Fung et al., 2014). La aproximación de los vectores de movimiento de las nubes es ejecutada por un microcontrolador empotrado, es decir, la gestión de los paneles solares puede ser optimizada para una salida máxima de electricidad. Finalmente, alumbrado público de bajo consumo puede ser conseguido con sistemas de bajo consumo de detección de movimiento equipados con microcontroladores de bajo consumo y dispositivos de comunicación inalámbricos (Adnan et al., 2015). De esta forma, las luces de la calle se encienden cuando la gente está en sus alrededores.

En este capítulo se ha implementado el SOM en una placa Arduino DUE, incluyendo el proceso de aprendizaje para implementar el proceso de detección automática de movimiento para la toma de decisiones en el detector en todo tipo de ambientes, evitando la computación no en línea y la comunicación con otros dispositivos.

La placa Arduino DUE que ha sido utilizada (Oxer y Blemings, 2009) es muy popular, económica, con un microcontrolador en una única placa y código abierto eficiente que permite el desarrollo fácil de proyectos (Lian et al., 2013; Cela et al., 2013; Ortega-Zamorano et al., 2015; Kornuta et al., 2012). También se propone un cambio en la representación de los datos utilizados en la programación del Arduino de la representación en punto flotante normalmente utilizada en este tipo de sistemas a la representación en punto fijo, para obtener un sistema más rápido con menos recursos hardware. Esto permite la utilización del SOM en este tipo de dispositivos.

El capítulo se estructura como sigue. En la Sección 5.2 se describe brevemente el sistema microcontrolador y se detalla la propuesta de programación en punto fijo. Luego se introduce el modelo de detección de movimiento incluyendo el SOM, que es específicamente diseñado para satisfacer las capacidades de computación de los microcontroladores (Sección 5.3). La Sección 5.4 explica los detalles de la aplicación implementada. Los resultados obtenidos se muestran en la Sección 5.5. Por último, las conclusiones se comentan en la Sección 5.6.

5.2. Descripción del sistema microcontrolador (μC)

El modelo de detección de movimiento basado en un SOM se ha implementado en un microcontrolador Arduino DUE. Los detalles del sistema implementado se describen a continuación, con énfasis en la comparación entre el uso de una representación en punto fijo o punto flotante. La Subsección 5.2.1 describe el hardware Arduino, la Subsección 5.2.2 ofrece una visión general del software de detección de movimiento, y en la Subsección 5.2.3 se discuten las opciones de implementar operaciones aritméticas.

5.2.1. La placa Arduino

Arduino es un microcontrolador en una única placa diseñado para hacer el proceso de utilización de la electrónica más accesible en proyectos multidisciplinares (Oxer y Blemings, 2009). El hardware consiste en una simple placa de código abierto diseñada sobre un microcontrolador de núcleo de 32-bit Atmel ARM, y el software incluye un compilador de lenguaje de programación estándar que se ejecuta en ordenador estándar y un cargador de arranque para cargar el código compilado en el microcontrolador. Arduino

Figura 5.1: Placa Arduino DUE utilizada para la implementación del modelo de detección de movimiento basado en SOM.

es un descendiente de la plataforma de código abierto *Wiring* y está programada utilizando un lenguaje basado en él (sintaxis y librerías), similar a C++ con algunas pequeñas modificaciones y simplificaciones, y un entorno de desarrollo basado en procesado integrado. La placa Arduino puede comprarse pre-ensamblada o para montarla por uno mismo, y la información del diseño del hardware está disponible. La longitud máxima y la anchura de una placa Arduino UNO son 10.2 y 5.3 cm, respectivamente, con el conector USB y toma de corriente que se extiende más allá de la dimensión anterior.

El Arduino DUE está basado en el SAM3X8E ARM Cortex-M 3 CPU (Atmel, 2016), y tiene 54 pines digitales de entrada/salida (de los cuales 12 pueden ser usados como salidas PWM), 12 entradas analógicas, 4 UARTs (puertos serie), un reloj de 84 MHz, una conexión compatible con USB OTG, 2 DAC (digital a analógico), y botones de reinicio y borrado. El SAM3X tiene 512 KB (2 bloques de 256 KB) de memoria flash para almacenar código, también viene con un gestor de arranque preinstalado que se almacena en una memoria ROM dedicada. La cantidad disponible de SRAM es de 96 KB en dos bancos contiguos de 64 y 32 KB. En la Figura 5.1 se muestra una imagen de una placa Arduino DUE.

El Arduino DUE tiene varios servicios para comunicarse con un ordenador, otro Arduino u otros microcontroladores, y diferentes dispositivos como teléfonos, tabletas, cámaras y más. El SAM3X proporciona un hardware UART y tres hardware USART para comunicación serie TTL (3.3V).

5.2.2. Fases de inicialización y ejecución del algoritmo

La implementación del modelo propuesto para detectar movimiento basado en SOM comprende dos fases: la fase de inicialización que genera el estado inicial del modelo, y la fase de ejecución en la que el microcontrolador actualiza el modelo y toma decisiones basándose en los datos de entrada. El fotograma del video de entrada se divide en varios bloques de píxeles que no se solapan, es decir, un modelo SOM se asocia a cada bloque.

La fase de inicialización genera el estado inicial del modelo SOM asociado a cada bloque de píxeles. Para hacer esto, los prototipos de todas las neuronas

de la SOM son inicializados al color medio de los píxeles que pertenecen al bloque de píxeles en el primer fotograma del video de entrada.

La fase de ejecución está dividida en dos diferentes procesos: el proceso de aprendizaje y el proceso de decisión. Para cada bloque de píxeles, el proceso de aprendizaje resume la información del color de todos los píxeles del bloque en un vector de entrada que es aplicado como una muestra de entrenamiento al SOM asociado al bloque. Después, el proceso de decisión estima si cada bloque individual contiene objetos en movimiento con la ayuda del modelo SOM asociado al bloque. Más detalles sobre los procesos de aprendizaje y decisión se proporcionan en la Sección 5.3.

5.2.3. Representación del tipo de datos

Los microcontroladores son dispositivos con una fuerza de cómputo limitada, por lo que, para incrementar la velocidad del proceso de aprendizaje, se ha decidido emplear una representación del tipo de datos en punto fijo. Nótese que el punto flotante es el más utilizado habitualmente en la representación de los tipos de datos en este tipo de dispositivos, pero esta representación no es siempre la más efectiva.

El cambio del paradigma en la representación del tipo de datos involucra un cambio en el tipo de variables utilizadas en la implementación software del modelo SOM. La representación en punto flotante es almacenada en una variable de tipo "flotante" ("float") o "doble" ("double") con un tamaño de 4 u 8 bytes, respectivamente, dependiendo de la precisión que se necesite. Por otro lado, la representación en punto fijo es almacenada en una variable de tipo "entero" ("integer") con un tamaño de 4 bytes.

Este cambio de paradigma implica cambios profundos en la forma en que se programa el modelo SOM, pero a cambio ofrece un proceso de aprendizaje más rápido y una representación de variables de menor tamaño. La Tabla 5.1 muestra el tiempo computacional (en μs) que se necesitan para el cálculo de una operación aritmética básica { + , - , * , / } con los tipos de variable mencionados (entero, flotante y doble) en el microcontrolador Arduino DUE.

Tipo Variable	Operaciones Básicas			
	+	-	*	/
Entero	59,8	59,8	71,7	99,9
Flotante	3965,4	4146,9	3751,8	5269,8
Doble	5113,2	5139,4	4763,3	13635,2

Tabla 5.1: Tiempo de computación que se necesita para las operaciones aritméticas básicas dependiendo del tipo de variable (entero, flotante y doble) utilizado en el microcontrolador Arduino DUE.

5.3. Modelo de detección de movimiento

El sistema de detección de movimiento propuesto en este capítulo está basado en la subdivisión del fotograma de entrada en varios bloques rectangulares del mismo tamaño que no se solapan. Se aprende un modelo de color para cada bloque mediante medias de una SOM, es decir, las anomalías de color pueden ser medidas en cada región de forma separada. Después, las anomalías de color son analizadas para determinar si están asociadas a objetos de primer plano en movimiento. La Subsección 5.3.1 explica la disposición de la subdivisión del fotograma, la Subsección 5.3.2 describe el modelo SOM, y la Subsección 5.3.3 detalla cómo se analiza las anomalías medidas. La Subsección 5.3.4 ofrece los detalles sobre el almacenamiento de los SOMs. El algoritmo utilizado para calcular la función exponencial es explicado en la Subsección 5.3.4. Por último, la Subsección 5.3.6 compara las implementaciones de punto fijo y punto flotante del modelo propuesto.

5.3.1. Subdivisión del fotograma

La mayoría de los enfoques actuales para la detección de movimiento construyen un modelo de color para cada píxel (Bouwmans, 2014b). Sin embargo, esto no es factible para los microcontroladores debido a sus limitaciones de hardware. Por lo tanto, se propone una subdivisión del fotograma de entrada en bloques rectangulares que no se solapan, es decir, se aprende un modelo de color por cada región.

Se asume que el fotograma del video de entrada tiene $N_{row} \times N_{col}$ píxeles, y que para cada píxel se obtiene un vector de color RGB $\mathbf{y_h} \in [0,1]^3$, donde $\mathbf{h} \in \{1,...,N_{row}\} \times \{1,...,N_{col}\}$ son las coordenadas del píxel. Después, el fotograma de entrada es dividido en $B_{row} \times B_{col}$ bloques que no se solapan de tamaño $\frac{N_{row}}{B_{row}} \times \frac{N_{col}}{B_{col}}$ píxeles, donde N_{row} es un entero múltiplo de B_{row} y N_{col} es un entero múltiplo de B_{col}.

Para cada bloque es necesario resumir la información de color proporcionada por el dato del color del píxel $\mathbf{y_h}$ de una manera rápida. Aquí se propone calcular la media del color de cada bloque:

$$\mathbf{x_r} = \frac{1}{N_{block}} \sum_{\mathbf{h} \in \mathcal{B}_\mathbf{r}} \mathbf{y_h} \tag{5.1}$$

donde $\mathbf{x_r} \in [0,1]^3$, $\mathcal{B}_\mathbf{r}$ es un conjunto de píxeles que pertenecen al bloque con coordenadas $\mathbf{r} \in \{1,...,B_{row}\} \times \{1,...,B_{col}\}$, y N_{block} es el número de píxeles por bloque:

$$N_{block} = \frac{N_{row} N_{col}}{B_{row} B_{col}} \tag{5.2}$$

Para cada fotograma de entrada y bloque, la media del color del bloque $\mathbf{x_r}$ es proporcionada al mapa autoorganizado asociado al bloque como muestra

de entrenamiento, como se ve a continuación.

5.3.2. Mapa autoorganizado

Ahora se describe el modelo SOFM de Kohonen que es el utilizado para aprender el modelo de color de un bloque del fotograma de entrada. Sea M el número de neuronas del mapa autoorganizado asociado a un cierto bloque del fotograma de entrada. Las neuronas están organizadas en una cuadrícula de tamaño $a \times b$, donde $M = ab$. La distancia topológica entre las neuronas i e i', localizadas en las posiciones $(y_1, y_2) \in \mathbb{N}^2$ y $(y_1', y_2') \in \mathbb{N}^2$ en el espacio de la cuadrícula, viene dada por:

$$d\left(i, i'\right) = \sqrt{\left(y_1 - y_1'\right)^2 + \left(y_2 - y_2'\right)^2} \tag{5.3}$$

Cada neurona i tiene un vector prototipo $\mathbf{w}_i$ que representa un grupo de muestras de entrada. Nótese que $\mathbf{w}_i \in [0,1]^3$, donde se consideran vectores de entrada tridimensionales de valores reales que codifica colores en el espacio de color RGB. En el paso de tiempo n, una nueva muestra $\mathbf{x}(n)$ que representa el color medio para el bloque se presenta a la red, y una neurona ganadora es declarada:

$$Winner\left(\mathbf{x}(n)\right) = \arg \min_{j \in \{1,\ldots,M\}} \left\|\mathbf{x}(n) - \mathbf{w}_j(n)\right\| \tag{5.4}$$

Después, los prototipos de todas las unidades son ajustados, para $i \in \{1, ..., M\}$:

$$\mathbf{w}_i(n+1) =$$

$$\mathbf{w}_i(n) + \eta(n)\,\Lambda\left(i, Winner\left(\mathbf{x}(n)\right)\right)\left(\mathbf{x}(n) - \mathbf{w}_i(n)\right) \tag{5.5}$$

donde $\eta(n)$ es una tasa de aprendizaje de deterioro y la función de vecindad Λ varía con el paso de tiempo n y depende de un *radio de vecindad* de deterioro $\Delta(n)$:

$$\eta(n+1) \leq \eta(n) \tag{5.6}$$

$$\Lambda\left(i, Winner\left(\mathbf{x}(n)\right)\right) =$$

$$\exp\left(-\left(\frac{d\left(i, Winner\left(\mathbf{x}(n)\right)\right)}{\Delta(n)}\right)^2\right) \tag{5.7}$$

$$\Delta(n+1) \leq \Delta(n) \tag{5.8}$$

En el tiempo de inicialización $n = 0$, cada prototipo $\mathbf{w}_i$ es ajustado a la muestra observada $\mathbf{x}(0)$ para $i \in \{1, ..., M\}$:

$$\mathbf{w}_i(0) = \mathbf{x}(0) \tag{5.9}$$

El campo receptivo de la neurona i, es decir, la región del espacio de entrada que es representada por i, es definida como:

$$F_i = \left\{ \mathbf{x} \in \mathbb{R}^D \mid i = Winner(\mathbf{x}) \right\} \tag{5.10}$$

El vector de cuantificación es uno de los principales objetivos de los mapas autoorganizados. Interesa el error de cuantificación q_k asociado a la entrada $\mathbf{x}_k$:

$$q_k = \min_{j \in \{1,...,M\}} \|\mathbf{x}_k - \mathbf{w}_j\| \tag{5.11}$$

El rendimiento global de un mapa para esta tarea está comúnmente definido por el error cuadrático medio (Hsu y Halgamuge, 2003; Yin, 2008; Beaton et al., 2010; Dlugosz et al., 2010):

$$MSE = \frac{1}{K} \sum_{k=1}^{K} q_k^2 \tag{5.12}$$

5.3.3. Análisis de anomalías

A medida que el mapa autoorganizado asociado a cada bloque aprende la información del color correspondiente a ese bloque, es posible estimar si el bloque contiene una parte sustancial de objetos en movimiento. Esto se hace de manera rápida considerando el error de cuantificación $q_{n,\mathbf{r}}$ en el paso de tiempo n del actual color medio del bloque $\mathbf{x}_{n,\mathbf{r}}$, representado por el mapa autoorganizado asociado al bloque:

$$q_{n,\mathbf{r}} = \min_{j \in \{1,...,M\}} \|\mathbf{x}_{n,\mathbf{r}} - \mathbf{w}_{j,n,\mathbf{r}}\| \tag{5.13}$$

donde $\mathbf{w}_{j,n,\mathbf{r}} \in [0,1]^3$ se ajusta para el prototipo en el paso de tiempo n de la j-ésima neurona del mapa autoorganizado asociado al bloque con coordenadas $\mathbf{r}$. Se declara que el bloque contiene objetos en movimiento si y solo si $q_{n,\mathbf{r}}$ es superior a un umbral T:

$$\text{Bloque } \mathbf{r} \text{ contiene objetos en movimiento} \Leftrightarrow q_{n,\mathbf{r}} > T \tag{5.14}$$

donde $T > 0$ es un parámetro configurable del sistema. La razón que hay tras esto es que los objetos en movimiento normalmente tienen un color que se diferencia significativamente del color del fondo.

5.3.4. Almacenamiento del modelo SOM

El número de bytes utilizado para representar el modelo SOM en cada bloque de píxeles depende de la representación del tipo de datos. Empleando la representación de punto fijo se permite utilizar 32 bits para cada variable desde que son almacenadas en una variable "entera". En este caso, como los valores del modelo SOM oscilan entre 0 y 1, la precisión de este tipo de variables es $2^{-32} = 2{,}328 \cdot 10^{-10}$.

Tomando en cuenta que la memoria SRAM disponible suma 96 KB para almacenar todas las variables del algoritmo y que los modelos SOM son las variables que más memoria consumen, la memoria SRAM ha sido dividida en dos partes. La primera almacena los modelos SOM de todos los bloques de píxeles con 80 KB, y la otra parte consta del resto de variables involucradas en la ejecución del algoritmo con 16 KB. Por tanto, el máximo número de bloques de píxeles que pueden ser almacenados en el sistema implementado viene dado por la siguiente ecuación:

$$N_{blocks} \leq \frac{80KB}{M \cdot 4Byte}, \tag{5.15}$$

donde $N_{blocks} = B_{row}B_{col}$ es el número de bloques de píxeles y $M = a \times b$ es el número de neuronas en la SOM de cada bloque de píxeles, que ha sido configurado a $M = 3 \times 4 = 12$ porque ofrece una buena compensación entre la habilidad de las SOMs de representar distribuciones complejas de colores de entrada y la carga computacional que requiere entrenar las SOMs.

Por tanto, el máximo número de bloques de píxeles que pueden ser almacenados es de 1706 bloques para variables con un tamaño de 4 bytes.

5.3.5. Cálculo de la función exponencial

El cálculo de la función exponencial puede ser ejecutado utilizando la unidad aritmética y lógica específica (Arithmetic and Logic Unit, o ALU) mediante el uso de la librería específica "math.h" para evaluar las funciones exponenciales involucradas en el modelo. El tiempo computacional de este procedimiento es igual a $58{,}9\mu s$ en el microcontrolador utilizado.

Se ha implementado una aproximación para ejecutar la función exponencial para reducir el tiempo computacional para esta función. Esta reducción del tiempo permite actualizar más bloques de píxeles en un tiempo dado, de este modo se incrementa el máximo número de bloques de píxeles que pueden ser procesados.

La aproximación se ha realizado mediante una búsqueda de tabla seguida de una interpolación lineal. Este método ha sido estudiado ampliamente en trabajos previos (Ortega-Zamorano et al., 2014a).

La tabla contiene los valores de la función exponencial para valores equispaciados de la variable independiente. Sin embargo, debido a que se necesitan

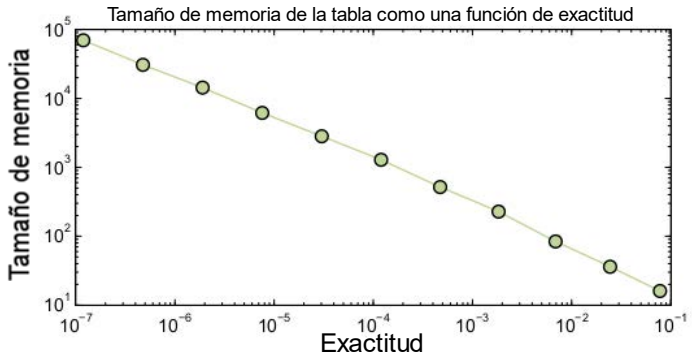


Figura 5.2: Tamaño de memoria necesario de acuerdo a la precisión de la aproximación del método optimizado en base a la tabla más la interpolación lineal de valores adyacentes.

valores de precisión alto para la correcta ejecución del algoritmo, el cálculo de la aproximación de función se complementa con un procedimiento de interpolación lineal utilizando dos valores tabulados adyacentes (más bajo y más grande) con respecto a los valores de entrada de la variable independiente. En este caso el tiempo de computación que se necesita se reduce a 1,437μs, lo que significa una reducción del 97,5 % en comparación con la librería específica "math.h"

El almacenamiento de valores de la tabla requiere grandes cantidades de memoria dependiendo de la precisión de la aproximación. La Figura 5.2 muestra el tamaño de memoria necesario de acuerdo a la precisión de la aproximación del método optimizado en base a la tabla más la interpolación lineal de valores adyacentes.

Se ha seleccionado un error máximo absoluto menor que $5 \cdot 10^{-5}$. Por tanto, el tamaño de memoria necesario para almacenar la tabla de búsqueda es igual a 4 KB. La Figura 5.3 muestra el error absoluto involucrado en el cálculo de la función exponencial negativa en el rango de 0 a 16.

5.3.6. Comparación entre las representaciones de punto fijo y punto flotante.

La Figura 5.4 muestra el tiempo de computación (eje y de la izquierda) en μs que se necesitan para actualizar el modelo SOM de un bloque de píxeles cuando se implementa con variables de tipo "entero" y "flotante" y el número de veces (eje y de la derecha) que las variables de tipo "entero" son más rápidas que las de tipo "flotante" como una función del número de neuronas del modelo SOM.

El tiempo de cálculo para actualizar un bloque de píxeles coloca un límite superior en la cantidad de píxeles que el sistema puede actualizar en

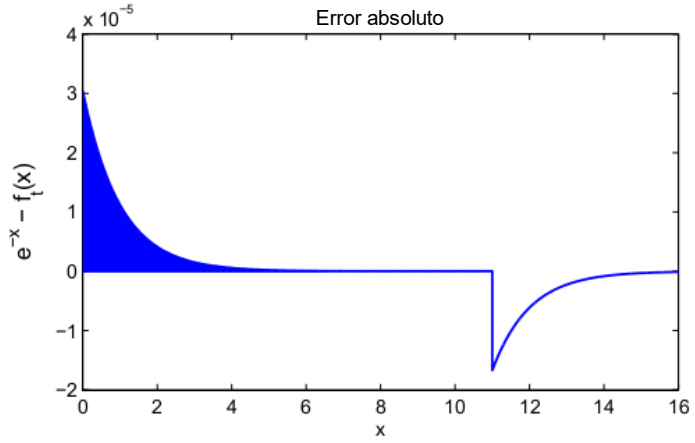


Figura 5.3: Error absoluto cometido en la aproximación de la función exponencial (ver texto para más detalles).

tiempo real. Actualmente las operaciones en tiempo real para visión por computador significa que cada fotograma debe ser procesado dentro de 30-40 ms (Pulli et al., 2012). Desde que las cámaras de video se utilizaron para experimentos capturan a 30 fotogramas por segundo, que es 33.33 ms por fotograma, un video completo debe ser procesado en menos de 1/30 s para que pueda ser considerado para operaciones de tiempo real. De esta forma, el máximo número de bloques de píxeles que puede ser procesado en el sistema implementado viene dado por la siguiente ecuación:

$$N_{blocks} \leq \frac{0{,}03333(s)}{T_{up}}, \tag{5.16}$$

donde N_{blocks} es el número de bloques de píxeles y T_{up} es el tiempo de computación para actualizar el modelo SOM de un bloque.

El número de neuronas (M) ha sido fijado a 12, T_{up} es igual a 93 μs para las variables de tipo "entero" y 969 μs para las variables de tipo "flotante". Por tanto, el máximo número de bloques que pueden ser actualizados es 358 bloques para variables de tipo "entero" y 34 bloques para variables de tipo "flotante".

5.4. Aplicación

Los sistemas de detección de intrusismo son ampliamente utilizados en todo tipo de premisas desde hogares a edificios públicos, es decir, hay muchos contextos donde este tipo de sistema podría ser desplegado. Se ha puesto énfasis en hacer el sistema sencillo de replicar para tener múltiples detectores de movimiento. En particular, el bajo coste y el bajo consumo energético han sido los focos de atención.

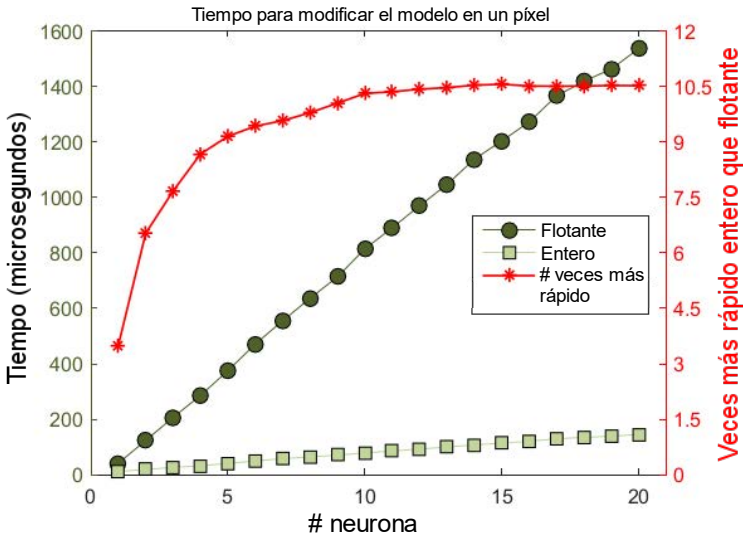


Figura 5.4: Tiempo de computación (izquierda del eje y) en μs que se necesitan para actualizar el modelo SOM de un bloque de píxeles con diferentes implementaciones de variables (Entero y Flotante) y el número de veces (derecha del eje y) que las variables de tipo "entero" son más rápidas que las variables de tipo "flotante" como una función del número de neuronas del modelo.

El sistema propuesto está compuesto de una cámara para obtener la imagen de la escena y un microcontrolador para decidir la existencia de movimiento inusual en la escena. Ambos han sido seleccionados para ser dispositivos de bajo coste y de bajo consumo. El microcontrolador elegido es el Arduino DUE (ver Subsección 5.2.1) y la cámara de video utilizada en la aplicación es la C429-RS232, que es un módulo con una cámara compacta e integrada. El módulo utiliza un sensor de color OmniVision^{TM} CMOS MT9V011 VGA, complementado con un chip de control Vimicro VC0706 para ofrecer un sistema de cámara de bajo coste y bajo consumo. Tiene interfaz serie RS232 para una conexión directa a un microcontrolador. La tasa de transferencia serie es de 115.2 Kbps para transferir color o imágenes monocromáticas en resolución VGA (640×480), QVGA (320×240), o QQVGA (160×120). La salida de video en tiempo real es proporcionada a 30 fps como señal CVBS, NTSC o PAL. C429-RS232 solo necesita 80mA de una fuente de alimentación de 5V.

La Figura 5.5 ilustra las tareas que son ejecutadas cuando se procesa un fotograma del video de entrada. Las cinco imágenes de la izquierda describen el proceso de un fotograma con no intrusión detectada, mientras que las cinco imágenes de la derecha corresponden a un fotograma donde se detecta una intrusión.

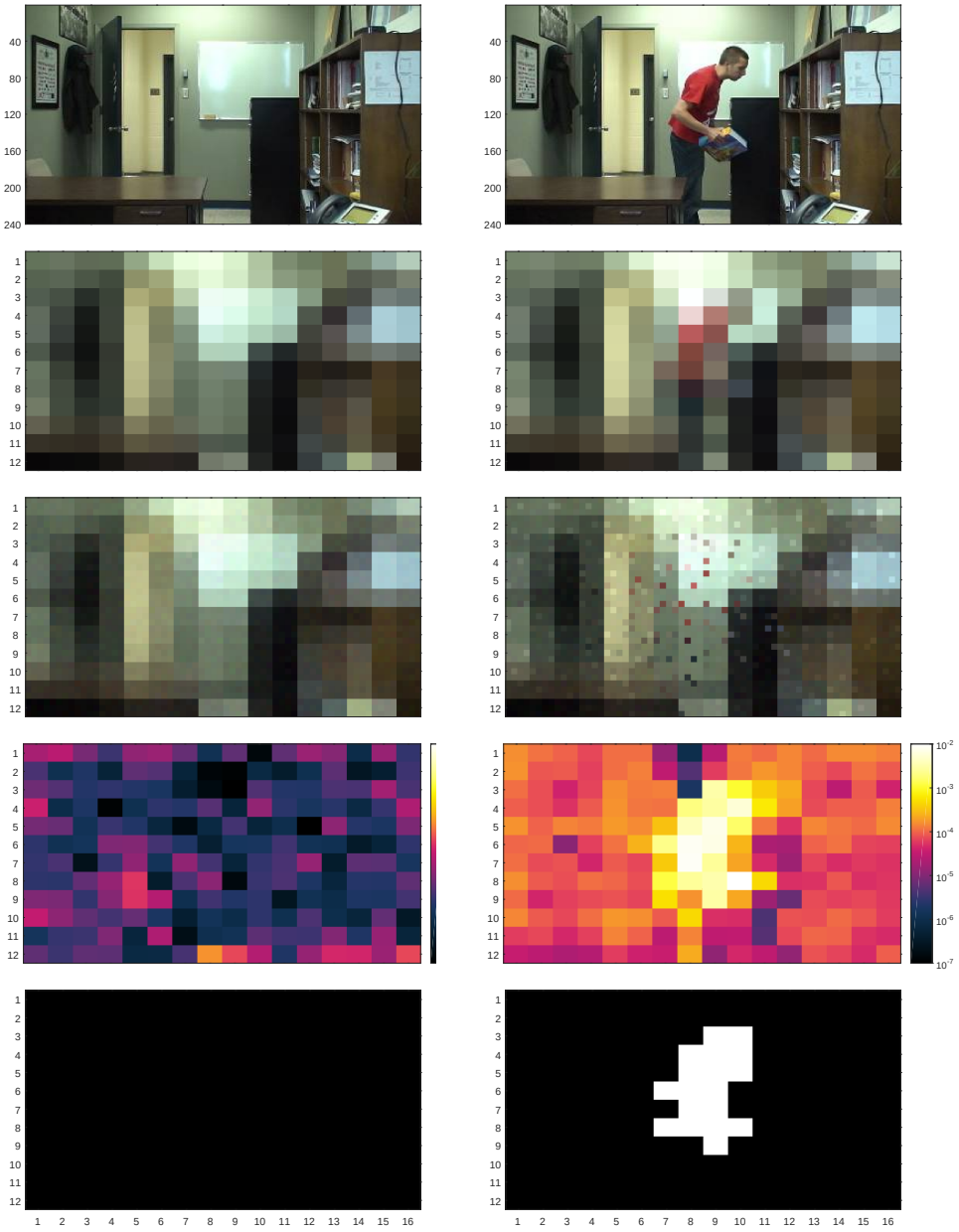


Figura 5.5: Pasos ejecutados en el procesamiento de imagen de un fotograma para una detección de intrusión (las cinco imágenes de la derecha) y otra escena sin detección (las cinco imágenes de la izquierda). (ver texto para más detalles).

Las imágenes de la primera fila muestran la imagen RGB que es capturada por la cámara. Estas imágenes son el punto de inicio del proceso de detección de intrusión.

Las imágenes de la segunda fila muestran las imágenes comprimidas que son enviadas al microcontrolador para que las procese. El tamaño máximo de las imágenes comprimidas está determinado por la ecuación 5.16. Sin embargo, en el caso del tamaño seleccionado es $12 \times 16 = 192$ bloques para demostrar que los tamaños más pequeños también pueden ser usados por esta aplicación con un decrecimiento no significativo de la eficiencia. Con este número de bloques se logra el procesado en tiempo real ya que el número de bloques es inferior al límite superior para el tiempo real, que es de 358 bloques, como se ha explicado anteriormente.

La tercera fila muestra los modelos SOM para cada píxel (ver Sección 5.3.2). Para cada uno de los 12×16 bloques de las imágenes comprimidas se muestra un mosaico con 3×4 pequeños rectángulos, donde cada uno de ellos representa el prototipo de una neurona del SOM asociado al bloque. Esto es porque las redes SOM utilizadas tienen una topología rectangular con $3 \times 4 = 12$ neuronas. Los rectángulos pequeños muestran el prototipo asociado a la neurona como un color RGB. Se puede observar que las neuronas asociadas al mismo bloque son bastante similares en el lado izquierdo cuando la intrusión no ha sido aún detectada. Por otro lado, las neuronas del mismo bloque son significativamente diferentes en el lado derecho cuando la intrusión se comienza a detectar. Esto es porque algunas neuronas aprenden los colores del objeto intruso.

La cuarta fila (y segunda por abajo) muestra el error de cuantificación de cada bloque de entrada de las imágenes capturadas dado por la ecuación 5.13. Se puede ver que los tonos máximo y mínimo utilizados en la barra de color son los mismos para ambas imágenes, es decir, los errores de cuantificación pueden ser comparados. Como se observa, el fotograma sin objetos intrusos (izquierda) ofrece unos errores de cuantificación más pequeños que el fotograma con el objeto intruso (derecha).

La quinta y última fila muestra la decisión de si cada bloque contiene objetos en movimiento, calculado con la ecuación 5.14. La decisión se basa dependiendo del valor del error cuantificación en cada instante de tiempo. Si el error mayor que un determinado umbral (T), entonces el bloque se declara como primer plano, es decir, un bloque con una intrusión detectada. En las imágenes, los bloques que muestran un error de cuantificación mayor que el umbral son pintados en blanco, mientras que los bloques con errores de cuantificación menores que el umbral son pintados en negro. Se declara que un fotograma contiene una intrusión en una escena de video cuando dos o más bloques tienen un error de cuantificación mayor que el umbral.

Para determinar el valor del umbral se ha analizado una secuencia de video y se ha obtenido el histograma de los errores de cuantificación máximos

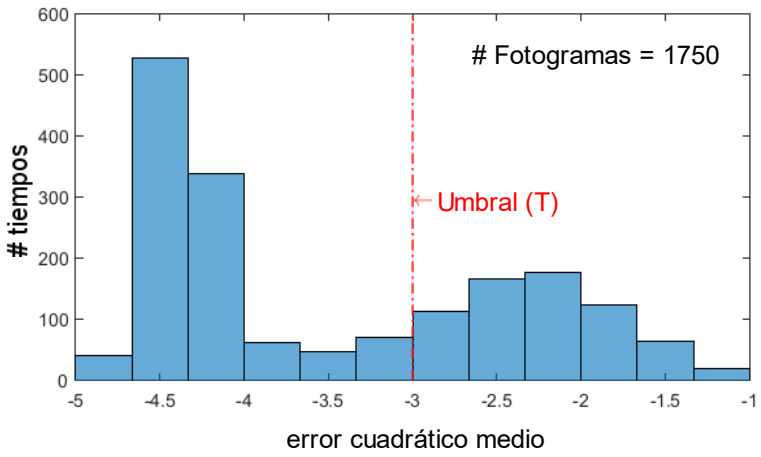


Figura 5.6: Histograma de máximos errores de cuantificación de todos los bloques para una secuencia de video formada por 1750 fotogramas. El umbral que separa los errores de cuantificación de los bloques que no presentan movimiento (izquierda) y los bloques con movimiento (derecha) se muestra con una línea vertical de puntos y rayas.

para todos los bloques de un fotograma. En la Figura 5.6 se puede observar que el histograma muestra dos modos. El modo de la izquierda corresponde a la ausencia de intrusión, mientras que el de la derecha está asociado a intrusiones. Como se ve, la mayoría de los fotogramas sin detección de intrusión tienen un error de cuantificación en el rango de 10^{-4} a 10^{-5}, mientras que los fotogramas con detección de intrusión el error de cuantificación es alrededor de 10^{-2}. Por esta razón se ha seleccionado el valor del umbral T como 10^{-3}.

5.5. Resultados

En esta sección se ha probado el sistema implementado con diferentes videos de referencia bien conocidos (Wang et al., 2014b) para demostrar la utilidad del esquema propuesto. Cada video RGB en bruto viene asociado con una máscara de verdad (ground truth o GT), que es una secuencia en blanco y negro que establece qué regiones de cada fotograma realmente corresponde a objetos en movimiento. La máscara de verdad solo se utiliza para medir el rendimiento de la detección de los métodos competidores; no se le facilita a los sistemas de detección de ninguna manera. Para obtener unos resultados reproducibles con videos conocidos, se ha conectado el microcontrolador Arduino por puerto USB con una comunicación serie al ordenador. Bajo esta configuración el ordenador es programado para simular una cámara. Es decir, el ordenador envía el video de la misma manera que la cámara lo hace, por lo que el microcontrolador no nota la diferencia.

El detector tradicional puede ser implementado de diferentes formas (Park et al., 2015; Naghiyev et al., 2014), aunque el más útil ha sido frecuentemente los sensores infrarrojos pasivos (PIR). El detector tradicional busca cambios abruptos en la iluminación global de la escena $\bar{y}$:

$$\bar{y} = \frac{1}{3N_{row}N_{col}} \sum_{\mathbf{h}\in\mathcal{B}_\mathbf{r}} \left(y^1_\mathbf{h} + y^2_\mathbf{h} + y^3_\mathbf{h}\right) \tag{5.17}$$

donde $y^j_\mathbf{h}$ es el valor del j-ésimo canal de color en el píxel con coordenadas $\mathbf{h}$.

Para medir los cambios en $\bar{y}$ a lo largo del paso de tiempo n, el tiempo medio de $\bar{y}$ puede ser estimado en el tiempo n como sigue:

$$\hat{y}\left(n+1\right) = \hat{y}\left(n\right) + \eta\left(n\right)\left(\bar{y}\left(n\right) - \hat{y}\left(n\right)\right) \tag{5.18}$$

donde $\eta(n)$ se refiere a la tasa de aprendizaje ya introducida en (5.5). En el tiempo de inicialización $n = 0$ la estimación es ajustada a la iluminación global observada:

$$\hat{y}\left(0\right) = \bar{y}\left(0\right) \tag{5.19}$$

El movimiento es detectado en el tiempo n siempre que la iluminancia global actual difiere de la media estimada $\hat{y}(n)$ por más de un umbral T:

$$e_n = \left|\bar{y}\left(n\right) - \hat{y}\left(n\right)\right| \tag{5.20}$$

$$\text{El fotograma contiene objetos en movimiento} \Leftrightarrow e_n > T \tag{5.21}$$

Además, se han seleccionado de la literatura varios métodos de referencia de detección de primer plano a nivel de píxel que tienen una implementación pública y razonablemente bien testeada, para ejecutar las comparaciones con ellos. Estos métodos han sido ejecutados en un ordenador estándar con un microprocesador de 3 GHz y 8GB RAM, ya que necesitan un alto nivel de computación para ser ejecutados en un microcontrolador. El primer algoritmo que se ha considerado es el método notado como WrenGA (Wren et al., 1997), que es el más antiguo y modela un modelo probabilístico con una gaussiana. Otro métodos gaussianos son GrimsonGMM (Stauffer y Grimson, 1999), que utiliza dos mixturas de gaussianas; y ZivkovicGMM (Zivkovic, 2004; Zivkovic y van der Heijden, 2006), que tiene un número no fijo de distribuciones gaussianas. Además, también se ha considerado un método basado en redes neuronales, notado como MaddalenaSOBS (Maddalena y Petrosino, 2008). Estos métodos están disponibles en la librería BGS versión 1.3.0, que está accesible en su página web[1]. Además, se ha seleccionado el

[1] https://github.com/andrewssobral/bgslibrary

Métodos	Parámetros
MFBM	Rasgos, $F = \{1, 2, 3\}$ Tamaño de paso, $\alpha = 0{,}01$
GrimsonGMM	Umbral, $T = 12$ Tasa de aprendizaje, $\alpha = 0{,}0025$ Número de gaussianas, $K = 3$
MaddalenaSOBS	Sensibilidad, $s_1 = 75$ Sensibilidad de entrenamiento, $s_0 = 245$ Tasa de aprendizaje, $\alpha_1 = 75$ Paso de entrenamiento, $N = 100$
WrenGA	Umbral, $T = 12$ Tasa de aprendizaje, $\alpha = 0{,}005$
ZivkovicGMM	Tasa de aprendizaje, $\alpha = 0{,}001$ Número de gaussianas, $K = 3$ Umbral, $T = 30$

Tabla 5.2: Valores considerados de los parámetros para los métodos competidores, formando el conjunto de configuraciones experimentales.

método MFBM (López-Rubio y López-Rubio, 2015) que está basado en la teoría de la aproximación estocástica. Los valores configurados por cada método son seleccionados de las recomendaciones de los autores y se muestran en la Tabla 5.2.

Debe mencionarse que el problema de la detección de movimiento que los tradicionales detectores tratan de resolver es un problema de clasificación binaria. La clase positiva está formada por aquellos fotogramas donde hay objetos en movimiento. Por otro lado, la clase negativa está compuesta de aquellos fotogramas donde no existen objetos en movimiento. Por tanto, se han considerado medidas de rendimiento de clasificación binaria recomendadas en la base de datos de referencia para la detección de movimiento ChangeDetection 2014[2]:

$$Exhaustividad = \frac{TP}{TP + FN} \tag{5.22}$$

$$Especificidad = \frac{TN}{TN + FP} \tag{5.23}$$

$$FPR = \frac{FP}{FP + TN} \tag{5.24}$$

[2]http://www.changedetection.net

Video	SOM	Ilu	FrameDiff	MFBM	GrimsonGMM	MaddalenaSOBS	WrenGA	ZivkovicGMM
Office	56,0036	**347,2222**	62,7510	8,7140	18,0037	12,6464	19,0158	32,3135
PETS2006	56,0036	**347,2222**	62,7510	2,0630	4,1255	2,7120	5,9493	5,9392
Highway	56,0036	**347,2222**	62,7510	9,4857	18,7410	14,2416	20,0945	20,0905
Pedestrians	56,0036	**347,2222**	62,7510	9,8522	20,4964	12,6251	28,9922	30,7618
Sofa	56,0036	**347,2222**	62,7510	10,5132	20,2561	15,3184	25,0976	25,4189
Canoe	56,0036	**347,2222**	62,7510	9,3113	20,1149	14,3254	30,5428	31,4138
Fountain02	56,0036	**347,2222**	62,7510	6,3019	12,9921	8,8917	14,5588	22,4193
Fall	56,0036	**347,2222**	62,7510	2,5021	4,4226	3,7189	7,2360	6,7668
Media	*56,0036*	***347,2222***	*62,7510*	*7,3429*	*14,8940*	*10,5599*	*18,9359*	*21,8905*

Tabla 5.3: Máximo número de fotogramas por segundo de los métodos competidores sobre las secuencias testeadas (mayor es mejor). Nótese que SOM, Ilu y FrameDiff se ejecutan en un Arduino DUE en tiempo real, mientras que el resto de los métodos se ejecutan en un ordenador estándar. Los mejores resultados de cada secuencia están remarcados en **negrita**.

$$FNR = \frac{FN}{TP + FN} \tag{5.25}$$

$$PBC = 100\frac{FN + FP}{TP + FP + FN + TN} \tag{5.26}$$

$$\text{Precisión} = \frac{TP}{TP + FP} \tag{5.27}$$

$$F - medida = 2\frac{Recall \cdot Precision}{Recall + Precision} \tag{5.28}$$

Nótese que, ya que el objetivo de este trabajo es la detección global de movimiento, las medidas de rendimiento han sido calculadas a nivel de fotograma y a nivel de píxel. Para este fin, se declara que un fotograma contiene movimiento cuando la fracción de píxeles que pertenecen a objetos de primer plano es mayor que $\frac{2}{192}$, que es el mismo criterio que el considerado en la Sección 5.4.

Los resultados cuantitativos se muestran como sigue. Las Tablas 5.3 y 5.4 presentan los fotogramas por segundo y el número de instrucciones ejecutadas por fotograma por cada método testeado sobre los videos evaluados, respectivamente. Se puede ver que los métodos a nivel de píxel (últimas cinco columnas) no alcanzan el tiempo real, incluso si se ejecutan en un ordenador estándar. Por otro lado, los métodos basados en Arduino pueden ejecutarse en tiempo real, incluida nuestra propuesta. Cada propuesta de Arduino tiene los mismos requisitos computacionales para todos los videos porque estos métodos ejecutan las mismas instrucciones para un tamaño de fotograma dado, independientemente del contenido del video. Además, bajo Arduino no hay memoria virtual u otra fuente de variabilidad en tiempo de ejecución.

Video	SOM	Ilu	FrameDiff	MFBM	GrimsonGMM	MaddalenaSOBS	WrenGA	ZivkovicGMM
Office	1,0483	**0,0873**	0,8235	294,1926	192,4942	299,4168	125,5993	111,4860
PETS2006	1,0483	**0,0873**	0,8235	1,341,5196	1,023,0307	1,478,7173	600,2945	540,3325
Highway	1,0483	**0,0873**	0,8235	263,7813	199,5114	266,5234	112,5730	103,7825
Pedestrians	1,0483	**0,0873**	0,8235	296,0330	187,0545	307,9948	125,8543	112,0348
Sofa	1,0483	**0,0873**	0,8235	263,0099	185,8824	269,5781	111,4275	100,1956
Canoe	1,0483	**0,0873**	0,8235	265,6015	190,8002	260,7247	113,1358	106,8618
Fountain02	1,0483	**0,0873**	0,8235	419,9596	305,8514	439,8973	179,6190	159,1311
Fall	1,0483	**0,0873**	0,8235	1,158,3201	975,5466	1,102,7957	492,2767	468,2882
Media	*1,0483*	***0,0873***	*0,8235*	*537,8022*	*407,5214*	*553,2060*	*232,5975*	*212,7641*

Tabla 5.4: Número de instrucciones ejecutadas por fotograma de los métodos competidores sobre las secuencias testeadas (en millones, menor es mejor). Nótese que SOM, Ilu y FrameDiff se ejecutan en un Arduino DUE en tiempo real, mientras que el resto de los métodos se ejecutan en un ordenador estándar. Los mejores resultados de cada secuencia están remarcados en **negrita**.

Video	SOM	Ilu	FrameDiff	MFBM	GrimsonGMM	MaddalenaSOBS	WrenGA	ZivkovicGMM
Office	0,8544	0,2088	0,9905	0,9905	0,9845	0,9928	**0,9940**	0,9606
PETS2006	0,4715	0,0976	**0,9009**	0,7293	0,7955	0,7789	0,4060	0,4632
Highway	0,9304	0,4485	0,9982	0,8670	0,9789	**1,0000**	0,9700	0,9665
Pedestrians	0,9540	0,1206	**0,9889**	0,9222	0,9587	0,9683	0,9206	0,9413
Sofa	0,9150	0,4511	**0,9985**	0,9872	0,9737	0,9962	0,9714	0,9714
Canoe	0,9637	0,3387	**1,0000**	0,9032	**1,0000**	**1,0000**	**1,0000**	**1,0000**
Fountain02	0,7192	0,0000	0,9557	0,5616	0,8177	**1,0000**	0,6305	0,5123
Fall	0,6371	0,5747	0,9966	0,7323	**1,0000**	**1,0000**	**1,0000**	**1,0000**
Media	*0,8057*	*0,2800*	***0,9787***	*0,8367*	*0,9386*	*0,9670*	*0,8616*	*0,8519*

Tabla 5.5: Exhaustividad de los métodos competidores sobre las secuencias testeadas (mayor es mejor). Los mejores resultados de cada secuencia están remarcados en **negrita**.

Video	SOM	Ilu	FrameDiff	MFBM	GrimsonGMM	MaddalenaSOBS	WrenGA	ZivkovicGMM
Office	0,5810	**0,9866**	0,0126	0,0140	0,1326	0,0140	0,0140	0,1732
PETS2006	0,9944	0,9983	0,3952	0,9915	0,9786	0,9957	**1,0000**	**1,0000**
Highway	0,9753	**1,0000**	0,0229	**1,0000**	0,9681	0,3617	0,9787	0,9787
Pedestrians	**1,0000**	**1,0000**	**1,0000**	**1,0000**	**1,0000**	**1,0000**	**1,0000**	**1,0000**
Sofa	**0,8309**	0,8138	0,0024	0,4707	0,6772	0,1511	0,6413	0,6402
Canoe	**1,0000**	0,9266	0,0000	**1,0000**	0,0000	0,0000	0,0000	0,0000
Fountain02	0,9500	0,9951	0,0206	**1,0000**	0,9384	0,0000	**1,0000**	**1,0000**
Fall	0,5834	0,4162	0,0022	**0,8870**	0,0000	0,0000	0,0000	0,0000
Media	*0,8644*	***0,8921***	*0,1827*	*0,7954*	*0,5869*	*0,3153*	*0,5793*	*0,5990*

Tabla 5.6: Especificidad de los métodos competidores sobre las secuencias testeadas (mayor es mejor). Los mejores resultados de cada secuencia están remarcados en **negrita**.

Video	SOM	Ilu	FrameDiff	MFBM	GrimsonGMM	MaddalenaSOBS	WrenGA	ZivkovicGMM
Office	0,4190	**0,0134**	0,9874	0,9860	0,8674	0,9860	0,9860	0,8268
PETS2006	0,0056	0,0016	0,6048	0,0085	0,0214	0,0043	**0,0000**	**0,0000**
Highway	0,0247	**0,0000**	0,9770	**0,0000**	0,0319	0,6383	0,0213	0,0213
Pedestrians	**0,0000**	**0,0000**	**0,0000**	**0,0000**	**0,0000**	**0,0000**	**0,0000**	**0,0000**
Sofa	**0,1691**	0,1862	0,9976	0,5293	0,3228	0,8489	0,3587	0,3598
Canoe	**0,0000**	0,0734	1,0000	**0,0000**	1,0000	1,0000	1,0000	1,0000
Fountain02	0,0500	0,0049	0,9794	**0,0000**	0,0616	1,0000	**0,0000**	**0,0000**
Fall	0,4166	0,5838	0,9978	**0,1130**	1,0000	1,0000	1,0000	1,0000
Media	*0,1356*	***0,1079***	*0,8180*	*0,2046*	*0,4131*	*0,6847*	*0,4207*	*0,4010*

Tabla 5.7: Tasa de falsos positivos (False Positive Rate o FPR) de los métodos competidores sobre las secuencias testeadas (menor es mejor). Los mejores resultados de cada secuencia están remarcados en **negrita**.

Video	SOM	Ilu	FrameDiff	MFBM	GrimsonGMM	MaddalenaSOBS	WrenGA	ZivkovicGMM
Office	0,1456	0,7912	0,0095	0,0095	0,0155	0,0072	**0,0060**	0,0394
PETS2006	0,5285	0,9024	**0,0991**	0,2707	0,2045	0,2211	0,5940	0,5368
Highway	0,0696	0,5515	0,0018	0,1330	0,0211	**0,0000**	0,0300	0,0335
Pedestrians	0,0460	0,8794	**0,0111**	0,0778	0,0413	0,0317	0,0794	0,0587
Sofa	0,0849	0,5489	**0,0015**	0,0128	0,0263	0,0038	0,0286	0,0286
Canoe	0,0362	0,6613	**0,0000**	0,0968	**0,0000**	**0,0000**	**0,0000**	**0,0000**
Fountain02	0,2808	1,0000	0,0443	0,4384	0,1823	**0,0000**	0,3695	0,4877
Fall	0,3629	0,4253	0,0033	0,2677	**0,0000**	**0,0000**	**0,0000**	**0,0000**
Media	*0,1944*	*0,7198*	***0,0213***	*0,1633*	*0,0614*	*0,0330*	*0,1384*	*0,1481*

Tabla 5.8: Tasa de falsos negativos (False Negative Rate o FNR) de los métodos competidores sobre las secuencias testeadas (menor es mejor). Los mejores resultados de cada secuencia están remarcados en **negrita**.

Video	SOM	Ilu	FrameDiff	MFBM	GrimsonGMM	MaddalenaSOBS	WrenGA	ZivkovicGMM
Office	**20,04**	44,5	43,11	43,2725	38,4719	43,1373	43,0696	38,0663
PETS2006	34,71	47,48	20,05	20,2447	**15,6841**	16,4627	43,9377	39,7108
Highway	6,66	35,55	7,12	12,2864	**2,1969**	4,8820	2,9292	3,2547
Pedestrians	4,401	46,79	**1,099**	6,1404	3,2581	2,5063	6,2657	4,6366
Sofa	**9,28**	40,28	38,05	22,4100	14,7621	34,9489	16,3628	16,4073
Canoe	**3,502**	41,65	31,49	6,1856	36,0825	36,0825	36,0825	36,0825
Fountain02	22,81	50,12	68,23	8,9178	8,6172	79,6593	**7,5150**	9,9198
Fall	38,35	50,54	60,54	**17,4116**	60,5070	60,5070	60,5070	60,5070
Media	*17,8062*	*44,6138*	*33,7111*	***17,1086***	*22,4475*	*34,7732*	*27,0837*	*26,0731*

Tabla 5.9: Probabilidad de Clasificación Errónea (Probability of Bad Classification o PBC) de los métodos competidores sobre las secuencias testeadas (menor es mejor). Los mejores resultados de cada secuencia están remarcados en **negrita**.

Video	SOM	Ilu	FrameDiff	MFBM	GrimsonGMM	MaddalenaSOBS	WrenGA	ZivkovicGMM
Office	0,8905	**0,9511**	0,5697	0,5677	0,5974	0,5683	0,5686	0,6030
PETS2006	0,9937	0,9848	0,8559	0,9959	0,9906	0,9981	**1,0000**	**1,0000**
Highway	0,9981	**1,0000**	0,930	**1,0000**	0,9973	0,9498	0,9982	0,9982
Pedestrians	**1,0000**	**1,0000**	**1,0000**	**1,0000**	**1,0000**	**1,0000**	**1,0000**	**1,0000**
Sofa	**0,9815**	0,7823	0,6197	0,7293	0,8133	0,6290	0,7964	0,7959
Canoe	**1,0000**	0,866	0,6851	**1,0000**	0,6392	0,6392	0,6392	0,6392
Fountain02	0,9799	0,0000	0,3124	**1,0000**	0,7721	0,2034	**1,0000**	**1,0000**
Fall	0,7109	0,4906	0,3943	**0,8088**	0,3949	0,3949	0,3949	0,3949
Media	***0,9443***	*0,7593*	*0,6709*	*0,8877*	*0,7756*	*0,6728*	*0,7997*	*0,8039*

Tabla 5.10: Precisión de los métodos competidores sobre las secuencias testeadas (mayor es mejor). Los mejores resultados de cada secuencia están remarcados en **negrita**.

Video	SOM	Ilu	FrameDiff	MFBM	GrimsonGMM	MaddalenaSOBS	WrenGA	ZivkovicGMM
Office	**0,8721**	0,3425	0,7233	0,7217	0,7436	0,7228	0,7234	0,7409
PETS2006	0,6395	0,1776	0,8778	0,8420	**0,8824**	0,8750	0,5775	0,6331
Highway	0,9631	0,6192	0,9630	0,9287	**0,9880**	0,9742	0,9839	0,9821
Pedestrians	0,9764	0,2153	**0,9944**	0,9595	0,9789	0,9839	0,9587	0,9697
Sofa	**0,9471**	0,5722	0,7648	0,8389	0,8863	0,7711	0,8753	0,8750
Canoe	**0,9815**	0,487	0,8131	0,9492	0,7799	0,7799	0,7799	0,7799
Fountain02	**0,8295**	0,0000	0,4709	0,7192	0,7943	0,3381	0,7734	0,6775
Fall	0,672	0,5293	0,5651	**0,7686**	0,5662	0,5662	0,5662	0,5662
Media	***0,8601***	*0,3679*	*0,7716*	*0,8410*	*0,8274*	*0,7514*	*0,7798*	*0,7781*

Tabla 5.11: F-medida de los métodos competidores sobre las secuencias testeadas (mayor es mejor). Los mejores resultados de cada secuencia están remarcados en **negrita**.

La exhaustividad, la especificidad, la tasa de falsos positivos (FPR) y la tasa de falsos negativos (FNR) de los métodos competidores sobre las secuencias probadas se muestran en las Tablas 5.5, 5.6, 5.7 y 5.8, respectivamente. Por último, la probabilidad de clasificación errónea (PCB), la precisión y la F-medida de los métodos probados sobre los videos probados se detallan en las Tablas 5.9, 5.10 y 5.11, respectivamente. De todas estas medidas seleccionadas, la F-medida se puede considerar como una evaluación global fiable de un método, ya que caracteriza el rendimiento de un clasificador en el espacio detección-precisión (Bouwmans, 2014a). Como se puede observar el modelo propuesto obtiene el mejor resultado medio en términos de F-medida. Como se ve en las Tablas 5.7 y 5.8, Ilu logra una baja tasa de falsos positivos, pero tiene una gran tasa de falsos negativos, y este desequilibrio obstaculiza su desempeño (Tabla 5.11). Por otro lado, FrameDiff tiene una baja tasa de falsos negativos, pero tiene una tasa alta de falsos positivos, y nuevamente esto produce un rendimiento general bastante malo.

Otro aspecto a destacar es que los métodos seleccionados de la literatura basados en píxeles (las últimas cinco columnas de las Tablas 5.3-5.11) obtienen resultados similares en la mayoría de casos. Esto ocurre porque estos métodos están diseñados para detectar objetos de primer plano a nivel de píxel, que es diferente de la detección de movimiento a nivel de fotograma. La detección de movimiento a nivel de fotograma es un problema más simple, por lo que la precisión a nivel de píxel no es necesaria para lograr un buen rendimiento. También se debe mencionar que las características inherentes a cada video probado tienen un gran impacto en los resultados.

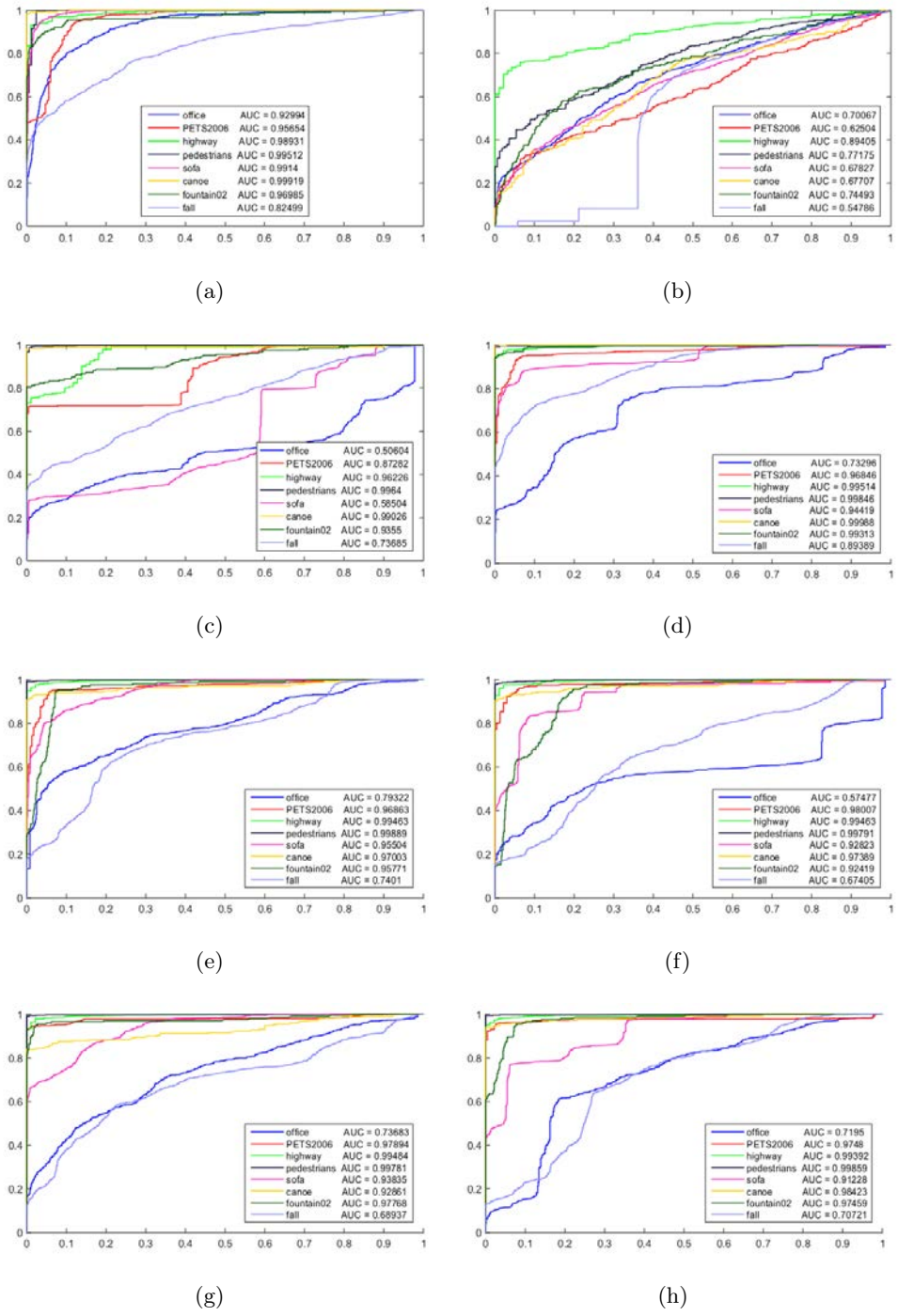


Figura 5.7: Curvas ROC correspondientes a los ocho videos de referencia analizados por los diferentes métodos probados. La primera fila muestra el sistema propuesto (a) y el tradicional detector (b). FrameDiff (c) y MFBM (d) se encuentran en la segunda fila. La tercera fila muestra las curvas ROC de los métodos GrimsonGMM (e) y MaddalenaSOBS (f). Por último, la cuarta fila presenta los métodos WrenGA (g) y ZivkovicGMM (h). El eje x se corresponde con la tasa de falsos positivos, mientras que el eje y se corresponde con la tasa de verdaderos positivos. Sus correspondientes áreas bajo la curva (AUC, mayor es mejor) se muestra en la leyenda dentro de las gráficas.

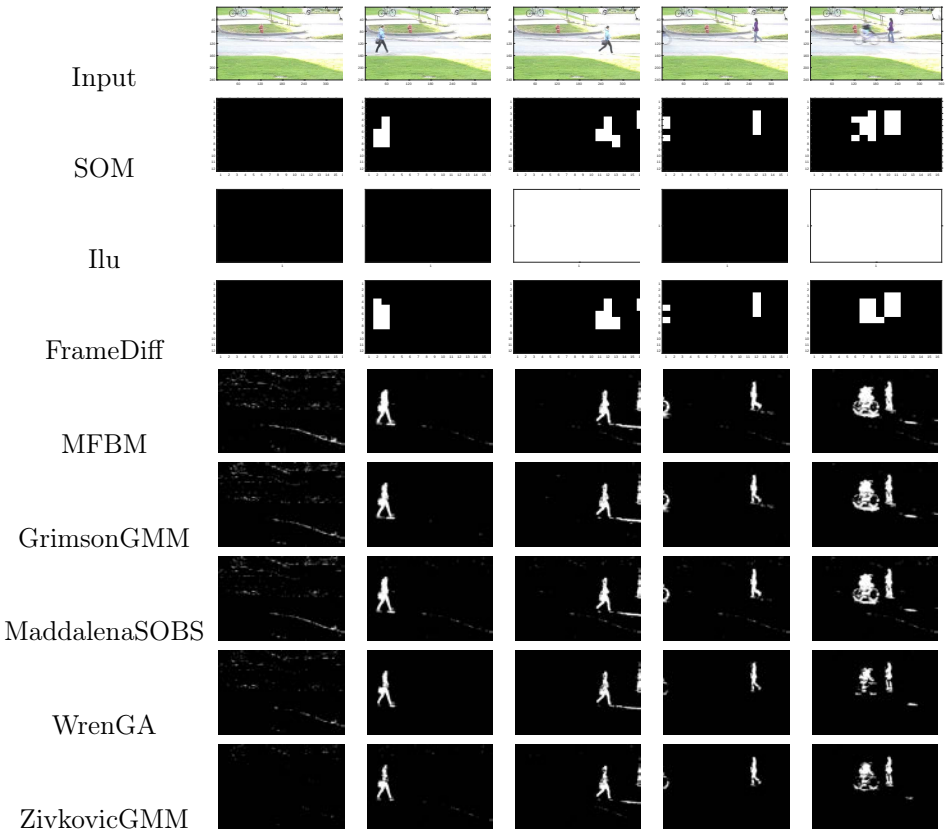


Figura 5.8: Ejemplos de detección de movimiento para el video Pedestrians. Primera fila: video RGB en bruto. Siguientes filas: decisión de detección por el sistema propuesto (SOM), el tradicional detector (Ilu), FrameDiff, MFBM, GrimsonGMM, MaddalenaSOBS, WrenGA y ZivkovicGMM, respectivamente.

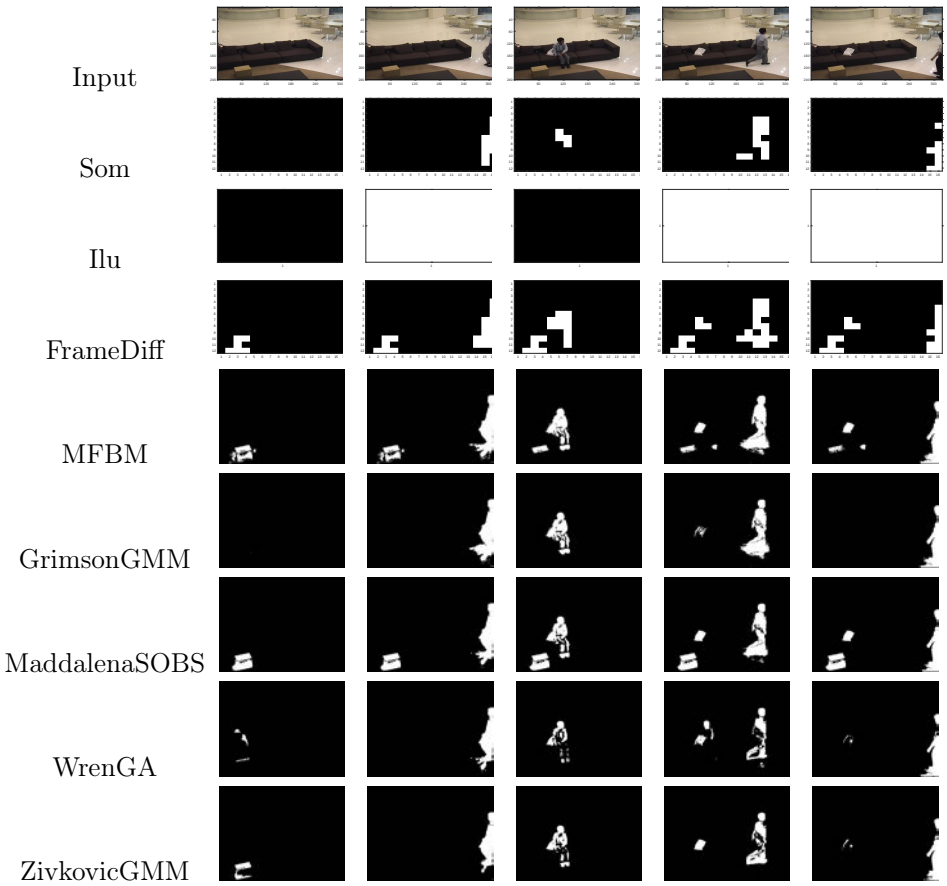


Figura 5.9: Ejemplos de detección de movimiento para el video Sofa. Primera fila: video RGB en bruto. Siguientes filas: decisión de detección por el sistema propuesto (SOM), el tradicional detector (Ilu), FrameDiff, MFBM, GrimsonGMM, MaddalenaSOBS, WrenGA y ZivkovicGMM, respectivamente.

Para tener una evaluación más precisa del rendimiento cuantitativo de las propuestas, la Figura 5.7 muestra las curvas de funcionamiento del receptor (receiver operating curves o ROC) para el sistema propuesto (a), los detectores tradicionales y de diferencia de fotogramas ejecutados en Arduino (b-c) y los métodos competidores ejecutados en un ordenador estándar (d-g) para todos los videos. Las gráficas representan la dependencia entre la tasa de verdaderos positivos (TPR, mayor es mejor), también conocida como exhaustividad, y la tasa de falsos positivos (FPR, menor es mejor), en varios ajustes de umbral. Nótese que una clasificación perfecta correspondería a la esquina superior izquierda de las gráficas. También se ha calculado el área bajo la curva (Area Under Curve o AUC, mayor es mejor) como una medida de la calidad de un clasificador binario, ya que es la probabilidad de que

un caso positivo seleccionado aleatoriamente reciba una puntuación mayor que un caso negativo seleccionado aleatoriamente. La propuesta SOM logra muchos mejores resultados que los otros detectores basados en Arduino, mientras que su rendimiento es similar con respecto a aquellos basados en ordenadores.

Por último, desde un punto de vista cualitativo, las Figuras 5.8 y 5.9 describen las decisiones de detección por cada método competidor en varios fotogramas de los videos "Pedestrians" y "Sofa", respectivamente. La primera fila muestra cinco fotogramas de la secuencia real capturada por la cámara de video, y la segunda fila muestra la decisión de detección para cada bloque en esos cinco fotogramas para el sistema propuesto de acuerdo a la ecuación (5.14). La tercera fila muestra la decisión para un sensor tradicional ejecutado en una placa Arduino, teniendo en cuenta que un fotograma en negro significa que no se ha detectado movimiento (clase negativa) y que un fotograma entero en blanco significa que se ha detectado movimiento (clase positiva). La cuarta fila se corresponde con el algoritmo de la diferencia de fotogramas ejecutado en Arduino. Las siguientes filas representan la salida de los algoritmos basados a nivel de píxel en un ordenador estándar. Como se ve, el método propuesto supera a los métodos ejecutados en Arduino, mientras que es competitivo con respecto a los métodos de ordenador. Esto confirma lo que anteriormente se ha comentado en los resultados cuantitativos.

5.6. Conclusiones

El algoritmo SOM ha sido implementado con éxito en un microcontrolador Arduino DUE. El SOM ha sido adaptado para superar las limitaciones impuestas por los recursos de memoria y velocidad de cómputo del dispositivo hardware. La implementación correcta del algoritmo ha sido verificada, y se ha detectado que, como la precisión se incrementa para evitar efectos de redondeo, el microcontrolador necesita más tamaño de memoria. Además, se ha llevado a cabo un detallado estudio de las diferencias de usar las representaciones de punto flotante o punto fijo, concluyendo que los mejores resultados pueden ser obtenidos con una representación de punto fijo de 32 bits destacando el cálculo 10 veces más rápido en la arquitectura neuronal más grande (más neuronas) que usando la tradicional representación de punto flotante. El cambio del paradigma de la representación del tipo de datos también permite usar arquitecturas SOM con más neuronas y procesar imágenes con más resolución, obteniendo en el sistema propuesto una detección de movimiento basada en bloques más precisa.

La implementación del algoritmo SOM ha sido empleada como un detector de movimiento obteniéndose un sistema versátil y barato con el que es posible ejecutar videovigilancia de forma eficiente. El proceso de aprendizaje ha sido implementado en el chip, por lo que el procedimiento de toma de

decisiones del detector está adaptado en tiempo real a los cambios observados en la escena. De esta forma los errores de decisión producidos por la evolución del ambiente capturado son reducidos de forma significativa.

La eficiencia del sistema propuesto es significativamente mayor que otros métodos de detección de movimiento tradicionales. Tiene una alta tasa de éxito con bajos falsos positivos. Éstos pueden ser reducidos a casi 0 ajustando el umbral, debido a la favorable dependencia entre las tasas de verdaderos y falsos positivos mostradas en las curvas ROC.

Como conclusión general, se ha mostrado la idoneidad del algoritmo SOM para su aplicación en una detección de movimiento usando un microcontrolador Arduino DUE. Por tanto, el presente estudio demuestra el potencial de la metodología propuesta para su aplicación a sistemas económicos en escenarios reales. El rendimiento de detección se puede mejorar aún más mediante el empleo de recursos informáticos y dispositivos ad hoc más potentes que la placa del microcontrolador que se considera aquí.

Capítulo 6

Redes neuronales superficiales y profundas para la clasificación de células sanguíneas

Lo que sabemos es una gota de agua; lo que ignoramos es el océano.

Isaac Newton

Resumen: El diagnóstico temprano de algunas enfermedades depende fuertemente de la precisión de la detección de glóbulos rojos en muestras sanguíneas. En la mayoría de los casos, las técnicas de procesamiento de imágenes aceleran y mejoran la precisión de esta detección. En este capítulo, la transformada del círculo de Hough es empleada para la detección de glóbulos rojos. Tras esto, se proponen redes neuronales artificiales para la clasificación de las células detectadas. Además, se lleva a cabo un estudio sobre los méritos relativos de las arquitecturas neuronales superficiales y profundas para esta tarea de clasificación. Específicamente, la aplicación de redes neuronales (MLP) como una técnica de clasificación estándar es comparada con nuevas propuestas relacionadas con el aprendizaje profundo como las redes neuronales convolucionales (CNNs). Los diferentes experimentos ejecutados revelan el alto ratio de clasificación de las propuestas, y muestran resultados prometedores tras la aplicación de las CNNs.

6.1. Introducción

El procesamiento de imágenes digitales es de importancia en varios campos de la medicina, desde imágenes que son obtenidas en test médicos como las imágenes de rayos X, tomografía computarizada, resonancia magnética, imagen de ultrasonido e imagen de medicina nuclear, a imágenes que son obtenidas en laboratorio con microscopio. En este sentido, el procesamiento de imágenes digitales permite procesar la imagen para obtener una mejor visibilidad, para enfatizar las partes requeridas o para hacer análisis y predicciones. Además, la segmentación de imágenes a través del procesamiento es esencial para la detección de patologías (McAuliffe et al., 2001; Davis y Boyers, 1992; Nogueira y Teófilo, 2014; Vishnuvarthanan et al., 2018).

Actualmente, la mayoría de las imágenes obtenidas en laboratorio con microscopio son digitalizadas tras esto y procesadas por ordenadores para hacer más fácil y más rápido el proceso de análisis de la imagen. Además de otros campos, la microscopía óptica de sangre ofrece muestras de imágenes cuyo estudio puede aportar información muy valiosa sobre la salud de los pacientes. Las técnicas aplicadas en hematología para contar las células sanguíneas en muestras de sangre, como las técnicas de centrifugación o los contadores de células hematológicas, son imprecisos e inexactos en la mayoría de los casos. Por tanto, el resultado puede cambiar significativamente de acuerdo a la técnica de medida utilizada (Cornbleet, 2016; Imeri et al., 2008). Además, los contadores automáticos de células se utlizan con menos frecuencia que los manuales en los laboratorios. Es por ello que las operaciones en el laboratorio deben ser optimizadas por contadores automáticos de células, considerándose los manuales para propósitos de validación (Lantis et al., 2003).

Durante un largo tiempo, la mayoría de las investigaciones han sido ejecutadas para desarrollar herramientas o técnicas que reducen el error en los métodos manuales, el coste y la lentitud en el conteo de células sanguíneas para la detección de patologías (Krause, 1990; Schmitt et al., 1992). En una muestra de sangre, el porcentaje ocupado por los glóbulos rojos en relación con el total de células es conocido como hematocrito. El valor de hematocrito es muy importante para la detección temprana de varias enfermedades. Un incremento apreciable o reducción del hematocrito puede ser un indicador de enfermedad como anemia. En muchos casos, estas enfermedades no son graves pero podrían causar otros problemas severos. Por tanto, una detección temprana sería vital (Wennecke, 2004; Tang et al., 2000; Ali et al., 2018; Balafar, 2014).

En este capítulo se realiza un análisis de imágenes de microscopía óptica de sangre detectando los glóbulos rojos (red blood cells o RBC) en muestras sanguíneas usando la transformada de Hough y redes neuronales. La técnica de la transformada de Hough proporciona la detección de cada objeto o

célula. Esta técnica es usada frecuentemente en procesamiento de imágenes debido a su eficacia y baja complejidad para la detección de formas y características. En el presente, es útil en estudios de imágenes médicas (Ecabert y Thiran, 2004; Philip et al., 1994; Bagui y Zoueu, 2014; Mazalan et al., 2013; Molina-Cabello et al., 2018).

Las redes neuronales han sido utilizadas para una amplia variedad de tareas en medicina, desde imágenes médicas, procesamiento de señales a investigaciones biomédicas. En particular, los perceptrones multicapa (multilayer perceptrons o MLP) son considerados como aproximadores universales por lo que pueden adaptarse para resolver diferentes problemas reales dentro del campo de las imágenes médicas (Clark, 1991; Egbert et al., 1990; Baxt, 1995). En esta propuesta, el método clasifica las células detectadas entre glóbulos rojos y otros elementos como glóbulos blancos u objetos espurios, entre otros. Recientemente, las redes neuronales convolucionales, que son una evolución del MLP, han sido aplicadas con éxito al reconocimiento de imágenes y clasificación con mayor tasa de precisión que técnicas tradicionales (Ortiz et al., 2016). Es por ello por lo que en este capítulo se realiza una comparativa entre los dos tipos de redes neuronales (MLP y CNN).

6.2. Metodología

El algoritmo propuesto tiene una primera etapa de detección donde los glóbulos rojos son extraídos de la imagen, seguido de un procedimiento de clasificación, como se muestra en la Figura 6.1. En la Subsección 6.2.1 se detalla el modelo de detección, mientras que la etapa de clasificación donde los glóbulos rojos son seleccionados se describe en la Subsección 6.2.2.

Figura 6.1: Esquema del funcionamiento del algoritmo propuesto.

6.2.1. Detección de los glóbulos rojos

La detección de los glóbulos rojos que aparecen en las imágenes está basada en la transformada del círculo de Hough (Circle Hough Transform o CHT). Esta técnica es usada para detectar las formas circulares en una imagen y se usa ya que la geometría de los glóbulos rojos es parecida a un círculo. La CHT es calculada como sigue. Como se sabe, la ecuación para un

círculo en dos dimensiones es:

$$(x-a)^2 + (y-b)^2 = r^2 \tag{6.1}$$

donde (a, b) es el centro del círculo, y $r > 0$ es el radio. La CHT toma como entrada un conjunto de N puntos de borde en la imagen:

$$\mathcal{S} = \left\{(x_i, y_i) \in \mathbb{R}^2 \,|\, i \in \{1, ..., N\}\right\} \tag{6.2}$$

Cada círculo tentativo viene dado por sus tres parámetros $(a, b, r) \in \mathbb{N}^3$, y una puntuación es calculada como el recuento de puntos de borde que se encuentran en ese círculo. Para este propósito, la ecuación paramétrica del círculo se considera:

$$(x, y) = (a + r\cos\theta, b + r\sin\theta) \tag{6.3}$$

Reordenando la ecuación de la siguiente manera:

$$(a, b) = (x + r\cos\theta, y + r\sin\theta) \tag{6.4}$$

Por último, para cada punto de borde en $\mathcal{S}$, cada valor tentativo del radio r, y cada ángulo θ, se lleva a cabo una votación para el círculo tentativo resultante (a, b, r), dado por (6.4). Los círculos tentativos con el mayor número de votos son declarados como círculos detectados.

La escala de la imagen es desconocida, por lo que una primera búsqueda es ejecutada para tener una aproximación de los radios r de las diferentes células sanguíneas presentes en la imagen. El tamaño de la imagen es reescalado aplicando un factor de 10 veces para reducirla, es decir, el tamaño en píxeles de la imagen es reducido 100 veces, 10 veces en cada dimensión. Después, se utiliza un rango de radios entre 2 y 10 píxeles para buscar círculos en la imagen reducida, y se cuenta el número de círculos detectados con cada uno de esos radios. El radio que presenta una mayor acumulación de círculos es seleccionado para la segunda búsqueda, que se realiza en mayor profundidad.

En esta fase, la CHT es aplicada nuevamente en la imagen original (no la imagen reducida) para hacer una nueva búsqueda de círculos. Se aplica un rango alrededor del radio más frecuente previamente encontrado, incrementando su tamaño en 10 para ajustarlo al tamaño original, siendo este rango un intervalo de radio tentativo. En particular, se define un rango de tamaño 10 alrededor del radio más frecuente.

6.2.2. Clasificación de glóbulos rojos

Después de que las células sanguíneas sean detectadas, dos propuestas son consideradas para clasificarlas en glóbulos rojos u otro tipo de células sanguíneas.

El primer modelo predictivo está basado en el perceptrón multicapa (MLP), que es una red neuronal superficial que puede ser empleada para tareas de clasificación. En este diseño, la red MLP está compuesta de una capa de entrada, donde cada patrón de entrada es una tupla representando el color mediano de cada célula detectada (es decir, tiene tres valores que pertenecen al valor RGB), una capa oculta con M neuronas y una capa de salida con una neurona cuyo valor representa la probabilidad de ser glóbulo rojo o no. El número de neuronas de la capa oculta ha sido fijado a $M = 6$. Se ha descubierto empíricamente que el rendimiento de clasificación no tiene diferencias significativas cuando este parámetro se incrementa.

Por otro lado, tres redes neuronales convolucionales profundas han sido analizadas: *Alexnet* (Krizhevsky et al., 2012), *GoogLeNet* (Szegedy et al., 2015) y *ResNet-50* (He et al., 2016). Estos modelos, cada uno con una diferente arquitectura, fueron desarrollados usando varias capas convolucionales y otras totalmente conectadas, y su objetivo era clasificar el conjunto de datos IMAGENET[1]. En este caso, la entrada de la red son las imágenes asociadas a las células detectadas, donde la capa de salida tiene 2 neuronas para clasificar las imágenes en 2 categorías: glóbulo rojo y no glóbulo rojo.

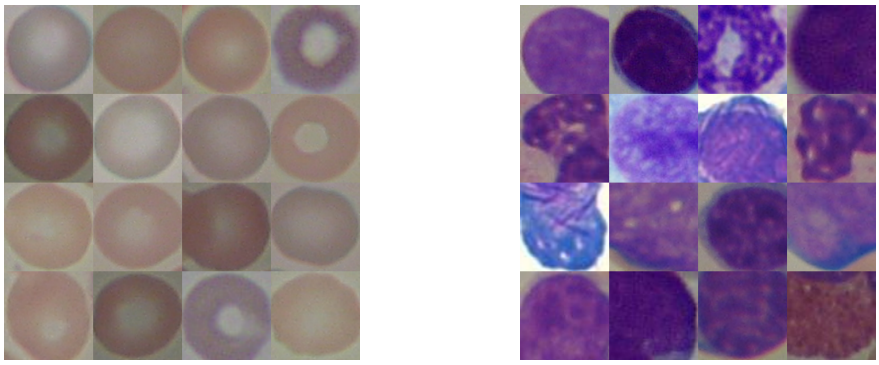
En ambos modelos, se ha considerado un conjunto balanceado de muestras de entrada para entrenar la red, donde solo dos clases son posibles: glóbulo rojo u otro caso. Las imágenes han sido seleccionadas de los conjuntos de imágenes considerados y cada una de ellas muestra diferentes células sanguíneas. En el caso de las CNNs, la arquitectura de red necesita un tamaño de imagen de 227x227 píxeles en el caso de *Alexnet*, y 224x224 píxeles en los casos de *GoogLeNet* y *ResNet-50*. Sin embargo, las muestras de células no tienen los tamaños requeridos. Para resolver este problema, se ha aplicado un proceso de redimensionamiento. Algunas diferentes muestras de estas imágenes se pueden ver en la Figura 6.2.

Además, ya que la escala de las células sanguíneas no es siempre la misma en todas las imágenes, se necesita el tamaño real de las células para compensar estos efectos. Por tanto, el tamaño real de cada célula tiene que ser calculado. Se sabe que el radio medio real de un glóbulo rojo es de 3.9 micrómetros aproximadamente, es decir, un diámetro medio de 7.8 micrómetros (Fedosov et al., 2010).

6.3. Experimentos

En esta sección, la metodología propuesta es aplicada a un conjunto de imágenes de muestras sanguíneas de tamaños heterogéneos. Diez imágenes han sido escogidas para evaluar la propuesta. Estas imágenes muestran varias células sanguíneas. También se consideran imágenes con diferente ilumina-

[1] `http://www.image-net.org`



(a) (b)

Figura 6.2: Ejemplo de imágenes de células sanguíneas utilizadas para el entrenamiento de las redes neuronales. El proceso de redimensionamiento ya está aplicado en cada región mostrada. (a) muestra glóbulos rojos y (b) exhibe células que no son glóbulos rojos.

ción, varias escalas y objetos que se solapan. Se han obtenido del conjunto de datos ASH Image Bank[2]. Las imágenes seleccionadas de este repositorio son nombradas como siguen: *1050*, *1051*, *1068*, *1071*, *1072*, *60095*, *60475*, *60550*, *60802* y *60974*.

Además, se han considerado otras 10 imágenes del conjunto de datos Acute Lymphoblastic Leukemia Image Database for Image Processing (ALL-IDB) [3]. En este caso las imágenes seleccionadas de este repositorio son: *Im021_1*, *Im024_1*, *Im035_0*, *Im037_0*, *Im048_1*, *Im049_1*, *Im067_0*, *Im077_0*, *Im085_0* y *Im104_0*.

Por otro lado, para entrenar las redes, se han etiquetado varias imágenes que han sido tomadas del conjunto de datos utilizado. Sin embargo, ya que no hay suficientes células que no son glóbulos rojos presentes en las imágenes, se han aplicado técnicas de aumento de datos (data augmentation) para incrementar su cantidad.

Además, el centroide de cada glóbulo rojo de cada imagen ha sido etiquetado manualmente para testear el rendimiento de la detección y clasificación de este tipo de células, usando esta información como máscara de verdad (ground truth).

Para el entrenamiento, el 90 por ciento de los datos forman el conjunto de entrenamiento y el 10 por ciento restante comprenden el conjunto de test, siguiendo una estrategia de 10 pliegues. Este proceso se repite 10 veces.

[2] `http://imagebank.hematology.org`
[3] `http://crema.di.unimi.it/~fscotti/all/`

Cada vez el orden de las imágenes es aleatoriamente seleccionado y también las imágenes de células de entrenamiento correspondiente a una imagen son elegidas aleatoriamente. Se han empleado 20 imágenes de glóbulos rojos y otras 20 de otro tipo de células en cada caso. Después del entrenamiento de las redes, el conjunto de test se utiliza para medir el rendimiento de la propuesta. Se consigue un rendimiento robusto de la propuesta, tal y como muestran los resultados.

Desde un punto de vista cuantitativo, se han seleccionado varias medidas bien conocidas para comparar el rendimiento de la detección y la clasificación de las células de la sangre. Se consideran la exactitud espacial (S), la exactitud (Acc) y la F-medida (Fm). Estas medidas proporcionan valores en el intervalo $[0, 1]$, donde un mayor valor es mejor, y representan el porcentaje de aciertos del sistema.

Verdaderos positivos (TP), verdaderos negativos (TN), falsos negativos (FN), falsos positivos (FP), la precision (PR), la exhaustividad (RC), la especificidad (SP) y el porcentaje de clasificación errónea (PWC) también son utilizados. Además de todas estas medidas, la exactitud espacial, la exactitud y la F-medida proporcionan una buena evaluación del rendimiento general de un método dado, mientras que los FN deben ser considerados contra los FP (más bajo es mejor), y PR contra RC (mayor es mejor).

La definición de cada medida puede ser descrita como sigue:

$$S = \frac{TP}{TP + FN + FP} \qquad Acc = \frac{TP + TN}{TP + FP + FN + TN} \tag{6.5}$$

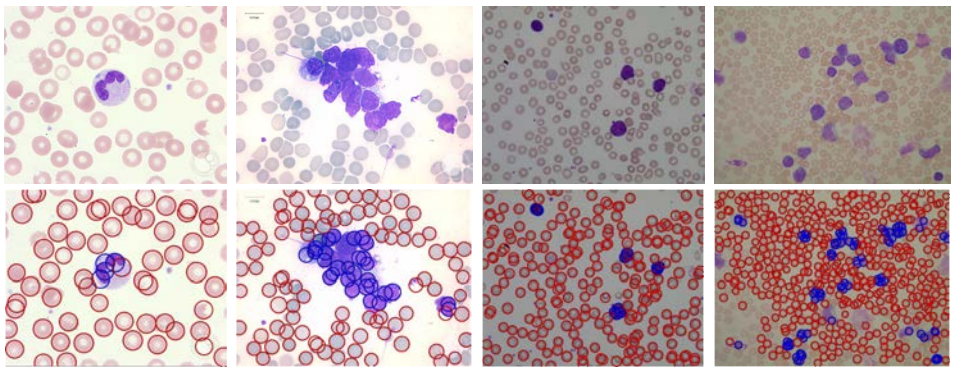
$$RC = \frac{TP}{TP + FN} \qquad PR = \frac{TP}{TP + FP} \qquad SP = \frac{TN}{FP + TN} \tag{6.6}$$

$$FNR = \frac{FN}{TP + FN} \qquad FPR = \frac{FP}{FP + TN} \tag{6.7}$$

$$\text{F-m} = 2 * \frac{PR * RC}{PR + RC} \tag{6.8}$$

$$PWC = 100 * \frac{FN + FP}{TP + FP + TN + FN} \tag{6.9}$$

Los centroides de las máscaras de verdad contra los centroides de las células estimadas han sido comparados para comprobar el rendimiento de la detección de glóbulos rojos. Se ha considerado que un centroide estimado está correctamente detectado si tiene una distancia de 1.5 micrómetros del centroide de la máscara de verdad. Se ha decidido este tipo de comparación porque se puede estimar la escala de la imagen de test y es bien conocido que el radio de un glóbulo rojo es 3.9 micrómetros (Fedosov et al., 2010). Además, solo se han considerado aquellos centroides de la máscara de verdad



(a) (b) (c) (d)

Figura 6.3: Detección y clasificación de glóbulos rojos. La primera fila muestra las imágenes en bruto y la segunda fila exhibe el resultado proporcionado por la propuesta. Cada columna presenta una imagen del conjunto de datos probado. Las columnas (a), (b), (c) y (d) corresponden a las imágenes *1072*, *60550*, *Im024_1* y *Im048_1*, respectivamente. Cada célula detectada es indicada con un círculo y el color muestra su clasificación: rojo para los glóbulos rojos y azul para otro tipo de células.

que aparecen dentro de la imagen. De acuerdo a estas restricciones, el test de detección no considera la medida TN. Por otro lado, el test del rendimiento de la clasificación se considera solo para las células detectadas, es decir, las células sanguíneas que no son detectadas por la CHT no son consideradas.

Tabla 6.1: Rendimiento de la detección de glóbulos rojos. La primera columna indica la imagen del conjunto de datos y el resto de columnas muestran el rendimiento medio logrado por la propuesta para diferentes medidas. Cada fila representa una imagen testeada del conjunto de datos y su rendimiento medio, y la última fila muestra la media de los valores de cada medida.

Imagen	TP	FP	FN	RC	FNR	PR	Fm	S
1050	63	7	2	0,969	0,031	0,900	0,933	0,875
1051	70	9	26	0,729	0,271	0,886	0,800	0,667
1068	129	50	12	0,915	0,085	0,721	0,806	0,675
1071	55	10	7	0,887	0,113	0,846	0,866	0,764
1072	52	11	4	0,929	0,071	0,825	0,874	0,776
60095	41	18	1	0,976	0,024	0,695	0,812	0,683
60475	199	32	11	0,948	0,052	0,861	0,902	0,822
60550	82	40	6	0,932	0,068	0,672	0,781	0,641
60802	77	24	4	0,951	0,049	0,762	0,846	0,733
60974	97	27	5	0,951	0,049	0,782	0,858	0,752
Im021_1	194	121	9	0,956	0,044	0,616	0,749	0,599
Im024_1	196	42	7	0,966	0,034	0,824	0,889	0,800
Im035_0	513	50	14	0,973	0,027	0,911	0,941	0,889
Im037_0	87	4	22	0,798	0,202	0,956	0,870	0,770
Im048_1	382	112	112	0,773	0,227	0,773	0,773	0,630
Im049_1	321	89	90	0,781	0,219	0,783	0,782	0,642
Im067_0	526	35	22	0,960	0,040	0,938	0,949	0,902
Im077_0	410	28	6	0,986	0,014	0,936	0,960	0,923
Im085_0	317	109	14	0,958	0,042	0,744	0,838	0,720
Im104_0	548	46	26	0,955	0,045	0,923	0,938	0,884
Media				0,915	0,085	0,818	0,858	0,757

La detección y clasificación de los glóbulos rojos de la propuesta muestra un buen rendimiento desde un punto de vista cualitativo, como se ve en la Figura 6.3. Se puede apreciar cómo la propuesta considera varias células como glóbulos rojos en la etapa de detección. Sin embargo, tras la aplicación del proceso de clasificación, la propuesta corrige estas predicciones. También debe destacarse la detección de células que se solapan o aquellas que aparecen en el borde de la imagen. Además, otro problema puede observarse cuando la propuesta detecta dos (o más) centroides para un único glóbulo rojo.

Desde un punto de visto cuantitativo, el rendimiento de la detección de la propuesta se muestra en la Tabla 6.1. La media de la detección de glóbulos rojos es 0.76, principalmente porque la CHT no solo detecta glóbulos rojos, sino también cualquier elemento circular que esté presente en la imagen. Por tanto, es posible detectar varios glóbulos rojos en la ubicación de

glóbulos blancos más grandes, por tanto es esencial tener una fase posterior que detecte correctamente o clasifique estos círculos como glóbulos rojos.

El rendimiento del modelo de clasificación MLP y la mejor CNN de acuerdo a su exactitud media se muestra en las Tablas 6.2 y 6.3, respectivamente. Además, la exactitud de los cuatro modelos es comparada, como se puede observar en la Figura 6.4. Aunque todos los modelos obtienen muy buenos ratios de clasificación de glóbulos rojos, en general, las propuestas CNN superan levemente los resultados logrados por la red neuronal superficial. Esto sucede porque al proporcionar más información de entrada (la imagen completa de cada círculo en lugar de un vector con tres elementos con el color medio) la CNN puede discernir mejor qué círculo corresponde a un glóbulo rojo, a pesar del considerable incremento de complejidad de esta segunda propuesta. Posiblemente, el desarrollo de CNNs adaptadas a este problema específico logra mejorar el resultado de la clasificación en imágenes con tamaños de células más pequeños (imagen *60974*) y con iluminación pobre (imagen *1068*).

Tabla 6.2: Rendimiento de la clasificación de glóbulos rojos empleando la red neuronal superficial básica (MLP). La primera columna indica la imagen del conjunto de datos y las restantes columnas muestran el rendimiento medio logrado por la propuesta para diferentes medidas. Cada fila representa una imagen testeada del conjunto de datos y sus rendimientos medios, y la última fila muestra la media de los valores de cada medida.

Imagen	TP	TN	FP	FN	RC	SP	FPR	FNR	PWC	PR	Fm	S	Acc
1050	65	5	0	0	1,000	1,000	0,000	0,000	0,000	1,000	1,000	1,000	1,000
1051	73	6	0	0	1,000	1,000	0,000	0,000	0,000	1,000	1,000	1,000	1,000
1068	166	1,6	0	13,4	0,925	1,000	0,000	0,075	7,403	1,000	0,961	0,925	0,926
1071	58	6,1	0	0,9	0,985	1,000	0,000	0,015	1,385	1,000	0,992	0,985	0,986
1072	57	3,9	0	2,1	0,964	1,000	0,000	0,036	3,333	1,000	0,982	0,964	0,967
60095	50	10	1	0	1,000	0,909	0,091	0,000	1,639	0,980	0,990	0,980	0,984
60475	231	2	0	1	0,996	1,000	0,000	0,004	0,427	1,000	0,998	0,996	**0,996**
60550	91	31	0	2	0,978	1,000	0,000	0,022	1,613	1,000	0,989	0,978	**0,984**
60802	91	11	0	0	1,000	1,000	0,000	0,000	0,000	1,000	1,000	1,000	1,000
60974	123	3	0	0	1,000	1,000	0,000	0,000	0,000	1,000	1,000	1,000	1,000
Im021_1	302	15	0	7	0,977	1,000	0,000	0,023	2,160	1,000	0,989	0,977	0,978
Im024_1	226	11	0	1	0,996	1,000	0,000	0,004	0,420	1,000	0,998	0,996	**0,996**
Im035_0	562	2	0	0	1,000	1,000	0,000	0,000	0,000	1,000	1,000	1,000	1,000
Im037_0	89	2	0	0	1,000	1,000	0,000	0,000	0,000	1,000	1,000	1,000	1,000
Im048_1	406	84,7	0	4,3	0,990	1,000	0,000	0,010	0,869	1,000	0,995	0,990	0,991
Im049_1	343	66,7	0	1,3	0,996	1,000	0,000	0,004	0,316	1,000	0,998	0,996	**0,997**
Im067_0	557	5	0	2	0,996	1,000	0,000	0,004	0,355	1,000	0,998	0,996	0,996
Im077_0	440	2	0	0	1,000	1,000	0,000	0,000	0,000	1,000	1,000	1,000	1,000
Im085_0	421	6	0	1	0,998	1,000	0,000	0,002	0,234	1,000	0,999	0,998	0,998
Im104_0	587,6	6,6	1,4	0,4	0,999	0,856	0,144	0,001	0,302	0,998	0,998	0,997	0,997
Media					0,993	0,954	0,046	0,007	0,894	0,997	0,995	0,990	0,991

Tabla 6.3: Rendimiento de la clasificación de glóbulos rojos empleando la red neuronal convolucional profunda Alexnet. La primera columna indica la imagen del conjunto de datos y las restantes columnas muestran el rendimiento medio logrado por la propuesta para diferentes medidas. Cada fila representa una imagen testeada del conjunto de datos y sus rendimientos medios, y la última fila muestra la media de los valores de cada medida.

Imagen	TP	TN	FP	FN	RC	SP	FPR	FNR	PWC	PR	Fm	S	Acc
1050	65	5	0	0	1,000	1,000	0,000	0,000	0,000	1,000	1,000	1,000	1,000
1051	73	6	0	0	1,000	1,000	0,000	0,000	0,000	1,000	1,000	1,000	1,000
1068	166	3	0	12	0,933	1,000	0,000	0,067	6,630	1,000	0,965	0,933	**0,934**
1071	58	7	0	0	1,000	1,000	0,000	0,000	0,000	1,000	1,000	1,000	**1,000**
1072	57	4,8	0	1,2	0,979	1,000	0,000	0,021	1,905	1,000	0,990	0,979	**0,981**
60095	50	10	1	0	1,000	0,909	0,091	0,000	1,639	0,980	0,990	0,980	0,984
60475	229,7	2	1,3	1	0,996	0,673	0,327	0,004	0,983	0,994	0,995	0,990	0,990
60550	91	30,9	0	2,1	0,977	1,000	0,000	0,023	1,694	1,000	0,989	0,977	0,983
60802	91	11	0	0	1,000	1,000	0,000	0,000	0,000	1,000	1,000	1,000	1,000
60974	123	3	0	0	1,000	1,000	0,000	0,000	0,000	1,000	1,000	1,000	1,000
Im021_1	302	15	0	7	0,977	1,000	0,000	0,023	2,160	1,000	0,989	0,977	0,978
Im024_1	225,9	11	0,1	1	0,996	0,992	0,008	0,004	0,462	1,000	0,998	0,995	0,995
Im035_0	562	2	0	0	1,000	1,000	0,000	0,000	0,000	1,000	1,000	1,000	1,000
Im037_0	89	2	0	0	1,000	1,000	0,000	0,000	0,000	1,000	1,000	1,000	1,000
Im048_1	406	86	0	3	0,993	1,000	0,000	0,007	0,606	1,000	0,996	0,993	**0,994**
Im049_1	341,5	66,9	1,5	1,1	0,997	0,978	0,022	0,003	0,633	0,996	0,996	0,992	0,994
Im067_0	556,7	6,3	0,3	0,7	0,999	0,965	0,035	0,001	0,177	0,999	0,999	0,998	**0,998**
Im077_0	440	2	0	0	1,000	1,000	0,000	0,000	0,000	1,000	1,000	1,000	1,000
Im085_0	421	6,8	0	0,2	1,000	1,000	0,000	0,000	0,047	1,000	1,000	1,000	**1,000**
Im104_0	589	7	0	0	1,000	1,000	0,000	0,000	0,000	1,000	1,000	1,000	**1,000**
Media					0,992	0,976	0,024	0,008	0,847	0,998	0,995	0,991	**0,992**

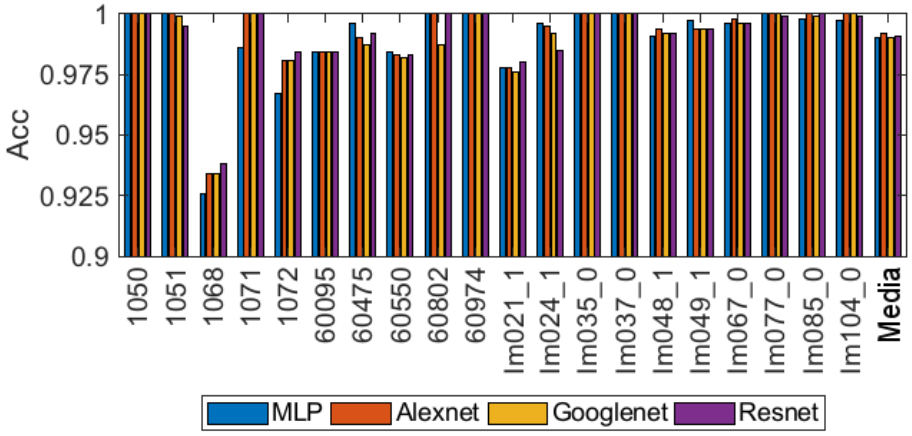


Figura 6.4: Comparación de la exactitud entre el rendimiento logrado por la red neuronal superficial básica y las redes neuronales convolucionales profundas en cada imagen del conjunto de datos. Las barras del total representan la exactitud media obtenida.

6.4. Conclusiones

Se ha presentado una metodología para la detección de glóbulos rojos en imágenes médicas basada en el uso de la transformada del círculo de Hough (CHT) para la detección y redes neuronales para la clasificación. El módulo de clasificación es necesario ya que no todos los círculos detectados corresponden a glóbulos rojos, lo que implica un proceso de identificación para asegurar qué círculo corresponde a un glóbulo rojo o no. Específicamente, dos tipos de alternativas han sido probadas para esta clasificación, una red neuronal superficial básica (multilayer perceptron, MLP) y las redes neuronales convolucionales profundas (CNN). En el caso de la CNN, se han considerado tres diferentes modelos: Alexnet, GoogLeNet y ResNet-50. La evaluación de las diferentes propuestas se ha ejecutado sobre un conjunto de imágenes médicas heterogéneas. Se ha observado que los resultados del primer módulo de detección son notables, con tasas de éxito cercanas al 75 %, mientras que las tasas de clasificación son cercanas al 99 % de media para ambos MLP y CNN. Hay cierta ventaja en favor de las CNNs. Sin embargo, la implementación de CNNs ad hoc asociadas con estas tareas podría mejorar los resultados obtenidos.

Capítulo 7

Un gas neuronal autoorganizado basado en las divergencias de Bregman

Las "leyes del pensamiento" no solo dependen de las propiedades de las células cerebrales, sino del modo en que están conectadas.

Marvin Minsky

Resumen: En este capítulo se propone un nuevo modelo de gas neuronal autoorganizado que se ha denominado gas neuronal jerárquico creciente de Bregman (Growing Hierarchical Bregman Neural Gas o GHBNG). La propuesta está basada en el gas neuronal jerárquico creciente (GHNG) en el que las divergencias de Bregman son incorporadas para calcular la neurona ganadora. Este modelo se ha aplicado a la detección de anomalías en secuencias de video junto con una red Faster R-CNN como módulo detector de objetos. Los resultados experimentales no solo confirman la efectividad del GHBNG para la detección de objetos anómalos en secuencias de video, sino también sus capacidades de autoorganización.

7.1. Introducción

Como ya se ha descrito en capítulos anteriores, los mapas autoorganizados (Self-organizing Map o SOM) (Kohonen, 1982) han sido ampliamente aplicados para el agrupamiento de datos desde su publicación. El SOM interpreta un mapeo entre los datos con alto número de dimensiones y una

representación del espacio con menor número de dimensiones preservando la topología de los datos de entrada. Muchos modelos neuronales de tipo SOM se han propuesto a lo largo de los años, que se basan en una topología de red fija entre las neuronas (Kohonen, 2013). El gas neuronal creciente (Growing Neural Gas o GNG) (Fritzke, 1995) es una red neuronal autoorganizada que aprende un grafo dinámico con número de neuronas y conexiones variable. Este grafo representa los datos de entrada de una manera más flexible y plástica que la topología fija del mapa, mejorando las capacidades de visualización y comprensión de los datos.

Estos modelos autoorganizados tienen sus versiones jerárquicas, como el mapa autoorganizado jerárquico creciente (Growing Hierarchical Self Organizing Map o GHSOM) para el SOM (Rauber et al., 2002) y el gas neuronal jerárquico creciente (Growing Hierarchical Neural Gas o GHNG) para el GNG (Palomo y López-Rubio, 2017), en el que una neurona puede ser expandida en un nuevo mapa o grafo en una capa posterior de la jerarquía dependiendo del error de cuantificación asociado a esa neurona o al grafo al que pertenece. Los modelos jerárquicos pueden reflejar relaciones jerárquicas presentes en los datos de entrada de una manera más clara.

Otro posible problema presente en estos modelos autoorganizados es el uso de la distancia euclídea para calcular la neurona ganadora, ya que esta distancia puede no ser la más adecuada para todas las distribuciones de entrada. De ahí que las divergencias de Bregman fueran tenidas en cuenta para el GHSOM (López-Rubio et al., 2014), ya que son adecuadas para el agrupamiento porque su minimizador es la media (Banerjee et al., 2005). Además, la distancia euclídea cuadrada es un caso particular de las divergencias de Bregman. Por tanto, usando las divergencias de Bregman se puede puede especificar la divergencia más adecuada a los datos de entrada. En este capítulo se propone una nueva red neuronal autoorganizada denominada gas neuronal jerárquico creciente de Bregman (Growing Hierarchical Bregman Neural Gas o GHBNG), que se basa en el modelo GHNG y el que se consideran las divergencias de Bregman.

Por otro lado, la proliferación en años recientes de una gran cantidad de información visual en forma de secuencia de datos ha llevado a un crecimiento en el campo de la videovigilancia inteligente. En particular, una de las tareas más importantes a considerar es la detección automática de objetos en movimiento que no son muy frecuentes en la escena y que pueden ser considerados como anomalías. Además, la aparición de las redes neuronales profundas para la detección de objetos en una imagen ha significado un momento crucial en la detección de objetos en secuencias de video (Wang, 2016). Por tanto, es posible usar redes pre-entrenadas con miles de datos y un gran número de tipos de objetos para detectar objetos en movimiento en una escena, proporcionando resultado más estables que aquellos obtenidos por las propuestas clásicas.

Para mostrar las posibles aplicaciones de la propuesta, se ha aplicado el GHBNG a la detección de anomalías en secuencias de video capturadas por cámaras IP estáticas. Los objetos en movimiento en cada fotograma son obtenidos por una red Faster R-CNN (Ren et al., 2017). Después, el GHBNG estima los objetos que son considerados como anómalos tras una fase previa de entrenamiento.

El resto del capítulo tiene la siguiente estructura: la Sección 7.2 describe de manera exhaustiva el modelo GHBNG. La Sección 7.3 presenta varios experimentos que demuestran la capacidad autoorganizativa del modelo GHBNG, además de su aplicación a la detección de objetos anómalos en secuencias de video. Finalmente, la Sección 7.4 concluye el capítulo.

7.2. El modelo GHBNG

Una red neuronal autoorganizada denominada gas neuronal jerárquico creciente de Bregman (GHBNG) se define como una red de gas neuronal jerárquico creciente (Growing Hierarchical Neural Gas o GHNG) (Palomo y López-Rubio, 2017) en el que las divergencias de Bregman son incorporadas para calcular la neurona ganadora. Una red GHBNG puede verse como un árbol de redes de gas neuronal creciente (Growing Neural Gas o GNG) (Fritzke, 1995) donde se establece un mecanismo para controlar el crecimiento de cada grafo GNG. Este mecanismo distingue entre una fase de crecimiento donde se añaden más neuronas hasta que no se obtiene una mejora significativa en el error de cuantificación, y una fase de convergencia donde no se pueden crear más unidades. Por tanto, cada grafo contiene un número variable de neuronas, es decir, su tamaño puede aumentar o disminuir durante el aprendizaje. Además, cada grafo es el hijo de una unidad del nivel superior, excepto para el grafo del nivel de arriba (raíz) que no tiene padre. Un ejemplo de la estructura de un modelo GHBNG se muestra en la Figura 7.1. Nótese que la estructura es la misma que el GHNG ya que la diferencia entre estos dos modelos autoorganizados reside en la forma de calcular la neurona ganadora de acuerdo a la divergencia de Bregman utilizada.

La definición del GHBNG está organizada en dos subsecciones. Primero se presenta una revisión de las divergencias de Bregman. Después, el modelo básico para un grafo y el correspondiente algoritmo de aprendizaje son explicados (Subsección 7.2.3). Por último se detalla cómo los nuevos grafos son creados para lograr una jerarquía de grafos (Subsección 7.2.4).

7.2.1. Revisión de las divergencias de Bregman

Ahora se hace una revisión de los fundamentos de las divergencias de Bregman y su aplicación al agrupamiento. Sea $\phi : \mathcal{S} \rightarrow \mathbb{R}$ una función

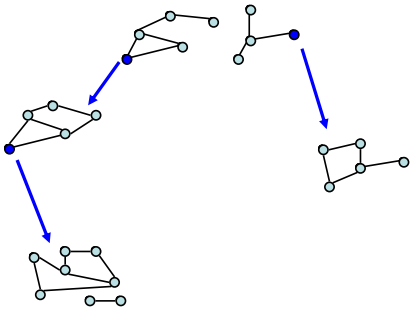


Figura 7.1: Estructura de un modelo GHBNG con cuatro grafos. Las neuronas padre se muestran en un tono más oscuro.

de valores reales estrictamente convexa definida sobre un conjunto convexo $\mathcal{S} \subseteq \mathbb{R}^D$, donde D es la dimensión de los datos de entrada (Bregman, 1967; Censor y Zenios, 1998; Villmann y Haase, 2011). Se asume que ϕ es derivable en el interior relativo $\mathrm{ri}(\mathcal{S})$ del conjunto $\mathcal{S}$ (Banerjee et al., 2005). Después, la divergencia de Bregman $D_\phi : \mathcal{S} \times \mathrm{ri}(\mathcal{S}) \to [0, +\infty)$ correspondiente a ϕ se define como

$$D_\phi(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) - \phi(\mathbf{y}) - (\mathbf{x} - \mathbf{y})^T \nabla\phi(\mathbf{y}) \tag{7.1}$$

donde $\mathbf{x} \in \mathcal{S}$ y $\nabla\phi(\mathbf{y})$ se ajusta para el vector gradiente de ϕ evaluado en $\mathbf{y} \in \mathrm{ri}(\mathcal{S})$. Las Tablas 7.1 y 7.2 muestran las divergencias de Bregman que se han considerado, que son: distancia euclídea cuadrada (Squared Euclidean distance), divergencia I-Generalizada (Generalized I-divergence), distancia de Itakura-Saito (Itakura-Saito distance), pérdida exponencial (Exponential loss) y pérdida logística (Logistic loss).

Divergencia	$\mathcal{S}$	$\phi(\mathbf{x})$
Distancia euclídea cuadrada	$\mathbb{R}^D$	$\|\mathbf{x}\|^2$
Divergencia I-Generalizada	$\mathbb{R}_+^D$	$\sum_{k=1}^D x_k \log x_k$
Distancia de Itakura-Saito	$\mathbb{R}_+^D$	$-\sum_{k=1}^D \log x_k$
Pérdida exponencial	$\mathbb{R}^D$	$\sum_{k=1}^D \exp x_k$
Pérdida logística	$(0,1)^D$	$\sum_{k=1}^D (x_k \log x_k + (1 - x_k) \log(1 - x_k))$

Tabla 7.1: Divergencias de Bregman consideradas ($\mathcal{S}$ y $\phi(\mathbf{x})$). $\mathbb{R}_+^D$ se refiere al conjunto de vectores de tamaño D con componentes reales estrictamente positivas.

Divergencia	$D_\phi(\mathbf{x}, \mathbf{y})$
Distancia euclídea cuadrada	$\|\mathbf{x} - \mathbf{y}\|^2$
Divergencia I-Generalizada	$\sum_{k=1}^{D} \left(-x_k + y_k + x_k \log \frac{x_k}{y_k}\right)$
Distancia de Itakura-Saito	$\sum_{k=1}^{D} \left(-1 + \frac{x_k}{y_k} - \log \frac{x_k}{y_k}\right)$
Pérdida exponencial	$\sum_{k=1}^{D} \left(\exp x_k - \exp y_k - (x_k - y_k) \exp y_k\right)$
Pérdida logística	$\sum_{k=1}^{D} \left(x_k \log \frac{x_k}{y_k} + (1 - x_k) \log \frac{1-x_k}{1-y_k}\right)$

Tabla 7.2: Divergencias de Bregman consideradas ($D_\phi(\mathbf{x}, \mathbf{y})$). $\mathbb{R}_+^D$ se refiere al conjunto de vectores de tamaño D con componentes reales estrictamente positivas.

Las divergencias Bregman son adecuadas para agrupar porque su minimizador es la media. Esta es la principal contribución de (Banerjee et al., 2005), donde se prueba que la clase de medidas de distorsión con respecto a un conjunto de centroides que admiten un procedimiento de minimización iterativo es precisamente el de la divergencias de Bregman. Además, también se demuestra que cada divergencia de Bregman está asociada de manera única a una familia exponencial regular de funciones de densidad de probabilidad, que se definen a continuación. De esta forma, una única función de densidad de probabilidad puede ser conectada al grupo asociado a un centroide dado, que permite la agrupación probabilística suave. Por otro lado, la maximización de expectativa puede ejecutarse con una complejidad computacional reducida para las divergencias generales de Bregman, es decir, las divergencias específicas de Bregman pueden ser diseñadas para adaptarse a la aplicación en cuestión.

La propiedad de que la media es el minimizador de una divergencia de Bregman se formaliza a continuación. Dada una distribución de entrada por $\mathbf{x}$ se cumple la siguiente condición (Villmann y Haase, 2011):

$$\boldsymbol{\mu} = E[\mathbf{x}] = \arg \min_{\mathbf{y}} E[D_\phi(\mathbf{x}, \mathbf{y})] \tag{7.2}$$

Sea N el número de grupos, y sea $\boldsymbol{\mu}_i$ el vector medio del i-ésimo grupo $\mathcal{C}_i$, $i \in \{1, ..., N\}$. Luego un punto $\mathbf{x}$ pertenece a $\mathcal{C}_i$ si $\boldsymbol{\mu}_i$ minimiza la divergencia con respecto a $\mathbf{x}$:

$$\mathcal{C}_i = \left\{ \mathbf{x} \in \mathcal{S} \mid i = \arg \min_{j \in \{1,...,N\}} D_\phi(\mathbf{x}, \boldsymbol{\mu}_j) \right\} \tag{7.3}$$

Por tanto, se puede reescribir (7.2) para hacer una partición $\mathcal{S}$ en N grupos $\mathcal{C}_i$:

$$\boldsymbol{\mu}_i = E\left[\mathbf{x} \mid \mathcal{C}_i\right] = \arg\min_{\mathbf{y}} E\left[D_\phi\left(\mathbf{x}, \mathbf{y}\right) \mid \mathcal{C}_i\right] \tag{7.4}$$

La ecuación anterior implica que la media del grupo $\mathcal{C}_i$ minimiza la divergencia de Bregman a las muestras $\mathbf{x}$ que pertenecen al grupo.

7.2.2. Modelo básico

Para las aplicaciones de agrupamiento y redes autoorganizadas es necesario aprender un vector de pesos $\mathbf{w}_i$ de cada grupo i (Mwebaze et al., 2011), es decir, $\mathbf{w}_i$ estima el vector medio del grupo $\boldsymbol{\mu}_i$. El gradiente descendiente estocástico ha sido propuesto en Villmann y Haase 2011 para minimizar $E\left[D_\phi\left(\mathbf{x}, \mathbf{z}\right)\right]$:

$$\triangle \mathbf{w}_i = -\eta \frac{\partial D_\phi\left(\mathbf{x}, \mathbf{w}_i\right)}{\partial \mathbf{w}_i} \tag{7.5}$$

donde η es un *tamaño de paso* adecuado.

Ahora se propone una diferente propuesta, denominada la estimación del vector medio del grupo $E\left[\mathbf{x} \mid \mathcal{C}_i\right]$ mediante aproximación estocástica (Kushner y Yin, 2003; Lai, 2003; Delyon et al., 1999; Sato y Ishii, 2000). Esta estrategia ha sido aplicada con éxito por los autores a otros modelos autoorganizados en López-Rubio et al. 2009; López-Rubio et al. 2011a; López-Rubio et al. 2011. El objetivo de la aproximación estocástica es encontrar el valor de algunos parámetros $\boldsymbol{\theta}$ que satisfacen

$$\zeta\left(\boldsymbol{\theta}\right) = 0 \tag{7.6}$$

donde ζ es una función cuyos valores no pueden ser obtenidos directamente. Lo que tenemos es una variable aleatoria z que es una estimación ruidosa de ζ:

$$E\left[z\left(\boldsymbol{\theta}\right) \mid \boldsymbol{\theta}\right] = \zeta\left(\boldsymbol{\theta}\right) \tag{7.7}$$

Bajo estas condiciones, el algoritmo Robbins-Monro procede iterativamente:

$$\boldsymbol{\theta}\left(n+1\right) = \boldsymbol{\theta}\left(n\right) + \eta\left(n\right) z\left(\boldsymbol{\theta}\left(n\right)\right) \tag{7.8}$$

donde n es el paso de tiempo.

En este caso, el parámetro variante $\boldsymbol{\theta}\left(n\right)$ es el i-ésimo vector de pesos:

$$\boldsymbol{\theta}\left(n\right) = \mathbf{w}_i\left(n\right) \tag{7.9}$$

Como se ha comentado anteriormente, el objetivo es estimar la expectativa condicional $\boldsymbol{\mu}_i = E[\mathbf{x} \mid \mathcal{C}_i]$ por aproximación estocástica, por lo que se puede tomar

$$\zeta(\mathbf{w}_i) = \boldsymbol{\mu}_i - \mathbf{w}_i \tag{7.10}$$

$$z(\mathbf{w}_i) = \frac{\mathbb{I}(\mathbf{x} \in \mathcal{C}_i)}{P(\mathcal{C}_i)}(\mathbf{x} - \mathbf{w}_i) \tag{7.11}$$

donde $\mathbb{I}$ se refiere a la función indicadora y $P(\mathcal{C}_i)$ es la probabilidad a priori del grupo $\mathcal{C}_i$. Nótese que

$$\mathbf{x} \notin \mathcal{C}_i \Rightarrow z(\mathbf{w}_i) = \mathbf{0} \tag{7.12}$$

Por lo tanto, se tiene que (7.11) satisface la condición (7.7):

$$E[z(\mathbf{w}_i) \mid \mathbf{w}_i] = P(\mathcal{C}_i) E[z(\mathbf{w}_i) \mid \mathcal{C}_i, \mathbf{w}_i] +$$

$$P(\bar{\mathcal{C}}_i) E[z(\mathbf{w}_i) \mid \bar{\mathcal{C}}_i, \mathbf{w}_i] =$$

$$E[\mathbf{x} - \mathbf{w}_i \mid \mathcal{C}_i, \mathbf{w}_i] = E[\mathbf{x} \mid \mathcal{C}_i] - \mathbf{w}_i = \boldsymbol{\mu}_i - \mathbf{w}_i \tag{7.13}$$

donde $\bar{\mathcal{C}}_i$ es el complementario del grupo $\mathcal{C}_i$.
Por tanto, la ecuación (7.8) dice

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + \eta(n) z(\mathbf{w}_i(n)) \tag{7.14}$$

Si se toma

$$\eta(n) = P(\mathcal{C}_i) \epsilon(n) \tag{7.15}$$

entonces (7.14) puede ser reescrita como

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n) +$$

$$\epsilon(n) P(\mathbf{x}(n) \in \mathcal{C}_i)(\mathbf{x}(n) - \mathbf{w}_i(n)) \tag{7.16}$$

donde $\epsilon(n)$ es la tasa de aprendizaje en el paso de tiempo n, y no se necesita el valor de la probabilidad a priori $P(\mathcal{C}_i)$. El término $\epsilon(n)P(\mathbf{x}(n) \in \mathcal{C}_i)$ se supone que es ϵ_b para la neurona ganadora, ϵ_n para sus vecinos directos, y cero en otro caso, con $\epsilon_b > \epsilon_n > 0$. Es decir, $P(\mathbf{x}(n) \in \mathcal{C}_i)$ se supone que es máximo para la neurona ganadora, más pequeño para sus vecinos inmediatos, y cero para el resto de unidades.

7.2.3. Modelo de grafo

Cada grafo del GHBNG está formado por H neuronas ($H \geq 2$) y una o más conexiones directas entre ellas. Ambas neuronas y conexiones pueden ser creadas y destruidas durante el proceso de aprendizaje. No es necesario que el grafo esté conectado, como se mencionó anteriormente. El conjunto de entrenamiento para el grafo se denota por $\mathcal{S}$, con $\mathcal{S} \subset \mathbb{R}^D$, donde D es la dimensión del espacio de entrada. Cada unidad $i \in \{1, ..., H\}$ tiene un prototipo asociado $\mathbf{w}_i \in \mathbb{R}^D$ y un error variable $e_i \in \mathbb{R}$, $e_i \geq 0$. Cada conexión tiene una edad asociada, que es un entero no negativo. El conjunto de conexiones se denomina $A \subseteq \{1, ..., H\} \times \{1, ..., H\}$.

El mecanismo de aprendizaje para un grafo del GHBNG está basado en el GNG original (Fritzke, 1995), pero incluye un procedimiento novedoso para controlar el crecimiento del grafo. Primero, se realiza una fase de crecimiento donde se le permite al grafo crecer, hasta que se cumple una condición que indica que un mayor crecimiento no proporcionaría mejoras significativas en el error de cuantificación. Tras esto, se ejecuta una fase de convergencia donde la no se permite la creación de ninguna unidad para llevar a cabo un ajuste del grafo. El algoritmo de aprendizaje viene dado por los siguientes pasos:

1. Se comienza con dos neuronas ($H = 2$) unidas por dos conexiones, una en cada sentido. Cada prototipo es inicializado a una muestra seleccionada aleatoriamente de $\mathcal{S}$. Las variables de error se inicializan a cero. La edad de las conexiones también se inicializan a cero.

2. Se selecciona una muestra de entrenamiento $\mathbf{x}_n \in \mathbb{R}^D$ aleatoria de $\mathcal{S}$.

3. Se busca la unidad más cercana q y la segunda unidad más cercana s en términos de las divergencias de Bregman:

$$q = \arg \min_{i \in \{1,...,H\}} D_\phi \left(\mathbf{x}\left(n\right), \mathbf{w}_i\left(n\right) \right) \tag{7.17}$$

$$s = \arg \min_{i \in \{1,...,H\} - \{q\}} D_\phi \left(\mathbf{x}\left(n\right), \mathbf{w}_i\left(n\right) \right) \tag{7.18}$$

4. Se incrementa la edad de todas las conexiones que parten de q.

5. Se añade la distancia euclídea cuadrada entre $\mathbf{x}_n$ y la unidad más cercana q a la variable de error e_q:

$$e_q\left(n+1\right) = e_q\left(n\right) + \left\| \mathbf{w}_q\left(n\right) - \mathbf{x}(n) \right\|^2 \tag{7.19}$$

6. Actualizar q y todos sus vecinos topológicos directos con tamaño de paso ϵ_b para la unidad q y ϵ_n para sus vecinos, donde $\epsilon_b > \epsilon_n$:

$$\epsilon\left(n,i\right) = \begin{cases} \epsilon_b & \text{iff } n = q \\ \epsilon_n & \text{iff } (n \neq q) \wedge (n,q) \in A \\ 0 & \text{iff } (n \neq q) \wedge (n,q) \notin A \end{cases} \tag{7.20}$$

$$\mathbf{w}_i\left(n+1\right) = \left(1 - \epsilon\left(n,i\right)\right)\mathbf{w}_i\left(n\right) + \epsilon\left(n,i\right)\mathbf{x}(n) \tag{7.21}$$

7. Si q y s están conectadas por una conexión, entonces ajustar la edad de esta conexión a cero. En otro caso, crearla.

8. Eliminar las conexiones con una edad superior a a_{max}. Después eliminar todas las neuronas que no tienen conexiones salientes.

9. Si el actual paso de tiempo n es un entero múltiplo de un parámetro λ y el grafo está en la fase de crecimiento, entonces hacer una copia de seguridad del grafo completo e insertar una nueva unidad como sigue. Primero determinar la unidad r con el máximo error y la unidad z con el mayor error entre todos los vecinos directos de r. Luego crear una nueva unidad k, añadir conexiones conectando k con r y z, y eliminar la conexión original entre r y z. Tras esto, decrementar las variables de error e_r y e_z multiplicándolas por una constante α, e inicializar la variable de error e_k a un nuevo valor de e_r. Finalmente, preparar el prototipo de k para estar a medio camino entre los de r y z, como sigue:

$$\mathbf{w}_k\left(n\right) = \frac{1}{2}\left(\mathbf{w}_r\left(n\right) + \mathbf{w}_z\left(n\right)\right) \tag{7.22}$$

10. Si el grafo está en la fase de crecimiento y el actual paso de tiempo n satisface:

$$\mathrm{mod}\left(n, 2\lambda\right) = \left\lfloor \frac{3}{2}\lambda \right\rfloor \tag{7.23}$$

donde $\lfloor\cdot\rfloor$ se refiere para redondear hacia $-\infty$, entonces se realiza una comprobación para ver si el crecimiento del grafo ha producido una mejora del error de cuantificación. La media del error de cuantificación por neurona de la copia de seguridad y de la actual versión del grafo se

calculan como la suma de sus variables de error dividido por el número de neuronas H. Sea MQE_{old} y MQE_{new} la media de los errores de cuantificación de la copia de seguridad y de la versión actual del grafo, respectivamente. Si la siguiente condición se cumple, entonces la actual versión es destruida, la copia de seguridad es restaurada, y el grafo entra en la fase de convergencia:

$$\frac{MQE_{old} - MQE_{new}}{MQE_{old}} < \tau \tag{7.24}$$

donde $\tau \in [0, 1]$ es un parámetro que controla el proceso de crecimiento. Cuanto mayor es τ, más significativa es la mejora en el error de cuantificación para continuar la fase de crecimiento. Por tanto, mayores valores de τ se asocian con grafos más pequeños, y viceversa.

11. Decrementar todas las variables de error e_i multiplicándolas por una constante d.

12. Si el número máximo de pasos de tiempo ha sido alcanzado, entonces parar. En otro caso, ir al paso 2.

7.2.4. Modelo jerárquico

Como se ha mencionado anteriormente, el GHBNG está definido como un árbol de grafos. A continuación se detalla este procedimiento para aprender cada jerarquía. El proceso comienza entrenando el grafo raíz con el conjunto de muestras de entrenamiento general. Cada tiempo que un grafo debe ser entrenado con el conjunto de entrenamiento $\mathcal{S}$, esto se hace de acuerdo al algoritmo especificado en la Subsección 7.2.3. Si el número de neuronas resultantes es $H = 2$, entonces el grafo es podado porque es muy pequeño para representar cualquier característica importante de la distribución de entrada. En otro caso, se crea un nuevo grafo para cada unidad i y se llama al proceso de entrenamiento recursivamente con el campo receptivo de la unidad i como el conjunto de entrenamiento:

$$\mathcal{S}_i = \left\{ \mathbf{x} \in \mathcal{S} \mid i = \arg \min_{j \in \{1,\dots,H\}} D_\phi \left(\mathbf{x}, \mathbf{w}_j \right) \right\} \tag{7.25}$$

Este proceso recursivo continúa hasta que se alcanza un número de niveles preespecificado. La eliminación de los grafos con menos de 3 neuronas y la división del conjunto de entrenamiento dado por (7.25) trabajan de forma conjunta para lograr una jerarquía parsimoniosa, es decir, una con un reducido número de grafos y neuronas. Esto es porque los grafos más pequeños en

Descripción del parámetro	Valor
Tamaño de paso para la unidad ganadora	$\epsilon_b = 0{,}2$
Tamaño de paso para la unidad vecina	$\epsilon_n = 0{,}006$
Edad máxima para una conexión	$a_{max} = 50$
Inserción de nuevas unidades	$\lambda = 100$
Número máximo de unidades	$H_{max} = 50$
Reducción de las variables de error	$\alpha = 0{,}5$
Deterioro de la variable de error	$d = 0{,}995$

Tabla 7.3: Selección de parámetros para el modelo GHBNG.

el árbol no pueden tener muchas neuronas porque sus conjuntos de entrenamiento son pequeños. Además, muchos de los gráficos creados eventualmente serán eliminados inmediatamente después de su entrenamiento, es decir, el hecho de que un grafo sea creado por cada unidad no implica un crecimiento incontrolado.

7.3. Resultados experimentales

7.3.1. Configuración experimental

Los experimentos mostrados se han ejecutado en un ordenador personal con 64 bits con un microprocesador Intel Core i7 2.90 GHz, 8 GB RAM y hardware standard. Todos los entrenamientos se ejecutaron utilizando 2 épocas, independientemente del número de muestras de entrenamiento M. Ya que el GHBNG está basado en el GHNG, esta nueva propuesta tiene los mismos parámetros que este modelo. De igual forma, el GHNG está basado en el GNG, es decir, la configuración de parámetros es la misma que la recomendada en el artículo original del GNG (Fritzke, 1995), cuyos valores se muestran en la Tabla 7.3. Como se mencionó con anterioridad, el parámetro τ (que también está presente en el modelo GHNG) controla el proceso de crecimiento donde cuanto menor sea τ, mayor será el tamaño de la arquitectura. Este parámetro debe ser configurado para cada experimento manteniendo $\tau \in [0, 1]$.

7.3.2. Experimentos de autoorganización

Este conjunto de experimentos ha sido diseñado para comprobar las capacidades de autoorganización del GHBNG antes de las diferentes divergencias de Bregman. Se han seleccionado dos diferentes distribuciones de entrada bidimensionales ($D = 2$), una con la forma del número ocho y otra con la forma de la letra M. El entrenamiento se ha ejecutado utilizando $M = 10,000$

muestras de entrada y $N = 20,000$ pasos de tiempo para cada distribución de entrada y divergencia de Bregman. Se han elegido dos valores diferentes del parámetro τ (0,1 y 0,2) para mostrar el efecto de este parámetro en el tamaño final de la arquitectura.

Los GHBNGs resultantes para cada divergencia de Bregman y valor de τ se muestran en las Figuras 7.2 y 7.3 para las distribuciones de entrada del número ocho y de la letra M, respectivamente. En esas gráficas las neuronas se representan por círculos y las conexiones entre las neuronas mediante líneas rectas, donde el color y tamaño tanto de las neuronas como de las conexiones es diferente dependiendo de la capa a la que pertenecen. Se ha dibujado un máximo de tres capas para evitar gráficas abarrotadas. Nótese que los GHBNGs para $\tau = 0,2$ (primera fila) se logran arquitecturas con menos neuronas que utilizando $\tau = 0,1$ (segunda fila) y por tanto comete más errores cuando se adapta a su correspondiente forma. Si se pone atención en $\tau = 0,1$ (segunda fila), cada divergencia de Bregman se adapta correctamente a la forma de las dos distribuciones de entrada, aunque para la letra M algunas conexiones están en el lugar incorrecto, especialmente para Itakura-Saito.

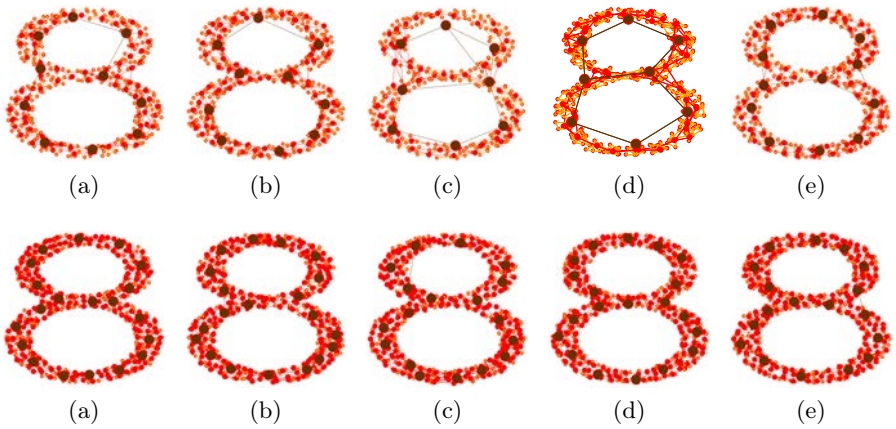


Figura 7.2: Resultados del GHBNG para la distribución de entrada del número ocho utilizando $\tau = 0,2$ (primera fila), $\tau = 0,1$ (segunda fila) y las diferentes divergencias de Bregman: (a) distancia euclídea cuadrada, (b) divergencia I-Generalizada, (c) distancia de Itakura-Saito, (d) pérdida exponencial y (e) pérdida logística. Las neuronas se representan por círculos y las conexiones entre las neuronas se representan por líneas rectas, donde las capas superiores se muestran en tonos más oscuros y con un tamaño más grande que las capas más profundas.

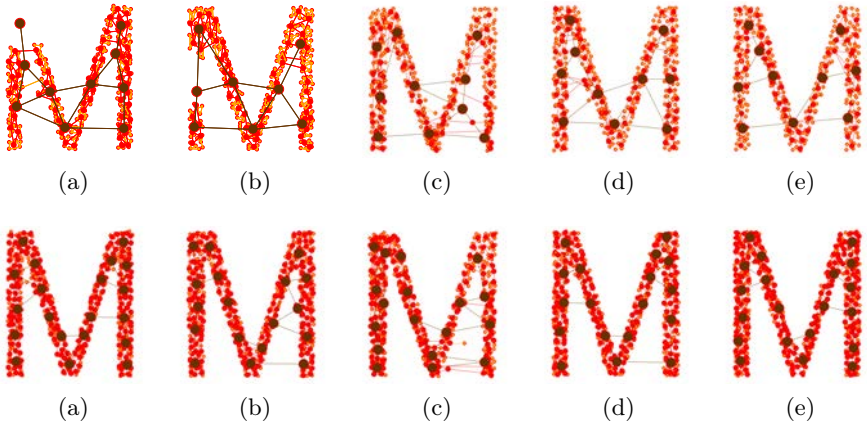


Figura 7.3: Resultados del GHBNG para la distribución de entrada de la letra M utilizando $\tau = 0{,}2$ (primera fila), $\tau = 0{,}1$ (segunda fila) y las diferentes divergencias de Bregman: (a) distancia euclídea cuadrada, (b) divergencia I-Generalizada, (c) distancia de Itakura-Saito, (d) pérdida exponencial y (e) pérdida logística. Las neuronas se representan por círculos y las conexiones entre las neuronas se representan por líneas rectas, donde las capas superiores se muestran en tonos más oscuros y con un tamaño más grande que las capas más profundas.

7.3.3. Detección de anomalías en secuencias de video

Se ha utilizado el modelo desarrollado en la detección de objetos anómalos en secuencias de video. El sistema está compuesto de una cámara estática en el escenario, capturando un video del mismo.

Con un módulo detector de objetos Faster R-CNN (Ren et al., 2017) (es una técnica de aprendizaje profundo que utiliza regiones con redes neuronales convolucionales), se pueden reconocer 20 clases de objetos diferentes incluidas en el conjunto de datos PASCAL VOC 2007 (Everingham et al., 2018). La entrada del detector Faster R-CNN es una imagen y su salida es un conjunto de probabilidades y su área para objeto detectado, donde cada probabilidad muestra el nivel de pertenencia a cada clase. Solamente se utilizan las probabilidades indicadas, por lo que, dado el fotograma t, el conjunto de probabilidades de salida $\mathbf{q}_{i,t}$ del módulo detector de objetos es:

$$\mathbf{q}_{i,t} = (q_{i,t,1}, ..., q_{i,t,K}) \in \mathbb{R}^K \tag{7.26}$$

donde i es uno de los objetos detectados en el fotograma t, $q_{i,t,k} \in [0, 1]$, $C_k \in Classes$ y el número de clases de objetos es K (en este caso, $K = 20$). Además, para estos experimentos se ha incorporado una tarjeta gráfica Titan X como dispositivo hardware.

En este contexto, los animales se han considerado como objetos anómalos,

mientras que el resto de clases detectadas son considerados como objetos no anómalos. Para entrenar los modelos de clasificación de anomalías, se ha ejecutado el sistema con un video que no muestra ningún objeto anómalo (es decir, no hay animales en él) y, para cada fotograma, se han obtenido las probabilidades de pertenencia de cada clase para cada objeto detectado por el módulo de detección. Tras esto, con esta información se ha entrenado un modelo GHBNG por cada una de las divergencias de Bregman. De acuerdo a la definición del módulo detector de objetos, se tiene una distribución de entrada de 20 dimensiones ($D = 20$). El entrenamiento se ha ejecutado utilizando $M = 650$ muestras de entrada, $N = 1,300$ pasos de tiempo para cada divergencia de Bregman y el valor del parámetro τ ha sido fijado a 0,1. Un esquema de los pasos para entrenar los modelos GHBNG se puede observar en la Figura 7.4.

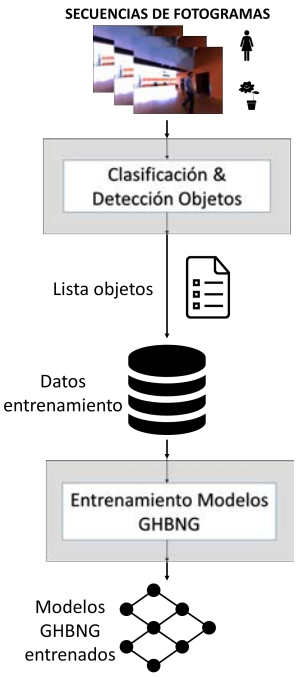


Figura 7.4: Esquema del proceso de entrenamiento de los modelos GHBNG.

Luego se ha ejecutado un video que presenta objetos anómalos en el mismo escenario (los videos con anomalías[1] y sin ellas [2]). En cada fotograma,

[1] `http://www.lcc.uma.es/~miguelangel/resources/fixed_camera/video_with_anomalies.rar`

[2] `http://www.lcc.uma.es/~miguelangel/resources/fixed_camera/video_without_anomalies.rar`

como en el paso anterior, el detector de objetos proporciona las probabilidades de pertenencia de cada objeto detectado y su salida es suministrada al módulo de clasificación de anomalías con un modelo GHBNG entrenado. Después, este modelo GHBNG calcula cuánto de anómalo es cada objeto detectado. Estos valores se obtienen de la distancia mínima del objeto al prototipo del modelo. Se ha considerado esta distancia como un valor negativo, por lo que un objeto será más anómalo que otro si su resultado es menor. Es decir, dado el conjunto de probabilidades $\mathbf{q}_{i,t}$ correspondientes al fotograma t, el módulo de detección de anomalías calcula el valor $v_{i,t} \in \mathbb{R}$ correspondiente a cómo de anómalo es el objeto i en t. Por último, el módulo de detección de anomalías indica el valor v_t correspondiente al objeto más anómalo en t para establecer si existe realmente un objeto anómalo en el fotograma t (por ejemplo, un perro) o no (por ejemplo, personas o una planta). El funcionamiento de este proceso se puede observar en la Figura 7.5.

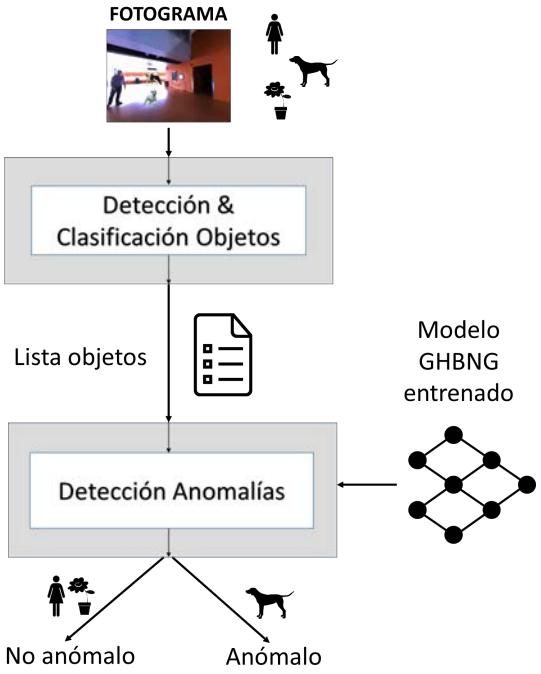


Figura 7.5: Esquema del funcionamiento del sistema de detección de anomalías basado en un modelo GHBNG.

Se ha ejecutado 10 veces el video con los objetos anómalos seleccionado con los diferentes modelos GHBNG y el resultado del v_t mediano producido

por cada modelo se puede observar en la Figura 7.6. Cada imagen muestra el objeto más anómalo en cada fotograma con su valor producido por cada modelo, respectivamente. Estos objetos están divididos en dos grupos: los objetos realmente anómalos (animales) y los objetos que realmente no son anómalos (resto de clases). Las dos líneas punteadas que aparecen se corresponden con los valores de la mayor salida de todos los objetos realmente anómalos y la mínima salida de todos los objetos que realmente no son anómalos. Cuanto menor sea el número de objetos entre estas dos líneas (mayor heterogeneidad) mejor es el modelo, y un valor entre ambas líneas podría considerarse como umbral de esos modelos para clasificar un objeto como anómalo. Es decir, un objeto entre estas dos líneas punteadas podría ser considerado como anómalo o no anómalo, dependiendo de la elección del umbral.

Como se puede observar en la Figura 7.6 los modelos de pérdida logística y de la divergencia I-Generalizada clasifican correctamente los objetos detectados en el video. Por otro lado, los modelos de la distancia euclídea cuadrada y de la pérdida exponencial logran el peor rendimiento. Además, se puede considerar un modelo mejor que otro si las salidas mostradas de los conjuntos anómalo y no anómalo están cerca entre sí, respectivamente, más aún si un conjunto está muy lejos del otro. Por tanto, la elección del umbral podría seleccionarse mejor y el modelo podría ser adecuado en un amplio rango de escenarios. Por tanto, por ejemplo, como se ve en la Figura 7.6, el modelo de pérdida logística podría ser más apropiado que el modelo de divergencia I-Generalizada.

7.4. Conclusiones

En este capítulo se ha presentado un novedoso modelo autoorganizado jerárquico creciente basado en la propuesta de gas neuronal y las divergencias de Bregman. La principal característica es su flexibilidad y adaptabilidad a los datos, aprendiendo un grafo dinámico en cada nivel de su jerarquía. Además, el uso de diferentes medidas (divergencias de Bregman) para calcular la neurona ganadora proporciona más posibilidades para formar grupos, lo que es adecuado en entornos heterogéneos reales.

Las capacidades de autoorganización han sido probadas en la sección de experimentos de una manera cualitativa. Por otro lado, se ha considerado una aplicación más específica relacionada con la detección de objetos anómalos en la videovigilancia, con interesantes resultados que indican su viabilidad en este campo. Son destacables las posibilidades que este modelo ofrece, obteniendo los mejores resultados con las divergencias de la pérdida logística y la I-Generalizada.

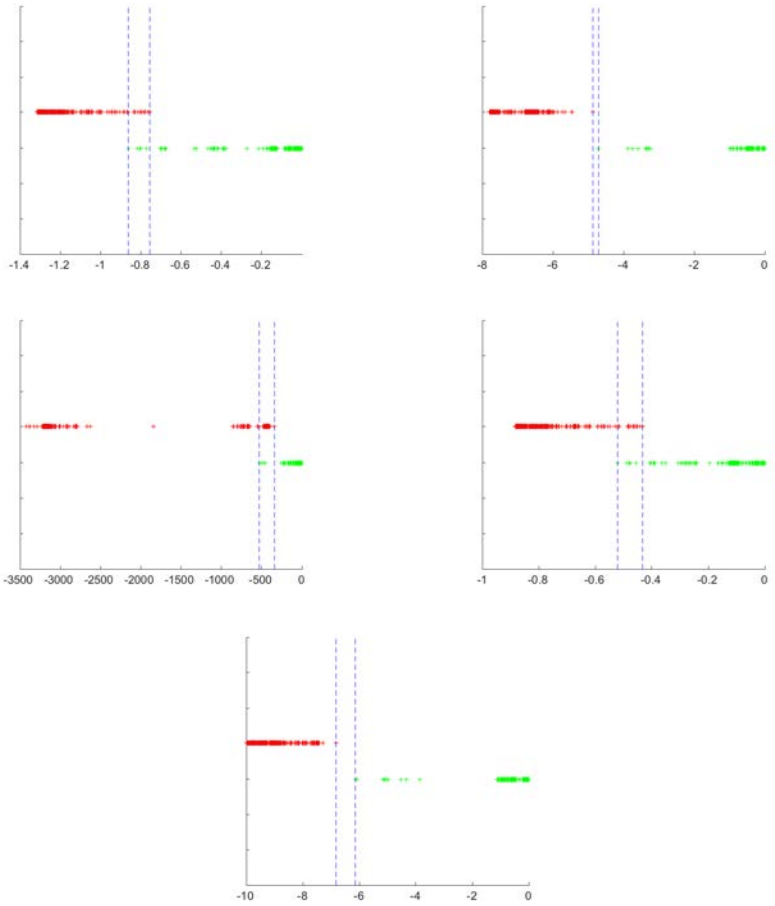


Figura 7.6: Salida más anómala para cada modelo de clasificación de anomalías en cada fotograma de la secuencia de video probada. La primera fila muestra las salidas de los modelos de distancia euclídea cuadrada y divergencia I-Generalizada. La segunda fila muestra las salidas de los modelos de distancia de Itakura-Saito y pérdida exponencial, respectivamente. La tercera fila muestra las salidas del modelo de pérdida logística. El eje x se corresponde con la salida del modelo para cada objeto, es decir, la cantidad de anomalía que determina para cada objeto. Para apreciar mejor los valores de cada objeto se han dispuesto los objetos anómalos y no anómalos en diferentes niveles del eje y: en rojo los anómalos y en verde los no anómalos. Las dos líneas azules punteadas se corresponden con los valores de la mayor salida de todos los objetos realmente anómalos y la mínima salida de todos los objetos que realmente no son anómalos.

Capítulo 8

Reducción del tamaño del fotograma para la detección de primer plano

El tamaño importa.
Anónimo

Resumen: En este trabajo se presenta un marco de trabajo para la reducción de la resolución de los fotogramas para reducir la carga computacional y mejorar la detección de primer plano en secuencias de video. Este marco consiste en tres diferentes etapas. Primeramente, el fotograma original del video es comprimido usando una función de interpolación específica. Segundo, una detección del primer plano del fotograma reducido es llevada a cabo por un modelo de fondo probabilístico llamado MFBM. Finalmente, las probabilidades de clase para el fotograma reducido son descomprimidas usando una interpolación bicúbica para estimar las probabilidades de clase del fotograma original. Los resultados experimentales aplicados a conocidas secuencias de video para demostrar la bondad de la propuesta.

8.1. Introducción

Tal y como se ha ido tratando a lo largo de la presente tesis, dentro del campo de la visión artificial, la investigación en sistemas de videovigilancia principalmente se centra en la detección, reconocimiento y seguimiento del movimiento de los objetos de primer plano en una secuencia de imágenes. Cualquier sistema de videovigilancia comienza su actividad detectando

los objetos en movimiento en la escena. Sin embargo, este proceso es más complejo que la resta del fotograma actual y el fondo de la imagen previamente calculado, que es considerado una propuesta candidata, pero hay varios problemas para ser resueltos que incrementan su complejidad. Factores desfavorables como los cambios de iluminación tanto abruptos como continuos, las sombras de objetos del fondo o el movimiento repetitivo de objetos estacionarios como las ramas de los árboles, deben ser tenidos en cuenta por los métodos desarrollados.

Hay varias propuestas en la literatura que tratan de gestionar estos problemas, como ya se describió en la Subsección 3.2. En (Friedman y Russell, 1997) una media temporal de la secuencia es usada para obtener una imagen del fondo. El filtro de Kalman es usado para cada píxel (Ridder et al., 1995) para hacer frente a la variabilidad de la iluminación de la escena. Además, en (Wren et al., 1997) se considera una distribución gaussiana para modelar el color de fondo de cada píxel, mientras que en (Grimson et al., 1998), el modelo previo es ampliado por una mixtura de distribuciones gaussianas. A diferencia de los dos métodos paramétricos previos, en (Elgammal et al., 2002) el fondo es modelado haciendo uso de un método no paramétrico, que es más robusto e invariante especialmente en escenas de exterior con mucha variabilidad en los objetos estacionarios del fondo. En (Haritaoglu et al., 2000) se presenta un modelo estadístico llamado W4 para representar cada píxel con tres valores: su mínimo y máximo valor, y la máxima diferencia de intensidad entre fotogramas observados consecutivos durante un periodo de entrenamiento.

Sin embargo, uno de los principales problemas de las técnicas de detección de primer plano basadas a nivel de píxel es que para el análisis de datos el modelo debe ser aplicado a cada píxel que pertenece a la escena, lo que implica una considerable alta carga computacional. Este tipo de propuestas están sujetas al desarrollo de modelos más complejos si se quiere mantener el mismo ratio de eficiencia y tiempo real. Por tanto, otras técnicas en el paradigma de consenso (Wang et al., 2014a) consiguen muy buenos resultados combinando las máscaras de varios métodos de detección de objetos, con la desventaja de no completar las necesidades temporales para el procesamiento en tiempo real.

A diferencia de otras propuestas que agrupan los datos por su similitud de color (Luque et al., 2009), el objetivo de este trabajo es presentar un marco de trabajo de resolución del fotograma que agrupa los datos de la vecindad de cada píxel y estima un prototipo para cada región. Por tanto, varios métodos de interpolación son estudiados para comprimir la secuencia. Desde que las secuencias de fotogramas normalmente son comprimidas con un codificador de video para reducir el tamaño y mejorar la tasa de transmisión, el uso de técnicas de interpolación podría aliviar los artefactos generados por la compresión, y mejorar levemente la salida de los métodos basados a nivel de

píxel.

Para analizar la propuesta de reducción de la resolución del fotograma, la técnica de detección probabilística del primer plano (López-Rubio y López-Rubio, 2015), que es un método basado a nivel de píxel, es considerado e incorporado en la propuesta para estudiar la calidad de la máscara de primer plano y la reducción de la carga computacional obtenida con nuestra metodología.

El resto del trabajo se estructura como sigue: la sección 8.2 expone la metodología de la propuesta, especificando los procesos de compresión y descompresión. La sección 8.3 muestra los resultados experimentales, mientras que la sección 8.4 presenta algunas conclusiones del trabajo.

8.2. Metodología

En esta sección se presenta un marco de trabajo para la reducción de la resolución del fotograma para el problema de la detección de primer plano. El modelo de fondo probabilístico que se ha usado como base es (López-Rubio y López-Rubio, 2015). Esta propuesta modela la distribución de los valores de las características del píxel $\mathbf{t}(\mathbf{x}) \in \mathbb{R}^D$ en las coordenadas del fotograma $\mathbf{x} \in \mathbb{Z}^2$ empleando una componente de mixturas de gaussianas $K(\mathbf{t}(\mathbf{x})|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ para el fondo, y una componente de mixturas uniforme $U(\mathbf{t}(\mathbf{x}))$ para el primer plano, donde D es el número de características de interés del píxel. El uso de una componente de mixturas uniforme tiene la ventaja de que todos los nuevos objetos de primer plano son modelados igualmente bien por la mixtura, sin tener importancia sus características. Por otro lado, el conjunto de características para ser usadas puede ser configurado convenientemente para la aplicación.

Nuestro objetivo es reducir la carga computacional del algoritmo base, mientras que al mismo tiempo la adaptación contra el ruido es a veces mejorada. El procedimiento propuesto está compuesto de tres etapas: primero el fotograma original del video es comprimido (Subsección 8.2.1), luego el modelo de fondo utilizado como base es aplicado al fotograma reducido del video, y finalmente las probabilidades de clase para el fotograma reducido son descomprimidas (Subsección 8.2.2).

8.2.1. Compresión

Considérese una secuencia de video con tamaño de fotograma $M \times N$ píxeles y cada píxel tiene D distintivas características como el color o la textura. El objetivo aquí es reducir el tamaño del fotograma a ser procesado por el algoritmo de fondo utilizado como base a $m \times n$ píxeles, donde $m < M$ y $n < N$, mientras que al mismo tiempo la máscara final de detección de primer plano es de tamaño $M \times N$ píxeles. Para cada píxel del fotograma de tamaño

reducido con coordenadas $\mathbf{x}$, $\mathbf{x} \in \{1, ..., m\} \times \{1, ..., n\}$, sus características $\mathbf{t}(\mathbf{x}) \in \mathbb{R}^D$ son calculadas de las características $\mathbf{t}(\mathbf{y})$ de la secuencia de video original:

$$\mathbf{t}(\mathbf{x}) = \varphi(\{\mathbf{t}(\mathbf{y}) \mid \mathbf{y} \in \mathcal{N}(\mathbf{x})\}) \tag{8.1}$$

$$\mathcal{N}(\mathbf{x}) \subset \{1, ..., M\} \times \{1, ..., N\} \tag{8.2}$$

donde $\mathcal{N}(\mathbf{x})$ es una vecindad apropiada del punto $\mathbf{x}' = \left(\frac{Mx_1}{m}, \frac{Nx_1}{n}\right)$ en el fotograma original de video y φ es una adecuada función de interpolación que toma un conjunto de vectores de características del fotograma original y muestra como salida un vector de características interpolado para el píxel del fotograma reducido. Por ejemplo, uno puede seleccionar φ para devolver el vector de características del vecino más próximo de $\mathbf{x}'$:

$$\mathbf{t}_{NN}(\mathbf{x}) = \mathbf{t}(\mathbf{y}_{NN}) \tag{8.3}$$

$$\mathbf{y}_{NN} = \arg \min_{\mathbf{y} \in \{1,...,M\} \times \{1,...,N\}} \left\|\mathbf{y} - \mathbf{x}'\right\| \tag{8.4}$$

Otra posibilidad es dividir la imagen original en bloques cuadrados no solapados de tamaño $B \times B$ píxeles, y luego calcular la media de los vectores de características sobre cada bloque:

$$\mathbf{t}_{AVG}(\mathbf{x}) = \frac{1}{B^2} \sum_{\mathbf{y} \in \mathcal{N}_{AVG}(\mathbf{x})} \mathbf{t}(\mathbf{y}) \tag{8.5}$$

$$\mathcal{N}_{AVG}(\mathbf{x}) = \{1 + B(x_1 - 1), ..., Bx_1\} \times \{1 + B(x_2 - 1), ..., Bx_2\} \tag{8.6}$$

También se consideran las interpolaciones bilinear y bicúbica calculadas de los datos del fotograma original en el punto $\mathbf{x}'$.

8.2.2. Descompresión

Los datos de las características reducidas $\mathbf{t}(\mathbf{x})$ son procesados por un modelo de fondo probabilístico como (López-Rubio y López-Rubio, 2015). El modelo genera las probabilidades de clase $P(i|\mathbf{t}(\mathbf{x})) \in [0, 1]$ de los valores observados $\mathbf{t}(\mathbf{x})$ para los píxeles del fotograma reducido, para las clases $i \in \{Back, Fore\}$. Tras esto, es necesario estimar las probabilidades de clase para los píxeles del fotograma original:

$$P(i|\mathbf{t}(\mathbf{y})) = \varphi'\left(\left\{P(i|\mathbf{t}(\mathbf{x})) \mid \mathbf{x} \in \mathcal{N}'(\mathbf{y})\right\}\right) \tag{8.7}$$

donde $\mathcal{N}'(\mathbf{y})$ es una vecindad adecuada del punto $\mathbf{y}' = \left(\frac{my_1}{M}, \frac{ny_1}{N}\right)$ en el fotograma reducido del video y φ' es una adecuada función de interpolación

Tabla 8.1: Resumen de las características clave del modelo usado por cada propuesta.

Nombre	Característica clave del modelo
ORIG	Tamaño original
AVG	Promedio de Bloques
NN	Vecino Más Cercano
LIN	Interpolación Bilinear
CUB	Interpolación Bicúbica

que toma un conjunto de probabilidades de clase del fotograma reducido y produce como salida una probabilidad de clase interpolada para el píxel del fotograma original. En los experimentos siempre se ha tomado φ' como una interpolación bicúbica, ya que produce estimaciones de probabilidades de clase suavizadas.

8.3. Resultados experimentales

En esta sección el rendimiento de la detección de primer plano y el tiempo de ejecución de los diferentes métodos de compresión y factores de compresión son analizados. Primero, el software y el hardware usados en los experimentos son detallados en la Subsección 8.3.1. Las secuencias de video testeadas son presentadas en la Subsección 8.3.2 y el conjunto de parámetros de cada método de compresión son especificados en la Subsección 8.3.3. Por último, los resultados se muestran en la Subsección 8.3.4.

8.3.1. Métodos

El método de detección de objetos utilizado como base es el algoritmo MFBM (López-Rubio y López-Rubio, 2015), que ha sido publicado previamente por nuestro grupo de investigación y que está basado en la teoría de la aproximación estocástica.

Varios métodos de compresión han sido probados, llamados: Promedio de Bloques (Blockwise average o AVG), Vecino Más Cercano (Nearest neighbor o NN), Interpolación Bilinear (LIN), e Interpolación Bicúbica (CUB). Se denota como el tamaño original del método (ORIG) si ninguna compresión es aplicada y cada píxel es individualmente procesado. El conjunto de características clave de cada método es mostrado en la Tabla 8.1.

No se ha utilizado ningún postprocesado adicional en ninguno de los métodos estudiados para hacer las comparativas lo más justas posibles. Todos los experimentos se han ejecutado en un ordenador personal de 64 bits con

Tabla 8.2: Valores considerados de los parámetros para los métodos competidores. Las combinaciones de ellos forman el conjunto de todas las configuraciones experimentales.

Método	Parámetros
MFBM	Tamaño de paso, $\alpha = 0{,}01$ Rasgos, $F = [1, 2, 3]$ Método de Compresión, $CompressionMethod = \{ORIG, AVG, NN, LIN, CUB\}$ Factor de Compresión, $\rho = \{1, 0{,}875, 0{,}75, 0{,}625, 0{,}5, 0{,}375, 0{,}25, 0{,}125\}$

un procesador de 8 núcleos Intel i7 3,60 GHz, 32 GB RAM y hardware convencional.

8.3.2. Secuencias

El conjunto de videos que han sido probados han sido elegidos del conjunto de datos 2014 del sitio web de ChangeDetection.net [1]. Las secuencias seleccionadas son los videos de la categoría Baseline, que está compuesta por videos con dificultades no especiales. Hay dos videos de exterior: *Highway* presenta una autovía con coches moviéndose de arriba abajo (320x240 píxeles y 1700 fotogramas), y *Pedestrians* muestra personas andando de izquierda a derecha y viceversa (360x240 píxeles y 1099 fotogramas). Además, hay dos secuencias de interior: *Office*, cuya particularidad es que una persona se mantiene estática en una habitación durante un intervalo de tiempo y luego continúa su movimiento (360x240 píxeles y 2050 fotogramas); y *PETS2006*, con gente moviéndose en una estación de tren (720x576 píxeles y 1200 fotogramas).

8.3.3. Selección de parámetros

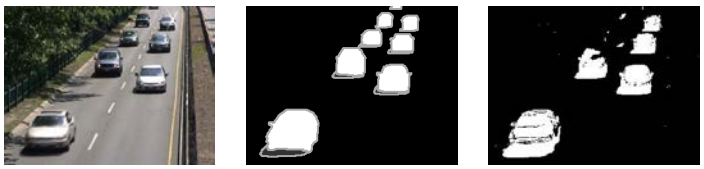
Se ha seleccionado un rango de valores para el parámetro Factor de Compresión (Compression Factor), que es el parámetro de test y que puede tomar diferentes valores. Para los parámetros del método MFBM hemos utilizado los valores recomendados por sus autores, por lo que estos parámetros son fijos. La combinación de los valores de parámetros forman el conjunto de configuraciones que se ha utilizado para comprobar el rendimiento de cada secuencia. Estos valores se muestran en la Tabla 8.2.

[1] http://changedetection.net/

Figura 8.1: Resultados de la propuesta para el video *Highway*.

(a) Un fotograma original, su máscara de verdad (Ground Truth o GT) y la salida del método MFBM, respectivamente.

Fotograma	GT	MFBM



(b) Máscaras de salida tras la aplicación del método MFBM con las combinaciones los Métodos de Compresión (primera columna) y Factores de Compresión (primera fila).

	0.875	0.75	0.625	0.5	0.375	0.25	0.125
AVG							
NN							
LIN							
CUB							

8.3.4. Resultados

El objetivo es determinar la influencia de los métodos de compresión analizados en la máscara de primer plano producida por el método de detección de objetos y su tiempo de ejecución.

Desde una perspectiva cualitativa, los experimentos muestran cómo los métodos de compresión afectan al resultado, tal y como se observa en la Figura 8.1. Cuando el Factor de Compresión disminuye, el resultado pierde detalle, apareciendo los objetos fuertemente pixelados, haciendo que parezcan cuadrados. Por otro lado, el proceso de compresión tiene consecuencias favorables: en muchos casos el resultado tiene un menor nivel de ruido, y el interior de los objetos está mejor definido que en el resultado original.

Desde un punto de vista cuantitativo, tres medidas de rendimiento han sido consideradas, denominadas exactitud, tiempo de ejecución y memoria utilizada. La mejor configuración para cada secuencia se muestra en la Figura 8.2. Del mismo modo, la Figura 8.3 muestra los resultados de cada método para las configuraciones probadas.

Como se puede observar en la Figura 8.2, conforme el Factor de Compre-

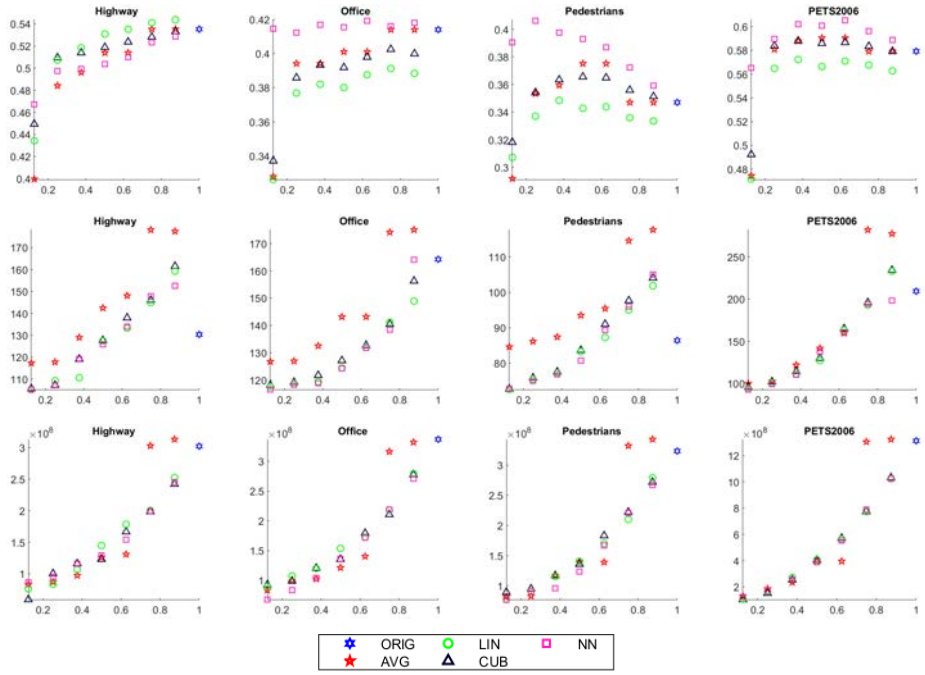


Figura 8.2: Comparativa de cada medida por factor de compresión: exactitud (primera fila), tiempo de ejecución (segunda fila), y memoria utilizada (tercera fila), en el eje y; todos ellos frente al factor de compresión, en el eje x. Cada video probado corresponde a una columna: Highway, Office, Pedestrians y PETS2006, respectivamente.

sión decrece, los requisitos de tiempo y memoria son menores. Además, todas las configuraciones probadas necesitan menos memoria que ORIG, excepto la compresión AVG que utiliza más memoria para valores mayores o iguales a 0.75 del Factor de Compresión. Este no es el caso del tiempo de ejecución, ya que existen muchas configuraciones con un tiempo de ejecución mayor que ORIG.

Además de esto, aplicar compresión a las imágenes no siempre significa una menor exactitud. Hay numerosas configuraciones que exhiben una exactitud similar o incluso superior que la configuración original.

La prueba más interesante es el video PETS2006 porque este video tiene el tamaño de fotograma más grande de las secuencias probadas. Como puede verse en la Figura 8.3 las diferencias en la memoria utilizada y en los tiempos de ejecución son mayores que en las otras secuencias.

La memoria utilizada por el algoritmo es muy similar con cada Método de Compresión y el mismo Factor de Compresión (excepto la compresión AVG, que utiliza más memoria para valores del Factor de Compresión superiores o iguales a 0.75), mientras que el tiempo de ejecución y la exactitud varían significativamente dependiendo del Factor de Compresión. En la Figura 8.4

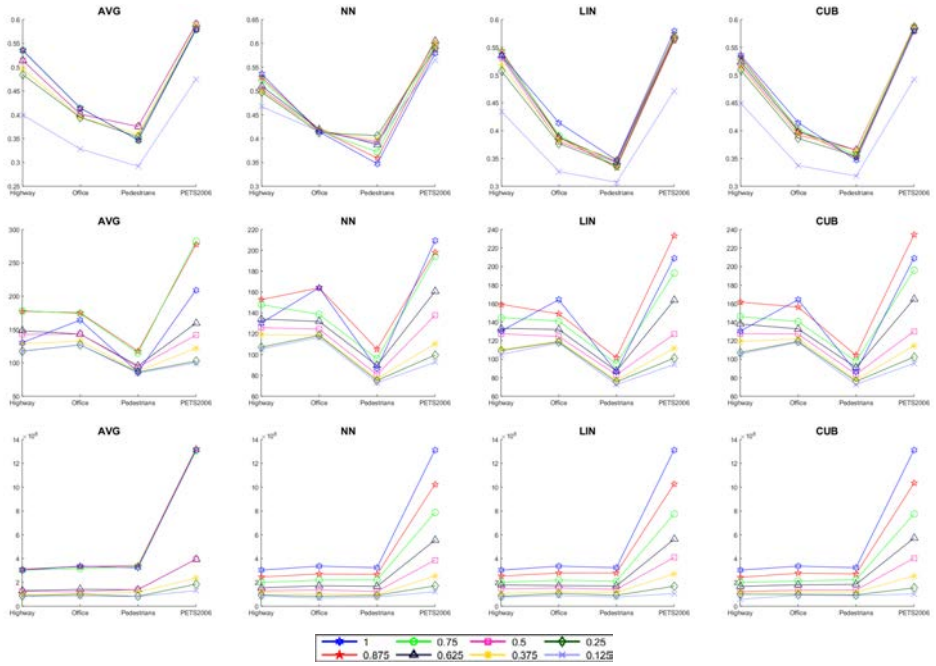


Figura 8.3: Comparativa de cada medida por video. Primera, segunda y tercera filas muestran la exactitud, el tiempo de ejecución (en segundos) y la memoria utilizada (en bytes), en el eje y, respectivamente. Cada columna se corresponde con uno de los métodos de compresión utilizados. El eje x de cada figura se corresponde con las secuencias probadas. Nótese que los valores de cada método están conectados entre ellos con líneas para una mejor comparación de los métodos en cada video, pero esto no significa que los videos estén relacionados.

la exactitud obtenida y el tiempo de ejecución necesitados se muestran para cada método y secuencia.

Los videos Office, Pedestrians y PETS2006 obtienen rendimiento similar. El método NN ofrece el mejor compromiso entre exactitud y tiempo de ejecución. CUB y AVG presentan una exactitud similar pero AVG emplea más tiempo. LIN es rápido pero su exactitud es peor que las otras. Sin embargo, la comparativa de la secuencia Highway ofrece que el método LIN es el mejor.

De acuerdo con estos resultados, el método NN aplicado a un sistema de detección de primer plano hará decrecer el uso de la memoria y podría reducir el tiempo de ejecución sin afectar a la exactitud de manera significativa. En algunos casos, la exactitud podría incluso ser mejorada.

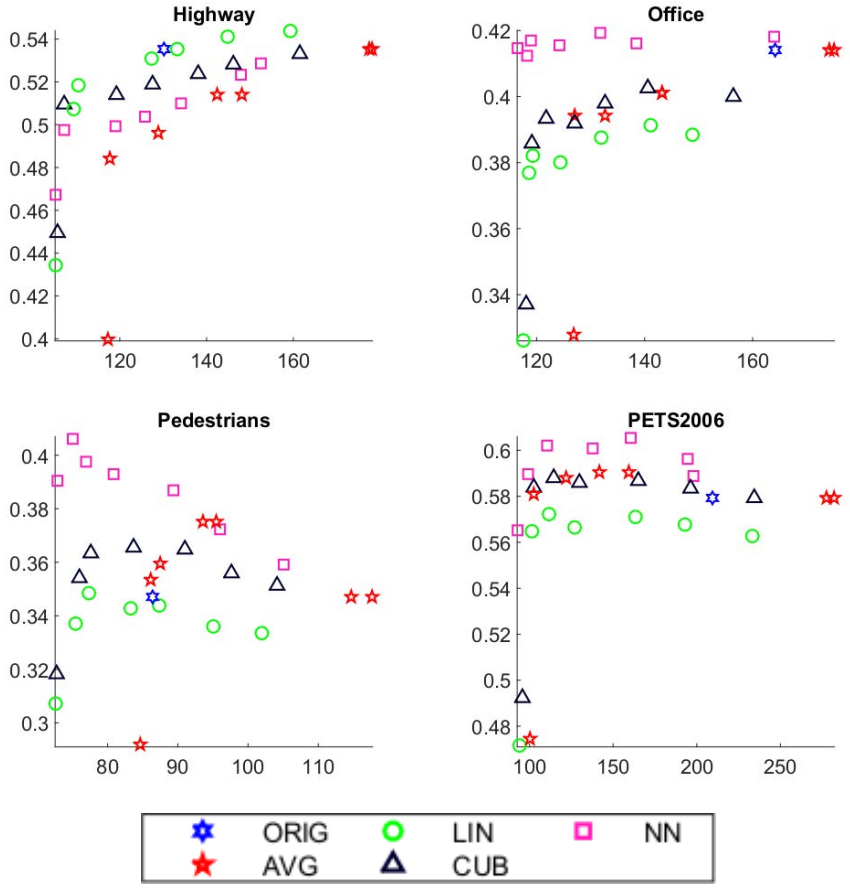


Figura 8.4: Exactitud y tiempo de ejecución (en segundos) para todas las configuraciones probadas y videos. La exactitud se corresponde con el eje y, mientras que el tiempo de ejecución se corresponde con el eje x.

8.4. Conclusiones

En este capítulo se ha presentado un método para la reducción del tamaño del fotograma para la detección de primer plano en videos. El método está dividido en tres etapas, que serían la compresión del fotograma original del video utilizando una función específica de interpolación, la detección del primer plano en el fotograma comprimido mediante un modelo de fondo probabilístico, y descomprimir las probabilidades de clase para el fotograma reducido para estimar las probabilidades de clase del fotograma original. Para el proceso de compresión se ha utilizado el Promedio de Bloques (Blockwise average o AVG), Vecino Más Cercano (Nearest neighbor o NN), Interpolación Bilinear (LIN), e Interpolación Bicúbica (CUB), además del modelo probabilístico de fondo MFBM López-Rubio y López-Rubio (2015) para la detección de primer plano.

Se han seleccionado cuatro vídeos diferentes y bien conocidos para los experimentos, donde la exactitud, el tiempo de ejecución y la memoria utilizada fueron analizados para varias configuraciones. Estos resultados ofrecieron un rendimiento similar o mejor que los obtenidos por el mismo método sin aplicar ningún método de compresión (ORIG), con la ventaja de decrementar significativamente la carga computacional del algoritmo.

Parte II

Cámara PTZ

Esta segunda parte de la tesis presenta los trabajos desarrollados en los que se hace uso de una cámara PTZ. Así, en esta parte se presentan dos trabajos con esta particularidad. El primero de ellos hace un seguimiento del objeto más anómalo del escenario, siendo el propio sistema el que decide cuáles son los objetos anómalos y cuáles no, decisión que se basa en un modelo entrenado previamente con un video en el que no aparecen objetos anómalos. Por su parte, el segundo trabajo muestra un sistema que indica a la cámara los movimientos a realizar en función de la salida producida por un modelo de fondo no paramétrico y mejorada con un GNG.

Capítulo 9

Detección de objetos anómalos mediante búsqueda con cámaras PTZ

Inteligencia es la habilidad de adaptarse a los cambios.

Stephen Hawking

Resumen: Debido a la gran cantidad de información visual que se genera diariamente, las propuestas que analizan y procesan datos automáticamente son cada vez más necesarias. Este capítulo se centra en la detección de objetos anómalos en secuencias de video capturados con una cámara PTZ (pan-tilt-zoom o de movimiento horizontal, vertical y cambio de la distancia focal (movimiento de zoom), considerando como anomalías los objetos que no deberían aparecer en un escenario específico (por ejemplo, personas en una autovía). La metodología propuesta se compone de tres módulos diferentes. Una etapa de detección de objetos, donde se utilizan redes neuronales profundas para detectar los objetos que aparecen en la escena; un módulo de detección de anomalías, donde se considera una mixtura de distribuciones de Dirichlet para detectar automáticamente, y sin entrenamiento supervisado, aquellos objetos detectados que son anómalos; y por último, un controlador de cámara PTZ que permite seguir y centrarse el foco en el objeto más anómalo de la escena. Los resultados experimentales muestran el rendimiento y viabilidad de la propuesta.

9.1. Introducción

La proliferación en los últimos años de una gran cantidad de información visual en forma de secuencias de datos ha motivado el desarrollo e investigación de nuevas técnicas para analizar esos datos, procesarlos y extraer información relevante. Por tanto, el campo de la videovigilancia inteligente ha crecido significativamente e involucra cada vez más a una comunidad científica más grande. En particular, una de las tareas más importantes a considerar es la detección automática de objetos en movimiento que no son muy frecuentes en una escena. Como ejemplo, podría mencionarse que no es muy normal encontrarse animales si se analizan secuencias capturadas en una autovía. Por tanto, si esta detección ocurre, entonces puede considerarse como anomalía. La detección automática de cada anomalía es una de las principales áreas de investigación de la videovigilancia .

En general, la arquitectura de un sistema de videovigilancia comienza con una etapa de detección de objetos en movimiento (Aggarwal y Ryoo, 2011). Posteriormente, es necesario hacer un seguimiento de esos objetos a lo largo de la secuencia, lo que implica un emparejamiento entre los objetos detectados en cada fotograma y los objetos detectados previamente. Con los datos obtenidos en la etapa de seguimiento es posible analizar el comportamiento de los objetos y extraer información de alto nivel.

En este tipo de sistemas el principal obstáculo es la primera etapa, porque los métodos de procesado del video basados en el modelado del fondo de la escena generan errores en la segmentación. Inconvenientes como fondos multimodales, sombras proyectadas o la similitud entre los tonos de los píxeles de primer plano y del fondo son un ejemplo de los problemas encontrados en esta fase (Bouwmans, 2014b). Además, para reducir costes, el uso de cámaras PTZ (pan-tilt-zoom o de movimiento horizontal, vertical y cambio de la distancia focal (zoom)) ha crecido considerablemente, ya que permiten ver una escena mucho más grande que una simple cámara en lugar de utilizar varias fijas. En estas situaciones, los algoritmos de detección de primer plano tradicionales no son adecuados y se necesitan nuevas propuestas más complejas para solventar el problema, aunque con otro tipo de imágenes de entrada (López-Rubio y López-Rubio, 2015).

En los últimos años, la aparición de las redes de aprendizaje profundo para la detección de objetos en imágenes ha significado un punto de inflexión en la detección de objetos en secuencias de video. Por tanto, es posible utilizar redes pre-entrenadas con miles de datos y un gran número de tipos de objetos para detectar objetos en movimiento en una escena, proporcionando resultados más estables que aquellos obtenidos de las propuestas clásicas. Entre las diferentes propuestas encontradas en la literatura (Wang, 2016), un tipo de red neuronal convolucional (Convolutional Neural Network o CNN), la Faster RCNN (Ren et al., 2017) es utilizada, dada su eficiencia en el re-

conocimiento de objetos y su eficacia para analizar imágenes en tiempo real. La principal ventaja de este tipo de redes es que pueden ser aplicadas indistintamente del tipo de cámara utilizada (fija o PTZ). De esta etapa de reconocimiento de objetos, es posible seguir los objetos o realizar cualquier tipo de análisis de datos relevantes. Específicamente, la propuesta se centra en determinar automáticamente qué objetos pueden ser considerados anómalos en una escena.

Por tanto, se propone una metodología para la detección de objetos anómalos en la escena, utilizando redes de aprendizaje profundo y algoritmos de agrupamiento no supervisado, con el posterior control de la cámara PTZ para seguir el objeto más anómalo considerado. Las dos principales novedades de la propuesta son:

- El desarrollo de un algoritmo de aprendizaje no supervisado no paramétrico basado en la distribución de Dirichlet para determinar qué objetos son anómalos de los detectados por la Faster-RCNN. Nótese que no es necesario tener ningún dato etiquetado a mano para aprender qué objetos son anómalos, ya que el sistema analiza la escena y determina la probabilidad de que cada objeto detectado sea anómalo.
- El controlador de la cámara PTZ, para seguir al objeto más anómalo, calculando los comandos de las acciones horizontal, vertical o de zoom cuando sea necesario. Si no se ha detectado un objeto anómalo, entonces la cámara PTZ continúa con su movimiento previamente establecido. Nótese que cuando se construye un sistema de seguimiento centrado en la detección del objeto más anómalo, el uso de una red como la Faster-RCNN como detector de objetos proporciona mejores resultados que las anteriores propuestas.

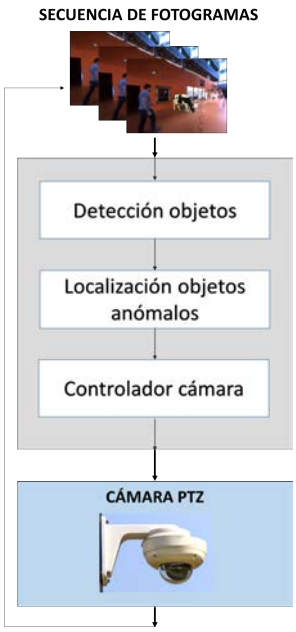
El resto del capítulo tiene la siguiente estructura. La Sección 9.2 describe el sistema de detección de anomalías propuesto. La Sección 9.3 presenta los experimentos que demuestran el rendimiento de la propuesta. Tras esto, algunas características importantes de la propuesta se discuten en la Sección 9.4. Por último, la Sección 9.5 concluye el capítulo.

9.2. El modelo

El sistema propuesto se puede describir como se muestra en la Figura 9.1. La cámara PTZ proporciona un fotograma al sistema de detección de anomalías propuesto. Este sistema está compuesto de tres módulos. El primero de ellos es el módulo de detección de objetos, donde su entrada es la imagen facilitada por la cámara y su salida es una lista de los objetos de la imagen detectados y clasificados. Después, esta salida es dada al módulo de localización de objetos anómalos, que determina cuál es el objeto detectado

más anómalo. Tras esto, de acuerdo con esta respuesta, el controlador de la cámara proporciona los comandos que considera oportunos a la cámara.

Figura 9.1: Esquema de la propuesta.



9.2.1. Detección de objetos

Antes de detectar los objetos anómalos en una escena, se tiene que buscar los objetos presentes en dicha escena. El detector de objetos de la propuesta está basado en una arquitectura de aprendizaje profundo debido a su mejor precisión a la hora de detectar clases. Tradicionalmente, el modelo de aprendizaje profundo más común es la red neuronal convolucional (Convolutional Neural Network o CNN) que es un tipo importante de red neuronal alimentada hacia adelante con especial éxito en aplicaciones donde la información objetivo puede ser representada por una jerarquía de características locales (Bengio et al., 2015).

Una CNN está definida como composición de varias capas convolucionales y varias capas totalmente conectadas. Cada capa convolucional es, en general, la composición de una capa no lineal y una capa de agrupación o submuestreo para obtener alguna invarianza espacial. Para imágenes, la capa no lineal de la CNN tiene ventaja, a través de conexiones locales y el reparto de peso, de una estructura 2D presente en los datos. Estas dos condiciones imponen una regularización muy fuerte en el número total de pesos en el modelo, lo que permite un exitoso entrenamiento del modelo mediante el uso de la retro-propagación.

En este caso, el aprendizaje profundo propuesto se compone de dos etapas: (i) detecta una región rectangular con un objeto en su interior; y, (ii) identifica la clase del objeto dentro de la región. Por tanto, se tiene que usar un modelo de regiones con características CNN (Regions with CNN features o R-CNN) que permite detectar y clasificar al mismo tiempo (Girshick et al., 2014). Como el principal objetivo de la propuesta es la detección de anomalías, se utiliza un modelo R-CNN pre-entrenado con buen rendimiento, lo que permite centrarse en el objetivo principal. Concretamente, se ha utilizado el modelo Faster-RCNN (Ren et al., 2017) que está compuesto de dos módulos: una red totalmente convolucional profunda que propone regiones y un detector Fast R-CNN (Girshick, 2015) que utiliza las regiones y produce la mínima región rectangular que envuelve a cada objeto (bounding box o BB) y las clases.

Nótese que para ambas partes, las primeras trece capas son compartidas y basadas en el modelo VGG-16 (Simonyan y Zisserman, 2014). Esas capas están pre-entrenadas con el conjunto de datos ImageNet y se utilizan para producir un conjunto de mapas de características. Estos mapas son usados como entradas a la red de propuesta de regiones (Region Proposal Network o RPN) que es una red pequeña que se desliza sobre los mapas de características para producir las regiones de interés. Por último, la parte Fast R-CNN toma dichas regiones y produce la regiones rectangulares y las clases correspondientes.

Durante el proceso de entrenamiento, todas las partes son entrenadas juntas en un proceso de extremo a extremo y las capas pre-entrenadas son ajustadas con los nuevos datos.

La salida del modelo es un conjunto de detecciones en el tiempo t:

$$\mathcal{D}_t = \{(\mathbf{h}_{i,t}, \mathbf{q}_{i,t}) \mid i = 1, ..., D_t\} \tag{9.1}$$

$$\mathbf{h}_{i,t} = (h_{i,t,1}, h_{i,t,2}, h_{i,t,3}) \in \mathbb{R}^3 \tag{9.2}$$

$$\mathbf{q}_{i,t} = (q_{i,t,1}, ..., q_{i,t,K}) \in \mathbb{R}^K \tag{9.3}$$

donde $(h_{i,t,1}, h_{i,t,2})$ son las coordenadas del píxel del i-ésimo objeto detectado en el actual fotograma, y $h_{i,t,3}$ es el área de la mínima región rectangular que envuelve a ese objeto, expresada en píxeles. Asociada a cada detección hay una probabilidad de pertenencia a cada clase de objeto, $q_{i,t,k} \in [0, 1]$, donde $C_k \in Classes$ y el número de clases de objetos es K.

9.2.2. Localización de objetos anómalos

Como la escena de video es grabada, entonces las detecciones son grabadas durante su duración. Las características posicionales $\mathbf{h}_{i,t}$ no pueden

ser usadas para evaluar si una detección es anómala porque cambian con los movimientos de la cámara. El ratio de aspecto de la región rectangular podría utilizarse, pero no es seguro porque las obstrucciones parciales cambian el ratio de aspecto. Por otro lado, las probabilidades de clase $\mathbf{q}_{i,t}$ deberían ser más estables con respecto a los movimientos de la cámara. Por lo que un modelo probabilístico de la ocurrencia de una detección puede ser expresada como sigue:

$$p\left(\mathbf{q}\right) = \sum_{j=1}^{M} \pi_j p_j\left(\mathbf{q}\right) \tag{9.4}$$

$$\sum_{j=1} \pi_j = 1 \tag{9.5}$$

donde $p\left(\mathbf{q}\right)$ es una mixtura probabilística de M distribuciones de Dirichlet $p_j\left(\mathbf{q}\right)$, y π_j es la probabilidad a priori de la componente j-ésima de la mixtura. La densidad de probabilidad para la componente j-ésima de la mixtura viene dada por:

$$p_j\left(\mathbf{q}\right) = \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_{jk}\right)}{\prod_{k=1}^{K} \Gamma\left(\alpha_{jk}\right)} \prod_{k=1}^{K} q_k^{\alpha_{jk}-1} \tag{9.6}$$

$$\log p_j\left(\mathbf{q}\right) = \left(\log \Gamma\left(\sum_{k=1}^{K} \alpha_{jk}\right)\right) - \left(\sum_{k=1}^{K} \log \Gamma\left(\alpha_{jk}\right)\right) + \sum_{k=1}^{K} \left(\alpha_{jk} - 1\right) \log q_k \tag{9.7}$$

Los parámetros de la distribución de Dirichlet $\boldsymbol{\alpha}_j \in \mathbb{R}^K$ son estimados por un algoritmo adecuado. Aquí se propone un estimador de densidad del núcleo de Dirichlet no paramétrico, donde todas las probabilidades a priori son iguales:

$$\pi_j = \frac{1}{M} \tag{9.8}$$

Aquí se confía en la parametrización alternativa de la distribución de Dirichlet:

$$s_j = \sum_{k=1}^{K} \alpha_{jk} \tag{9.9}$$

$$m_{jk} = \frac{\alpha_{jk}}{s_j} \tag{9.10}$$

$$\sum_{k=1}^{K} m_{jk} = 1 \tag{9.11}$$

$$E\left[\mathbf{q} \mid j\right] = \mathbf{m}_j \tag{9.12}$$

donde $\mathbf{m}_j$ es la media de la componente j-ésima de la mixtura y s_j es su parámetro de precisión. Se asume un parámetro de precisión común s para todas las componentes de la mixtura, es decir, los vectores de medias $\mathbf{m}_j$ son ajustados a los datos de entrenamiento y la precisión s se obtiene por minimización de la probabilidad logarítmica negativa media sobre el conjunto de validación:

$$ANLL = -\sum_{v=1}^{V} \log p\left(\mathbf{q}_v\right) \tag{9.13}$$

Los datos de entrenamiento y validación para el algoritmo podrían venir de una secuencia de video donde la cámara explora la escena libremente, moviéndose en todas las direcciones y cambiando el zoom.

Luego la detección con la menor probabilidad es asociada al objeto más anómalo observado en el tiempo t:

$$d_t = \arg \min_{i \in \{1,\ldots,D_t\}} p\left(\mathbf{q}_{i,t}\right) \tag{9.14}$$

La cámara PTZ puede seguir únicamente un objeto en cada momento. Por tanto, se está interesado en las coordenadas en píxeles (x_1, x_2) del objeto más anómalo en la escena, y el área x_3 de su región rectangular. Junto con su velocidad de cambio $\frac{dx_q}{dt} = x_{q+3}$ forman el vector de estado verdadero $\mathbf{x} \in \mathbb{R}^6$. El vector de valores observados se denota $\mathbf{z}_t \in \mathbb{R}^3$, y se define como:

$$\mathbf{z}_t = \mathbf{h}_{d_t} \tag{9.15}$$

donde d_t se calcula de (9.14).

9.2.3. Controlador de cámara

Por último, el centroide y el tamaño del objeto objetivo de primer plano es proporcionado al controlador para decidir el movimiento de la cámara con el propósito de seguir al objetivo seleccionado. El área de la región rectangular del objetivo y el centro de dicho rectángulo son considerados el tamaño y el centroide del objetivo, respectivamente.

Los movimientos especificados en el controlador son el horizontal (pan), vertical (tilt) y el zoom. La cámara se mueve hacia la izquierda o la derecha aplicando el movimiento horizontal; con el movimiento vertical la cámara se mueve hacia arriba o abajo; y el zoom entrante o zoom saliente puede ser

aplicado con el movimiento de zoom. Para cada tipo de movimiento se define una cantidad de movimiento ρ_δ donde $\delta \in \{pan, tilt, zoom\}$.

Además, hay varios movimientos no permitidos. Por ejemplo, el zoom entrante no puede ser aplicado indefinidamente. Por tanto, para evitar estas situaciones, existen unos posibles valores máximo y mínimo para el movimiento zoom. Por tanto, estos valores están asociados a los límites físicos de la cámara y se denotan como $\boldsymbol{\Psi_\delta}$ y $\boldsymbol{\psi_\delta}$, respectivamente. Por otro lado, el controlador considera que cada tipo de movimiento puede aplicar un comando de no moverse.

Considerando estas condiciones, en cada fotograma t de la secuencia de video la cámara realiza un cambio de posición por cada tipo de movimiento. Por tanto, sea $(\boldsymbol{\alpha}_t, \boldsymbol{\beta}_t, \boldsymbol{\gamma}_t)$ la posición de la cámara en coordenadas esféricas para las posiciones horizontal, vertical y de zoom. En el $(t+1)$-ésimo instante de tiempo la cámara estará en la siguiente posición:

$$(\boldsymbol{\sigma}_{t+1}, \boldsymbol{\beta}_{t+1}, \boldsymbol{\gamma}_{t+1}) = (\boldsymbol{\sigma}_t, \boldsymbol{\beta}_t, \boldsymbol{\gamma}_t) + (\Delta\boldsymbol{\sigma}_t, \Delta\boldsymbol{\beta}_t, \Delta\boldsymbol{\gamma}_t) \tag{9.16}$$

donde para cada tipo de movimiento δ existe un grado de cambio $\delta_t \in \{-\rho_\delta, 0, \rho_\delta\}$. Este δ_t es la orden del controlador para reducir, permanecer quieto o incrementar la posición de la cámara en ese movimiento en el instante t-ésimo, respectivamente.

Además, se consideran algunas situaciones donde la cámara no aplica ningún movimiento para realizar los menores movimientos posibles. Por ejemplo, cuando el centroide del objeto objetivo de primer plano está cerca de la posición horizontal y/o vertical del centro del fotograma, entonces el controlador no indica ningún movimiento horizontal y/o vertical, respectivamente. De la misma forma, cuando un objetivo seguido tiene un tamaño (en píxeles) que está entre un mínimo y un máximo valor de porcentaje con respecto al número de píxeles totales del fotograma, en este caso el controlador no aplica ningún zoom. Por tanto, se han definido un valor mínimo y máximo ϕ_δ and Φ_δ para cada tipo de movimiento δ, notándose que los valores para las posiciones horizontales y verticales indican la distancia (en grados) entre el centroide del objetivo y el centro del fotograma, y los valores para el zoom indican el porcentaje de tamaño del objetivo respecto del número de píxeles del fotograma.

Por último, el controlador trata de encontrar un objetivo cuando no ha encontrado ninguno, indicando diferentes comandos a la cámara. En este sentido, el controlador previene de situaciones donde un objetivo nunca será encontrado por la cámara. Por ejemplo, si la cámara está enfocando un área por donde no pasa nadie en la parte superior del escenario con un zoom entrante alto y no ejecuta ningún movimiento, sería bastante probable que la cámara nunca encontrara un nuevo objetivo. Para evitar estas situaciones, cuando el sistema no encuentra un objetivo, se ejecutan varios movimientos predeterminados hasta que se encuentra un nuevo objetivo. Estos comandos

son: realizar siempre un movimiento horizontal a la derecha, y mantener un valor medio para los movimientos vertical y de zoom entre sus límites máximos y mínimos $\boldsymbol{\Psi_\delta}$. Es decir, siguiendo con el ejemplo dado anteriormente, en esta situación la cámara empezaría a ejecutar un movimiento horizontal hacia la derecha, vertical hacia abajo y de zoom saliente; cuando la cámara alcanza una posición vertical media, se para el movimiento vertical; por otro lado, si la cámara alcanza una posición de zoom media, se queda quieta en relación con el movimiento de tipo zoom; sin embargo, la cámara continuará ejecutando el movimiento horizontal hacia la derecha hasta que un objetivo sea encontrado.

9.3. Resultados experimentales

Los experimentos computacionales que se han ejecutado y sus resultados se muestran en esta sección.

9.3.1. Métodos

La metodología propuesta está escrita en Matlab y utiliza el modelo Faster R-CNN basado en el código [1] liberado por sus autores. Nótese que se ha entrenado el modelo utilizando los parámetros por defecto definidos en el código. El módulo de localización de objetos anómalos también está escrito en Matlab. Y por último, para simular el funcionamiento de una cámara PTZ mediante un video panorámico de 360 grados, se ha utilizado la librería *virtualptz*, que es una propuesta reproducible (Chen et al., 2015) y nuestro controlador de cámara se ha desarrollado basado en ella. Esta librería está implementada en C++ y utiliza la librería OpenCV[2]. La librería *virtualptz* puede ser encontrada en su página web [3].

Además, para hacer una comparativa, se han considerado otros dos modelos de detección de anomalías: uno está basado en el algoritmo de las K-medias (K-means) (Lloyd, 1982; Arthur y Vassilvitskii, 2007), y el otro se inspira en los mapas autoorganizados (Self-Organizing Maps o SOM) (Kohonen, 1982; Kohonen y Honkela, 2007). El algoritmo de las K-medias es un método de agrupamiento cuyo objetivo es dividir el conjunto de datos en k grupos; mientras que el SOM es un tipo de red neuronal que adapta su estructura a las propiedades topológicas del conjunto de datos. Ambos métodos se han implementado en Matlab.

Los experimentos se han realizado en un ordenador con un microprocesador de 8 núcleos Intel i7 3.60 GHz, 32 GB RAM y una tarjeta gráfica Titan X.

[1] `https://github.com/ShaoqingRen/faster_rcnn`
[2] http://opencv.org/
[3] `https://bitbucket.org/pierre_luc_st_charles/virtualptz_standalone`

9.3.2. Secuencias

Los videos que se han seleccionado para ejecutar los experimientos presentan dos escenarios: el primer escenario (scenario3) es una sala y el segundo (scenario5) es una habitación. En ambos escenarios hay gente y animales moviéndose en ellos. Los videos que se han obtenido de estos escenarios son secuencias notadas como textitscenario3cow (compuesto por 1061 fotogramas), *scenario3dogcow* (531 fotogramas), *scenario3horse* (1061 fotogramas), *scenario3sheep* (1061 fotogramas) and *scenario5dog* (1835 fotogramas). Estos videos pueden descargarse de la página web [4] y se han obtenido de la edición de los videos que pertenecen al conjunto de datos utilizados en (Chen et al., 2015), cuyas secuencias se pueden ver en su página web[5]. La peculiaridad de los nuevos videos es que se han añadido diferentes animales sintéticos moviéndose en ellos para comprobar la viabilidad del módulo de localización de objetos anómalos. Por tanto, se considera como objeto anómalo un animal, y el resto de elementos que aparecen en el video son considerados no anómalos.

Para calcular el rendimiento del sistema propuesto desde un punto de vista cualitativo y cuantitativo, se ha obtenido la máscara de verdad (Ground Truth o GT) de cada secuencia de video probada. Esta máscara de verdad está formada por el centroide y la región rectangular mínima de cada objeto anómalo y el conjunto de movimientos apropiados de la cámara (horizontal, vertical y zoom) de acuerdo a los objetos anómalos que aparecen en ellas. El objetivo es comparar estos movimientos ideales con los que se obtienen de la ejecución del sistema durante la secuencia.

Además, la máscara de verdad de un fotograma tiene un conjunto de movimientos para cada objeto anómalo presente en él. Por ejemplo, si un fotograma muestra dos objetos anómalos, uno de ellos en el lado izquierdo y el otro en el lado derecho del fotograma, el movimiento ideal de la cámara podría ser seguir al objeto de la izquierda o el de la derecha. Esto se debe a que no se ha indicado ninguna restricción adicional para seleccionar el objeto anómalo que va a ser seguido y se seleccionará de acuerdo al resultado producido por los modelos de detección de anomalías.

Por último, como se ha dicho anteriormente, los videos son secuencias panorámicas de 360 grados, por lo que la cámara del sistema puede apuntar al inicio con un ángulo entre 0 y 360. Para ejecutar los experimentos con un amplio rango de videos se ha considerado como puntos de inicio los ángulos iguales a 0, 90, 180 y 270. Por tanto, el video notado como *scenario3dogcow_ 180* indica que el video seleccionado es la secuencia *scenario3dogcow* con un ángulo de inicio igual a 180.

[4] `http://www.lcc.uma.es/~miguelangel/papers/anomalous_object_detection_by_active_search_with_PTZ_cameras/dataset.rar`

[5] `https://drive.google.com/file/d/0B55Ba7lWTLh4QzNiOFFpSHFRNXM)`

9.3.3. Consideraciones previas

Es imposible detectar todos los tipos de objetos posibles ya que se debería tener un modelo entrenado con todos los tipos de objetos del mundo. Por tanto, se ha entrenado el modelo de detección de objetos para detectar los 20 tipos de objetos incluidos en el conjunto de datos PASCAL VOC 2007 (Everingham et al., 2018).

Además de los módulos de localización de objetos anómalos Dirichlet, K-medias y SOM, se ha considerado el módulo que no detecta ninguna anomalía, que es notado como *None*.

Por otro lado, además del modelo de controlador de cámara propuesto, se han considerado otros modelos que proporcionan la información necesaria al controlador de la cámara para obtener los mismos comandos de movimientos a la cámara en cada fotograma, independientemente de las anomalías detectadas, y se denotan como modelos de movimiento básico. El *básico derecho* produce un resultado donde siempre se le indica a la cámara el comando de hacer un movimiento horizontal hacia la derecha, mientras que el *básico izquierdo* produce un movimiento horizontal hacia la izquierda en cada fotograma del video. Otro modelo considerado es el *básico aleatorio*, que siempre produce un movimiento horizontal aleatorio. Por último, el resultado del *básico estático* es quedarse quieto.

Además, se han denominado dos categorías para agrupar los diferentes modelos de cada módulo: las categorías *Adaptable* y *No adaptable*. En relación a los modelos de localización de objetos anómalos, se han agrupado los modelos de Dirichlet, K-medias y SOM como pertenecientes a la categoría Adaptable, mientras que el modelo None pertenece a la categoría No adaptable. Respecto a los modelos de controlador de cámara, los modelos de movimiento básico se han etiquetado como No adaptables, y el modelo que se ha propuesto es etiquetado como Adaptable. Estos modelos posibles que pueden ser aplicados a los módulos de localización de objetos anómalos y controlador de cámara, respectivamente, se estructuran como sigue:

- Localización de objetos anómalos
  - No adaptable
    - None
  - Adaptable
    - Dirichlet
    - K-medias
    - SOM
- Controlador de cámara
  - No adaptable
    - Básico derecho
    - Básico izquierdo
    - Básico aleatorio
    - Básico estático
  - Adaptable
    - Propuesta

Por último, nótese que los modelos adaptables de localización de objetos anómalos calculan un valor para objeto detectado en cada fotograma que significa cuánto de anómalo es ese objeto. Una vez esta operación ha terminado, el modelo devuelve el valor que corresponde con el objeto más anómalo. En el caso de los modelos K-medias y SOM, este valor es la distancia (con signo cambiado) del objeto más anómalo al grupo más cercano y al prototipo más cercano, respectivamente.

9.3.4. Selección de parámetros

Primero se tiene que definir un conjunto de valores configurados para los parámetros relacionados con los diferentes métodos que se han empleado. Siguiendo las recomendaciones de los autores y la experiencia de trabajos previos se han considerado las posibles configuraciones de parámetros que se muestran en la Tabla 9.1. Debe resaltarse que los valores del módulo controlador de la cámara PTZ tienen que ser seleccionados con el propósito de dar atención a la baja tasa de fotogramas que tienen los videos de referencia.

Figura 9.2: Fotogramas correspondientes a la salida del módulo de detección y clasificación obtenidos de la secuencia scenario3dogcow_180 ejecutada con el movimiento básico derecho. Se muestran las regiones mínimas rectangulares en verde y su etiqueta correspondiente a cada objeto detectado y clasificado por el detector Faster-RCNN en los fotogramas 4, 84 y 146, respectivamente. La etiqueta contiene la clase del objeto y la probabilidad de pertenencia a esa clase. Se puede observar cómo el módulo de detección y clasificación detecta objetos que no existen en un fotograma.

Tabla 9.1: Configuraciones de parámetros consideradas para cada método utilizado.

Método	Parámetros
Faster RCNN	Umbral, $\tau = 0{,}80$
Dirichlet	Precisión, P $P = \{10^{0,2}(1{,}5849), 10^{0,4}(2{,}5119), 10^{0,6}(3{,}9811),$ $10^{0,8}(6{,}3096), 10^{1,0}(10), 10^{1,2}(15{,}8489),$ $10^{1,4}(25{,}1189), 10^{1,6}(39{,}8107), 10^{1,8}(63{,}0957),$ $10^{2}(100), 10^{2,2}(158{,}4893), 10^{2,4}(251{,}1886),$ $10^{2,6}(398{,}1072), 10^{2,8}(630{,}9573), 10^{3}(1000),$ $10^{3,2}(1584{,}8932), 10^{3,4}(2511{,}8864), 10^{3,6}(3981{,}0717),$ $10^{3,8}(6309{,}5734), 10^{4}(10000)\}$
K-medias	Número de grupos, N $N = \{2, 4, 6, 8, 10, 12, 14, 16, 18, 20,$ $22, 24, 26, 28, 30, 32, 34, 36, 38, 40\}$ Distancia, $D = SquaredEuclideandistance$ Máximo número de iteraciones, $MaxIter = 100$
SOM	Número de neuronas, $NumNeurons$ $NumNeurons = \{4, 9, 16, 25, 36, 49, 64, 81, 100, 121,$ $144, 169, 196, 225, 256, 289, 324, 361, 400, 441\}$ Número de pasos, $NumSteps = 100000$ Número de pasos por época, $NumStepsPerEpoch = 10000$ Tasa de aprendizaje inicial, $InitialLearningRate = 0{,}4$ Radio máximo, $MaxRadius = sqrt(NumNeurons)/8$ Tasa de aprendizaje de convergencia, $ConvergenceLearningRate = 0{,}01$ Radio de convergencia, $ConvergenceRadius = 1$
Virtualptz	Variación del movimiento horizontal, $\rho_h = 2$ Variación del movimiento vertical, $\rho_v = 2$ Variación del movimiento zoom, $\rho_z = 2$ Límite mínimo horizontal, $\psi_h = -180$ Límite máximo horizontal , $\Psi_h = 180$ Límite mínimo vertical , $\psi_v = 0$ Límite máximo vertical , $\Psi_v = 180$ Límite mínimo zoom , $\psi_z = 40$ Límite máximo zoom , $\Psi_z = 140$ Distancia mínima horizontal, $\phi_h = -10$ Distancia máxima horizontal, $\Phi_h = 10$ Distancia mínima vertical, $\phi_v = -20$ Distancia máxima vertical, $\Phi_v = 20$ Distancia mínima zoom, $\phi_z = 0{,}05$ Distancia máxima zoom, $\Phi_z = 0{,}30$

Tras esto, se ha decidido elegir un conjunto de valores óptimos de los parámetros para los modelos de detección de anomalías que han sido considerados. Para seleccionar estas configuraciones óptimas, primeramente se ha procesado la secuencia scenario3 (el video sin objetos anómalos) considerando el modelo de localización de objetos anómalos None y el modelo de controlador de cámara básico derecha, y se ha obtenido la salida de la detección y la clasificación del método para cada fotograma. De esta información se ha seleccionado un conjunto de 500 objetos aleatorios y se ha usado como datos de entrenamiento de los modelos de detección de anomalías. Aunque no existe ningún objeto anómalo en el video, el módulo de detección y clasificación detecta varios objetos que no existen en la secuencia (en este caso trenes, aviones, mesas...). La Figura 9.2 muestra algunos fotogramas con este comportamiento.

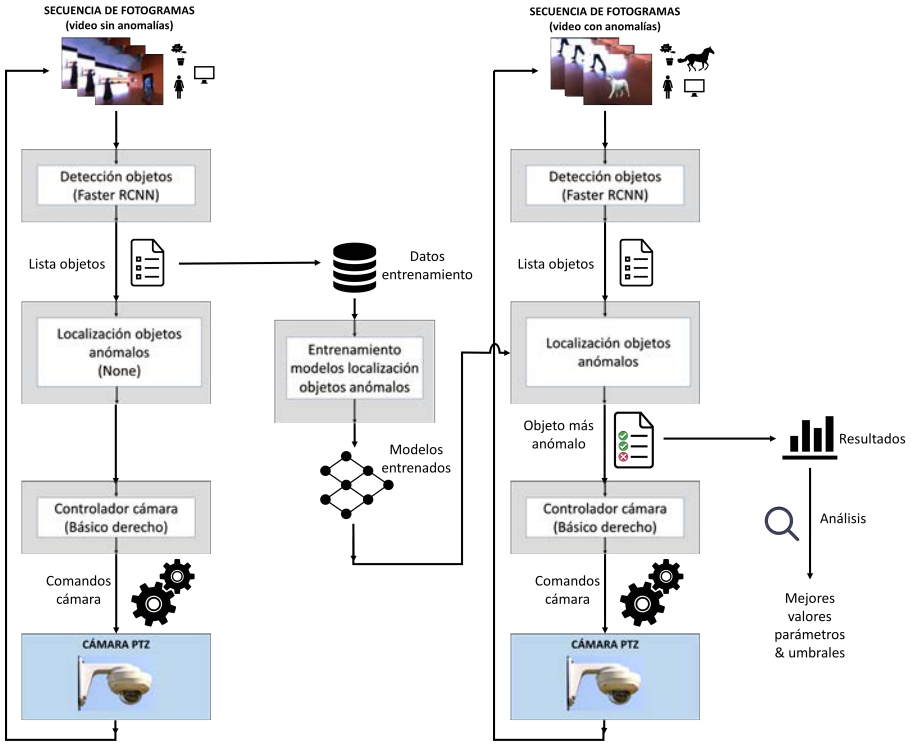
Los valores de los parámetros de los métodos que determinan cómo de anómalo es un objeto han sido seleccionados de acuerdo a un estudio preliminar del rendimiento de los diferentes modelos de detección de anomalías considerando un conjunto de posibles configuraciones de esos parámetros. Para hacer esto, se ha procesado el video scenario3dogcow_0 aplicando cada modelo adaptable de localización de objetos anómalos y el modelo de controlador de cámara básico derecho. Para cada fotograma, se calcula la salida de cada método adaptable para cada objeto detectado con cada conjunto de configuraciones y solo se considera el más anómalo. Cuando ha finalizado este proceso, se ha estudiado la distribución de las salidas para los objetos anómalos y los no anómalos en el video. La Figura 9.3 muestra un esquema de este proceso.

Tabla 9.2: Configuración óptima de parámetros considerada para cada modelo de localización de objetos anómalos.

Método	Parámetros
Dirichlet	Precisión, $P = 10^{1,4}$ (25,1189) Umbral = -52,653
K-medias	Número de grupos, $N = 12$ Umbral = -0,82
SOM	Número de neuronas, $NumNeurons = 169$ Umbral = -0,82

De acuerdo con las definiciones de cada método, el valor proporcionado como salida por ellos normalmente muestra que los objetos anómalos presentan un menor valor que los no anómalos. Con esta consideración, la

Figura 9.3: Esquema del procedimiento para calcular los valores óptimos de los parámetros.

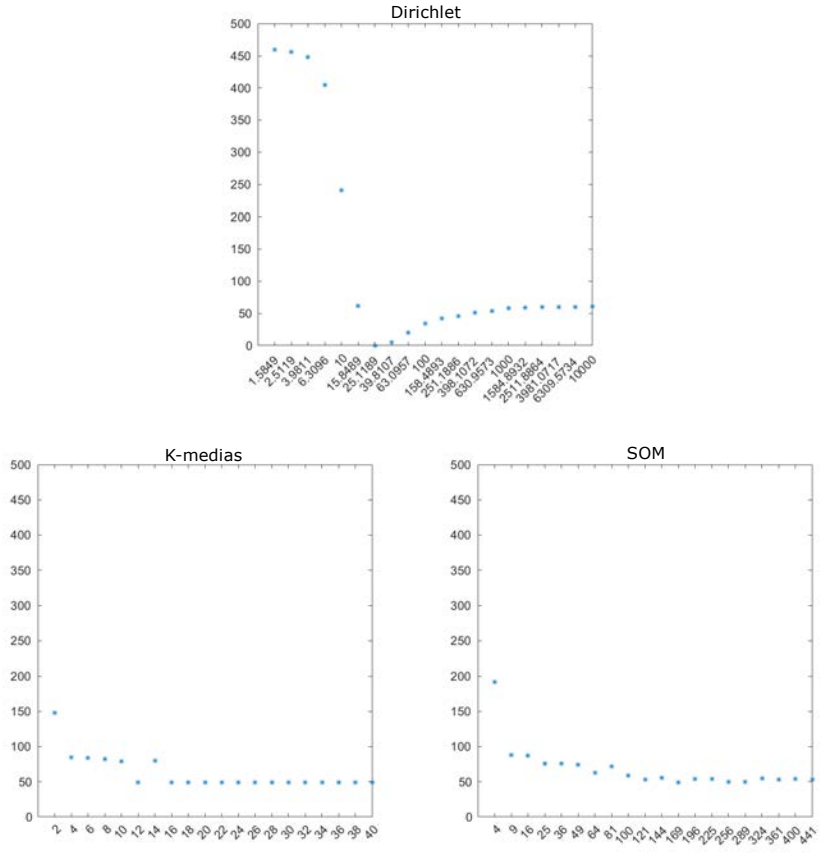


mejor configuración para un método es aquella que presenta una mayor heterogeneidad en las salidas. Un ejemplo de esto puede verse en la Figura 9.5. La primera imagen muestra una mayor heterogeneidad que las demás porque hay un menor número de objetos que pueden ser considerados como anómalos o no anómalos.

En este punto, se puede considerar un buen umbral aquellos valores entre la máxima salida para los objetos anómalos y la mínima salida para los objetos no anómalos. En este caso, se ha decidido que el mejor umbral es aquel que minimiza el número de aquellos objetos que son estimados como no anómalos pero que realmente lo son (falsos negativos). Es decir, se ha decidido que el umbral es el máximo valor de las salidas anómalas y, con este valor, no se tendrán falsos negativos. Por tanto, en la comparativa, el mejor modelo será aquel con el mínimo número de objetos no anómalos que son considerados como anómalos (falsos positivos). Los resultados de la comparativa de cada modelo con las configuraciones probadas pueden ser observados en la Figura 9.4. Para un mejor entendimiento y evitar problemas de solapamiento, se ha mostrado cada conjunto de objetos anómalos y no anómalos en un diferente nivel del eje y.

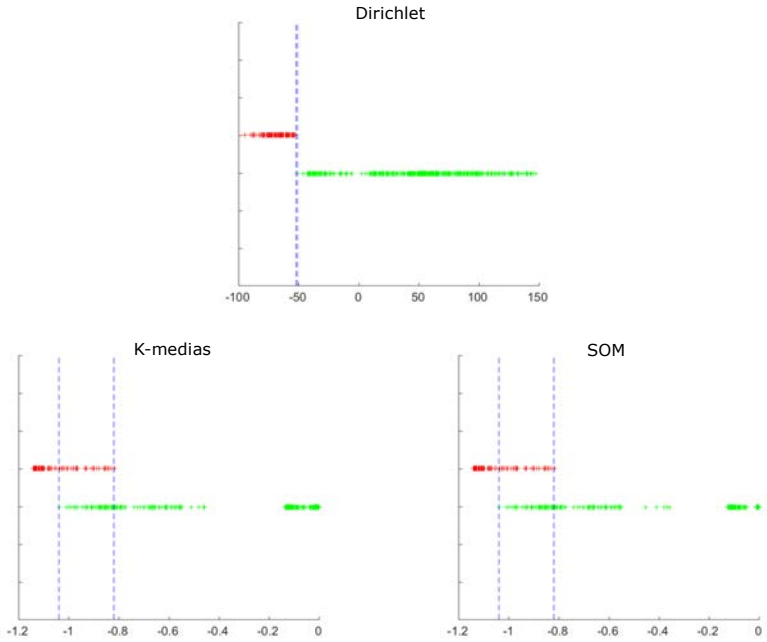
Como se muestra, los modelos K-medias y SOM siempre muestran un número similar de falsos positivos. Sin embargo, el modelo de Dirichlet ofrece varias configuraciones cuya salida es mejor que los otros modelos. Más aún, hay una configuración cuyo resultado es una clasificación perfecta de los objetos (primera imagen, datos correspondientes al valor 25,1189 en el eje x). La distribución de la salida de la mejor configuración probada de cada modelo puede observarse en la Figura 9.5. Como se ha mencionado anteriormente, se ha demostrado que la distribución de la salida en el modelo de Dirichlet es más heterogénea que en los otros métodos.

Figura 9.4: Número de falsos positivos (eje y) para cada configuración probada (eje x) de cada modelo de clasificación de anomalías. El eje x se corresponde con los parámetros precisión (Dirichlet), número de grupos (K-medias) y número de neuronas (SOM), respectivamente. Menor número de falsos positivos es mejor.



Por último, se ha utilizado la implementación de la mejor configuración de cada modelo en los siguientes experimentos. Los valores óptimos pueden verse en la Tabla 9.2.

Figura 9.5: Distribución de los valores proporcionados como salida por cada modelo de localización de objetos anómalos de acuerdo a la mejor configuración de la Figura 9.4. Es decir, Dirichlet con precisión = 25,1189, K-medias con 12 grupos y SOM con 169 neuronas. Para lograr una mejor comprensión y eliminar problemas de solapamiento, cada conjunto de objetos anómalos y no anómalos se muestra con un diferente nivel en el eje y; mientras que el eje x se corresponde con el valor de salida del modelo.



En este punto se pueden considerar dos estudios diferentes. El primero es una comparación entre el resultado producido por cada modelo de detección de anomalías considerando el mismo fotograma de entrada, es decir, el sistema emplea un controlador no adaptable para proporcionar siempre las mismas indicaciones a cada uno de los modelos de detección de anomalías. Y el segundo es una comparación del sistema implementando cada modelo utilizando el controlador de cámara adaptable que se ha propuesto, por lo que el primer fotograma del video será el mismo para todas las propuestas pero los siguientes fotogramas dependerán de las acciones que ellos hayan ejecutado con los fotogramas anteriores.

9.3.5. Comparación de modelos utilizando un controlador de cámara no adaptable

Para comparar la bondad de cada modelo detector de anomalías, se han procesado los mismos fotogramas con los modelos de anomalías Dirichlet,

K-medias y SOM. Primero se han procesado todas las secuencias testeadas considerando cada modelo adaptable de localización de objetos anómalos y el modelo de controlador de cámara básico derecho. Los videos son procesados fotograma a fotograma, obteniendo el resultado del modelo de detección y clasificación y proporcionando este valor a cada modelo de detección de anomalías. Debido a que el modelo de controlador de cámara básico derecho siempre proporciona los mismos comandos, todos los modelos adaptables de localización de objetos anómalos reciben el mismo fotograma como entrada y para cada fotograma se obtiene la salida de cada modelo de anomalía, que será el objeto más anómalo en ese fotograma. Por último, debido a que se saben las clases reales y predichas de los objetos, se puede establecer cuál de ellos son estimados correctamente como objetos anómalos y cuáles son incorrectamente predichos. La Figura 9.6 muestra un esquema de este procedimiento.

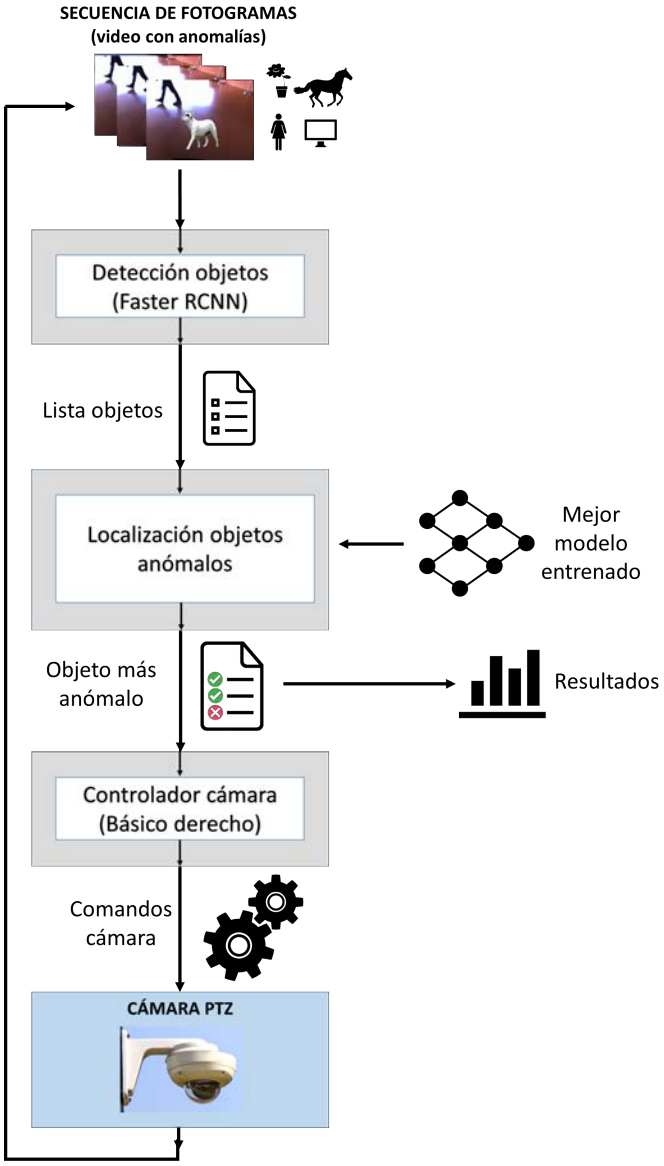
La Figura 9.7 muestra algunos resultados producidos por el sistema empleando el modelo de Dirichlet desde un punto de vista cualitativo. En las dos primeras filas y en el segundo fotograma de la tercera fila se presentan varios fotogramas con un solo objeto anómalo y varios no anómalos en cada escenario. El caso de más de un objeto anómalo apareciendo en la escena se presenta en la cuarta fila. En esta situación, el sistema elegirá aquel objeto que es señalado por el modelo de Dirichlet. Por otro lado, el caso de que no exista ningún objeto anómalo en el fotograma se muestra en el primer fotograma de la tercera fila.

Desde un punto de vista cuantitativo, con esta comparación entre los datos de la máscara de verdad y el resultado de la propuesta se puede proporcionar la bondad del método y el resto de modelos utilizando varias medidas. Dado un video, se sabe el número de fotogramas donde aparecen objetos no anómalos y sin embargo son considerados como anómalos por el modelo (falsos positivos o FP), el número de fotogramas donde los objetos anómalos son considerados como anómalos (verdaderos positivos o TP), el número de fotogramas donde los objetos anómalos son considerados como no anómalos (falsos negativos o FN), y el número de fotogramas donde los objetos no anómalos son considerados como no anómalos (verdaderos negativos o TN). De acuerdo con estas consideraciones, se ha seleccionado la exactitud como la principal medida porque ofrece una buena aproximación de su bondad y se puede calcular fácilmente. La exactitud (Accuracy o Acc) es un valor entre 0 y 1, donde mayor es mejor. La exactitud se puede definir de la siguiente manera:

$$Acc = \frac{TP + TN}{TP + FP + FN + TN} \tag{9.17}$$

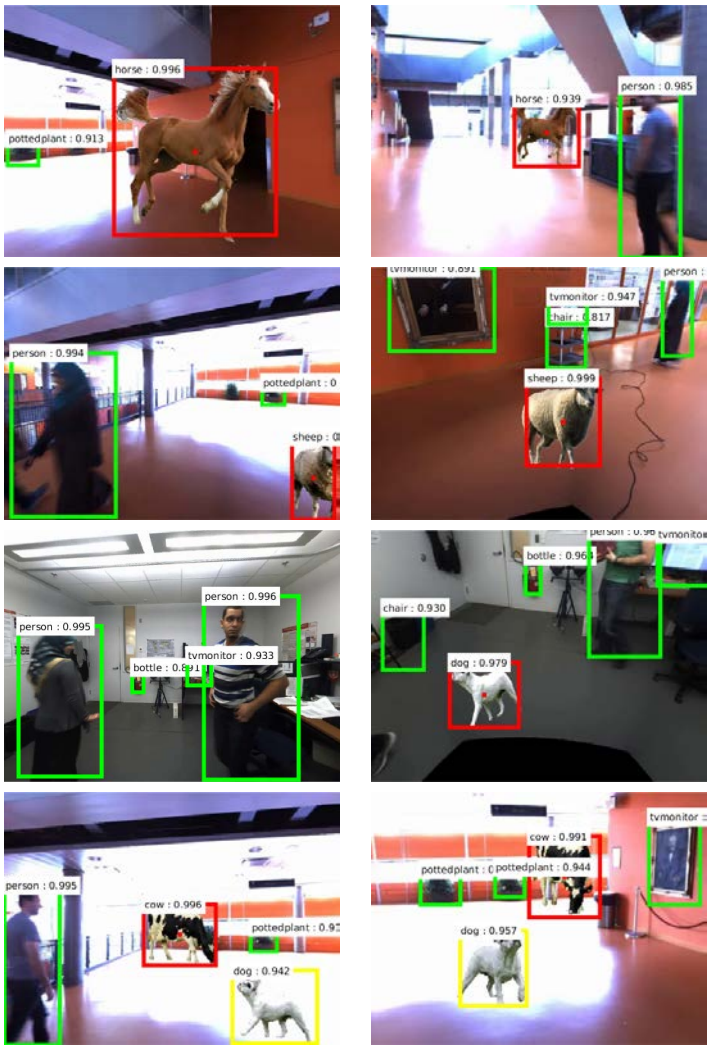
Por tanto, se puede comparar la exactitud de cada modelo de detección de anomalías y el resultado de esta comparativa puede observarse en la Tabla 9.3. Como se muestra, el modelo de Dirichlet logra el mejor rendimiento de

Figura 9.6: Esquema del procedimiento para comparar los diferentes modelos de detección de objetos anómalos.



todos los competidores considerados. Además, el modelo trabaja adecuadamente a pesar de que los datos de entrenamiento no tenían información sobre la secuencia scenario5; sin embargo, los resultados de los modelos restantes son peores.

Figura 9.7: Algunos fotogramas con los resultados de salida del módulo de detección y clasificación de objetos. Los objetos no anómalos detectados presentan una región mínima rectangular verde, el objeto más anómalo se muestra en rojo, y el resto de objetos anómalos, si existen, están coloreados en amarillo. Cada fila muestra dos fotogramas de un video probado. De arriba abajo (y de izquierda a derecha): scenario3horse_0 (608 y 875), scenario3sheep_0 (352 y 481), scenario5dog_270 (186 y 691) y scenario3dogcow_0 (188 y 329).



9.3.6. Comparativa de modelos utilizando un controlador de cámara adaptable

En este apartado se comparan todos los modelos de detección de anomalías considerados utilizando el controlador de cámara adaptable que se ha

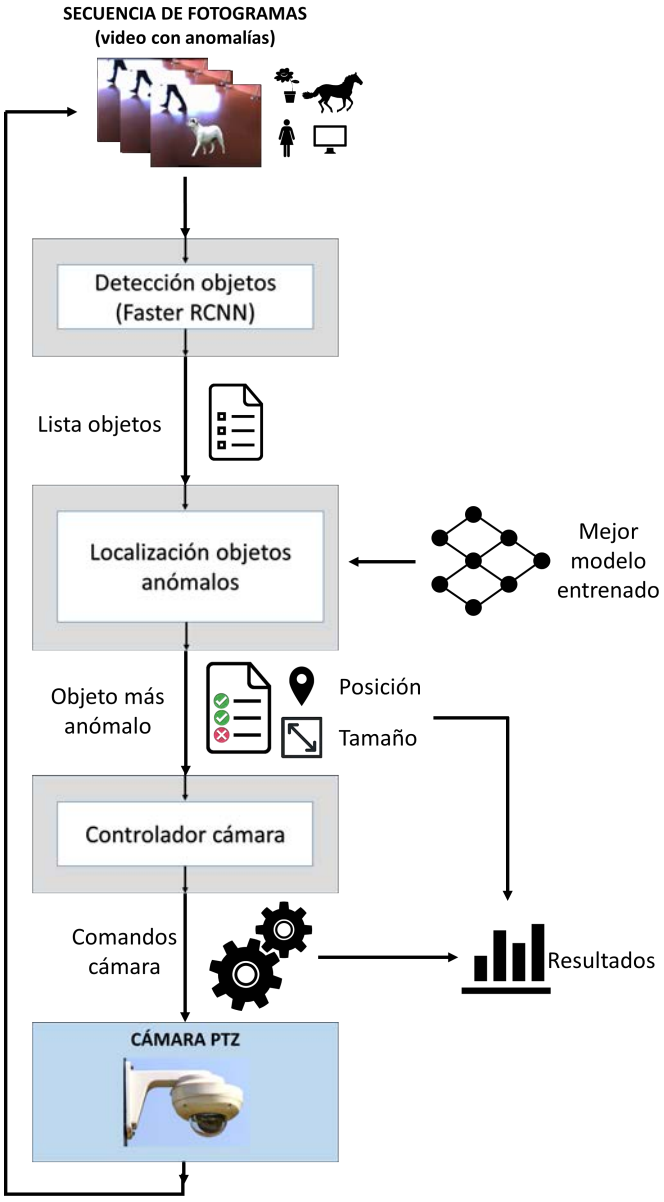
Tabla 9.3: Exactitud de cada modelo de detección de anomalías. Cada fila representa un video y la última fila muestra la exactitud media. La primera columna muestra el video testeado y las siguientes columnas presentan el rendimiento de exactitud de los modelos Dirichlet, K-medias y SOM, respectivamente. Los mejores resultados están resaltados en **negrita**.

Video	Dirichlet	k-medias	SOM
scenario3horse_0	**0,967**	0,768	0,857
scenario3horse_90	**0,992**	0,733	0,851
scenario3horse_180	**0,997**	0,757	0,846
scenario3horse_270	**0,997**	0,749	0,868
scenario3sheep_0	**0,989**	0,774	0,869
scenario3sheep_90	**0,998**	0,737	0,854
scenario3sheep_180	**0,996**	0,813	0,884
scenario3sheep_270	**0,980**	0,807	0,905
scenario5dog_0	**0,997**	0,310	0,369
scenario5dog_90	**0,998**	0,306	0,358
scenario5dog_180	**0,998**	0,316	0,364
scenario5dog_270	**0,998**	0,362	0,415
scenario3dogcow_0	**0,998**	0,800	0,918
scenario3dogcow_90	**0,996**	0,805	0,908
scenario3dogcow_180	**0,987**	0,745	0,862
scenario3dogcow_270	**0,983**	0,824	0,916
scenario3cow_0	**0,955**	0,761	0,869
scenario3cow_90	**0,962**	0,714	0,843
scenario3cow_180	**0,958**	0,732	0,835
scenario3cow_270	**0,973**	0,770	0,864
Media	**0,986**	0,647	0,736

propuesto. Por tanto, el primer fotograma del resultado producido por cada propuesta será el mismo para todos, pero los siguientes fotogramas dependerán del movimiento de la cámara indicado por cada modelo en cada instante. También se consideran las configuraciones que emplean el modelo de localización de objetos anómalos None y los modelos de controladores de cámara básicos. Nótese que la configuración compuesta por el modelo de localización de objetos anómalos None y el modelo de controlador de cámara propuesto no tiene sentido porque este controlador necesita información como entrada que ese modelo de anomalía no puede ofrecer (la posición y el tamaño de los objetos anómalos detectados considerados por el modelo de localización de objetos anómalos adaptable). La Figura 9.8 muestra la mecánica de esta comparativa.

En este caso, se ha considerado una medida que hemos denominado *Co-*

Figura 9.8: Esquema de la mecánica para comparar los diferentes modelos de detección de objetos anómalos utilizando el controlador de cámara adaptable propuesto.



bertura (*Coverage*) para proporcionar información sobre el número de fotogramas donde un objeto anómalo real (o varios) está presente en el fotograma, aunque el sistema lo haya detectado o no. Por tanto, esta medida muestra una idea sobre el rendimiento del sistema para localizar objetos anómalos. Esta medida es un valor entre 0 y 1, donde mayor es mejor. Puede

ser definida como sigue:

Sea N el número de fotogramas analizados y sea $\mathbf{o}_i$ una marca que representa si un objeto anómalo está presente en el fotograma i ($\mathbf{o}_i = 1$) o no ($\mathbf{o}_i = 0$):

$$Cobertura = \frac{1}{N}\sum_{i=1}^{N}\mathbf{o}_i \tag{9.18}$$

El rendimiento de los comandos proporcionados por el controlador también es calculado y se ha seleccionado la exactitud (Acc) y el error cuadrático medio (Mean Square Error o MSE) como medidas. En este caso, la exactitud se define como sigue:

Sea k el movimiento de la cámara observado (horizontal, vertical o zoom) de los K existentes movimientos de la cámara, sea $\mathbf{x}_k$ y $\mathbf{w}_k$ el movimiento de la cámara de la máscara de verdad y el movimiento sugerido por la propuesta, respectivamente, donde $\mathbf{x_k}, \mathbf{w_k} \in \{-1, 0, 1\}$, correspondiendo:

$-1 = \{\text{horizontal izquierda, vertical arriba, zoom hacia afuera}\}$

$0 = \{\text{horizontal quieto, vertical quieto, zoom quieto}\}$

$1 = \{\text{horizontal derecha, vertical abajo, zoom hacia adentro}\}$

para los movimientos horizontal, vertical y zoom, respectivamente. Además, sea $\mathbf{q}_k = 1$ si el modelo acierta el movimiento k de la cámara (por lo que $\mathbf{x_k} = \mathbf{w_k}$) y $\mathbf{q}_k = 0$ si el modelo falla ($\mathbf{x_k} \neq \mathbf{w_k}$):

$$Acc = \frac{1}{K}\sum_{k=1}^{K}\mathbf{q}_k \tag{9.19}$$

Por otro lado, el error cuadrático medio es un número positivo real, donde menor es mejor, y su definición es:

$$MSE = \frac{1}{K}\sum_{k=1}^{K}(\mathbf{x}_k - \mathbf{w}_k)^2 \tag{9.20}$$

Los resultados de la comparativa de cobertura pueden observarse en la Tabla 9.4. El modelo de Dirichlet obtiene el mejor rendimiento medio. Los modelos K-medias y SOM logran un rendimiento muy parecido entre ellos, pero menor comparándolo con Dirichlet. Por último, los movimiento básicos producen un resultado similar entre ellos, pero bastante lejos de los otros.

Como muestran los resultados, la posición inicial de la cámara y la posición de los objetos anómalos tienen un enorme impacto en los resultados de cada propuesta y video. Por ejemplo, los videos scenario3horse, scenario3sheep y scenario3dogcow muestran animales moviéndose alrededor del mismo área del escenario y los modelos k-means y SOM están observando este área porque consideran una planta como objeto anómalo y ambos

modelos están enfocándola, por lo que, cuando los animales aparecen, ambos modelos pueden cubrir muchos fotogramas con estos objetos anómalos; mientras tanto, la propuesta Dirichlet está buscando un objeto anómalo en el resto del escenario. Por tanto, el rendimiento del modelo de Dirichlet es peor que el de K-medias y SOM. Por otro lado, el video scenario3cow presenta animales en la otra parte del escenario, por lo que K-medias y SOM no encuentran estos animales porque están enfocando el área de la planta y esto produce malos resultados. Por último, la propuesta de Dirichlet funciona bien en otro escenario (scenario5dog) y obtiene un rendimiento similar, a pesar de que los datos de entrenamiento no tienen información sobre este escenario; sin embargo, K-medias y SOM ofrecen un rendimiento peor.

Tabla 9.4: Resultados de la cobertura de los métodos comparados. Cada fila representa un video y la última fila muestra la cobertura media. La primera columna muestra el video testeado y las siguientes columnas presentan el rendimiento de cobertura para Dirichlet, K-medias, SOM, (básico) derecho, (básico) izquierdo, (básico) aleatorio y (básico) estático, respectivamente. Los mejores resultados están resaltados en **negrita**.

Video	Dirichlet	K-medias	SOM	Derecho	Izquierdo	Aleatorio	Estático
scenario3horse_0	**0,321**	**0,321**	**0,321**	0,124	0,081	0,113	0,111
scenario3horse_90	0,284	**0,321**	**0,321**	0,134	0,076	0,000	0,000
scenario3horse_180	0,214	**0,321**	**0,321**	0,084	0,101	0,154	0,100
scenario3horse_270	0,137	**0,321**	**0,321**	0,069	0,121	0,116	0,150
scenario3sheep_0	0,503	**0,576**	**0,576**	0,112	0,161	0,212	0,219
scenario3sheep_90	0,456	**0,576**	**0,576**	0,156	0,169	0,068	0,056
scenario3sheep_180	**0,576**	**0,576**	**0,576**	0,238	0,155	0,242	0,216
scenario3sheep_270	0,558	**0,576**	**0,576**	0,194	0,161	0,121	0,142
scenario5dog_0	**0,412**	0,254	0,182	0,067	0,063	0,082	0,100
scenario5dog_90	**0,402**	0,148	0,182	0,051	0,073	0,016	0,029
scenario5dog_180	**0,386**	0,148	0,148	0,078	0,049	0,116	0,095
scenario5dog_270	**0,447**	0,122	0,208	0,080	0,068	0,082	0,090
scenario3dogcow_0	0,498	**0,667**	0,523	0,193	0,305	0,576	0,593
scenario3dogcow_90	0,449	**0,523**	**0,523**	0,146	0,248	0,068	0,189
scenario3dogcow_180	**0,540**	0,523	0,523	0,182	0,102	0,000	0,085
scenario3dogcow_270	0,479	**0,523**	**0,523**	0,275	0,180	0,000	0,000
scenario3cow_0	**0,447**	0,000	0,000	0,187	0,221	0,000	0,000
scenario3cow_90	**0,444**	0,000	0,000	0,221	0,136	0,000	0,000
scenario3cow_180	0,414	0,000	0,000	0,191	0,188	0,548	**0,574**
scenario3cow_270	**0,388**	0,176	0,000	0,194	0,243	0,121	0,293
Media	**0,418**	0,334	0,320	0,149	0,145	0,132	0,152

Además, se ha calculado el rendimiento de los comandos proporcionados a la cámara. Estos resultados se muestran en las Tablas 9.5 y 9.6 y muestran el rendimiento logrado considerando solo aquellos fotogramas donde está presente un objeto anómalo real en el fotograma y el módulo considera que existe un objeto anómalo en él. Esta consideración es para no adulterar

los resultados mostrados porque algunos problemas en la detección y clasificación de los objetos que pertenecen al fotograma producen una detección errónea y, por tanto, la salida del módulo de detección de anomalías no será la deseada. Como se ha mostrado, el modelo de Dirichlet logra la mejor exactitud y el segundo mejor error cuadrático medio. Ambos resultados están influenciados por la secuencia scenario3cow: los módulos K-medias y SOM no muestran ningún fotograma donde un objeto anómalo está en él, por lo que la exactitud y el error cuadrático medio para este video es 0 en ambas medidas, pero en el caso del error cuadrático medio este valor es el mejor rendimiento posible, por lo que la medida media para estas dos propuestas ha sido beneficiada.

Tabla 9.5: Resultados de exactitud de los comandos proporcionados a la cámara para los métodos comparadores solo considerando aquellos fotogramas en los que un objeto anómalo real está presente en el fotograma y el módulo considera que existe un objeto anómalo. Cada fila representa un video y la última fila muestra la exactitud media. La primera columna muestra el video testeado y las restantes columnas presentan el rendimiento de la exactitud para Dirichlet, K-medias, SOM, (básico) derecho, (básico) izquierdo, (básico) aleatorio y (básico) estático, respectivamente. Los mejores resultados están resaltados en **negrita**.

Video	Dirichlet	K-medias	SOM	Derecho	Izquierdo	Aleatorio	Estático
scenario3horse-0	**0,855**	**0,855**	0,849	0,527	0,554	0,550	0,499
scenario3horse-90	0,836	**0,849**	**0,849**	0,462	0,596	0,000	0,000
scenario3horse-180	**0,876**	0,849	0,849	0,532	0,586	0,528	0,428
scenario3horse-270	0,849	**0,864**	0,849	0,708	0,544	0,629	0,465
scenario3sheep-0	0,922	**0,947**	**0,947**	0,387	0,376	0,321	0,382
scenario3sheep-90	0,936	**0,947**	**0,947**	0,440	0,415	0,403	0,407
scenario3sheep-180	**0,949**	0,947	0,947	0,369	0,465	0,392	0,371
scenario3sheep-270	0,951	**0,952**	0,947	0,437	0,453	0,453	0,391
scenario5dog-0	0,891	0,852	**0,969**	0,284	0,258	0,218	0,333
scenario5dog-90	0,888	0,759	**0,969**	0,259	0,236	0,356	0,478
scenario5dog-180	**0,884**	0,759	0,754	0,242	0,281	0,131	0,080
scenario5dog-270	**0,893**	0,844	0,881	0,254	0,272	0,218	0,317
scenario3dogcow-0	**0,947**	0,922	0,905	0,435	0,451	0,431	0,360
scenario3dogcow-90	**0,925**	0,905	0,905	0,524	0,382	0,435	0,447
scenario3dogcow-180	**0,958**	0,905	0,905	0,372	0,475	0,000	0,333
scenario3dogcow-270	0,912	**0,913**	0,905	0,462	0,463	0,000	0,000
scenario3cow-0	**0,917**	0,000	0,000	0,458	0,447	0,000	0,000
scenario3cow-90	**0,897**	0,000	0,000	0,517	0,475	0,000	0,000
scenario3cow-180	**0,896**	0,000	0,000	0,498	0,558	0,417	0,355
scenario3cow-270	0,883	**0,900**	0,000	0,569	0,550	0,661	0,422
Media	**0,903**	0,749	0,719	0,437	0,442	0,307	0,303

También se han estudiado las matrices de confusión de la propuesta Di-

Tabla 9.6: Resultados del error cuadrático medio de los comandos proporcionados a la cámara para los métodos comparadores solo considerando aquellos fotogramas en los que un objeto anómalo real está presente en el fotograma y el módulo considera que existe un objeto anómalo. Cada fila representa un video y la última fila muestra el error cuadrático medio en media. La primera columna muestra el video testeado y las restantes columnas presentan el rendimiento de la exactitud para Dirichlet, K-medias, SOM, (básico) derecho, (básico) izquierdo, (básico) aleatorio y (básico) estático, respectivamente. Los mejores resultados están resaltados en **negrita**.

Video	Dirichlet	K-medias	SOM	Derecho	Izquierdo	Aleatorio	Estático
scenario3horse-0	**0,145**	0,148	0,151	1,000	1,004	0,633	0,501
scenario3horse-90	0,171	0,151	0,151	1,002	0,779	**0,000**	**0,000**
scenario3horse-180	**0,124**	0,151	0,151	0,760	0,732	0,748	0,572
scenario3horse-270	0,158	**0,136**	0,151	0,717	0,870	0,631	0,535
scenario3sheep-0	0,078	**0,053**	**0,053**	0,994	1,024	0,911	0,618
scenario3sheep-90	0,064	**0,053**	**0,053**	0,875	0,993	1,097	0,593
scenario3sheep-180	**0,051**	0,053	0,053	1,456	0,943	0,928	0,629
scenario3sheep-270	0,049	**0,048**	0,053	0,836	0,971	0,875	0,609
scenario5dog-0	0,109	0,208	**0,031**	1,011	1,142	0,982	0,667
scenario5dog-90	0,112	0,305	**0,031**	1,050	1,100	0,678	0,522
scenario5dog-180	**0,116**	0,305	0,310	1,086	0,978	1,042	0,920
scenario5dog-270	**0,107**	0,371	0,209	1,018	1,220	0,982	0,683
scenario3dogcow-0	**0,053**	0,086	0,110	0,908	0,977	0,836	0,640
scenario3dogcow-90	**0,075**	0,110	0,110	0,723	1,153	0,759	0,553
scenario3dogcow-180	0,042	0,110	0,110	1,170	0,877	**0,000**	0,667
scenario3dogcow-270	0,088	0,101	0,110	0,959	0,905	**0,000**	**0,000**
scenario3cow-0	0,083	**0,000**	**0,000**	1,007	1,108	**0,000**	**0,000**
scenario3cow-90	0,103	**0,000**	**0,000**	0,868	0,928	**0,000**	**0,000**
scenario3cow-180	0,104	**0,000**	**0,000**	0,893	0,809	0,914	0,645
scenario3cow-270	0,117	0,100	**0,000**	0,924	0,824	0,698	0,578
Media	0,098	0,124	**0,092**	0,963	0,967	0,636	0,497

richlet para examinar la comparativa de la exactitud de los comandos proporcionados a la cámara en más detalle. Esta información puede verse en la Figura 9.9. En general, los resultados presentados en las matrices de confusión son satisfactorios. Se puede observar cómo el movimiento estimado tiene un buen rendimiento para los movimientos horizontal y vertical, no así para el movimiento zoom. Esto puede observarse en la diagonal principal de cada matriz: cuanto mayores los valores de la exactitud y más blancos los cuadros de los elementos de la matriz mejor el rendimiento de cada movimiento. Además, los elementos finales de la antidiagonal presetan un valor prácticamente cero, es decir, el movimiento estimado no es el movimiento opuesto del deseado.

Además, se han estudiado aquellos casos de las matrices de confusión

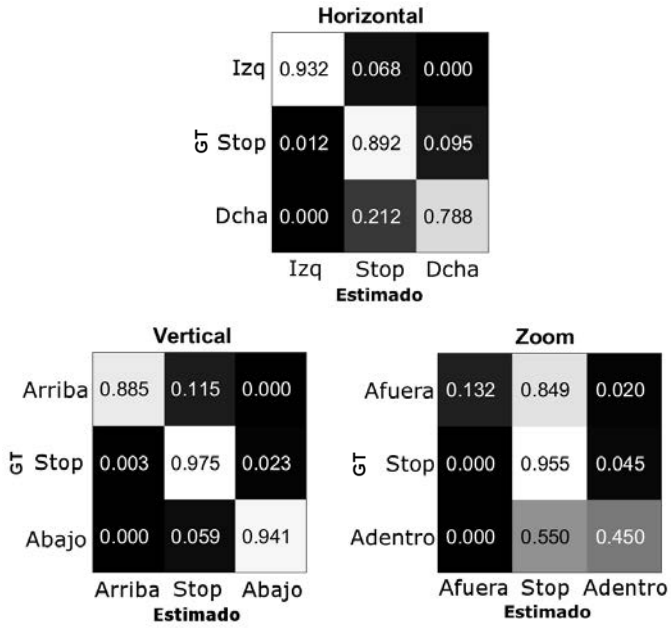
que muestran peores valores. El problema más grave que muestran es el movimiento zoom. Como se puede ver en la última fila de la Figura 9.9, el error en el movimiento zoom estimado es alto cuando la cámara debería moverse para seguir el objeto anómalo correctamente. Se han analizado los fotogramas que producen los valores obtenidos de la parte inferior derecha de la matriz para descubrir las razones de estos valores. La peor exactitud es el caso cuando $GT = Out$ y $Estimated = Stop$, y esto es producido porque los diferentes videos obtenidos de la secuencia scenario3horse están compuestos por bastantes fotogramas donde el caballo es detectado por el sistema con un tamaño cuya relación en píxeles respecto del número de píxeles del fotograma está entre los parámetros ϕ_z y Φ_z, pero muy próximos al valor Φ_z.

Por tanto, el movimiento zoom estimado es estar parado. Sin embargo, la máscara de verdad indica que el tamaño del caballo en esos fotogramas es mayor que el estimado y la proporción en relación al número total de píxeles del fotograma es un poco superior a Φ_z, es decir, la máscara de verdad indica ejecutar un movimiento de zoom hacia afuera. Esto se puede observar en la primera fila de la Figura 9.10. Por otro lado, el caso cuando $GT = In$ y $Estimated = Stop$ es análogo, porque los objetos detectados tienen un tamaño con una proporción mayor que ϕ_z, pero el tamaño de los objetos de acuerdo a la máscara de verdad es menor que ϕ_z, es decir, el movimiento deseado es hacer un zoom hacia adentro pero el estimado es permanecer quieto. La segunda fila de la Figura 9.10 muestra este problema. Por último, los casos horizontal y vertical son similares: la posición del centroide del objeto anómalo de acuerdo a la máscara de verdad y el centroide estimado están bastante cerca de ϕ_h, Φ_h, ϕ_v o Φ_v, respectivamente, donde uno de estos parámetros está entre ambos centroides. Por tanto, la máscara de verdad indica ejecutar un movimiento y la estimada es permanecer quieto, o viceversa. Este efecto se muestra en la tercera fila de la Figura 9.10. Sin embargo, el movimiento estimado no es el movimiento opuesto al deseado, por lo que es importante que esto se destaque.

9.4. Discusión

La detección de anomalías es un problema muy amplio, que puede ser tratado desde muchas perspectivas. Las propuestas clásicas están basadas en modelos artesanales de alto nivel del comportamiento normal, es decir, algo que no se ajuste en el modelo es considerado como anómalo. Esta es una propuesta frágil, ya que asume que el modelador humano está disponible para tener en cuenta todas las posibilidades que surgen en las situaciones normales de un escenario dado. Aquí se propone una vista no supervisada del problema, donde el modelo del comportamiento normal no es proporcionado por un humano, sino que es aprendido automáticamente. Esto es, el sistema puede ser desplegado en una amplia variedad de escenarios con reducida

Figura 9.9: Resultados de las matrices de confusión para cada tipo de movimiento del modelo Dirichlet, considerando solo aquellos fotogramas en los que un objeto anómalo real está presente en el fotograma y el módulo considera que existe un objeto anómalo ($GT = Y$ y $Approach = Y$). Cuanto más claros los cuadros de la diagonal principal (y más oscuros los cuadros restantes) mejor la exactitud del movimiento.



intervención humana.

Una elección de diseño importante al modelar anomalías es el tipo de características con respecto a las cuales se detectan las anomalías. En este capítulo se emplean las redes neuronales convolucionales para proporcionar las probabilidades de pertenencia a un preespecificado conjunto de clases de objeto con una alta invariancia con respecto a la postura de la cámara. Por tanto, las anomalías son detectadas con respecto a las clases de objeto. Una posible extensión es aumentar el número de rasgos que son empleados para aprender el modelo de actividad normal.

Una dificultad específica de la detección de anomalías con control de cámara PTZ es la generación de la máscara de verdad. Es muy difícil establecer los movimientos de la cámara de la máscara de verdad porque no hay una única mejor secuencia de movimientos. Esto es porque los objetos anómalos que aparecen en el campo de visión dependen de los movimientos previos de la cámara, que a veces son ejecutados sin objetos anómalos a la vista, es decir, buscando nuevos objetos anómalos. Algunos escenarios difíciles pueden surgir. Uno de ellos es un fotograma que presenta claramente dos o más ob-

Figura 9.10: Algunos fotogramas que muestran el error producido en los comandos de los movimientos de la cámara. Se muestra la región mínima rectangular y el centroide del objetivo detectado por el sistema propuesto (en rojo) y la máscara de verdad (en azul). La primera fila muestra los fotogramas 620 y 674 del video scenario3horse_0, respectivamente. La segunda fila muestra el fotograma 790 del video scenario3sheep_180 y el fotograma 1380 del video scenario5dog_90, respectivamente. La tercera fila muestra el fotograma 1470 de scenario5dog_0 y el fotograma 845 de scenario3sheep_0, respectivamente.

jetos anómalos a ser seguidos. Entonces es difícil decidir cuál de ellos seguir. Otro escenario complicado involucra cuando se sigue un objeto anómalo y se aplica zoom hacia adentro. El zoom que puede ser aplicado es complejo de determinar, porque un zoom mayor significa que se logra una confianza de detección de objetos mejorada, mientras que un zoom menor permite que otras anomalías puedan ser detectadas en el campo de visión más amplio. En este trabajo se han diseñado los experimentos con estas dificultades en mente, al tiempo que proporciona medidas de rendimiento cuantitativas y cualitativas para las comparaciones.

Los resultados experimentales obtenidos demuestran el rendimiento de

nuestro sistema propuesto, tanto por módulos como en conjunto. El diseño experimental se ha ejecutado para evaluar separadamente los módulos individuales y el sistema en general.

9.5. Conclusiones

Se ha diseñado un sistema de videovigilancia automático basado en una cámara PTZ. Detecta objetos anómalos con respecto al tipo de objeto. Las detecciones de objetos en bruto son proporcionadas por una red neuronal convolucional. El sistema no necesita datos etiquetados manualmente, ya que un modelo probabilístico no paramétrico de los tipos de objetos normalmente encontrados en la escena se aprende de manera no supervisada. Se ha propuesto para este propósito un estimador de densidad del núcleo basado en la distribución de Dirichlet. El modelo probabilístico logra una estimación de la probabilidad de que un objeto sea anómalo. Esta información es proporcionada a un módulo de control de la cámara, que sigue y se centra en el objeto más anómalo en el actual campo de visión.

La validación de la propuesta ha sido demostrada de varias maneras. El modelo de detección de anomalías propuesto ha sido evaluado usando tanto un controlador de cámara no adaptable como uno adaptable, para evaluar el rendimiento del módulo de detección de anomalías no considerando el control de la cámara. Además, el rendimiento de la detección de anomalías del algoritmo controlador de la cámara propuesto ha sido comparado con otros controladores no adaptables, es decir, la idoneidad del módulo controlador de cámara es independientemente medido. Los resultados de estos tests muestran que el sistema supera claramente las alternativas no adaptables.

Capítulo 10

Controlador neuronal para cámaras PTZ basado en detección de primer plano no panorámica

La economía de tiempo humano es la ventaja de las máquinas industriales.

Charles Babbage

Resumen: En este capítulo se presenta un controlador para cámaras PTZ basado en un modelo de red neuronal no supervisada. Tiene la ventaja de que la máscara de primer plano es generada por un subsistema de detección de primer plano no panorámico. Por tanto, el objetivo es optimizar los movimientos de la cámara PTZ para lograr la máxima cobertura de la escena observada en presencia de objetos en movimiento. Se aplica un gas neuronal creciente (growing neural gas o GNG) para mejorar la representación de los objetos en primer plano. Se ofrecen resultados cualitativos y cuantitativos usando varios conjuntos de datos, que demuestran la idoneidad de la propuesta.

10.1. Introducción

La mayoría de los sistemas de videovigilancia fueron construidos con una sola cámara estacionaria durante muchos años. Sin embargo, actualmente es posible encontrar diferentes tipos de cámaras y cualquier sistema de videovigilancia está compuesto frecuentemente por múltiples dispositivos que

intentan cubrir la mayor área posible (Xu y Song, 2010). Entre otros tipos o criterios, dos de los tipos de cámara más utilizados son la omnidireccional y la PTZ (pan-tilt-zoom o de movimiento horizontal, vertical y cambio de la distancia focal (zoom)). Los sistemas de videovigilancia convencionales normalmente constan de al menos una cámara omnidireccional y una PTZ.

Los sistemas de videovigilancia son capaces de supervisar toda la secuencia utilizando una sola cámara omnidireccional pero, debido a las limitada resolución de este tipo de cámaras panorámicas, la información en detalle de los objetos podría no ser recogida. Como resultado de ello, una cámara PTZ se usa para esas tareas, que requieren vistas de primer plano de alta resolución.

Las cámaras PTZ son adecuadas para la identificación y reconocimiento de objetos en escenas de campo lejano. Sin embargo, el uso práctico de las cámaras PTZ en escenarios del mundo real es complicado debido a varias razones (Lisanti et al., 2016). Se necesita una calibración en línea y continua de la cámara ya que los valores absolutos posicionales de las componentes horizontal, vertical y de zoom proporcionados por los activadores de la cámara no están sincronizados con el flujo de video en la mayoría de los casos. Además, alguna representación adaptativa del fondo se hace necesaria para hacer seguimiento del objetivo, ya que el fondo de la escena está cambiando continuamente debido al funcionamiento de la cámara (Sajid et al., 2016).

Los sistemas convencionales de cámara son normalmente fácilmente personalizables y permiten a los usuarios desplegar las infraestructuras de detección de acuerdo a sus necesidades, ajustando parámetros de la cámara como el campo de visión, modo de funcionamiento (visión nocturna/diurna, interior/exterior), la resolución (Konda et al., 2016). En la mayoría de los casos el diseño de la infraestructura de un sistema de videovigilancia está realizado manualmente, aunque el amplio rango de configuraciones y configuración de parámetros conduce a soluciones subóptimas, que implican una cobertura incompleta del área controlada, o, a la inversa, costes de despliegue más altos para lograr un resultado satisfactorio (Hörster y Lienhart, 2006). Como regla general, se puede asumir que el objetivo de un coordinador de cámara es garantizar la cobertura máxima del espacio observado, minimizando oclusiones y obteniendo la mejor visibilidad de los objetos de interés (Gil Whoan Chu y Myung Jin Chung, 2000).

En este capítulo se propone un método para cámaras PTZ basado en los modelos de gas neuronal creciente (Growing Neural Gas o GNG) para determinar automáticamente la posición y la configuración de las componentes horizontal, vertical y de zoom para optimizar la cobertura del primer plano.

Los algoritmos tradicionales de primer plano identifican los píxeles de primer plano porque sus características son diferentes de aquellas del fondo, pero esto origina falsas detecciones en las cámaras con movimiento. Además de otras propuestas basadas en la construcción de un modelo panorámico

de la escena, aquí se pone atención en los métodos no panorámicos (Kim et al., 2013; López-Rubio y López-Rubio, 2015) que son adecuados para las cámaras de libre movimiento. El uso del método neuronal filtra el ruido y los objetos espurios obtenidos en la máscara de primer plano. Además, los objetos en movimiento en la escena son representados con mayor precisión y robustez. Estas son los elementos clave para diseñar un controlador PTZ efectivo.

Los modelos GNG son un tipo de redes neuronales autoorganizadas y uno de los ejemplos de más éxito de aprendizaje no supervisado en un grafo. El modelo GNG original fue propuesto por Fritzke (Fritzke, 1995) y se ha convertido en un estándar para las aplicaciones de visión por computador (Palomo y López-Rubio, 2017) y robótica (Yanik et al., 2014), además de otros modelos autoorganizados para la detección de primer plano (Luque-Baena et al., 2015a) o segumiento de objetos (Luque-Baena et al., 2013) en secuencias de video.

10.2. Arquitectura del sistema

En esta sección se describe la arquitectura del sistema de control de la cámara PTZ que se ha propuesto. Este sistema está formado por tres módulos: un procedimiento de detección de primer plano (Subsección 10.2.1), un modelo de aprendizaje no supervisado para aprender la distribución de los objetos de primer plano (Subsección 10.2.2), y un módulo de control para mover la cámara (Subsección 10.2.3).

10.2.1. Detección de primer plano

La primera tarea que debe ser completada es la detección de los píxeles que pertenecen al primer plano. Esto significa que una máscara binaria debe ser calculada para cada fotograma del video entrante, es decir, los píxeles de primer plano son marcados como verdaderos en esa máscara. Para hacer esto, se ha utilizado un modelo de fondo para cámaras en movimiento (López-Rubio y López-Rubio, 2015). En cada instante de tiempo t, la salida del algoritmo es una marca binaria $f_{i,j} \in \{true, false\}$ para cada píxel, donde (i, j) son las coordenadas del píxel. Para esto se construye un conjunto de entrenamiento, que está compuesto de las coordenadas de todos los píxeles de primer plano:

$$\mathcal{S}_t = \{(i, j) \mid f_{i,j} = true\} \subset \mathbb{R}^2 \tag{10.1}$$

Este conjunto de entrenamiento es proporcionado a la versión modificada del modelo neuronal GNG, como se especifica a continuación.

10.2.2. Modelo neuronal

Para localizar los objetos de primer plano más relevantes en la escena, se propone entrenar un GNG (Fritzke, 1995) en línea por episodios, es decir, cada episodio es un fotograma del video entrante. El GNG presenta un número variable de neuronas H, que son añadidas y eliminadas de la red a medida que se ejecuta el proceso de aprendizaje. Las neuronas están conectadas mediante enlances unidireccionales, es decir, el grafo resultante podría tener varias componentes conectadas. Se modifica el GNG para procesar los datos de entrada entrantes mediante episodios, es decir, en el episodio t se presenta un nuevo conjunto de entrenamiento $\mathcal{S}_t$, que está formado de vectores D-dimensionales de valores reales, $\mathcal{S}_t \subset \mathbb{R}^D$. Como se ve en la Subsección 10.2.1, para la aplicación propuesta $D = 2$. Cada neurona $i \in \{1, ..., H\}$ tiene un centroide asociado $\mathbf{w}_i \in \mathbb{R}^D$, una edad (que es un número natural) y una variable de error $e_i \in \mathbb{R}$, $e_i \geq 0$. Cada conexión también tiene una edad. Se denota por A el conjunto de todas las conexiones, $A \subseteq \{1, ..., H\} \times \{1, ..., H\}$.

El algoritmo de aprendizaje viene dado por los siguientes pasos:

1. En el episodio inicial $t = 0$ empieza con dos neuronas ($H = 2$) unidas por una conexión. Cada prototipo es inicializado a una muestra tomada aleatoriamente de $\mathcal{S}_0$. Las variables de error son inicializadas a cero. La edad de la conexión y de las neuronas también son inicializadas a cero.

2. Se toma una muestra de entrenamiento $\mathbf{x} \in \mathbb{R}^D$ aleatoriamente de $\mathcal{S}_t$.

3. Encontrar la neurona más cercana q y la segunda neurona más cercana s en términos de la distancia euclídea:

$$q = \arg \min_{i \in \{1,...,H\}} \|\mathbf{w}_i - \mathbf{x}\| \tag{10.2}$$

$$s = \arg \min_{i \in \{1,...,H\} - \{q\}} \|\mathbf{w}_i - \mathbf{x}\| \tag{10.3}$$

4. Incrementar la edad de todos las conexiones salientes de q.

5. Añadir la distancia euclídea cuadrada entre $\mathbf{x}$ y la neurona más cercana q a la variable de error e_q:

$$\Delta e_q = \|\mathbf{w}_q - \mathbf{x}\|^2 \tag{10.4}$$

6. Actualizar q y todos sus vecinos topológicos directos con tamaño de paso ϵ_b para la neurona q y ϵ_n para sus vecinos, donde $\epsilon_b > \epsilon_n$:

$$\epsilon(i) = \begin{cases} \epsilon_b & \text{iff } i = q \\ \epsilon_n & \text{iff } (i \neq q) \wedge (i,q) \in A \\ 0 & \text{iff } (i \neq q) \wedge (i,q) \notin A \end{cases} \tag{10.5}$$

$$\Delta \mathbf{w}_i = \epsilon(i)(\mathbf{x} - \mathbf{w}_i) \tag{10.6}$$

7. Si q y s están conectadas, entonces ajustar la edad de ese eje a cero. En otro caso, crearlo.

8. Ajustar la edad de q y s a cero, e incrementar la edad de todas las demás neuronas.

9. Eliminar los ejes con una edad superior a a_{max}. Después eliminar todas las neuronas que no tienen conexiones, y aquellas neuronas cuya edad es superior a a_{max}.

10. Si λ muestras han sido procesadas desde la creación de la última neurona y el actual número de neuronas H es menor que el máximo H_{max}, entonces insertar una nueva neurona como sigue. Primero se determina la neurona r con el error máximo y la neurona z con el error más grande entre todos los vecinos directos de r. Después crear una nueva neurona k, insertar ejes conectando k con r y s, y eliminar el eje original entre r y z. Tras esto, decrementar las variables de error e_r y e_z multiplicándolas por una constante α, e inicializar la variable de error e_k al nuevo valor de e_r. Por último, configurar el prototipo de k para que esté a medio camino entre r and z, como sigue:

$$\mathbf{w}_k = \frac{1}{2}(\mathbf{w}_r + \mathbf{w}_z) \tag{10.7}$$

11. Decrementar todas las variables de error e_i multiplicándolas por una constante d.

12. Eliminar todas las neuronas que no han ganado durante los últimos N_k pasos, y sus conexiones.

13. Si se ha alcanzado el número máximo de muestras a ser procesadas para el episodio actual, entonces ir al paso 13. En otro caso, ir al paso 2.

14. Si se ha procesado el último episodio, entonces parar. En otro caso, incrementar el contador de episodios t, cargar el siguiente episodio e ir al paso 2.

10.2.3. Control de cámara

El último paso del sistema es el módulo de control de cámara. En cada instante de tiempo t, se calcula el conjunto de componentes conectadas del grafo directo asociado al GNG. Del conjunto de todas las conexiones $A \subseteq \{1, ..., H\} \times \{1, ..., H\}$, el conjunto de componentes conectadas $\mathcal{S}_t \in 2^{\{1,...,H\}}$ es una partición del conjunto de todas las neuronas $\{1, ..., H\}$, es decir, cada conjunto $S_{i,t} \in \mathcal{S}_t$ está formado por neuronas que están conectadas por una cadena de conexiones en A. Estas componentes conectadas están asociadas a los objetos de primer plano que aparecen en la escena en el tiempo t. De ellas se toma la más grande, es decir, la componente conectada asociada al objeto de primer plano más grande. Después se calcula el centroide de esa componente:

$$\boldsymbol{\mu}_t = \frac{1}{\left|\hat{S}_t\right|} \sum_{i \in \hat{S}_t} \mathbf{w}_i \tag{10.8}$$

donde $\hat{S}_t$ se refiere a la componente conectada más grande de $\mathcal{S}_t$. Por último, la cámara se mueve hacia el centroide $\boldsymbol{\mu}_t$.

Además, se ha asumido que el tamaño de la componente conectada es calculada como un círculo, considerando el centroide como el centro del círculo y su radio como la distancia media de cada neurona al centroide:

$$\boldsymbol{r}_t = \frac{1}{\left|\hat{S}_t\right|} \sum_{i \in \hat{S}_t} \|\mathbf{w_i} - \boldsymbol{\mu_t}\| \tag{10.9}$$

Por tanto, el tamaño de la componente conectada es estimado como sigue:

$$\boldsymbol{\omega}_t = \boldsymbol{\pi} \boldsymbol{r}_t^2 \tag{10.10}$$

El módulo de control de cámara, dado el centroide y el tamaño de la componente conectada más grande, envía a la cámara los comandos que debe ejecutar para seguir los objetos de primer plano rastreados. Los comandos que el modulo puede mandar a la cámara son los movimientos horizontal, vertical y de zoom. Con el movimiento horizontal la cámara se mueve hacia la izquierda o la derecha; la cámara puede moverse hacia arriba o abajo con el movimiento vertical; y aplicar zoom hacia adentro y afuera. La cantidad de movimientos de cada tipo están cuantificados, es decir, hay un tamaño mínimo de movimiento ρ_δ para cada tipo de movimiento $\delta \in \{pan, tilt, zoom\}$. Además, se tienen que evitar movimientos prohibidos. Por ejemplo, la cámara no puede aplicar zoom indefinidamente porque tiene definidos unos

valores máximo y mínimo. Por otro lado, el no aplicar movimiento también está considerado para cada tipo de movimiento. Además, la cámara hará un movimiento por cada tipo de movimiento en el instante de tiempo t del video. Por tanto, si se representa la posición horizontal, vertical y de zoom de la cámara en ese fotograma como $(\boldsymbol{\alpha}_t, \boldsymbol{\beta}_t, \boldsymbol{\gamma}_t)$ mediante coordenadas esféricas, en el $(t+1)$-ésimo fotograma la posición de la cámara será:

$$(\boldsymbol{\sigma}_{t+1}, \boldsymbol{\beta}_{t+1}, \boldsymbol{\gamma}_{t+1}) = (\boldsymbol{\sigma}_t, \boldsymbol{\beta}_t, \boldsymbol{\gamma}_t) + (\Delta\boldsymbol{\sigma}_t, \Delta\boldsymbol{\beta}_t, \Delta\boldsymbol{\gamma}_t) \tag{10.11}$$

donde la variación de cada tipo de movimiento δ es $\delta_t \in \{-\rho_\delta, 0, \rho_\delta\}$ para una decisión del módulo de control para decrementar, parar o incrementar la posición de la cámara en ese tipo de movimiento, respectivamente.

Además, cada tipo de movimiento tiene un máximo y un mínimo valor posible como un límite físico de la cámara. Por ejemplo, no se puede aplicar zoom hacia adentro todo lo que se quiera. Estos valores máximo y mínimo se denotan por $\boldsymbol{\Psi_\delta}$ y $\boldsymbol{\psi_\delta}$, respectivamente.

Para ejecutar los menores movimientos posibles, se han definido varios escenarios donde no se aplica movimiento a la cámara. Es decir, cuando las coordenadas horizontal y vertical del centroide del objeto seguido está cerca de las coordenadas vertical y horizontal del centro del fotograma, entonces el módulo de control no proporciona ningún comando de movimiento vertical y horizontal, respectivamente. Además, no se aplica el movimiento de zoom si el tamaño (en píxeles) del objeto objetivo está entre un valor mínimo y máximo del porcentaje respecto del número total de píxeles del fotograma. Es decir, para cada tipo de movimiento δ se ha especificado un valor mínimo y máximo: ϕ_δ and Φ_δ, respectivamente. Los valores para los movimientos horizontal y vertical indican la distancia (en grados) del centro del fotograma al centroide; y el valor del movimiento zoom indica el porcentaje del número de píxeles del fotograma que son ocupados por el objetivo.

Por último, cuando no se ha encontrado un objetivo el módulo de control trata activamente de buscar uno moviendo la cámara para evitar estados del sistema donde la cámara nunca encontrará un objetivo.

10.3. Resultados experimentales

En esta sección se muestran los resultados experimentales que se han ejecutado y sus resultados. El software y hardware que se ha utilizado se describen en la Subsección 10.3.1. Después se especifican las secuencias de video probadas en la Subsección 10.3.2. Por su parte, la Subsección 10.3.3 detalla los parámetros configurados del software y, por último, se comentan los resultados obtenidos de los experimentos en la Subsección 10.3.4.

10.3.1. Métodos

Se ha utilizado un algoritmo no panorámico de detección de objetos de primer plano para el sistema de control propuesto. El método empleado es trabajo previo del grupo de investigación (López-Rubio y López-Rubio, 2015). Este método que se denomina *nonpan* está implementado en Matlab y utiliza archivos MEX escritos en C++ para las partes que más tiempo demandan del microcontrolador. La implementación está disponible en su página web [1]. Por otro lado, la implementación del modelo GNG está escrita en Matlab.

El módulo de control de cámara está basado en la librería *virtualptz* (Chen et al., 2015). Con ella, a través de un video panorámico de 360 grados, se simula el funcionamiento de una cámara PTZ, y está disponible en su página web [2]. Su implementación está escrita en C++ y utiliza la librería OpenCV[3].

Los experimentos se han ejecutado en un ordenador personal de 64 bits con un microprocesador Intel i7 3,60 GHz de ocho núcleos, 32 GB RAM y hardware estándar. La implementación de la propuesta no necesita ningún recurso gráfico específico.

10.3.2. Secuencias

Tres videos se han utilizado para ejecutar los experimentos, que están disponibles en la página web de virtualptz. Las tres secuencias son escenas de interior y se llaman *scenario3*, *scenario4* y *scenario5*. Las dos primeras se corresponden con la misma localización (una espaciosa sala con gente andando en ella) y son muy similares, por lo que solamente se ha utilizado la secuencia *scenario3* (3500x1750 píxeles y 566 fotogramas). Por otro lado, la secuencia *scenario5* (3500x1750 píxeles y 1957 fotogramas) muestra una habitación con gente moviéndose en ella y haciendo diferentes acciones.

10.3.3. Selección de parámetros

Se ha definido un conjunto de valores de configuración de los parámetros de los métodos para ejecutar los experimentos. Estos parámetros fijados han sido elegidos de las recomendaciones de los autores del GNG y de experiencias propias. Son mostrados en la Tabla 10.1. En particular, los valores relacionados para el control de la cámara PTZ han sido seleccionados para atender la baja tasa de fotogramas de los videos de referencia.

[1] `http://www.lcc.uma.es/~ezeqlr/nonpan/nonpan.html`
[2] `https://bitbucket.org/pierre_luc_st_charles/virtualptz_standalone`
[3] http://opencv.org/

Tabla 10.1: Valores considerados de los parámetros.

Métodos	Parámetros
Nonpan	Características, $F = \{[19\ 20\ 22]\}$
	Tamaño de paso, $\epsilon = 0{,}03$
	Umbral, $\tau = 0{,}999$
Modelo GNG	Unidades máximas, $H_{max} = 100$
	Lambda, $\lambda = 100$
	Número de pasos, $N = 20000$
	Epsilon B, $\epsilon_b = 0{,}2$
	Epsilon N, $\epsilon_n = 0{,}006$
	Alpha, $\alpha = 0{,}5$
	A max, $a_{max} = 50$
	D, $d = 0{,}995$
	Pasos para eliminar neuronas no activas, $N_k = 1000$
Virtualptz	Variación del movimiento horizontal, $\rho_\sigma = 2$
	Variación del movimiento vertical, $\rho_\beta = 2$
	Variación del movimiento zoom, $\rho_\gamma = 2$
	Límite mínimo horizontal, $\psi_\sigma = -180$
	Límite máximo horizontal, $\Psi_\sigma = 180$
	Límite mínimo vertical, $\psi_\beta = 0$
	Límite máximo vertical, $\Psi_\beta = 180$
	Límite mínimo zoom, $\psi_\gamma = 40$
	Límite máximo zoom, $\Psi_\gamma = 140$
	Distancia mínima horizontal, $\phi_\sigma = -10$
	Distancia máxima horizontal, $\Phi_\sigma = 10$
	Distancia mínima vertical, $\phi_\beta = -20$
	Distancia máxima vertical, $\Phi_\beta = 20$
	Distancia mínima zoom, $\phi_\beta = 0{,}10$
	Distancia máxima zoom, $\Phi_\beta = 0{,}40$

10.3.4. Resultados

El funcionamiento de la propuesta con uno de los videos seleccionados se muestra en la Figura 10.1. En ella se muestra un fotograma capturado por la cámara PTZ, junto con el resultado obtenido después de la ejecución del algoritmo nonpan, el estado del modelo GNG en ese momento y la componente conectada más grande que debe ser seguida. Esta última imagen también muestra el centroide de la componente (seguida) más grande.

Tras esto, dependiendo del centroide y del tamaño de la componente

Figura 10.1: Descripción gráfica del funcionamiento del método propuesto. De izquierda a derecha, las columnas muestran un fotograma de una secuencia, la máscara binaria producida por el método no panorámico de detección de primer plano, el estado del modelo de red neuronal en este momento (los círculos rojos muestran las neuronas y las líneas azules representan las conexiones entre las neuronas), la componente seleccionada (es decir, la componente conectada más grande) y su centroide (representado por un asterisco verde). La primera y segunda filas muestran el fotograma 78 del scenario3, cada fila con una diferente inicialización del GNG en el primer fotograma.

seleccionada, el módulo de control indicará los diferentes comandos (movi-

mientos horizontal, vertical y de zoom) a la cámara.

El estado del GNG depende directamente de sus estados previos y de la salida del método nonpan. Cuando más rápido sea el movimiento de la cámara y las acciones de los agentes, menos cercano será el GNG con respecto al deseado. Esto puede ser apreciado en la Figura 10.2. La evolución del estado del GNG mantiene componentes de neuronas conectadas a pesar de que pertenecen a diferentes componentes. Este efecto es más fuerte cuanto mayor actividad de los agentes. Sin embargo, la propia naturaleza del GNG proporciona una aproximación eficiente del centroide del objetivo y una aproximación adecuada a su tamaño.

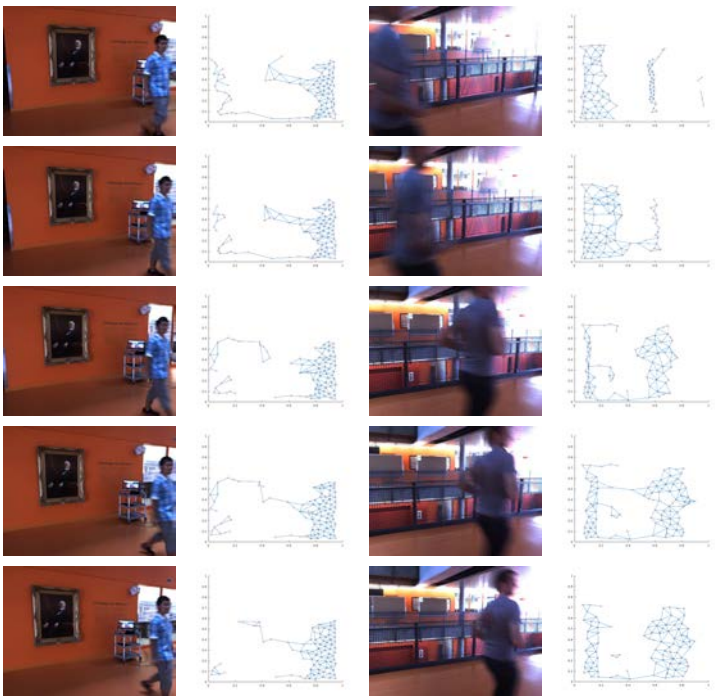


Figura 10.2: Evolución gráfica del GNG. La primera y segunda columnas se corresponden con los fotogramas 176 a 180 del video scenario3 y su correspondiente estado del GNG. La tercera y cuarta columnas se corresponden a los fotogramas 506 a 510 del video scenario3 y su correspondiente estado de GNG.

Además, en las dos primeras filas se puede observar cómo el resultado y la evolución del video puede ser diferente dependiendo de la inicialización de la red neuronal GNG. Esto tiene influencia en el módulo controlador y sus decisiones pueden ser diferentes con el mismo fotograma. Esto normalmente ocurre cuando el fotograma presenta más de una persona o un alto nivel de ruido en la salida producida por el método nonpan.

Se han elegido algunos fotogramas como máscara de verdad con el propó-

Tabla 10.2: Resultados de exactitud. La primera columna se corresponde con un video de referencia (las dos secuencias probadas con dos diferentes inicializaciones) y las columnas restantes indican la exactitud (aciertos/intentos) para los movimientos horizontal, vertical y de zoom considerando los 25 fotogramas de referencia seleccionados para cada video. Cada fila muestra un video y su rendimiento de exactitud y la última fila indica la exactitud total considerando los cuatro videos que han sido ejecutados.

Video	Horizontal	Vertical	Zoom
Scenario3 (1)	16/25 (0,64)	22/25 (0,88)	22/25 (0,88)
Scenario3 (2)	23/25 (0,92)	18/25 (0,72)	19/25 (0,76)
Scenario5 (1)	11/25 (0,44)	12/25 (0,48)	11/25 (0,44)
Scenario5 (2)	17/25 (0,68)	18/25 (0,72)	7/25 (0,28)
Total	**67/100 (0,67)**	**70/100 (0,70)**	**59/100 (0,59)**

sito de determinar el rendimiento. Es por ello por lo que se ha utilizado una máscara de verdad que contiene información del centroide y la región mínima rectangular de una persona. Por tanto, los fotogramas seleccionados como referencia presentan solo una persona en diferentes situaciones. Se muestran los comandos determinados por el controlador de acuerdo a la máscara de verdad de la posición de los objetos seguidos y la posición estimada del GNG.

Esta propuesta ha sido ejecutada con los dos videos probados y dos diferentes inicializaciones con el propósito de tener un rango más amplio de fotogramas de referencia, por lo que se tienen cuatro secuencias. Con cada una de ellas se han seleccionado 25 fotogramas aleatorios del conjunto de fotogramas de referencia. Como se puede observar en la Figura 10.3 los resultados cualitativos ofrecidos por el GNG son similares a la máscara de verdad. Además, la decisión del módulo de control es bastante similar para los datos de máscara de verdad y del GNG en cada uno de los fotogramas de referencia.

Para tener un punto de vista cuantitativo sobre el rendimiento de la propuesta se ha seleccionado la exactitud, calculando los aciertos de la decisión y los intentos. El rendimiento obtenido se muestra en la Tabla 10.2. Se puede considerar que el rendimiento más importante en los videos probados es la exactitud en el movimiento horizontal y, en un menor grado, el movimiento de zoom. Esto es porque las personas se mueven de izquierda a derecha y viceversa, más lejos o más cerca, pero casi siempre cerca de la cámara. De acuerdo a estos resultados, la mayoría de los errores en la decisión de los movimientos son producidos por el ruido del resultado obtenido del método nonpan y del estado del GNG. Esto se puede observar en las dos primeras

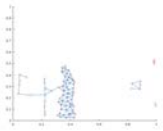
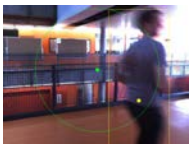
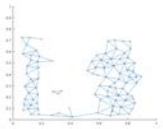
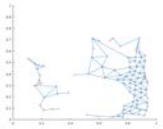
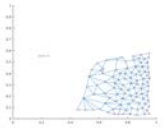
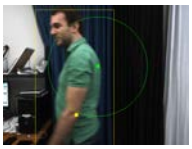
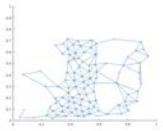
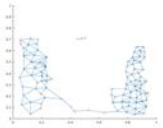
Fotograma con objetivo	GNG	GT	Estimado
		Izquierda No vertical Zoom hacia adentro	Izquierda Abajo Zoom hacia adentro
		Derecha Abajo Zoom hacia afuera	No horizontal No vertical Zoom hacia afuera
		Derecha No vertical No zoom	Derecha No vertical No zoom
		Derecha Abajo No zoom	Derecha No vertical No zoom
		Izquierda Abajo Zoom hacia afuera	No horizontal No vertical No zoom
		Derecha No vertical No zoom	No horizontal No vertical Zoom hacia afuera

Figura 10.3: Resultados cuantitativos para algunos fotogramas de referencia. La primera columna muestra un fotograma con el centroide y la región mínima rectangular de la máscara de verdad del objetivo, ambos coloreados en amarillo; y el centroide y el tamaño del objetivo indicados por el módulo de control, en verde. La segunda columna representa el estado del GNG en ese momento. Las últimas dos columnas muestran las direcciones dadas por el controlador a la máscara de verdad del objetivo (GT) y el objetivo detectado por el GNG (Estimado), respectivamente. La primera y segunda filas se corresponden con los fotogramas 58 y 510 del video scenario3, y la tercera y cuarta filas representan los fotogramas 171 y 192 para la misma secuencia pero con diferente inicialización y máscara de verdad objetivo. La quinta y sexta filas muestran los fotogramas 165 y 249 de la secuencia scenario5.

filas de la Figura 10.3: el GNG tiene algunas partes bien diferenciadas pero están conectadas, por lo que el GNG considera un objetivo más grande y su centroide es desplazado, produciendo una decisión de movimiento dife-

rente de la decisión de la máscara de verdad, lo que conlleva a una menor exactitud. Además, debe resaltarse el hecho de que los errores opuestos (por ejemplo, la máscara de verdad indica un movimiento a la izquierda y el GNG a la derecha) solo hayan aparecido unas pocas veces: 3 veces para las 100 decisiones de movimiento horizontal probadas y 2 veces para las 100 decisiones de movimiento de zoom testeadas.

10.4. Conclusión

Se ha presentado un controlador neuronal para cámaras PTZ, que está basado en un gas neuronal creciente (GNG), para optimizar la cobertura máxima del área de la escena en presencia de objetos de primer plano. Los objetos en movimiento son detectados utilizando un algoritmo no panorámico de detección de primer plano que ofrece una máscara binaria de primer plano. El modelo GNG representa la máscara de primer plano con el objetivo de eliminar ruido y objetos espurios, además de proporcionar mayor robustez al módulo de control de cámara.

Se ha utilizado la librería *virtualptz* para simular el rendimiento de una cámara PTZ real. Varias secuencias de video públicamente disponibles se han considerado en este estudio. En particular, algunos resultados cualitativos se obtienen en comparación con la máscara de verdad (movimientos horizontal, vertical y de zoom en cada fotograma) de la escena. Dentro de este esquema se obtienen algunos resultados prometedores (alrededor del 65 % de exactitud). Se debe tener en cuenta que lograr exactamente los mismos movimientos que la máscara de verdad no siempre es necesario para obtener la mejor cobertura en una escena. Esto significa que la evaluación del rendimiento de este tipo de sistemas es complicada.

Parte III

Aplicaciones al transporte

Esta tercera parte de la tesis presenta los trabajos desarrollados sobre transporte inteligente. Así, en esta parte se engloban trabajos de distinta índole como lo es la clasificación de los vehículos que aparecen en secuencias de tráfico. En uno de los capítulos muestra la aplicación de técnicas tradicionales como la segmentación y posterior extracción de rasgos o características; en el siguiente capítulo se emplea una fase de segmentación y varias redes convolucionales en lugar de la extracción de características, ello unido a un estudio del redimensionado de las imágenes para proveerlas en el formato necesario a cada red convolucional, además de la conveniencia de utilizar superresolución en dicho redimensionado; y el último capítulo de esta parte hace uso de un modelo de red que detecta y clasifica objetos, tras lo que se estima la contaminación generada por los vehículos detectados.

Capítulo 11

Detección de tipos de vehículos mediante grupos de redes convolucionales y superresolución

La formulación de un problema es más importante que su solución.

Albert Einstein

Resumen: La detección y clasificación automática de vehículos en secuencias de tráfico es una tarea típica que es ejecutada en muchos sistemas prácticos de videovigilancia. La llegada del aprendizaje profundo ha facilitado el diseño de estos sistemas. Sin embargo, las limitaciones en la resolución de las cámaras de videovigilancia implican que los vehículos no sean claramente definidos en los fotogramas del video entrante, lo que dificulta el rendimiento de clasificación de las redes neuronales convolucionales de aprendizaje profundo. En este capítulo se presenta un método para superar este reto, que está basado en varios pasos. Una segmentación inicial es seguida de un postprocesado de las imágenes segmentadas para resolver el solapamiento de vehículos y los diferentes tamaños de vehículos. Después se emplea un algoritmo de superresolución para mejorar la definición de las ventanas de imagen para ser suministradas a las redes neuronales. Por último, las salidas de un grupo de tales redes se integran para obtener un rendimiento de reconocimiento mejorado por el consenso de las redes del grupo. Varios test computacionales utilizando conocidos videos de referencia demuestran la efectividad de la propuesta, incluso en situaciones difíciles. Por tanto, el sistema de clasificación de vehículos propuesto supera muchas

limitaciones de sencillas aplicaciones de redes neuronales convolucionales, ya que cada subsistema propuesto aborda diferentes dificultades que surgen en los datos de video de tráfico reales.

11.1. Introducción

Los sistemas de videovigilancia son ampliamente utilizados en cualquier esquina del mundo, lo que posibilita la recolección de una gran cantidad de datos que pueden ser utilizados para numerosos propósitos. El tráfico es uno de los tipos de escenario que pueden ser analizados automáticamente por estos sistemas para detectar congestiones, incidentes e infracciones en el cumplimiento de las necesidades de la carretera (Ren et al., 2016; García et al., 2017). La descripción de las secuencias es un importante punto a ser considerado para lograr información significativa del tráfico de la carretera. La clase, posición y velocidad de los vehículos, al menos, son aspectos necesarios para obtener resultados útiles (Mithun et al., 2016).

La detección del fondo y el primer plano en las secuencias de video es el primer paso en la mayoría de los sistemas de videovigilancia (Thurnhofer-Hemsi et al., 2018; López-Rubio et al., 2018a,b; Ortega-Zamorano et al., 2016), y en particular en videovigilancia de tráfico. Los algoritmos basados en modelos de Markov (Baumgartner et al., 2016), distribuciones gaussianas individuales (Wren et al., 1997) o redes autoorganizadas (Maddalena y Petrosino, 2008) son ejemplos de los métodos desarrollados para modelar el fondo de una escena. Hay casos en los que las características del escenario son conocidas y se pueden aplicar técnicas más específicas para mejorar el rendimiento de la detección. Por ejemplo, se puede modelar una escena con condiciones de nieve y niebla (Sen-Ching y Kamath, 2004).

Los siguientes pasos consisten en la detección y el consiguiente seguimiento de los objetos a lo largo de la secuencia. Un ejemplo puede ser un sistema de seguimiento de peatones (Lacabex et al., 2016). En este caso, cada objeto es etiquetado para identificar su clase, es decir, coche, camión o moto son tipos de vehículos vistos en un video de una carretera. Cada clase debe ser identificada de manera única, lo que es un proceso complejo que requiere un proceso previo de extracción de características. El clasificador utiliza como entradas las características significativas que pueden ser extraídas de los objetos para mejorar la predicción de este módulo. Como ejemplo, la textura o el brillo se pueden utilizar (Wang et al., 2016). El aprendizaje profundo también tiene aplicación para este propósito (Kato et al., 2016), detectando las características principales de una manera intrínseca en las primeras capas de la red neuronal.

En este capítulo se propone un nuevo método de detección de vehículos en secuencias de video de carreteras. Basándose en un sistema de seguimien-

to de vehículos previo (Luque-Baena et al., 2015b), se ha desarrollado un nuevo algoritmo que integra redes neuronales convolucionales (Convolutional Neural Networks o CNN) para clasificar los vehículos que aparecen en un típico escenario de tráfico. Las CNNs son una técnica del estado del arte para clasificar imágenes que aprenden las características locales óptimas de una imagen para una tarea de clasificación dada (Sanchez et al., 2016; Koziarski y Cyganek, 2017; Ortega-Zamorano et al., 2017; Lin et al., 2017). Se pueden configurar para reconocer objetos o estructuras específicos (Zhang et al., 2017; Cha et al., 2017; Rafiei y Adeli, 2015), para mejorar la calidad de la seguridad (Rafiei et al., 2017), o para evaluar el estado de un objeto (Rafiei y Adeli, 2018). Sin embargo, cuando el problema es complejo, el aprendizaje en grupo ha sido probado para mejorar los resultados (Islam y Yao, 2008). La principal idea es combinar diferentes salidas para obtener mejores resultados. Hay ejemplos en una amplia variedad de campos (reconocimiento de imágenes, medicina, etc) que muestran un grupo de redes neuronales que un conjunto de redes neuronales permitirá extraer características de mayor calidad y puede producir una mejora del método original (Abuassba et al., 2017; Ortiz et al., 2016; Fernández et al., 2017).

La CNN utilizada en este trabajo es conocida como AlexNet (Krizhevsky et al., 2012) y ha sido utilizada previamente en tareas de supervisión de tráfico (Amato et al., 2017; Wshah et al., 2016). La entrada estándar de una red neuronal, Alexnet en particular, es una imagen donde el objeto de interés ocupa la mayor parte del área de la imagen. Una situación típica es cuando la entrada puede ser afectada por oclusiones. Por ejemplo, en Amato et al. 2017 se muestra cómo los árboles bloquean parcialmente la visión de un vehículo. El punto clave es que la oclusión puede estar causada por objetos que no pueden ser confundidos con los objetos a reconocer.

El preprocesamiento de las imágenes de video antes del procesamiento por parte de la CNN es un desafío importante. Hay casos donde el flujo de la cámara es segmentado manualmente (Amato et al., 2017) o se implementa una estructura especial de sensor (Wshah et al., 2016) para satisfacer las condiciones de entrada de la red. El propósito de este trabajo es centrarse en este problema y tratar de solventarlo para la detección de tipos de vehículos en escenas de tráfico. La imagen adquirida presenta a menudo oclusiones parciales producidas por otros vehículos. Por tanto, se debe ejecutar un procedimiento de segmentación para obtener las regiones de primer plano y dividirlas en vehículos. Solo un vehículo debe aparecer en cada imagen de entrada, por lo que se deben determinar ventanas rectangulares y ejecutar después un escalado apropiado para satisfacer las necesidades de la CNN (en el caso de Alexnet, imágenes RGB de 227×227 píxeles).

Se deben considerar dos factores importantes para la obtención de resultados razonables. Primero, es difícil encontrar un escalado apropiado de los datos originales del video para satisfacer las necesidades de tamaño de

la CNN porque el tamaño de los vehículos varía de motos a camiones en gran medida. Es necesario gestionar adecuadamente este problema para obtener una tasa de reconocimiento alta. Segundo, las imágenes de vehículos segmentados tienen pocos píxeles debido a la pobre resolución de los videos de tráfico. Para la gestión de esta situación, las técnicas de superresolución (superresolution o SR) pueden utilizarse para mejorar la resolución de la imagen de entrada y, por tanto, mejorar la habilidad de reconocimiento de la CNN. La SR puede reconstruir una imagen de alta resolución basándose en un conjunto de imágenes de baja resolución (Maiseli et al., 2015) o de una sola imagen de baja resolución (Yang et al., 2014a). Se puede ver como un método para superar las limitaciones de los dispositivos de adquisición de imágenes (Tian y Ma, 2011).

El resto del capítulo se organiza como sigue: la Sección 11.2 presenta la arquitectura del sistema donde se integra esta propuesta y la Sección 11.3 establece el infraestructura de clasificación que describe cómo se aplican la superresolución y la red neuronal convolucional. La Sección 11.4 muestra los resultados experimentales sobre una secuencia pública de videovigilancia de tráfico y la Sección 11.5 concluye el capítulo.

11.2. Arquitectura del sistema

El sistema de seguimiento de vehículos que se ha desarrollado (Figure 11.1) puede dividirse en tres diferentes pasos: uno inicial donde los objetos son detectados y extraídos de la secuencia de fotogramas, un segundo paso donde los objetos son seguidos a lo largo de la secuencia de video, y por último, en el tercer paso los objetos son clasificados de acuerdo a los diferentes tipos de vehículos considerados. En esta sección se explica la estructura de cada etapa.

La etapa de detección y extracción de objetos tiene dos partes. Primero, se consigue una segmentación inicial basada en un método de detección de objetos (López-Rubio y Luque-Baena, 2011) que utiliza mixturas de distribución uniforme y gaussianas multivariadas con matrices de covarianza completa, y desarrolla un proceso de actualización del modelo que está basado en la aproximación estocástica. Este método es robusto para el modelado del fondo. El marco de trabajo de la aproximación estocástica ha sido probado como un buen método de aprendizaje para algoritmos de tiempo real en línea que descartan los datos de entrada cuando son procesados. El fondo y el primer plano de la escena son modelados utilizando una gaussiana y una componente uniforme, respectivamente. Bajo este modelo probabilístico, se utiliza el algoritmo de aproximación estocástica de Robbins-Monro para la actualización de las ecuaciones (Robbins y Monro, 1951). Por tanto, el resultado de esta parte inicial de la etapa de detección es una máscara binaria que separa los píxeles de primer plano de los píxeles del fondo.

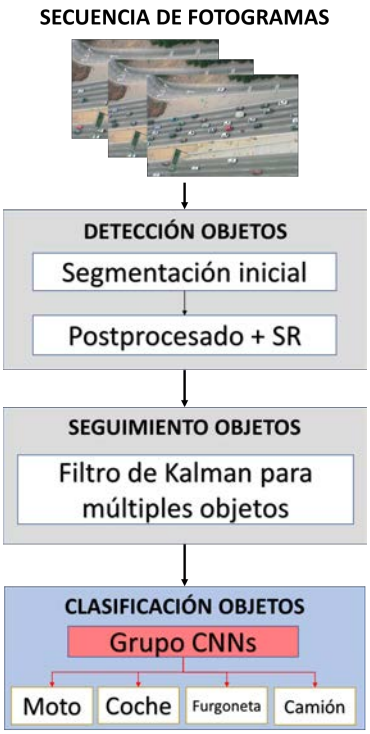


Figura 11.1: Esquema del sistema de seguimiento de vehículos propuesto.

Además, se ejecuta un proceso de postprocesado para las imágenes segmentadas, para corregir imperfecciones en la máscara binaria debido a cambios de iluminación, efectos de camuflaje, y otros inconvenientes. Se aplican algunos operados morfológicos básicos como la erosión y la dilatación. Por tanto, se pueden eliminar los falsos positivos (píxeles de fondo que son asignados a primer plano) que están presentes en la imagen segmentada, y también los falsos negativos (píxeles de primer plano que están asignados al fondo) que aparecen en el interior de los objetos. Primero se aplica la erosión y después la operación de dilatación para recuperar el tamaño original de los objetos. Hay casos donde se tienen objetos solapados porque están muy cerca unos de otros. Es necesario dividirlos para separarlos en cada vehículo individual. Tras esto, se ejecuta un procedimiento de dos pasos para escalar la imagen segmentada para satisfacer los requisitos de la red neuronal. Aplicando un proceso de superresolución se obtiene una imagen de alta resolución, que es seguida de un algoritmo de nitidez. La nitidez de una imagen es un proceso que consiste en la mejora de la imagen enfatizando la definición del borde mediante la aplicación de filtros convolucionales específicos (Ngocho y Mwangi, 2016). El objetivo final es escalar y mejorar la calidad de las imágenes segmentadas y eliminar objetos espurios.

La etapa de seguimiento de objetos utiliza las detecciones de objetos

obtenidas en la etapa anterior para estimar sus trayectorias a lo largo del tiempo. Para la etapa de seguimiento se ha implementado una versión del filtro de Kalman para múltiples objetos (Reid, 1979). Este modelo estadístico procesa las detecciones de objetos de la etapa de detección de objetos, que son utilizados para estimar la posición de cada objeto seguido en el siguiente fotograma. Para este propósito se extraen las principales características de cada objeto detectado en el fotograma actual, y después se desarrolla una búsqueda sobre la pequeña ventana centrada en el centroide del objeto actual para calcular la posición más probable del objeto seguido en el siguiente fotograma.

Por último, existe una etapa de clasificación de objetos para determinar las clases de los objetos seguidos. Se emplea un modelo de clasificación basado en un grupo de CNNs para determinar el tipo de los vehículos presentes en la imagen. Dependiendo de sus características, se han considerado cuatro clases: moto (moto), coche (car), furgoneta (van) y camión (truck). En la siguiente sección se muestran más detalles del modelo de clasificación.

11.3. Marco de clasificación

11.3.1. Red individual

La CNN que se ha empleado en este trabajo es AlexNet (Krizhevsky et al., 2012). Esta red fue desarrollada para clasificar el conjunto de datos IMAGENET[1], que es un conjunto de imágenes de alta resolución. AlexNet está compuesta por 5 capas convolucionales y capas totalmente conectadas, donde las neuronas que pertenecen a la capa totalmente conectada tienen conexiones con todas las neuronas de entrada. Además, tras las dos primeras capas convolucionales se utilizan capas de normalización. Por último, después de estas capas de normalización y la quinta capa convolucional, se han colocado capas de agrupación. En este trabajo se ha utilizado una réplica del modelo original basada en Caffe [2].

Se ha considerado un conjunto de muestras de entrada para entrenar la red. Este conjunto de entrenamiento está compuesto del mismo número de muestras para cada clase de vehículo. Por otro lado, las imágenes han sido seccionadas de las secuencias de test y cada una de ellas muestra vehículos moviéndose por el video. Varios vehículos de este conjunto de datos se muestran en la Figura 11.2 ordenados por su tipo.

Para entrenar la CNN se necesita un conjunto de datos con todas las imágenes con el mismo tamaño (256x256 píxeles). Sin embargo, las muestras de vehículos no tienen las mismas dimensiones (por ejemplo, los camiones son más grandes que las motos), lo que implica que las regiones de imágenes

[1] `http://www.image-net.org`
[2] `https://github.com/BVLC/caffe/tree/master/models/bvlc_alexnet`

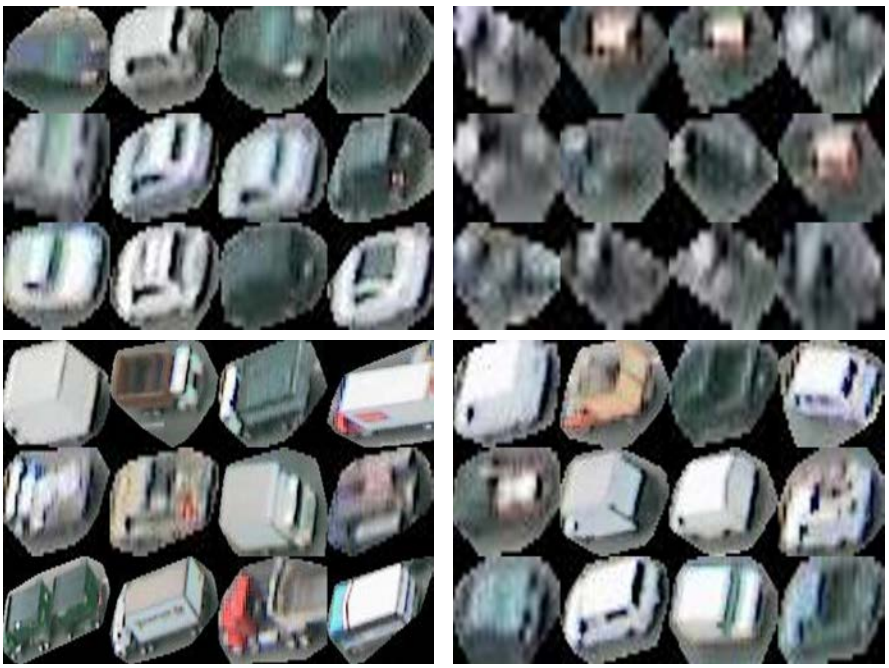


Figura 11.2: Tipos de vehículos. Se aplica una operación de escalado para mostrarlos convenientemente. La primera fila muestra coches y motos, y la segunda representa camiones y furgonetas, respectivamente.

deben ser redimensionadas para servir de entrada a la red neuronal. Es decir, se necesita aplicar un proceso para adaptar el conjunto de datos al tamaño requerido.

Para obtener el mejor rendimiento posible, se ha considerado la estrategia del escalado centrado (Centered Scale), que es la mejor de las estrategias propuestas en Molina-Cabello et al. 2017c. Esta estrategia redimensiona la imagen aplicando una escala preservando el ratio entre las imágenes originales, colocándola en el centro de la imagen mientras el resto de los píxeles se rellenan con color negro. Las estrategias de no redimensionado o no centradas tienen el inconveniente de que los objetos pequeños como motos y coches normales pueden estar fuera de los límites. Un ejemplo de la aplicación del método del escalado centrado se encuentra en la Figura 11.3. Tras este proceso, dada una imagen de poca resolución $\mathbf{z}$ de tamaño $n \times m$ píxeles, el método de escalado produce una imagen $\mathbf{Z}$ de tamaño 256×256 píxeles.

Las redes de aprendizaje profundo ofrecen el mejor rendimiento si el número de patrones de entrenamiento es alto. Cuando esta condición no se satisface es posible ejecutar un proceso llamado aumento de datos (data augmentation). En esta propuesta se toman 10 regiones de 227x227 píxeles para cada muestra que sirve como entrada a la red, donde cada región tiene un desplazamiento aleatorio sobre la original región de 256x256 píxeles. Además

Figura 11.3: Redimensionado con el método de escalado centrado para 256x256 píxeles. Las clases coche, camión, moto y furgoneta se presentan de la primera a la cuarta fila, respectivamente.

de modificar la entrada de la red, es necesario modificar la salida. La única variación en este punto es cambiar el número de clases de salida a 4, porque hay cuatro tipos de vehículos a ser detectados: coches, motos, camiones y furgonetas.

11.3.2. Superresolución

Como se ha visto anteriormente, las imágenes de los vehículos a ser clasificados son normalmente mucho más pequeños que el tamaño de 256×256 píxeles que el modelo AlexNet de la Subsección 11.3.1 necesita. Para obtener versiones ampliadas de las imágenes iniciales de los vehículos se aplica un método de superresolución de imágenes (single image superresolution o SISR), en lugar del sencillo redimensionado de la imagen. El método SISR proporciona una mejor calidad visual que el redimensionado sencillo, lo que mejora el redimiento del reconocimiento de objetos de la etapa de clasificación. Por tanto, el SISR puede ser integrado en la estrategia de centrado escalado, propuesta previamente en la Subsección 11.3.1, que emplea un redimensionado de la imagen con el vehículo segmentado.

El algoritmo de la transformación del filtro de la mediana (Median Filter Transform o MFT) ha sido elegido para obtener imágenes de alta resolución a partir de entradas de baja resolución. El MFT (López-Rubio, 2016) está basado en la aplicación del filtro de la mediana en ventanas con forma de paralelogramo. Además, el MFT tiene la ventaja de que puede reducir el ruido presente en la imagen segmentada. En este trabajo el MFT se emplea para agrandar las imágenes de vehículos antes de su presentación a las CNNs.

Dada una imagen original de un vehículo de tamaño $n \times m$ píxeles, el tamaño $N \times M$ de la imagen con superresolución depende de la estrategia de redimensionado. En la estrategia de escalado centrado se preserva la relación de aspecto y la distancia focal (zoom) se ajusta al vehículo más ancho del conjunto de entrenamiento, es decir, $N = \frac{256n}{m_{widest}}$, $M = \frac{256m}{m_{widest}}$, donde m_{widest} es el número de columnas del vehículo más ancho.

Una vez que se determina el tamaño $N \times M$ de la imagen con superresolución, entonces se ejecuta el MFT. Se considera una imagen con baja

resolución (low resolution o LR) $\mathbf{z}$ de tamaño $n \times m$ píxeles, con coordenadas de píxeles $\mathbf{x} = (x_1, x_2) \in \mathbb{Z}^2$. Las coordenadas del píxel en la imagen de alta resolución (high resolution o HR) $\mathbf{Z}$ de tamaño $N \times M$ píxeles son $\mathbf{y} = (y_1, y_2) \in \mathbb{Z}^2$. El píxel en $\mathbf{x}$ en la imagen LR es asociado a las coordenadas $\alpha\mathbf{x} = (\alpha x_1, \alpha x_2) \in \mathbb{R}^2$ en la imagen HR, donde $\alpha > 1$ es el (posiblemente no entero) factor de zoom. El MFT de $\mathbf{z}$ se define como la siguiente imagen HR:

$$\mathbf{Z}(\mathbf{y}) = \text{median}\left(\left\{\psi\left(\mathbf{y}, \mathbf{A}_1, \mathbf{b}_1\right), ..., \psi\left(\mathbf{y}, \mathbf{A}_H, \mathbf{b}_H\right)\right\}\right) \tag{11.1}$$

$$\forall i \in \{1, ..., H\}, \ \psi\left(\mathbf{y}, \mathbf{A}_i, \mathbf{b}_i\right) = \text{median}\left(\zeta\left(\mathbf{y}, \mathbf{A}_i, \mathbf{b}_i\right)\right) \tag{11.2}$$

donde:

- H es una constante, que se llama número de baldosas.
- $\mathbf{A}_i$ son matrices 2×2 seleccionadas aleatoriamente de una distribución apropiada $p(\mathbf{A})$, que se define por la medida de probabilidad de Haar para la parte de rotación de $\mathbf{A}$ y la de Jeffreys antes de parte de escalado de $\mathbf{A}$.
- $\mathbf{b}_i$ son vectores 2×1 seleccionadas aleatoriamente de una distribución uniforme $p(\mathbf{b})$.
- $\zeta\left(\mathbf{y}, \mathbf{A}_i, \mathbf{b}_i\right)$ es un conjunto de valores de píxeles tomados de la imagen LR que se define por

$$\zeta(\mathbf{y}, \mathbf{A}, \mathbf{b}) =$$

$$\left\{\mathbf{z}(\mathbf{x}) \mid \text{round}\left(\mathbf{A}\alpha\mathbf{x} + \mathbf{b}\right) = \text{round}\left(\mathbf{A}\mathbf{y} + \mathbf{b}\right)\right\} \tag{11.3}$$

donde round redondea las componentes de un vector de dos elementos de valor real al entero más cercano, y α es el factor de zoom. El conjunto $\zeta(\mathbf{y}, \mathbf{A}, \mathbf{b})$ se compone de todos los píxeles LR que pertenecen al paralelogramo donde el píxel HR $\mathbf{y}$ pertenece, de acuerdo a la baldosa del plano definido por $\mathbf{A}$ y $\mathbf{b}$. Más detalles sobre el MFT se pueden encontrar en López-Rubio 2016, así como las definiciones de $p(\mathbf{A})$ y $p(\mathbf{b})$.

Después de aplicar el MFT se aplica una etapa de postprocesado adicional. Se trata de un operador de afilado de bordes, que mejora el contraste de los bordes, produciendo un mejor rendimiento de reconocimiento de objetos de las CNNs.

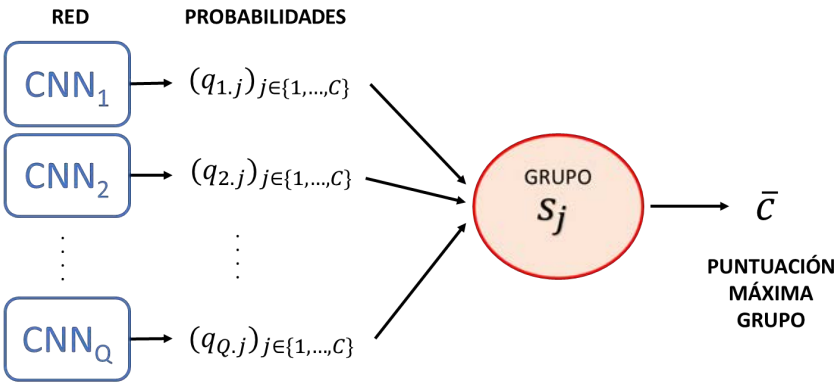


Figura 11.4: Grupo de CNNs del marco de trabajo propuesto.

11.3.3. Grupo de redes

A continuación se describe el grupo (ensemble) de CNNs del marco de trabajo propuesto para mejorar el rendimiento de la clasificación de las CNNs individuales consideradas anteriormente. En la Figura 11.4 se muestra un esquema del modelo de grupo. El objetivo general del grupo es combinar las salidas de varias CNNs, es decir, se toma una decisión por consenso sobre el tipo del vehículo. La decisión por consenso es más probable que sea correcta que alguna de las decisiones hechas por las CNNs individuales, ya que el mecanismo de consenso compensa algunos errores de clasificación cada vez que la mayoría de las salidas individuales de cada CNN sean buenas. Los grupos son muy populares en numerosos campos de la informática como la optimización (Rostami et al., 2017; Rostami y Neri, 2016), porque un grupo puede lograr una mejor estimación del mejor valor de los parámetros a ser optimizados (Iacca et al., 2015, 2014).

Sea $\mathbf{Z}$ la imagen con superresolución correspondiente a un objeto a ser clasificado. Sea Q el número de CNNs que forman el grupo, y C el número de posibles clases c_j, donde $j \in \{1, ..., C\}$. En este caso, $C = 4$. La i-ésima CNN produce un vector de probabilidades de clase para la imagen:

$$\mathbf{q}_i(\mathbf{Z}) = (P(c_j|\mathbf{Z}))_{j\in\{1,...,C\}} = (q_{i,j}(\mathbf{Z}))_{j\in\{1,...,C\}} \tag{11.4}$$

Entonces los siguientes métodos alternativos pueden ser considerados para unir la información que procede de los miembros del grupo para lograr las puntuaciones de clase $s_j(\mathbf{Z})$:

- Grupo de media. La puntuación de una clase del grupo es la media de las puntuaciones individuales de esa clase:

$$s_j(\mathbf{Z}) = \text{mean}\{q_{i,j}(\mathbf{Z}) | i \in \{1, ..., Q\}\} \tag{11.5}$$

- Grupo de mediana. La puntuación de una clase del grupo es la mediana de las puntuaciones individuales de esa clase:

$$s_j(\mathbf{Z}) = \text{median}\left\{q_{i,j}(\mathbf{Z}) \,|i \in \{1,...,Q\}\right\} \tag{11.6}$$

- Grupo de máximo. La puntuación de una clase del grupo es el máximo de las puntuaciones individuales de esa clase:

$$s_j(\mathbf{Z}) = \max\left\{q_{i,j}(\mathbf{Z}) \,|i \in \{1,...,Q\}\right\} \tag{11.7}$$

- Grupo de votos. La puntuación de una clase del grupo es el número de CNNs para el cual la clase alcanza la puntuación individual máxima:

$$s_j(\mathbf{Z}) = \sum_{i \in \{1,...,Q\}} \mathbb{I}\left(j = \arg \max_{j \in \{1,...,C\}} q_{i,j}(\mathbf{Z})\right) \tag{11.8}$$

  donde $\mathbb{I}$ se refiere a la función indicadora.

Después de que las puntuaciones de clase sean calculadas, se predice que el objeto descrito por $\mathbf{Z}$ pertenece a la clase que logra la máxima puntuación del grupo:

$$\tilde{c}(\mathbf{Z}) = \arg \max_{j \in \{1,...,C\}} s_j(\mathbf{Z}) \tag{11.9}$$

También se consideran grupos complejos como una opción adicional. El objeto descrito en $\mathbf{Z}$ es clasificado de acuerdo a la salida del clasificador φ que acepta las salidas de las CNNs como entradas:

$$\tilde{c}(\mathbf{Z}) = \varphi\left(q_{1,1}(\mathbf{Z}), ..., q_{Q,C}(\mathbf{Z})\right) \tag{11.10}$$

En la siguiente sección, se prueba experimentalmente la propuesta.

11.4. Resultados experimentales

En esta sección se muestran los diferentes tests que se han ejecutado y sus resultados.

11.4.1. Secuencias

Para probar la propuesta que se muestra en este capítulo, se han seleccionado varias secuencias que muestran dos autovías con diferentes tipos de vehículos moviéndose en ellas, presentando algunos problemas como oclusiones y objetos que se solapan. Los videos seleccionados son los denominados *sb-camera2-0750am-0805am* (*SB*) y *lankershim-camera3-0830am-0845am* (*Lankershim*). Estas secuencias se han tomado del conjunto de datos

del programa Next Generation Simulation (NGSIM), proporcionado por la Federal Highway Administration (FHWA).

11.4.2. Test de rendimiento

La propuesta trata de clasificar los vehículos detectados en una secuencia en cuatro diferentes clases dependiendo de las características de los vehículos. Las posibles clases son moto, coche, furgoneta y camión. Para entrenar la red y testearla, se han seleccionado varios vehículos con su trayectoria y se han etiquetado. La trayectoria de un objeto segmentado O_i es un conjunto de características $\{\mathbf{f}_i^j \in \mathbb{R}^4 \mid j \in [1..MaxFrame]\}$ que está compuesto por todos los fotogramas j donde este vehículo aparece en el video entero. Después, se forma el conjunto de imágenes que se usan para entrenar y testear el rendimiento de la propuesta. Las imágenes de este conjunto se corresponden con imágenes de cada trayectoria seleccionada previamente.

Para obtener un rendimiento de la bondad de la propuesta que sea robusto se ha empleado una estrategia de 10 pliegues (10-fold), donde el conjunto de entrenamiento tiene el 90 por ciento de los datos y el conjunto de test tiene el restante 10 por ciento. Además, cada conjunto tiene el mismo número de imágenes de cada clase. Este proceso se ha repetido 100 veces.

Tras el entrenamiento se utiliza el conjunto de test para medir el rendimiento de la propuesta. Para comparar los distintos métodos entre ellos desde un punto de vista cuantitativo, se han seleccionado varias medidas bien conocidas. La medida más importante que se ha considerado es la exactitud (Accuracy o Acc), que proporciona valores en el intervalo $[0, 1]$, donde mayor es mejor, y representa el porcentaje de aciertos del sistema.

Sea K los objetos existentes y k el objeto observado, se denota por $\mathbf{x}_k$ y $\mathbf{w}_k$ la clase real y la predicha del objeto k, respectivamente, donde $\mathbf{x_k}, \mathbf{w_k} \in \{1, 2, 3, 4\}$, correspondiente a $1 = moto, 2 = car, 3 = van$ y $4 = truck$. Además, si el modelo acierta con la clasificación del objeto k, (es decir, $\mathbf{x_k} = \mathbf{w_k}$), esto se denota por $\mathbf{q}_k = 1$; y se denota por $\mathbf{q}_k = 0$ cuando el modelo produce un error en su predicción (es decir, $\mathbf{x_k} \neq \mathbf{w_k}$). Por tanto, la definición de la exactitud es:

$$Acc = \frac{1}{K} \sum_{k=1}^{K} \mathbf{q}_k \tag{11.11}$$

El factor Kappa (κ), la especificidad (specificity) y la sensibilidad (sensitivity) también se emplean y sus definiciones son:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{11.12}$$

$$Sensibilidad = \frac{TP}{TP + FN} \tag{11.13}$$

$$Especificidad = \frac{TN}{FP + TN} \tag{11.14}$$

donde p_o es el acuerdo relativo observado entre calificadores (en este caso, $p_o = Acc$), y p_e es la probabilidad hipotética de acuerdo casual, por lo que, para K categorías, N números de objetos y n_{ki} el número de veces que el evaluador i predijo la categoría k:

$$p_e = \frac{1}{N^2} \sum_{k=1}^{K} n_{k1} n_{k2} \tag{11.15}$$

11.4.3. Métodos

Para obtener un rendimiento mejor que el presentado en Molina-Cabello et al. 2017c, se ha desarrollado un grupo compuesto por varias redes entrenadas utilizando la estrategia de redimensionado de escalado centrado. También se ha estudiado la aplicación del proceso de superresolución en la etapa de redimensionado. Desde un punto de vista cualitativo, la diferencia en el redimensionado normal y el que utiliza superresolución se puede observar en la Figura 11.5. La imagen con superresolución presenta una aproximación más suavizada y no está pixelada.

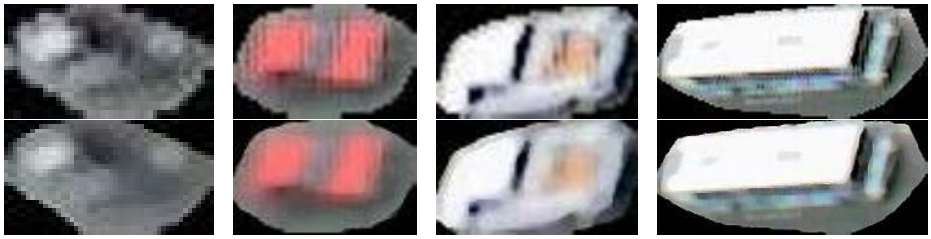


Figura 11.5: Detalle de la diferencia entre varios vehículos segmentados y las mismás imágenes con el proceso de superresolución aplicado. La primera y segunda fila muestran las imágenes con la imagen segmentada original y la imagen con superresolución, respectivamente. Cada columna muestra una moto, un coche, una furgoneta y un camión, respectivamente.

En ambos casos (aplicando superresolución o no), el conjunto de imágenes se ha dividido en dos grupos aleatorios: el 90 por ciento de los datos se ha utilizado como entrenamiento de las redes que componen el grupo y el restante 10 por ciento de los datos se utiliza para el test del grupo. El conjunto de entrenamiento también se divide en 10 subgrupos siguiendo una estrategia de 10 pliegues, es decir, en este proceso se obtienen 10 redes con diferente conjunto de entrenamiento entre cada una. Cada uno de los conjuntos propuestos tiene el mismo número de imágenes por clase. El conjunto de datos del video SB está formado por 192 imágenes, donde el conjunto de

datos de entrenamiento y el de test tienen 120 y 12 imágenes seleccionadas aleatoriamente, respectivamente, con el mismo número de imágenes por clase; mientras que el conjunto de datos del video Lankershim está compuesto por 187 imágenes, donde el conjunto de entrenamiento y el de test tienen 140 y 16 imágenes seleccionadas aleatoriamente, respectivamente, con el mismo número de imágenes por clase.

Una vez que se han entrenado 10 redes con el proceso de los k-pliegues, se construye un grupo con ellas. De esta forma, dada una imagen de la que nos gustaría saber su clase, los resultados producidos por cada red (la clase proporcionada y las probabilidades de pertenencia a cada clase) son combinados, proporcionando un nuevo resultado. Se han implementado las estrategias propuestas en la Subsección 11.3.3. En el caso de los grupos complejos que aprenden una función de agregación utilizando algoritmos de clasificación supervisados, las alternativas consideradas en este trabajo para este clasificador φ son:

- MLP (Rumelhart et al., 1986): Los perceptrones multicapa (Multilayer Perceptrons) son un tipo de red neuronal artificial que simulan los procesos de aprendizaje biológico mediante una serie de estrategias de ajuste de peso.

- SVM (Cortes y Vapnik, 1995): Las máquinas de soporte vectorial (Support Vector Machines) son modelos de aprendizaje que clasifican datos encontrando el mejor hiperplano que divide las muestras proporcionadas en cada clase.

- NB (Friedman y Russell, 1997): El clasificador de las redes bayesianas (Naive Bayes) es una propuesta probabilística que considera que las observaciones o muestras son estadísticamente independientes entre ellas.

- DT (Quinlan, 1986): Los árboles de decisión (Decision Trees) son técnicas de modelado predictivo que representan los datos de decisión en forma de árbol. Para predecir una respuesta, la muestra sigue las decisiones desde el nodo raíz hacia abajo hasta un nodo hoja (clase).

- RF (Breiman, 2001): Los bosques aleatorios (Random Forests) son métodos de aprendizaje de grupo que trabajan en combinación con los árboles de decisión donde, dada una observación, toman una salida por consenso.

Todos estos métodos se han implementado en Matlab. Los valores de los parámetros de cada método se encuentran especificados en la Tabla 11.1 y son aquellos que se han encontrado que rinden bien en los experimentos o

que aparecen en la literatura [3] [4].

Método	Parámetro
MLP	Número de neuronas en cada capa oculta, $h = [4, 5]$
SVM	Función de núcleo, $k = lineal$(producto punto)
	Método usado para encontrar el hiperplano separador,
	$m =$ (Optimización Secuencia Mínima)
NB	Modelo de distribución predictor dentro de cada clase, $pdm = gaussiano$
DT	Umbral de p-valor, $t = 0{,}05$
	Distribución de probabilidad a priori de clase, $p = 0{,}25$
RF	Número de árboles, $nt = 50$

Tabla 11.1: Configuraciones de cada grupo complejo considerado.

Para obtener una medida significativa del rendimiento de cada grupo, se ha repetido el proceso de entrenamiento de las redes y la construcción de los grupos 100 veces. En este caso se han entrenado las redes utilizando un proceso de ajustado de pesos con una tasa de aprendizaje inicial de 0.001, actualizando y bajando la tasa de aprendizaje cada número de épocas específico. Los parámetros que se han configurado son el tamaño de la mini-carga (mini-batch) para cada iteración de entrenamiento, que se denota por MiniBatchSize y el número máximo de épocas a utilizar para el entrenamiento, que se denota por MaxEpochs. Los posibles valores de estos parámetros son MaxEpochs = $\{50, 100\}$ y MiniBatchSize = $\{100, 256\}$. Los restantes valores de los parámetros de la red neuronal convolucional se pueden encontrar en la Tabla 11.2.

11.4.4. Resultados

El rendimiento de cada método se muestran en las Tablas 11.3 y 11.4. Se puede observar que los grupos producen una mejor clasificación que las redes individuales, con un rendimiento mayor con más de un punto de diferencia, en algunos casos incluso más de dos puntos.

Si se analizan las diferentes estrategias de grupo, muestran una exactitud similar entre ellas. Sin embargo, podría decirse que el grupo de bosques aleatorios (RF Ensemble) es el mejor y que el grupo de árboles de decisión (DT Ensemble) es el peor de los grupos analizados de acuerdo a los experimentos ejecutados.

Además, los tests muestran que el rendimiento de una red es mayor cuando se utiliza un mayor valor de los parámetro de tamaño de la mini-carga

[3] `https://es.mathworks.com/matlabcentral/fileexchange/55946-deep-multilayer-perceptron-neural-network-with-back-propagation`

[4] `https://es.mathworks.com/matlabcentral/fileexchange/62061-multi-class-svm`

Parámetro	Valor
Máximo número de épocas (MaxEpochs)	$\{50, 100\}$
Tamaño de la mini-carga (MiniBatchSize)	$\{100, 256\}$
Función que resuelve el entrenamiento de la red	Optimizador del gradiente estocástico descendiente con momento (Stochastic gradient descent with momentum optimizer o SGDM)
Momento	0,9
Tasa de aprendizaje inicial	0,001
Bajada de la tasa de aprendizaje durante el entrenamiento	Actualización de la tasa de aprendizaje cada periodo de bajada de la tasa de aprendizaje multiplicado por el factor de bajada de la tasa de aprendizaje
Factor de bajada de la tasa de aprendizaje	0,1
Periodo de bajada de la tasa de aprendizaje	$\frac{maxEpochs}{3}$
Decaimiento de peso	0,004

Tabla 11.2: Configuración del proceso de entrenamiento de las redes neuronales convolucionales.

y máximo número de épocas (MiniBatchSize y MaxEpochs), produciendo mejores grupos. Sin embargo, la mejora de exactitud de un grupo no es la misma que la mejora de las redes que lo componen.

Otro punto que debe ser destacado es el uso del proceso de superresolución. Se puede apreciar que el entrenamiento de las redes con un menor número de épocas utilizando superresolución produce peores redes que cuando no se usa el proceso de superresolución; aún produciendo los grupos un mejor rendimiento que los otros. La Tabla 11.5 muestra los rendimientos logrados de acuerdo a diferentes medidas para el mejor grupo en varias de las configuraciones probadas, mientras que la Figura 11.6 exhibe la matriz de confusión para dichas configuraciones.

Además, se ha estudiado el impacto del número de redes que componen el grupo en el rendimiento obtenido. De acuerdo a las informaciones de las Tablas 11.3 y 11.4, se ha ejecutado el entrenamiento con la configuración con el menor número de recursos para establecer cómo la superresolución afecta al rendimiento de los diferentes grupos. Es decir, se ha analizado la configuración los valores de parámetros MiniBatchSize=100 y MaxEpochs=50.

Parámetros	Superresolución	No				Sí				Media
	MiniBatchSize	100		256		100		256		
	MaxEpochs	50	100	50	100	50	100	50	100	
Métodos	Red	0,889	0,896	0,870	0,888	0,880	0,896	0,844	0,892	0,882
	Media	0,897	0,895	0,883	0,898	0,895	0,902	0,858	0,900	0,891
	Mediana	0,900	0,890	0,885	**0,902**	0,890	0,900	0,858	0,895	0,890
	Máximo	0,898	0,902	0,878	0,895	0,902	**0,908**	0,857	0,900	0,893
	Votos	0,903	0,892	**0,887**	0,898	0,892	0,898	0,857	0,893	0,890
	MLP	0,897	0,898	0,883	0,893	0,898	0,898	0,868	0,900	0,892
	SVM	0,897	0,903	0,872	0,898	0,903	0,900	0,868	0,898	0,893
	NB	**0,915**	0,902	**0,887**	0,900	0,902	0,907	**0,877**	**0,905**	**0,899**
	DT	0,878	0,898	0,857	0,875	0,898	0,885	0,852	0,898	0,880
	RF	0,895	**0,908**	0,882	0,900	**0,908**	0,905	0,863	0,900	0,895

Tabla 11.3: Valores de exactitud para cada método (mayor es mejor) para el video *SB*. La primera, segunda y tercera filas muestran la configuración de los parámetros utilizada en el entrenamiento de las redes que componen los grupos: el uso o no del proceso de superresolución, el tamaño de la minicarga a utilizar por cada iteración de entrenamiento y el máximo número de épocas a utilizar para el entrenamiento, respectivamente. La siguiente fila muestra la exactitud media producida por las redes que componen los grupos, y las filas restantes representan la exactitud de cada grupo analizado. La primera columna representa los parámetros analizados y los métodos, y las columnas restantes se corresponden al entrenamiento de los grupos utilizando los valores de los parámetros indicados. El mejor resultado de cada configuración se destaca en **negrita**.

Los resultados pueden observarse en la Figura 11.7. En cada subfigura, cada línea representa la exactitud de un grupo utilizando el número de redes que indica el eje x, es decir, el grupo de la i-ésima columna está compuesto por i CNNs. Se puede observar que con un número bajo de redes el rendimiento de los grupos varía de forma abrupta; sin embargo, en general, cada vez que se agregan más redes, el rendimiento del conjunto es mayor y se va ajustando. Por otro lado, añadir una red al conjunto, aunque el rendimiento de esta red fuera alto, no tiene por qué producir una mejora del rendimiento del conjunto.

Tras esto, se ha ejecutado un test de significación estadística para comparar el rendimiento de la clasificación de los dos mejores grupos de acuerdo a la Tabla 11.3 y una red básica con la misma configuración. Es decir, se ha comparado la red entrenada con la configuración Superresolución=No, MiniBatchSize=100 y MaxEpochs=50, y el grupo NB creado con esta configuración.

Parámetros	Superresolución	No				Sí				
	MiniBatchSize	100		256		100		256		
	MaxEpochs	50	100	50	100	50	100	50	100	Media
Métodos	Red	0,807	0,800	0,816	0,836	0,807	0,810	0,822	0,831	0,816
	Media	0,821	0,816	0,830	0,843	0,838	0,838	0,833	0,838	0,832
	Mediana	0,821	0,814	0,825	0,840	0,839	0,833	0,831	0,841	0,830
	Máximo	0,816	**0,820**	0,828	**0,851**	0,825	0,826	0,830	0,834	0,829
	Votos	0,819	0,814	**0,831**	0,840	0,836	0,833	0,828	0,836	0,830
	MLP	0,820	0,809	0,820	0,841	0,836	**0,843**	0,830	0,841	0,830
	SVM	0,819	0,818	0,814	0,839	0,841	0,838	**0,840**	0,838	0,831
	NB	0,799	0,800	0,813	0,829	0,814	0,809	0,824	**0,850**	0,817
	DT	**0,826**	0,806	0,804	0,838	0,831	0,805	0,825	0,835	0,821
	RF	0,823	0,816	0,825	0,836	**0,844**	0,835	0,839	0,848	**0,833**

Tabla 11.4: Valores de exactitud para cada método (mayor es mejor) para el video *Lankershim*. La primera, segunda y tercera filas muestran la configuración de los parámetros utilizada en el entrenamiento de las redes que componen los grupos: el uso o no del proceso de superresolución, el tamaño de la mini-carga a utilizar por cada iteración de entrenamiento y el máximo número de épocas a utilizar para el entrenamiento, respectivamente. La siguiente fila muestra la exactitud media producida por las redes que componen los grupos, y las filas restantes representan la exactitud de cada grupo analizado. La primera columna representa los parámetros analizados y los métodos, y las columnas restantes se corresponden al entrenamiento de los grupos utilizando los valores de los parámetros indicados. El mejor resultado de cada configuración se destaca en **negrita**.

Para completar el test de significación estadística, el primer paso es analizar si los valores de rendimiento producidos por ambas propuestas para las imágenes de test seleccionadas son distribuidos de forma gaussiana. Los valores correspondientes a la red se han elegido de la exactitud mediana de las redes utilizadas en la construcción de los grupos. Por otro lado, los valores correspondientes al grupo son su exactitud. Por tanto, se ha considerado una cantidad total de 100 valores (uno por repetición del proceso del entrenamiento de las redes y la construcción de los grupos).

Se ha seleccionado como test de normalidad el test de Lilliefors y su resultado informa que ambos conjuntos de datos no pueden suponerse que están distribuidos con una gaussiana, con un 5 % de nivel de significación. Por tanto, para comprobar la hipótesis nula de la igualdad de las medias de los valores de rendimiento de ambos métodos, se ha considerado el test no paramétrico de suma de rangos de Wilcoxon. Su resultado indica que hay diferencia significativa entre ambos métodos con un nivel de significación del

	Video	SB		Lankershim	
Parámetros	Superresolución	No	Sí	No	Sí
	MiniBatchSize	100			
	MaxEpochs	50			
Medidas	Exactitud (Acc)	0,9150	0,9083	0,8263	0,8438
	Factor Kappa (κ)	0,7733	0,7556	0,5367	0,5833
	Sensibilidad	0,9387	0,9287	0,8471	0,8630
	Especificidad	0,9744	0,9714	0,9443	0,9505

Tabla 11.5: Rendimiento de los mejores grupos con la configuración de parámetros MiniBatchSize=100 y MaxEpochs=50. La primera fila muestra la secuencia seleccionada; la segunda, tercera y cuarta filas muestran la configuración de parámetros utilizada en el entrenamiento de las redes que componen los grupos: el uso o no del proceso de superresolución, el tamaño de la mini-carga para utilizar en cada iteración de entrenamiento y el máximo número de épocas a utilizar para el entrenamiento, respectivamente. Las siguientes filas muestran (en media) la exactitud, el factor Kappa, la sensibilidad y la especificidad de cada grupo analizado. La primera columna representa los parámetros analizados y los métodos y las columnas restantes se corresponden con el mejor grupo utilizando los valores de los parámetros indicados para las secuencias indicadas, es decir, en este caso los grupos son NB, RF, DT y RF, respectivamente.

5 %, con un p-valor de 0,0244.

Además, también se ha comparado el rendimiento de los dos mejores grupos con Superresolución=No y Superresolución=Sí. Es decir, se ha seleccionado el grupo NB de acuerdo a la configuración Superresolución=No, MiniBatchSize=100 y MaxEpochs=50; y el grupo Máximo de acuerdo a la configuración Superresolución=Sí, MiniBatchSize=100 y MaxEpochs=100. Una vez más, el test de Lilliefors indica que las medias de los valores de rendimiento de ambos métodos no puede suponerse que estén distribuidos por una gaussiana, con un nivel de significación del 5 %. Sin embargo, el resultado del test no paramétrico de suma de rangos de Wilcoxon es que no hay diferencia significativa entre ambos métodos con un nivel de significación del 5 %, con un p-valor de 0,248.

Resumiendo, los grupos producen una mejora en el rendimiento en relación con el uso de una sola CNN. Además, el uso de un proceso de resolución en el redimensionado de las imágenes de baja resolución a las dimensiones requeridas por la CNN es adecuado cuando se necesita un breve tiempo de

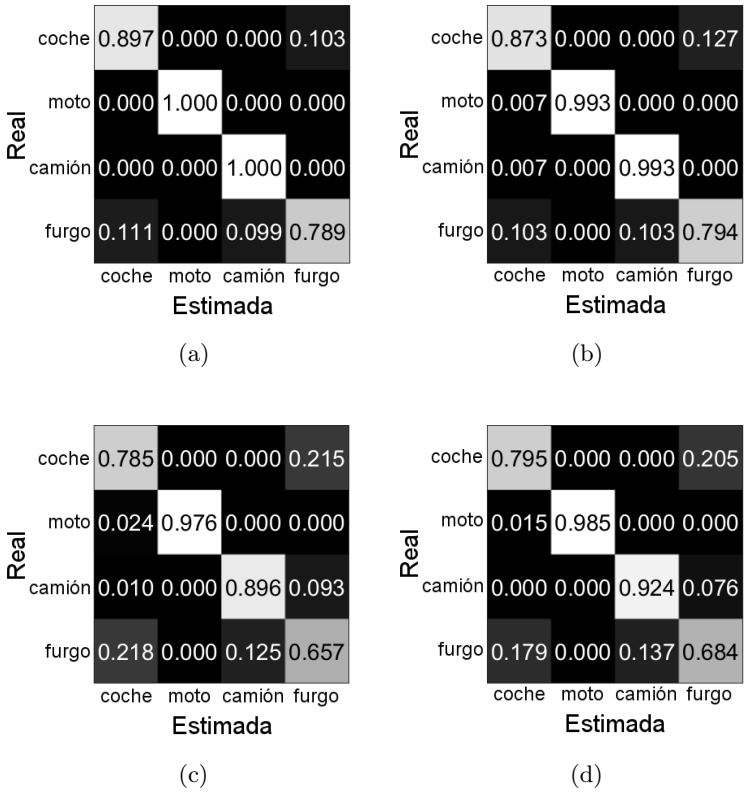


Figura 11.6: Matrices de confusión correspondientes a los mejores grupos con la configuración de parámetros MiniBatchSize=100 y MaxEpochs=50. (a) y (b) se corresponden con el video SB: (a) con la configuración Superresolución = No (grupo NB), mientras que (b) muestra Superresolución = Sí (grupo RF). Por su parte, (c) y (d) muestran el video Lankershim: (c) se corresponde con la configuración Superresolución = No (grupo DT) y (d) se refiere a Superresolución = Sí (grupo RF)

entrenamiento. Además, la superresolución proporciona un rendimiento similar sin diferencia significativa en los procesos de entrenamiento de redes con larga duración.

Por último, el mejor modelo de clasificación entrenado ha sido integrado en el sistema de videovigilancia de tráfico previamente descrito. Algunos resultados cualitativos se muestran en la Figura 11.8, donde la clasificación de vehículos puede ser observada en varios fotogramas de test.

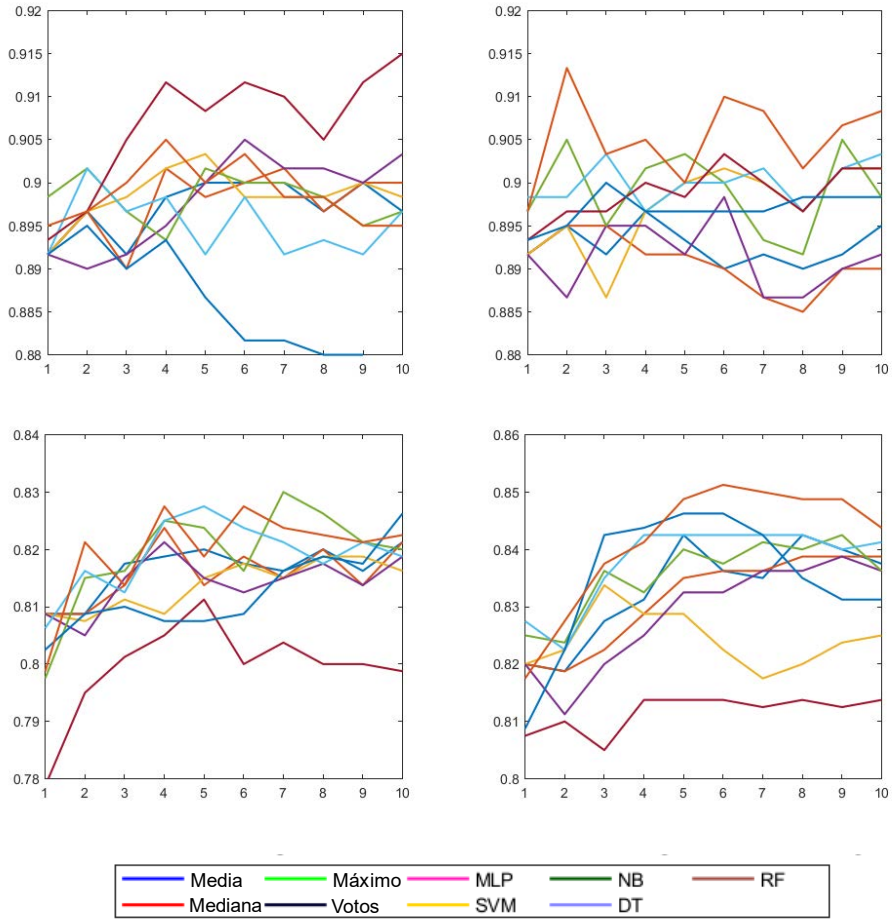
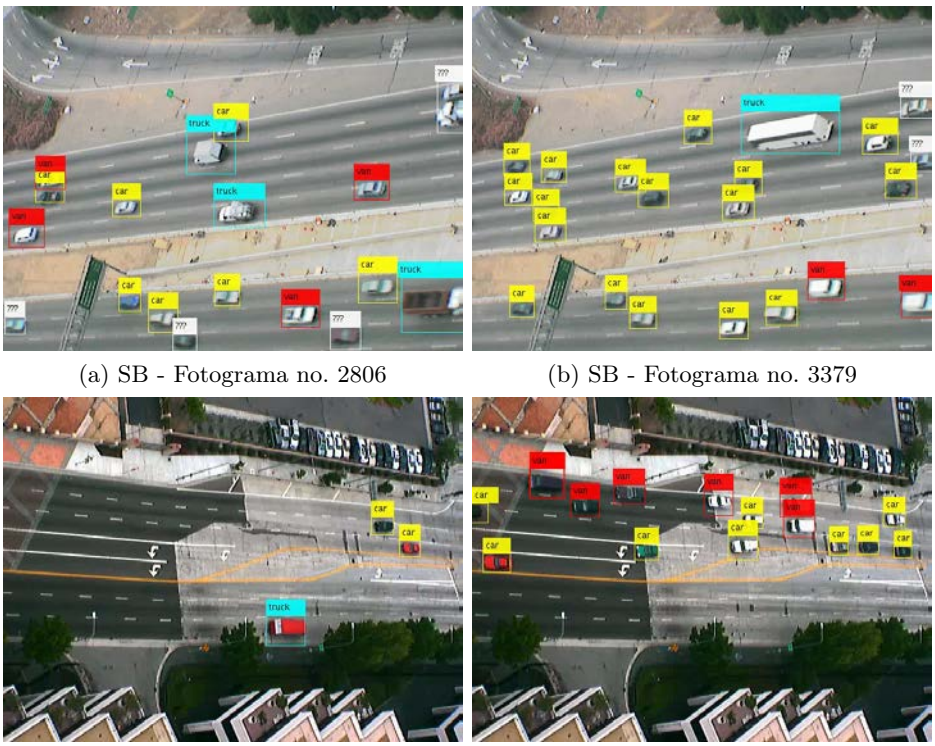


Figura 11.7: Rendimiento de los grupos estudiados de acuerdo al número de redes utilizadas en su construcción con la configuración de parámetros MiniBatchSize=100 y MaxEpochs=50. Las imágenes de la izquierda se corresponden a la configuración de entrenamiento de la red con Superresolución = No, mientras que las imágenes de la derecha se corresponden con la configuración Superresolución = Sí. Las imágenes de la primer fila se corresponden con la secuencia *SB*, y la segunda fila se corresponde con el video *Lankershim*. En cada figura, el eje x se corresponde con el número de redes utilizadas por el grupo, mientras que el eje y representa la exactitud.

11.5. Conclusiones

Se ha desarrollado una propuesta empleando un modelo de red Alexnet, que es una red neuronal convolucional de aprendizaje profundo, para clasificar los vehículos presentes en secuencias de tráfico. El sistema asigna cada vehículo detectado a una clase: coche, moto, camión o furgoneta. Debido a



(a) SB - Fotograma no. 2806 (b) SB - Fotograma no. 3379

(c) Lankershim - Fotograma no. 491 (d) Lankershim - Fotograma no. 570

Figura 11.8: Clasificación de vehículos de las secuencias *SB* y *Lankershim*.

que la red AlexNet necesita un conjunto de imágenes de entrenamiento donde todas las imágenes tengan el mismo tamaño, por lo que se han considerado diferentes propuestas de redimensionado de regiones para transformar el conjunto de datos para adaptar el tamaño de sus imágenes al tamaño requerido. Se ha estudiado una comparativa cuantitativa y los resultados demuestran que el método del centrado escalado es la mejor propuesta con una alta exactitud. Además, se han analizado diferentes estrategias de grupo usando varias CNNs que aplican el método del centrado escalado, con resultados prometedores. Debido a la baja resolución del conjunto de datos de entrenamiento y la ligera diferencia visual entre las clases coche y furgoneta, se ha estudiado el impacto del proceso de superresolución en el rendimiento de la clasificación. Los resultados indican que su uso es apropiado cuando se necesita un breve tiempo de entrenamiento. Por último, se ha integrado el mejor clasificador en el sistema de seguimiento de vehículos desarrollado anteriormente con una estrategia en línea para completar un sistema de clasificación de vehículos en tiempo real.

Capítulo 12

Clasificación de vehículos mediante un gas neuronal creciente

Lo importante es no dejar de hacerse preguntas.

Albert Einstein

Resumen: La supervisión del tráfico es una de las aplicaciones más populares de la videovigilancia automatizada. La clasificación de vehículos en tipos es importante para proporcionar información actualizada sobre las características del flujo de tráfico a los controladores humanos de tráfico. En este capítulo se propone un sistema de videovigilancia para ejecutar esta clasificación. Primeramente se ejecuta un proceso de extracción de características para obtener aquellos rasgos más significativos de los vehículos detectados. Tras esto, se emplea un conjunto de redes de gas neuronal creciente para determinar sus tipos. También se lleva a cabo una evaluación cualitativa y cuantitativa de la propuesta sobre unas secuencias de video de tráfico que sirven de referencia, con resultados favorables.

12.1. Introducción

Como ya se comentó en el capítulo anterior, el campo de la supervisión de tráfico ha generado gran expectación en los últimos años dentro de la comunidad de sistemas de transporte inteligente debido al incremento del desarrollo de hardware, el bajo coste de las tecnologías de sensores y la

mejora en el desarrollo y optimización de los algoritmos de procesado de datos. Específicamente, la detección de video y las soluciones de supervisión para aplicaciones de tráfico pueden ayudar a mejorar el rendimiento de la gestión del tráfico (Luque-Baena et al., 2015b; Kamijo et al., 2000; Cheng y Hsu, 2011). Por tanto, por ejemplo, si una alta frecuencia de vehículos pesados es detectada en una de las secciones de la carretera analizada, es posible redirigir el tráfico en un punto previo con el objetivo de evitar la congestión de tráfico.

Tal y como se describió en la Subsección 11.1, los sistemas de videovigilancia automática pueden dividirse en varias fases (Buch et al., 2011; Baumann et al., 2008). Un primer paso involucra la detección de los objetos en movimiento dentro de la escena; una segunda etapa realiza la supervisión de tareas para asociar el mismo vehículo detectado en todos los fotogramas de la secuencia en que aparece; y por último, una fase de detección de características para extraer conocimiento del movimiento de esos objetos, su comportamiento y apariencia.

Cada etapa construye sobre la anterior, lo que implica que se necesita tener implementadas las etapas de detección y seguimiento de objetos si se desea analizar los vehículos detectados. En este trabajo se combinan sistemas de detección y seguimiento de objetos con otras técnicas para lograr un sistema de clasificación de vehículos. Específicamente, un red neuronal autoorganizada se aplica para agrupar los píxeles del fondo y del primer plano para detectar qué píxeles están en movimiento dentro de la escena(López-Rubio et al., 2011b). Tras esto, se aplica un modelo de Kalman para múltiples objetos para determinar la trayectoria de cada vehículo que aparece en la escena (Rad y Jamzad, 2005).

Por tanto, el objetivo de este trabajo es clasificar los vehículos detectados en cuatro categorías: coche, moto, camión y furgoneta. Un proceso de extracción de características se requiere para obtener rasgos robustos y discriminantes que puedan diferenciar correctamente entre los grupos de vehículos. Este análisis ayudaría a gestionar y distribuir el tráfico más eficientemente en el área analizada. Otros trabajos de la literatura tienen el mismo propósito, aunque algunos aplican diferentes métodos de clasificación o agrupamiento (Crouzil et al., 2016; Huang et al., 2016) o empiezan desde otras metodologías asociadas a los sistemas de videovigilancia (Liang et al., 2015). En este caso, se considera el modelo de gas neuronal creciente (Growing Neural Gas model o GNG), ya que ha sido utilizado en diferentes problemas de clasificación, desde detección de novedades (Fink et al., 2015) a clasificación de texto (Wang y Shen, 2007), o incluso tareas relacionadas con la medicina o la biología como la detección de osteoporosis (Podolak y Jastrzebski, 2013).

12.2. Modelo

Esta sección describe un sistema de clasificación multiclase consistente en una combinación de varios modelos autoorganizados basados en el gas neuronal creciente.

12.2.1. Gas neuronal creciente

El gas neuronal creciente (Fritzke, 1995) es una red neuronal no supervisada adecuada para la cuantificación vectorial y el agrupamiento debido a su capacidad de encontrar una estructura topológica que refleje la topología de la distribución de entrada.

Sea $X = \{\vec{x} \in \mathbb{R}^d\}$ un conjunto de M patrones de entrenamiento en un espacio de d dimensiones. Sea GNG un modelo de gas neuronal creciente compuesto de N neuronas. Inicialmente, el GNG empieza colocando $N = 2$ neuronas aleatoriamente en el espacio de entrada, que se suponen vecinas en la estructura topológica y, por tanto, conectadas por un eje.

Durante la fase de entrenamiento, la topología es modificada de dos maneras. Primero, los centroides de los grupos que las neuronas representan $\vec{w}_i$ cambian para adaptar el mapa autoorganizado que el GNG está creando a la distribución de datos de entrada. Para este propósito se utiliza la regla de aprendizaje competitivo. Esta regla dice que en una iteración de entrenamiento t solo la neurona que mejor representa el patrón de entrenamiento presentado actualmente $\vec{x}_t$, es decir, la neurona cuyo centroide es más cercano a él, puede modificar sus propios datos internos, que en el caso de una red GNG son el centroide de la neurona $\vec{w}_{win}$ y una variable $error_{win}$ que contiene el error acumulado que sería obtenido si se ejecuta un vector de cuantificación.

$$win(t) = argmin_{1 \leq j \leq N}\{\|\vec{x}_t - \vec{w}_j(t-1)\|^2\} \tag{12.1}$$

$$\vec{w}_i(t) =$$

$$\begin{cases} \vec{w}_i(t-1) + \eta_{win}(t)\ (\vec{x}_t - \vec{w}_i(t-1)) & \text{if } i = win(t) \\ \vec{w}_i(t-1) & \text{otro caso} \end{cases} \tag{12.2}$$

$$error_{win}(t) = error_{win}(t-1) + \|\vec{w}_{win}(t-1) - \vec{x}_t\|^2 \tag{12.3}$$

donde $\|\cdot\|$ es la norma euclídea.

Para permitirle al mapa que autoorganice y mantenga la forma de la distribución de entrada que ha capturado, las neuronas que son vecinas de la neurona ganadora también actualizan un poco su posición del centroide. En

este caso se utiliza una tasa de aprendizaje más pequeña η_{neigh} para aquellas neuronas adyacentes a la ganadora. Por tanto, este nuevo caso es añadido a la Ecuación 12.2 y la ecuación final para modelar la actualización de la posición del centroide es

$$\vec{w}_i(t) = \vec{w}_i(t-1) +$$

$$\begin{cases} \eta_{win}(t)\ (\vec{x}_t - \vec{w}_i(t-1)) & \text{si } i = win(t) \\ \eta_{neigh}(t)\ (\vec{x}_t - \vec{w}_i(t-1)) & \text{si } i \in Neighbors(win(t)) \\ 0 & \text{en otro caso} \end{cases} \tag{12.4}$$

donde $\eta_{win} : \mathbb{N} \longrightarrow [0,1]$ y $\eta_{neigh} : \mathbb{N} \longrightarrow [0,1]$ son dos constantes o funciones de decrecimiento monótono que satisfacen $\forall t \in \mathbb{N} \quad \eta_{win}(t) > \eta_{neigh}(t)$

Segundo, durante la fase de entrenamiento se crean nuevas neuronas periódicamente y las neuronas inútiles, también conocidas como neuronas muertas, son eliminadas del mapa autoorganizado. Ésta es una de las ventajas de las redes neuronales crecientes como el GNG: el número de neuronas que son parte del modelo no tiene por qué ser fijado a priori por el usuario. Sin embargo, se puede definir un máximo número de neuronas y previene al GNG de crear demasiadas neuronas.

Cada λ iteraciones una nueva neurona es insertada en el mapa autoorganizado. La selección del lugar donde insertar la neurona se basa en el rendimiento de las neuronas existentes. Aquellas neuronas con un alto error acumulado rinden pobremente porque el grupo que representan es heterogéneo o contiene muchos más elementos que otros grupos. Por tanto, el centroide de la nueva neurona será colocado en el punto medio del eje que conecta la neurona u con el error más grande y la neurona v que es la vecina de u con el error acumulado más grande.

$$\vec{w}_{new} = \frac{\vec{w}_u + \vec{w}_v}{2} \tag{12.5}$$

Como *new* está entre u y v, el eje que conecta ambas es reemplazado por dos nuevos ejes: uno de u a *new* y otro desde *new* a v. La nueva neurona se asume que representa algunos de los patrones que pertenecían previamente a los grupos correspondientes a u y v, por lo que el error acumulado de las tres neuronas es actualizado convenientemente.

$$error_u(t) = \alpha \cdot error_u(t) \tag{12.6}$$

$$error_v(t) = \alpha \cdot error_v(t) \tag{12.7}$$

$$error_{new}(t) = error_u(t) \tag{12.8}$$

donde $\alpha \in [0, 1]$ se considera la fracción estimada de error acumulado que es reducida tras insertar la nueva neurona.

Por otro lado, los ejes entre las neuronas tienen una variable asociada *age* que es incrementada a medida que el entrenamiento avanza. La edad de los ejes que conectan neuronas que siguen ganando la competición, es decir, aquellas que no están muertas, son refrescadas y puestas a 0 (ver Subsección 12.2.1.1). Aquellos ejes cuya edad es mayor que un umbral dado a_{max} son eliminadas, ya que las neuronas que ellos conectan no ganaron la competición recientemente. Si algunas neuronas llegan a estar aisladas después de eliminar ejes, se consideran muertas y también son eliminadas del mapa.

12.2.1.1. Algoritmo de entrenamiento del GNG

1. La red es inicializada creando dos nodos posicionados aleatoriamente que están conectados por un eje. Sus errores acumulados son ajustados a 0.

2. En el instante de tiempo t, seleccionar aleatoriamente un vector $\vec{x}_t$ que no ha sido presentado previamente a la red, si es posible.

3. Determinar la neurona ganadora *win* usando la ecuación 12.1 y también la neurona subcampeona *rup* con un vector de referencia $\vec{w}_{rup}$ de forma que $\|\vec{x}_t - \vec{w}_{rup}(t-1)\|^2$ es la segunda más pequeña, para todas las neuronas N.

4. Actualizar el error acumulado de la neurona ganadora por medio de la ecuación 12.3.
   Actualizar la topología de la red usando la ecuación 12.4.

5. La edad de todos los ejes que conectan *win* con sus vecinos topológicos son incrementadas en uno.
   Si *win* y *rup* están conectadas por un eje, la edad de ese eje es puesta a 0. En otro caso, un nuevo eje es creado entre ellas.

6. Los ejes con edad mayor que a_{max} son eliminados. Si esto conlleva que algunas neuronas queden sin conexiones, aquellas neuronas aisladas son eliminadas.

7. En caso de que máximo número de nodos no se haya alcanzado y $\{\exists j \in \mathbb{N} - \{0\} \mid t = j \cdot \lambda\}$ entonces se crea una nueva neurona.

   a) Se determina la neurona u con el error más grande, además de su neurona vecina v con el error más largo. La nueva neurona es insertada entre ellas (ver ecuación 12.5)

b) El eje entre u y v es eliminado y dos nuevos ejes son añadidos, de new a u, y de new a v.

c) Los errores acumulados para las neuronas u, v y new son actualizados siguiendo las ecuaciones 12.6, 12.7 y 12.8.

8. Los errores acumulados son decrementados por un factor β

$$\forall i \in [1..N] \quad error_i(t) = error_i(t) - \beta \cdot error_i(t) \tag{12.9}$$

9. Si los requisitos para finalizar no han sido alcanzados, es decir, el máximo número de pasos de entrenamiento no ha sido alcanzado, ir al paso 2.

12.2.2. Sistema de clasificación

El sistema de clasificación multiclase propuesto está basado en una método de uno contra todos, que involucra el entrenamiento de una red neuronal GNG para cada una de las C clases a las que los patrones pueden pertenecer.

$$Clasificador = \{GNG_i, \quad 1 \leq i \leq C\} \tag{12.10}$$

Dado un patrón $\vec{x} \in \mathbb{R}^d$ es asignado a la clase i correspondiente al GNG que tiene la neurona más cercana a ese patrón en el espacio de entrada.

$$clase(\vec{x}) = argmin_{1 \leq i \leq C}\{\|\vec{x} - \vec{w}^i_{winner}\|^2\} \tag{12.11}$$

donde $\vec{w}^i_{winner}$ es la neurona ganadora (ver ecuación 12.1) de la red neuronal GNG GNG_i, que es entrenada para detectar patrones que pertenecen a la clase i.

El máximo número de neuronas que forman los diferentes mapas de autoorganización han sido ajustados al mismo valor, $N = N_{max}$. De esta forma ninguna de las redes particulares GNG_i tiene una ventaja durante la fase de entrenamiento.

12.3. Resultados experimentales

En esta sección se presentan los resultados que se han obtenido de los tests. La secuencia seleccionada para testear la propuesta es el video de tráfico denominado *US-101 Highway* que está disponible en el conjunto de datos del programa Next Generation Simulation (NGSIM), proporcionado por Federal Highway Administration (FHWA). Esta secuencia presenta varias dificultades que deberían tratarse, como una perspectiva de una escena de exterior, objetos que se solapan u oclusiones.

El conjunto de datos, que contiene información sobre las trayectorias de varios vehículos detectados en una secuencia de video, presenta una pequeña

Figura 12.1: Diferentes vehículos y sus trayectorias detectadas por la propuesta. De arriba abajo: vehículo 422 corresponde a la clase moto, la trayectoria 408 es una furgoneta, la 2776 es un camión y la 426 es un coche.

cantidad de vehículos etiquetados y una gran cantidad de ellos sin etiquetar. Por tanto, la propuesta trata de clasificar los vehículos que aparecen en la secuencia en 4 posibles clases: moto, coche, furgoneta y camión. Un ejemplo de estas diferentes clases se pueden observar en la Figura 12.1, que muestra 4 vehículos y sus correspondientes trayectorias, un vehículo por cada clase.

Entre las distintas características que pueden ser extraídas tras la segmentación de una imagen, las seleccionadas para los test han sido el área, el perímetro, la anchura y el alto de cada objeto. Características como la posición del objeto en la escena son irrelevantes cuando se intenta identificar su tipo y no se toman en consideración. Por tanto, para cada objeto segmentado O_i un conjunto $\{\vec{x}_i^f \in \mathbb{R}^4 \mid f \in [1..MaxFrame]\}$ define su trayectoria durante la secuencia de video. Para hacer funcionar el clasificador propuesto de forma adecuada, se hace necesario elegir un representante de cada objeto. La descripción del objeto elegido $\vec{x}_i^f$ es aquella cuyo área coincide con la mediana de los valores de área en los miembros de O_i. Debido a la robustez de la mediana estadística, aquellas descripciones de objetos correspondientes a los fotogramas en los que la segmentación ha fallado o varios objetos se solapan nunca son seleccionados porque están lejos del valor de la mediana. Además, solo las trayectorias O_i con un cardinal mínimo han sido consideradas, en un intento de evitar problemas de solapamiento de objetos.

La metodología de mapas autoorganizados que se ha usado está compuesta de 4 redes neuronales, una por clase, que representa cada tipo de vehículo

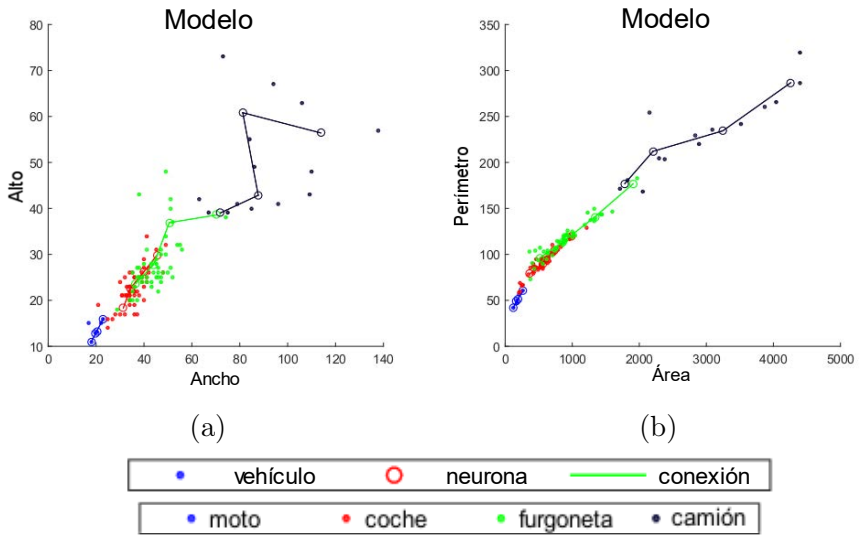


Figura 12.2: Conjunto de entrenamiento y la red neuronal modelada utilizando 4 neuronas por clase. La columna (a) muestra los datos ordenados por área y perímetro, mientras que la columna (b) muestra los datos ordenados por ancho y alto.

considerado. Cada red consiste de 4 neuronas. Este número de neuronas es bajo porque la propuesta está basada en una estrategia de aprendizaje en línea, es decir, se logra un funcionamiento en tiempo real.

Para comprobar la bondad de la propuesta se han separado los datos etiquetados en dos grupos aleatorios: el primero con el 90 por ciento de los datos es usado para entrenar el modelo, y el segundo con el restante 10 por ciento de los datos es utilizado para comparar las etiquetas de tipo de los vehículos con las predichas por la propuesta. La división de los grupos se ha ejecutado aplicando una selección aleatoria por grupos. Es decir, cada grupo tiene el mismo número de objetos.

Así, el proceso para obtener una clasificación de los objetos es como sigue. Primero, los datos etiquetados se dividen en 2 grupos: datos de entrenamiento y datos de test. Después, se crea el modelo y se entrena con los datos de entrenamiento. Tras esto, el modelo es probado con los datos de test. Por último, los datos no etiquetados son clasificados. Este proceso se ha ejecutado 10 veces.

La distribución de un modelo con su modelo de neuronas de cada clase se puede observar en la Figura 12.2: (a) muestra la distribución organizada por área y perímetro, y (b) muestra la misma distribución organizada por ancho y alto.

Desde un punto de vista cualitativo, los resultados de clasificación producidos por el modelo implementado se muestran en la Figura 12.3. Los

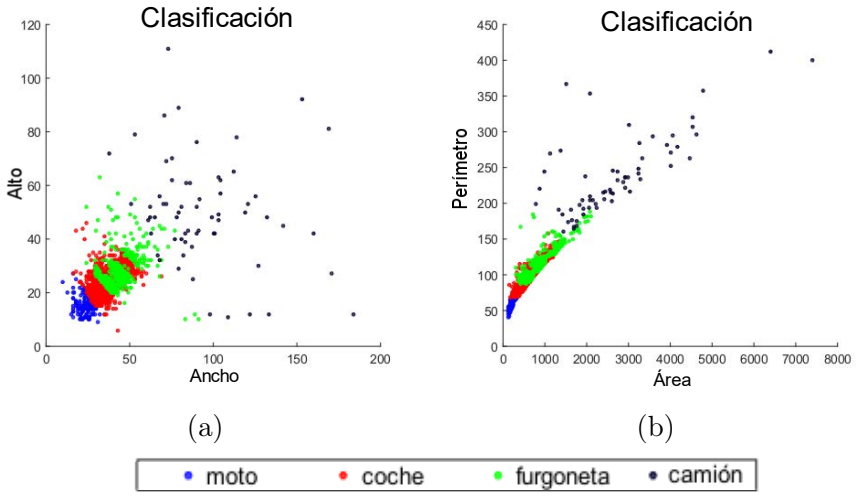


Figura 12.3: Clasificación producida por el modelo de la Figura 12.2. La columna (a) muestra los datos ordenados por área y perímetro. La columna (b) muestra los datos ordenados por ancho y alto.

resultados organizados por área y perímetro de cada vehículo se muestran en (a), mientras que la misma clasificación organizada por ancho y alto se muestra en (b).

Por otro lado, los resultados cuantitativos del rendimiento de la propuesta se observan en la Tabla 12.1. En ella se ven varias medidas bien conocidas como la exactitud (Accuracy o Acc) o el error cuadrático medio (Mean Square Error o MSE). La exactitud es un valor entre 0 y 1, donde mayor es mejor. Por su parte, el error cuadrático medio es un número positivo real, donde menor es mejor.

Sea k el objeto observado de los K objetos existentes, sean $\mathbf{x}_k$ y $\mathbf{w}_k$ las clases real y predicha del objeto, respectivamente, donde $\mathbf{x_k}, \mathbf{w_k} \in \{1, 2, 3, 4\}$, correspondiente con $1 = moto, 2 = car, 3 = van$ y $4 = truck$. Además, sea $\mathbf{q}_k = 1$ si el modelo acierta la clasificación del objeto k (por lo que $\mathbf{x_k} = \mathbf{w_k}$) y $\mathbf{q}_k = 0$ si el modelo falla ($\mathbf{x_k} \neq \mathbf{w_k}$):

$$Acc = \frac{1}{K} \sum_{k=1}^{K} \mathbf{q}_k \tag{12.12}$$

$$MSE = \frac{1}{K} \sum_{k=1}^{K} (\mathbf{x}_k - \mathbf{w}_k)^2 \tag{12.13}$$

Además, se han considerado otras medidas del rendimiento de clasificación (Moschou et al., 2007). El índice Rand (Rand Index, Jain y Dubes 1988) mide la similitud entre el grupo asociado a la correcta clasificación y el grupo asociado a las etiquetas predichas. Ofrece valores en el intervalo $[0, 1]$ (mayor

Medida	Mediana	Media	Mejor	Peor
Exactitud	0,7142	0,6785	0,8571	0,5000
Error cuadrático medio	0,2857	0,3214	0,1428	0,5000
Índice Rand	0,6978	0,7000	0,8021	0,6043
Estadístico gamma de Hubert	0,2470	0,2737	0,5153	0,0739
Entropía del grupo total	0,5228	0,5290	0,3718	0,7305
Entropía de la clase total	0,5252	0,5294	0,3718	0,6364
Entropía total	0,5407	0,5292	0,3718	0,6834

Tabla 12.1: Medidas cuantitativas de los resultados. Cada fila es una medida y cada columna representa la mediana, la media, el mejor y el peor resultado para cada medida, respectivamente.

es mejor), donde 1 indica una clasificación perfecta. El estadístico gamma de Hubert (Hubert's Gamma Statistic, Jain y Dubes 1988) tiene valores entre -1 y 1 (mayor es mejor), donde 1 significa una correlación perfecta entre las etiquetas reales y las predichas. Por último, la entropía del grupo total, la entropía de la clase total y la entropía total (He et al., 2004) miden la información sobre las etiquetas correctas que está contenida en las etiquetas predichas. Estas tres medidas tienen valores en el intervalo $[0, 1]$, donde menor es mejor.

Los resultados obtenidos están influenciados por el proceso de segmentación (donde se detectan los vehículos), y la etapa de seguimiento (se calcula la trayectoria de cada vehículo). Se aprecían algunos problemas de solapamiento, especialmente los camiones por su mayor tamaño, y esto causa errores en la clasificación. Otro aspecto a destacar es el bajo número de objetos etiquetados, particularmente motos.

12.4. Conclusiones

Este capítulo propone un nuevo método basado en el gas neuronal creciente (Growing Neural Gas o GNG) para determinar los tipos de los vehículos que aparecen en escenas de tráfico. Los vehículos detectados son clasificados en cuatro categorías (coches, motos, camiones y furgonetas) por el sistema, basado en un proceso de extracción de características que proporciona los datos de entrada para ejecutar la clasificación.

La arquitectura neuronal propuesta está compuesta por cuatro GNGs, es

decir, cada GNG representa uno de los tipos de vehículos considerados. Cada vehículo es clasificado en la clase asociada al GNG que mejor representa las características del vehículo. Por tanto, el sistema de clasificación multiclase propuesto está basado en una propuesta de uno contra todos, que involucra los cuatro GNGs. Debido a los requisitos de tiempo real se ha utilizado una estrategia de aprendizaje en línea para el entrenamiento de los cuatro GNGs, cuyo número de neuronas es limitado por esta misma razón.

Los resultados ofrecidos por la simulación de experimentos en el video muestran que la propuesta consigue unos resultados de clasificación satisfactorios. El rendimiento de la propuesta depende de los procesos de segmentación y seguimiento. Las evaluaciones cualitativas y cuantitativas de estos resultados muestran que la propuesta rinde con un alto nivel de detección de vehículos, mientras que el rendimiento del proceso de clasificación es satisfactorio.

Capítulo 13

Estimación de la contaminación en carreteras

La verdad es demasiado complicada como para permitir nada más allá de meras aproximaciones.

John von Neumann

Resumen: En este capítulo se presenta una metodología para estimar la contaminación en carreteras analizando secuencias de video de tráfico. El objetivo es tomar ventaja de la gran red de cámaras estáticas que es posible encontrar en el sistema de carreteras de cualquier estado o país para estimar la contaminación en cada área. Esta propuesta utiliza redes neuronales de aprendizaje profundo para la detección de objetos, y un modelo de estimación de la contaminación basado en la frecuencia de vehículos y su velocidad. Los experimentos muestran resultados prometedores que sugieren que el sistema puede ser utilizado solo o combinado con otros sistemas existentes para medir la contaminación en carreteras.

13.1. Introducción

Actualmente, los últimos avances en investigación relaciona con el campo de la videovigilancia de tráfico, han permitido estudiar otros factores muy interesantes del análisis y detección de vehículos en una carretera. Además, el crecimiento en la implementación y uso de cámaras IP, principalmente por razones de seguridad, está generando una gran cantidad de información que podría analizar el comportamiento normal de los vehículos, detectar patrones

anómalos (por ejemplo, conducción en la dirección opuesta) o estimar la contaminación del aire en entornos de tráfico.

La evaluación de la contaminación del aire causada por las emisiones de los vehículos y la previsión de la calidad del aire han sido gestionadas desde diferentes puntos de vista (Yang et al., 2016). Una de las propuestas consiste en medir la concentración de aire producida por el tráfico con sensores de supervisión, aunque no es altamente adecuado para supervisar áreas grandes debido al coste de la instalación y aplicación de los sensores. En Wang et al. 2015 los autores deciden determinar la contaminación del tráfico en las intersecciones de carreteras utilizando modelos híbridos que combinan redes neuronales de pequeña onda y algoritmos genéticos. Además, en Lozhkina y Lozhkin 2015 se analiza una comparativa entre dos modelos diferentes de emisión y dispersión.

A diferencia de otros modelos que estiman la contaminación del tráfico basada en sensores de calidad del aire, sensores ambientales y sensores que determinan la densidad del tráfico, solo las cámaras estáticas presentes en las carreteras se utilizan en este trabajo. La propuesta intenta estimar de forma optimista el nivel de contaminación del tráfico del análisis de vehículos en movimiento en las carreteras. Para hacer esto, en cada fotograma de la secuencia, se detecta el número de vehículos circulando y su velocidad. Con esta información, es posible estimar el nivel de contaminación producida en cada instante de tiempo.

Al igual que ocurre con lo descrito en el Capítulo 11, la metodología propuesta empieza con una fase de detección de los vehículos que aparecen en la escena (Bouwmans, 2014b). Debido a la mejor en la fuerza de los dispositivos hardware, el reciente desarrollo de las técnicas de aprendizaje profundo (que permiten abordar complejas tareas como el reconocimiento de objetos) está siendo incorporado progresivamente en el campo de la videovigilancia de tráfico (Xue et al., 2016). De hecho, muchas técnicas tradicionales para la detección de objetos en primer plano (mixtura de gaussianas (Zivkovic y van der Heijden, 2006), modelado estadístico del fondo (Luque et al., 2010), etc.) están siendo reemplazadas por redes neuronales profundas, que proporcionan tasas de acierto mucho más altas en la identificación y detección de objetos (Lecun et al., 2015). En este capítulo se utiliza la red Faster-RCNN para reconocer los vehículos en la escena (Ren et al., 2017). Tras esto, se considera una etapa de seguimiento para obtener las trayectorias parciales a lo largo de la carretera (Yilmaz et al., 2006).

La perspectiva de la cámara hace difícil el cálculo de la velocidad de cada vehículo. Es por ello por lo que se aplica una red neuronal autoorganizada para modelar la distribución de los vehículos y su tamaño para corregir la perspectiva. Utilizando la velocidad calculada y el número de vehículos es posible estimar la contaminación en cada fotograma, y consecuentemente, en la escena al completo.

El resto del capítulo es estructurado como sigue. La Sección 13.2 presenta la arquitectura de la propuesta, donde cada subsección describe cada parte en detalle. La Sección 13.3 muestra los resultados experimentales que se han llevado a cabo, mientras que la Sección 13.4 resume las conclusiones.

13.2. Arquitectura del sistema

La propuesta desarrollada puede ser descrita como se muestra en la Figura 13.1. Un fotograma de la secuencia de video es proporcionado al sistema, que está compuesto de tres módulos. El primero es un proceso de detección y seguimiento de los vehículos, donde se seleccionan los objetos deseados (vehículos en este caso) de los restantes objetos (personas, plantas y otros). El segundo es un módulo de estimación de la velocidad para calcular la velocidad de cada vehículo detectado. Y finalmente se puede estimar la contaminación producida por cada vehículo detectado considerando su velocidad con un módulo de estimación de la contaminación.

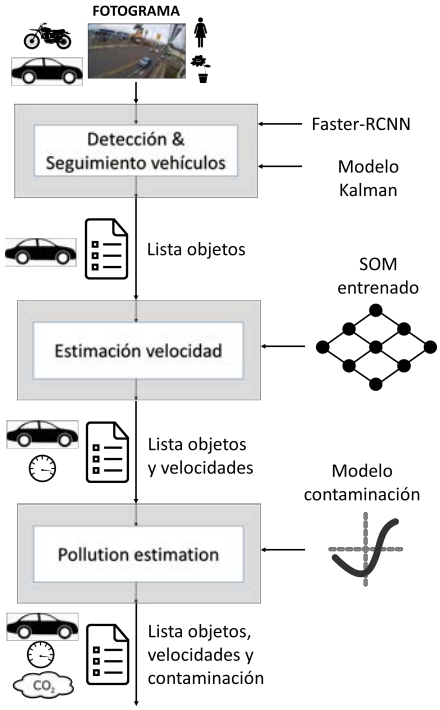


Figura 13.1: Esquema del funcionamiento de la propuesta.

13.2.1. Detección y seguimiento de vehículos

El módulo de detección y seguimiento de vehículos está compuesto de dos etapas. La primera es un proceso de detección y clasificación de objetos, y tras esto, se tiene un proceso de seguimiento para gestionar las trayectorias de los vehículos detectados.

El proceso de detección y clasificación de objetos está basado en una arquitectura de aprendizaje profundo. En este caso, se ha utilizado el modelo Faster-RCNN (Ren et al., 2017), que emplea redes neuronales convolucionales y proporciona el área y la clase de los objetos detectados. Se ha considerado un modelo preentrenado para detectar los 20 tipos de objetos incluidos en el conjunto de datos PASCAL VOC 2007 (Everingham et al., 2018).

Dada una imagen correspondiente al fotograma t de un video, la salida del modelo es un conjunto de objetos detectados, con su área y sus probabilidades de pertenencia a cada posible clase. Sea i uno de los objetos detectados en el fotograma t, la salida del módulo de detección de objetos es:

$$\mathbf{h}_{i,t} = (h_{i,t,1}, h_{i,t,2}, h_{i,t,3}, h_{i,t,4}) \in \mathbb{R}^3 \tag{13.1}$$

$$\mathbf{q}_{i,t} = (q_{i,t,1}, ..., q_{i,t,K}) \in \mathbb{R}^K \tag{13.2}$$

donde $(h_{i,t,1}, h_{i,t,2})$ son la esquina superior izquierda de la región rectangular mínima correspondiente al i-ésimo objeto detectado en el actual fotograma y $(h_{i,t,3}, h_{i,t,4})$ son el ancho y el alto, respectivamente, de esta región mínima, expresados en píxeles. Asociada a cada detección hay un probabilidad de pertenencia a cada clase, $q_{i,t,k} \in [0, 1]$, donde $C_k \in Classes$ y el posible número de clases de objetos es K. En este caso, $K = 20$.

Tras esto, en un fotograma t se aplica un umbral τ y solo se quieren considerar aquellos objetos con $q_{i,t,k}$ mayor que ese umbral, para considerar únicamente los vehículos que aparecen en el fotograma.

Una vez se han detectado los vehículos, se requiere una fase de seguimiento para obtener sus trayectorias parciales. Esta información es crucial para determinar la velocidad y contaminación en cada fotograma. Se aplica el filtro de Kalman para evaluar la correspondencia entre los objetos detectados en un fotograma y los objetos seguidos. Esta técnica está basada en un esquema de predicción - corrección de los centroides de los objetos a lo largo de la secuencia (Bar-Shalom, 1987).

13.2.2. Estimación de la velocidad

La mayoría de las cámaras localizadas en autovías capturan las imágenes con perspectiva, lo que causa que no haya homogeneidad en las distancias en cada parte del fotograma. Por tanto, para estimar las distancias reales

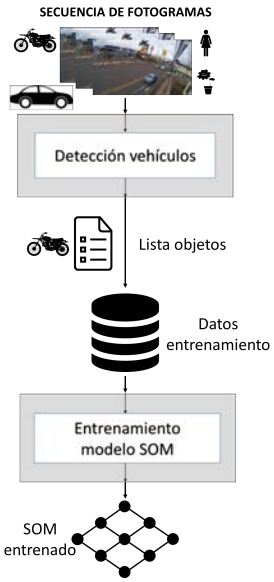


Figura 13.2: Esquema del proceso de entrenamiento del SOM.

en el escenario y la velocidad de los vehículos, se ha considerado un modelo de mapa autoorganizado (Self-Organizing Map o SOM) (Luque-Baena et al., 2015a). Se extrae un vector de características $\mathbf{z} \in \mathbb{R}^D$ de cada objeto detectado donde D es el número de rasgos elegidos. En este caso, la información geométrica representada por el área, y el alto y el ancho de su región rectangular mínima es suficiente para estimar las distancias en píxeles de cada vehículo. Estos valores forman el vector de características. Por tanto, el objetivo de la red es aprender una función de suavizado:

$$\mathcal{F} : \mathbb{R}^2 \rightarrow \mathbb{R}^D \quad \mathbf{y} = \mathcal{F}(\mathbf{x}) \tag{13.3}$$

donde $\mathbf{x}$ es una localización de píxel en el fotograma del video.

Las M unidades del mapa autoorganizado, que están dispuestas en una topología rectangular de tamaño $a \times b$, están representadas por dos prototipos, uno para los vectores de entrada $\mathbf{w} \in \mathbb{R}^2$ (coordenadas píxel) y otro para los vectores de salida $\mathbf{v} \in \mathbb{R}^D$ (características típicas del objeto en las coordenadas píxel $\mathbf{w}$). El prototipo de entrada es utilizado para calcular la unidad ganadora, mientras que el prototipo de salida es utilizado para estimar la función de suavizado $\mathcal{F}$. Por tanto, los vehículos que son asociados con la neurona i utilizarán la salida $\mathbf{v_i}$ para estimar la velocidad a la que están conduciendo en ese momento, y cuya localización dentro del fotograma será cercana a $\mathbf{w_i}$.

El entrenamiento del SOM está basado en el tamaño de las motos, ya que son muy parecidas entre sí de acuerdo a su relación de aspecto. En el caso de un coche, existe una mayor diferencia entre el coche más pequeño y

el más grande.

Primero, para generar el conjunto de datos de entrenamiento del SOM, se ha ejecutado el módulo de detección y clasificación de objetos con un video del escenario seleccionado. Para cada fotograma, se ha obtenido la lista de los objetos detectados con sus clases y regiones rectangulares mínimas (área y posición), y solo se han considerado aquellos objetos indicados como motos y con una alta probabilidad de pertenencia a la clase moto. Es decir, el objeto i en el fotograma t se considera que pertenece al conjunto de entrenamiento si:

$$\forall k \in K(\mathbf{q}_{i,t,k} \leqslant \mathbf{q}_{i,t,m}) \wedge (\mathbf{q}_{i,t,m} > \tau) \wedge (C_m = moto) \tag{13.4}$$

Además, un vehículo es descartado si una parte de su región rectangular mínima corresponde al borde la imagen.

Tras esto, con la región rectangular mínima de cada moto detectada se ha entrenado el modelo SOM, considerando un ancho estándar de una moto. Para estimar este valor, se puede seleccionar el número de motos con licencia por tamaño de motor. Después, se elige el modelo conocido que tiene más licencias actualmente con este tamaño de motor y se toma su ancho. Además, también se considera el alto de una persona conduciendo una moto. Por tanto, se puede estimar en cada área del fotograma la correspondencia entre metros y píxeles.

En la Figura 13.2 se puede observar un esquema del proceso de entrenamiento del SOM.

13.2.3. Estimación de la contaminación

Se ha considerado una estimación de la contaminación basada en el factor de emisión (emission factor o EF), que es una medida en unidades de g/km por PM_{10} y litro/100km de consumo de combustible. El modelo para estimar el factor de emisión está basado en las curvas del infome Production of Updated Emission Curves for Use in the National Transport Model, que está disponible en su página web [1]. Estas curvas están definidas por la siguiente ecuación:

$$y(x) = \frac{a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6}{x} \tag{13.5}$$

donde x es la velocidad en km/h.

Para estimar la contaminación producida por un coche, debido a que no se puede asegurar su tipo de combustible, se ha considerado la proporción de coches por tipo de combustible, gasolina o diésel (petrol_car_proportion y diesel_car_proportion, respectivamente), en autovías. Por tanto, el factor

[1] `https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/662795/updated-emission-curves-ntm.pdf`

de emisión producido por un coche, considerando las curvas de emisión para el tipo de combustible de vehículos de gasolina y diésel (y_p y y_d, respectivamente), se define como sigue:

$$\begin{aligned} EF(x) = petrol\_car\_proportion * y_p(x) \\ + diesel\_car\_proportion * y_d(x) \end{aligned} \tag{13.6}$$

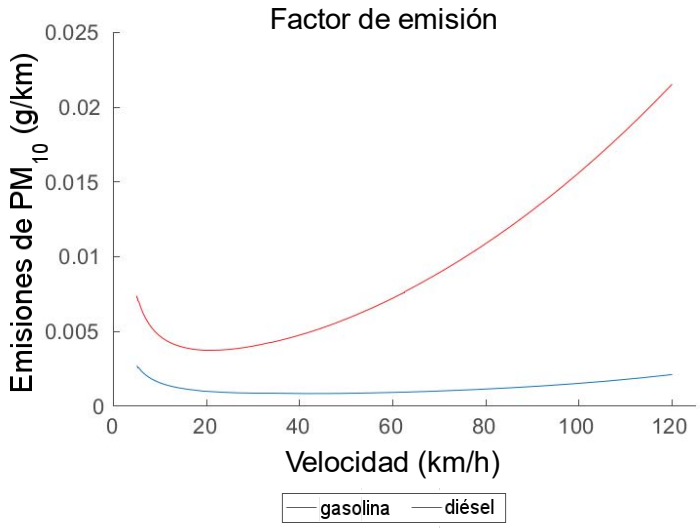


Figura 13.3: Curvas del factor de emisión para los tipos de coche gasolina y diésel correspondientes al módulo de estimación de contaminación.

13.3. Resultados experimentales

Los experimentos computacionales que se han ejecutado y sus resultados se muestran en esta sección. Primero, la Subsección 13.3.1 muestra el software y hardware utilizados. Después, en la Subsección 13.3.2 se han especificado las secuencias de video probadas. La configuración de los parámetros del software se puede observar en la Subsección 13.3.3. Por último, los resultados obtenidos en los experimentos se describen en la Subsección 13.3.4.

13.3.1. Métodos

La implementación del sistema se ha escrito en Matlab. Los módulos de estimación de la velocidad y de la contaminación han sido implementado por el grupo de investigación, mientras que el módulo de detección de vehículos

Tabla 13.1: Valores considerados de los parámetros.

Método	Parámetro
Faster RCNN	Umbral $\tau = 0{,}50$
	Modelo, $model\_dir$ = faster_rcnn_VOC0712_vgg_16layers
SOM	Número de pasos, $num\_steps = 100000$
	Número de pasos por época, $num\_steps\_per\_epoch = 10000$
	Número de neuronas, $num\_neurons = 25$
	Número de filas, $num\_rows\_map = 4$
	Número de columnas, $num\_cols\_map = 4$
	Tasa de aprendizaje inicial, $initial\_learning\_rate = 0{,}4$
	Radio máximo, $max\_radius = sqrt(num\_neurons)/8$
	Tasa de aprendizaje de convergencia, $convergence\_learning\_rate = 0{,}01$
	Radio de convergencia, $convergence\_radius = 1$
	Longitud normal de una moto, $usual\_moto\_lenght = 2{,}080$
	Longitud normal de una persona conduciendo,
	$usual\_person\_driving\_moto\_lenght = 1{,}700$
EF	Proporción de coches de gasolina en autovía, $petrol\_car\_proportion = 0{,}29$
	Proporción de coches diésel en autovía, $diesel\_car\_proportion = 0{,}71$
	$a\_petrol = 0{,}01185628$
	$b\_petrol = 0{,}00034047$
	$c\_petrol = 1{,}2576E-06$
	$d\_petrol = 1{,}0462E-07$
	$e\_petrol = -7{,}216E-10$
	$f\_petrol = 6{,}0976E-12$
	$g\_petrol = 0$
	$a\_diesel = 0{,}02918783$
	$b\_diesel = 0{,}0013909$
	$c\_diesel = 2{,}8984E-05$
	$d\_diesel = 6{,}175E-07$
	$e\_diesel = 9{,}9971E-09$
	$f\_diesel = -7{,}31E-11$
	$g\_diesel = 2{,}1786E-13$

está basado en la librería Faster R-CNN, que está disponible en su página web[2].

Los experimentos se han ejecutado en un ordenador personal de 64 bits con un microprocesador de ocho núcleos Intel i7 3,60 GHz, 32 GB RAM y una tarjeta gráfica Titan X.

[2] `https://github.com/ShaoqingRen/faster_rcnn`

13.3.2. Secuencias

Se ha considerado un escenario específico, que está disponible en la página web de *camaras viales*[3], para ejecutar los experimentos. El escenario seleccionado es *camara-guadalupe*, donde una cámara está grabando una carretera durante las 24 horas del día[4]. Se ha tomado un video de este escenario, el cual está compuesto por 36005 fotogramas con un tamaño de 1080x1920 píxeles. Este video puede descargarse de la web[5].

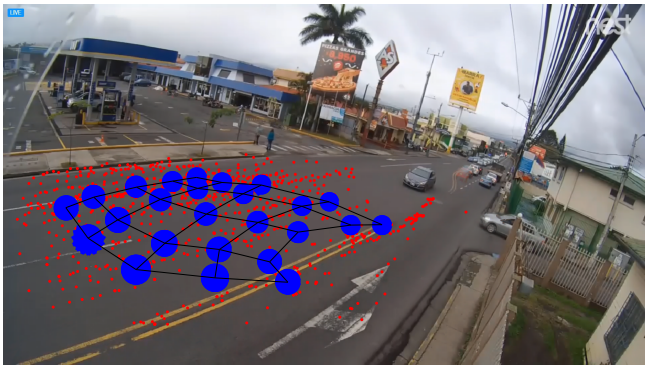


Figura 13.4: Datos de entrenamiento y el SOM entrenado con ellos mostrados en un fotograma del fondo del video seleccionado. Los puntos rojos son muestras de entrenamiento, los círculos azules son los prototipos del SOM y las líneas entre los prototipos son las conexiones entre ellos. Se puede observar que los prototipos más cercanos a la cámara son más grandes que el resto de prototipos, lo que significa que un mayor número de píxeles se corresponden con un metro real.

13.3.3. Selección de parámetros

La configuración de los parámetros de los diferentes módulos que componen el sistema se describen en esta subsección.

Primero, los valores de los parámetros del módulo Faster-RCNN son aquellos que los autores recomiendan como valores por defecto. Solamen-

[3] `https://www.camarasviales.com`

[4] `https://www.camarasviales.com/camara-guadalupe`

[5] `https://www.lcc.uma.es/~miguelangel/resources/fixed_camera/camarasviales-guadalupe_2018-01-18_23-30-00.mp4`

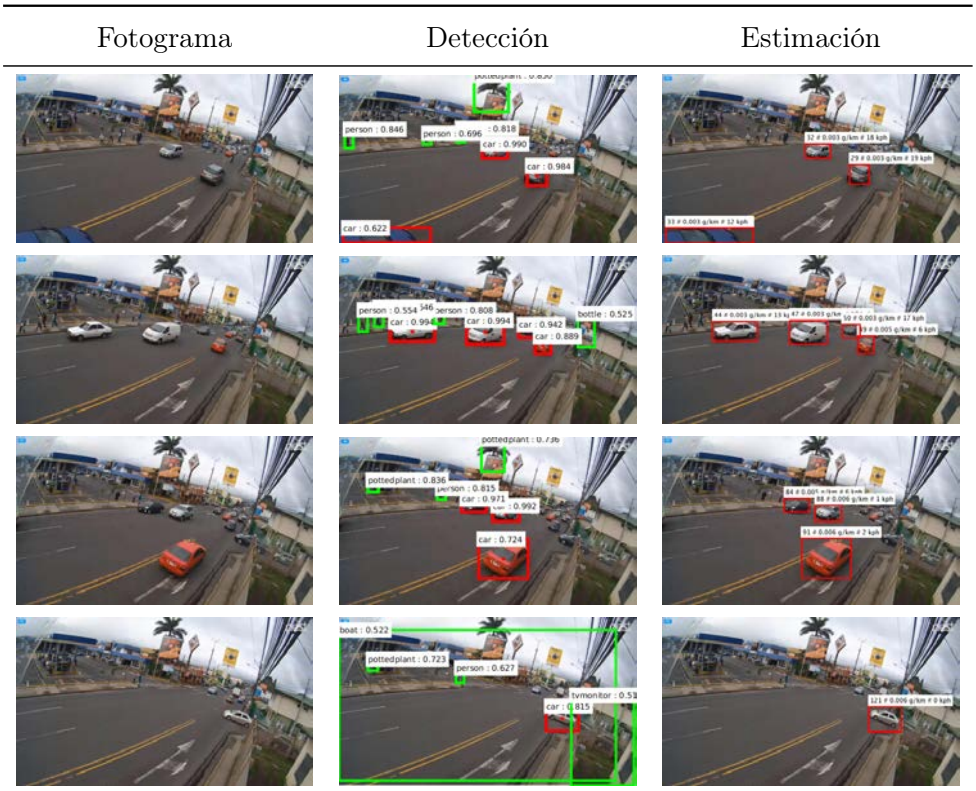


Figura 13.5: Descripción gráfica del funcionamiento de la propuesta. De izquierda a derecha, las columnas muestran un fotograma de una secuencia, los objetos detectados en él proporcionados por el módulo Faster-RCNN (detección de objetos), y los coches detectados con su trayectoria proporcionados por el modelo de Kalman, junto con la información de su velocidad y contaminación producida estimadas (Estimación). Las filas muestran los fotogramas 698, 1000, 1651 y 2563 de camara-guadalupe. Nótese que las regiones rectangulares mínimas rojas corresponden a objetos detectados no deseados y las verdes se corresponden con objetos que pertenecen a la clase *coche*.

te se ha modificado el umbral para reconocer el mayor número posible de coches.

Por otro lado, el parámetro más importante del módulo SOM es el número de neuronas y se ha seleccionado para cubrir las regiones del video con más actividad. Además, en el proceso de detección de motos para obtener el conjunto de entrenamiento del SOM, se ha empleado un umbral de la Faster-RCNN con $\tau = 0{,}90$ para conseguir unos datos de entrenamiento robustos. Por otro lado, para estimar el valor de la anchura estándar de una moto, se ha seleccionado el número de motos con licencia por tamaño del motor en

Reino Unido[6], y se dice que el tamaño medio de motor es aproximadamente 600 cc. Tras esto, se ha seleccionado el modelo de moto conocido con más licencias actualmente con este tamaño de motor[7], que se corresponde con la YAMAHA FZS 600. Por último, se ha tomado la anchura (longitud) de la YAMAHA FZS 600, y es de 2080 mm (2,08 metros)[8]. También se ha considerado la altura (longitud) de una persona conduciendo una moto y se ha estimado con un valor de 1700 mm (1,70 metros).

Por último, los valores de los parámetros que se han utilizado en el módulo de estimación de la contaminación se han obtenido del informe Production of Updated Emission Curves for Use in the National Transport Model, que está disponible en su página web[9] y se ha seleccionado la configuración correspondiente al año 2020. Además, de acuerdo con este informe, las curvas de emisión son adecuadas para valores de velocidad entre 5 y 120 km/h. La Figura 13.3 muestra las curvas de emisión que se han considerado para los tipos de vehículos de combustible gasolina o diésel (y_p y y_d, respectivamente).

Todos los valores de los parámetros se muestran en la Tabla 13.1.

13.3.4. Resultados

Para comprobar la idoneidad de la propuesta, se han estudiado los datos obtenidos desde un punto de vista cualitativo y cuantitativo.

Se ha utilizado la misma secuencia *camara-guadalupe* en el proceso de entrenamiento del SOM y el de estimación de la contaminación.

Primero, se ha ejecutado el proceso de entrenamiento del SOM para obtener el modelo SOM que proporcionará una estimación de la relación píxel-metro en cada área del fotograma. El SOM entrenado y el conjunto de entrenamiento se pueden observar en la Figura 13.4.

Tras esto, se ha ejecutado la propuesta para estimar la contaminación en el video seleccionado. Desde un punto de vista cualitativo, el funcionamiento del sistema propuesto se puede observar en la Figura 13.5. Se muestran algunos fotogramas aleatorios (primera columna) y su correspondiente salida tras la aplicación de la red Faster-RCNN (segunda columna) y el modelo de Kalman con la contaminación y velocidad estimadas (tercera columna). Como puede verse, dado un fotograma de la secuencia como entrada, la salida con la información obtenida de la Faster-RCNN (segunda columna) se corresponde con el paso de detección y clasificación de objetos, dentro del módulo de detección y seguimiento. Este paso produce varias deteccio-

[6] `https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/608185/veh0306.ods`

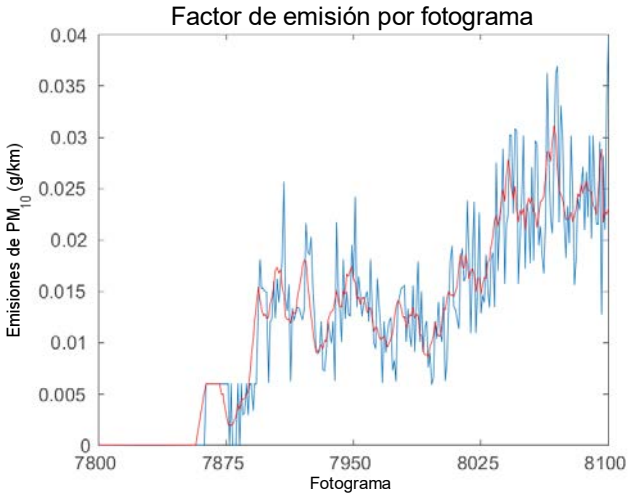
[7] `https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/666910/veh0120.ods`

[8] `https://en.wikipedia.org/wiki/Yamaha_FZS600_Fazer`

[9] `https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/662795/updated-emission-curves-ntm.pdf`

Figura 13.6: Factor de emisión por fotograma.

(a) Factor de emisión por fotograma en la secuencia para los fotogramas entre el 7800 y el 8100. La línea azul indica el factor de emisión por fotograma y la línea roja muestra dicho factor de emisión aplicando una ventada deslizante.



(b) Ejemplos de fotogramas con la información de salida proporcionada por la propuesta. Fotogramas 7800, 7948 y 8048 mostrando la trayectoria de los vehículos detectados junto con su velocidad y contaminación estimadas en esos fotogramas.

nes no deseadas (regiones rectangulares mínimas rojas) y solo se seleccionan aquellas correspondientes a la clase *coche* (regiones rectangulares mínimas verdes). Por último, considerando esta información, el sistema produce la salida correspondiente al fotograma de entrada proporcionando la información de la trayectoria de cada vehículo detectado y su estimación de velocidad y contaminación (tercera columna).

Por otro lado, el análisis cuantitativo se muestra en la Figura 13.6 (a), que muestra el factor de emisión para varios fotogramas seleccionados de la secuencia elegida. Como puede observarse, la propuesta estima la contaminación en cada fotograma y con esta información se pueden obtener aquellos momentos con un nivel de tráfico alto o bajo, como se puede ver en la Figura 13.6 (b). Otro punto importante a destacar son las subidas y bajadas de la estimación, que se corresponde con la línea azul de (a). Esto se debe a que

la región rectangular mínima correspondiente al vehículo i producida como salida por la red Faster-RCNN en el fotograma t no es prácticamente la msima que en el fotograma previo $t-1$. Este error produce un mal cálculo de cada centroide, y en la mayoría de los casos ambos centroides estarán lejos el uno del otro, por lo que el sistema proporcionará una alta velocidad para este vehículo i. Por tanto, para evitar estas subidas y bajadas, se ha empleado una ventana deslizante con un tamaño de 5 fotogramas para mostrar un factor de emisión suavizado por fotograma, que está representado por una línea roja en (b).

13.4. Conclusión

En este capítulo se ha presentado una metodología para la estimación de la contaminación en carreteras utilizando cámaras de tráfico estáticas. Inicialmente fue necesario detectar los vehículos presentes en la carretera (utilizando la red Faster-RCNN) para estimar después su velocidad y nivel de contaminación. Debido a que todas las escenas tienen perspectiva de visión, se ha utilizado un modelo neuronal autoorganizado para corregir y homogeneizar la correspondencia entre la distancia física y el número de píxeles en cada región de la imagen.

Los experimentos muestran que hay una clara correlación entre la estimación de contaminación en cada fotograma y el número de vehículos mostrados, como puede verse en la Figura 13.6. Estos resultados prometedores permiten hacer estudios de comparativas más extensivos con otras técnicas existentes. También es posible estudiar la factibilidad de usar esta propuesta combinada con otro tipo de sensores que incrementen su efectividad.

Parte IV

Conclusiones

Esta cuarta y última parte de la tesis muestra las conclusiones generales que se han obtenido tras la ejecución de los trabajos que se han llevado a cabo durante su desarrollo.

Capítulo 14

Conclusiones (Conclusions)

No entiendes realmente algo a menos que seas capaz de explicárselo a tu abuela.
(You do not really understand something unless you can explain it to your grandmother.)

Albert Einstein

Sólo podemos ver poco del futuro, pero lo suficiente para darnos cuenta de que hay mucho que hacer.
(We can only see a short distance ahead, but we can see plenty there that needs to be done.)

Alan Turing

Resumen: En este capítulo se describen las conclusiones generales a las que se han llegado tras la realización de la presente tesis y se indican una serie de posibles líneas de investigación futura.

14.1. Conclusiones y líneas de trabajo futuras

14.1.1. Conclusiones

A continuación se van a exponer las conclusiones generales que se pueden extraer como consecuencia de la realización de esta tesis.

Para empezar se debe destacar que su desarrollo ha permitido la publicación de numerosos trabajos tanto en revistas indexadas en el ranking *Journal*

Citation Reports como en congresos que aparecen en los rankings *GII-GRIN-SCIE (GGS) Conference Rating* y *CORE*, estando algunos de ellos situados en la parte noble de cada ranking.

En cuanto al trabajo desarrollado, en esta tesis se ha tratado de cubrir diversas áreas de la visión por computador y del procesamiento y análisis de imágenes y vídeo. Se han implementado tanto propuestas generales como concretas para su aplicación a determinados campos de la ciencia, utilizando hardware específico o genérico. Para llevar a cabo dicha implementación principalmente se han utilizado redes neuronales de diferentes tipos, pero también se ha combinado con otros modelos como la lógica difusa.

Como se ha comentado anteriormente en diversas ocasiones, los sistemas de videovigilancia pueden dividirse en tres etapas: la primera corresponde a la segmentación de los objetos en primer plano, la segunda trata de hacer un seguimiento de dichos objetos, y la tercera observa sus comportamientos o actividades. Dicho esto, la presente tesis hace énfasis en la primera de estas etapas, comenzando por el estudio y desarrollo de nuevos métodos de algoritmos de detección de objetos de primer plano. Dichas propuestas se corresponden con los Capítulos 3, 4, 5 y 6.

En el Capítulo 3 se ha diseñado un nuevo modelo de red neuronal, el cual es utilizado con el propósito de dar respuesta al modelado del fondo de la escena y detectar aquellos objetos de primer plano en cualquier tipo de escenario y sin la necesidad de un hardware potente. En el aspecto teórico este nuevo modelo ofrece unas características muy particulares ya que aúna las ventajas de los mapas autoorganizados (en el hecho de que no existen neuronas muertas) y al aprendizaje competitivo estándar (ya que no existe una topología que mantenga conectadas las neuronas). Con respecto a la parte práctica, este nuevo método basado a nivel de píxel logra un rendimiento superior o similar a otros del mismo tipo del estado del arte. Es de destacar el hecho de que el nuevo modelo obtenga el mejor resultado en la categoría Baseline del conjunto de datos ChangeDetection.net, ya que estos videos se han utilizado ampliamente para el estudio del problema de la detección de objetos en primer plano.

Por su parte, el Capítulo 4 ha abordado el desarrollo del modelo, en lugar de con uno totalmente nuevo, con la mejora de un método ya existente. Además, el problema se ha tratado desde una perspectiva más específica. Sin embargo, esta aparente simplicidad ha venido acompañada de la introducción del aprendizaje y empleo de la lógica difusa, una metodología que puede proporcionar soluciones sencillas y elegantes. En el caso concreto de este capítulo se ha centrado en tratar los cambios de iluminación para mejorar la salida del método, pero podría aplicarse a otros ámbitos de la videovigilancia. Otro hecho a destacar es el de la extracción de características a partir de la información que se posee. Este extra de información influye en la toma de decisiones para incrementar el rendimiento del método.

Ambos métodos que se acaban de comentar operan con cualquier tipo de hardware presente en un ordenador personal; sin embargo, por ejemplo, no funcionarían adecuadamente con un sistema hardware de bajo coste. En este sentido, en el Capítulo 5 se ha desarrollado un sistema de detección de movimiento con un hardware de este tipo. Si bien es cierto que el problema de la detección de movimiento es más simple que el de la detección de objetos en primer plano, también es cierto que en determinadas ocasiones no se requiere una solución tan sofisticada, siendo la principal limitación el coste económico del sistema a implantar.

En cuanto al Capítulo 6, en él se muestra un caso específico de segmentación, concretamente perteneciente al ámbito sanitario. Gracias al análisis de las características que presentan los objetos a segmentar y el escenario en el que se encuentran, se pueden desarrollar alternativas a las tradicionales genéricas. Éste es el caso de los glóbulos rojos en la sangre, en los que se observa que tienen una forma parecida a la de un círculo, de forma que, mediante el uso combinado de técnicas geométricas y de aprendizaje computacional, se puede obtener el resultado de la segmentación de dichas células rápidamente. Debe destacarse que la aplicación de principios matemáticos, como la transformada del círculo de Hough en este caso, pueden reducir y simplificar problemas que, a priori, pueden ser complejos. También resalta el hecho de que, aunque los métodos basados en soluciones tradicionales ofrezcan un muy buen rendimiento, pueden surgir nuevas propuestas que mejoren lo anterior, como es el caso de las técnicas basadas en aprendizaje profundo. Una ligera mejora del rendimiento puede no suponer un gran avance en muchos de los campos de investigación, pero en otros, como la medicina, puede suponer un punto de inflexión.

Todos estos modelos tienen una característica común: están basados a nivel de píxel. Esto quiere decir que, al igual que les ocurre a la mayoría de métodos del estado del arte, les afecta en gran medida el tamaño de las imágenes que van a tratar. De esta forma, las actuales dimensiones de los fotogramas suponen una complejidad añadida para su ejecución en tiempo real. Es cierto que la potencia de los ordenadores de hoy día mitigan esta dificultad, pero no lo suficiente. Es por ello por lo que, además de desarrollar nuevas alternativas que traten este problema, se deben buscar soluciones que permitan reutilizar y adaptar los métodos ya existentes al nuevo contexto. Ésta es la idea que alimenta al trabajo presentado en el Capítulo 8, en el que se elabora un marco de trabajo, basado en la compresión de imágenes, que pueda aplicarse a todos los métodos de detección de objetos de primer plano para mejorar sus tiempos de computación y poder lograr la ejecución en tiempo real. La compresión de imágenes es una solución que se ha aplicado para aligerar la carga computacional en diversas propuestas. De hecho, en el Capítulo 6 se emplea esta estrategia para obtener una segmentación preliminar de los glóbulos rojos.

El contexto general en el que se han implementado los sistemas de videovigilancia desde sus inicios ha constado de una cámara estática. Sin embargo, la evolución de la tecnología ha permitido disponer de otro tipo de dispositivos que ofrecen otras características adicionales, como son las cámaras PTZ. Es por ello por lo que en esta tesis se ha propuesto una segunda parte de la misma, formada por los Capítulos 9 y 10, centrada en los sistemas de videovigilancia que hacen uso de este tipo específico de cámara, de la que se desea que tenga un movimiento inteligente automático para cubrir el escenario en el que se encuentre instalada. Además, continuando con las etapas de las que se componen los sistemas de videovigilancia, podemos decir que ambos capítulos pertenecen a la segunda fase, en la que se hace un seguimiento de los objetos detectados. Así, el Capítulo 9 presenta un sistema que realiza un seguimiento del objeto de la escena que considera más anómalo, determinando entre si es anómalo o no gracias a un modelo no paramétrico entrenado únicamente con datos de la propia escena, sin necesidad de la intervención humana en ninguno de sus procesos. Este sistema hace uso de una unidad de procesamiento gráfico con capacidad para ejecutar modelos de aprendizaje profundo, por lo que se trata de un dispositivo superior al estándar. Sin embargo, el Capítulo 10 ofrece un método para que la cámara PTZ cubra aquellas zonas donde se esté registrando una mayor presencia de objetos en primer plano, actividad menos intensa que la realizada por el método del caso anterior pero que se puede ejecutar sin la necesidad de una unidad de procesamiento gráfico tan potente.

El Capítulo 7 podría decirse que aúna las características principales presentadas en los Capítulos 3 y 9, puesto que sigue la línea de trabajos de una elevada carga teórica, como en el Capítulo 3, con el diseño de un nuevo modelo de gas neuronal autoorganizado basado en las Divergencias de Bregman; y se estudia la aplicación de dicho modelo a la detección de anomalías, problema tratado en el Capítulo 9.

Por su parte, la tercera parte de esta tesis se ha considerado el estudio de la aplicación de soluciones al transporte inteligente mediante el uso de sistemas de videovigilancia de tráfico. Los Capítulos 11 y 12 son similares puesto que abordan el problema de la clasificación de los vehículos que circulan por una carretera, proponiendo el primero de ellos una solución eficaz pero compleja por el hardware que requiere, mientras que el segundo hace uso de unas técnicas que no requieren tanto procesamiento de cómputo pero que ofrecen un peor rendimiento. En el Capítulo 12, al igual que ocurre en los Capítulos 4 y 6, la extracción de determinadas características permite abordar el problema con mejor rendimiento; por su parte, en el Capítulo 11 la extracción de rasgos se produce automáticamente gracias a la utilización del aprendizaje profundo.

Siguiendo con las aplicaciones en los sistemas de videovigilancia de tráfico, y gracias a trabajos como los que se acaban de comentar, es posible

plantearse otros fines u objetivos que antes no habían sido considerados. Un ejemplo de esto puede ser la estimación de la contaminación producida por los vehículos que circulan por una carretera utilizando únicamente la información proporcionada por una cámara de videovigilancia estática, para lo que se propone el método que se presenta en el Capítulo 13. Cabe destacar la necesidad de datos de interés para acometer la implementación y contrastar la solución que se aporta. En el caso de este problema planteado, se hace vital contar con cierta información como el número de vehículos matriculados por tipo de combustible, la contaminación existente en cada instante de tiempo o incluso el propio video de tráfico. La publicación de informes y trabajos que ofrezcan datos abiertos de calidad es primordial para el avance de la Investigación.

Por último, el hito de mayor impacto durante la realización de la presente tesis ha sido la irrupción del aprendizaje profundo a mediados de su desarrollo. Esta nueva técnica ha supuesto una revolución en el procesamiento de imágenes y en el caso de esta tesis ha motivado que se considerara su estudio y posterior aplicación al trabajo que se estaba ejecutando. Prueba de ello son los Capítulos 6, 7, 9, 11 y 13, que emplean metodologías basadas en aprendizaje profundo.

Tal y como se puede comprobar, los estudios realizados son muy variopintos pero siempre dentro del campo del procesamiento de imágenes, hecho que indica la riqueza y amplitud de este área dentro de la Ciencia.

14.1.2. Líneas de trabajo futuras

Tras exponer las conclusiones generales, se pueden comentar diversas futuras líneas de investigación con el objetivo de mejorar y/o completar los estudios que componen esta tesis. Así, algunas de estas ideas serían las siguientes:

- Diseñar nuevos algoritmos de modelado de fondo de la escena y detección de objetos en primer plano para escenarios de un contexto específico, especialmente aquellos en los que hay numerosas dificultades que tener en consideración como son los fondos dinámicos o las condiciones climatológicas adversas, ya que en multitud de estos casos la respuesta de los algoritmos no es la deseada.

- Diseñar nuevos algoritmos de detección de objetos que puedan adaptarse a las necesidades hardware que cada situación requiera. Por ejemplo, los dispositivos hardware de bajo coste han abaratado la tecnología de forma que ahora se puede implantar un sistema donde antes hubiera sido impensable, por lo que sería deseable contar con métodos de detección de objetos específicos para este tipo de arquitecturas, de forma que se puedan acometer nuevos retos.

- Estudiar la detección (temprana) de enfermedades mediante el análisis de imágenes médicas. De igual forma que se ha hecho con el estudio de la detección de los glóbulos rojos en esta tesis, se podrían hacer estudios que contemple la detección de otro tipo de células, o bien se podría aplicar técnicas de aprendizaje profundo para la detección temprana de enfermedades.
- Analizar y predecir el estado de tráfico de una carretera en función del comportamiento de los vehículos que van circulando por dicha vía.
- Mejorar la estimación de la contaminación realizando un cálculo más exhaustivo de la trayectoria de los vehículos, lo que implica obtener la velocidad con mayor precisión y por ende la estimación de la contaminación. También se pueden incorporar modelos de estimación de la contaminación para cada clase de vehículo.
- Diseñar nuevos métodos de detección de anomalías. Sería deseable contar con sistemas que interpreten una escena y detecten las anomalías de forma automática. En el caso de una cámara estática en un contexto conocido puede resultar bastante sencillo, pero este trabajo se complica cuando se desconoce el medio en el que se encuentra la cámara, siendo mayor la dificultad cuando se cuenta con una cámara PTZ ya que lo deseable sería que se tuviera la mayor cobertura posible de aquellas zonas donde se están detectando las anomalías.

14.2. Conclusions and future lines of research

14.2.1. Conclusions

The general conclusions that can be extracted as a result of the attainment of this thesis are presented below.

For a start, it must be pointed out that its development has allowed the publication of many works in journals indexed in the *Journal Citation Reports* ranking and conferences which appear in the *GII-GRIN-SCIE (GGS) Conference Rating* and *CORE* rankings, some of them located in the superior part of each ranking.

Regarding the developed work, this thesis tries to cover several topics of computer vision and image and video processing and analysis. General approaches and specific proposals have been implemented for its application to different science fields, by employing specific or generic hardware. In order to obtain these implementations, different kinds of neural networks have been used. Also, they have been combined with other computational intelligence models, like fuzzy logic.

As it has commented before several times, surveillance systems can be divided into three steps: the first one corresponds to the segmentation of

the foreground objects, the second step tries to follow these objects, and the third one observes their behaviour or activities. That said, the presented thesis focuses on the first of these steps, by starting with the study and development of new foreground object detection algorithms. These proposals correspond to Chapters 3, 4, 5 and 6.

In Chapter 3, a new neural network model has been designed, which is used with the aim to provide a solution to the background scene modelling problem and to detect those foreground objects in any scenario without any powerful hardware needed. In the theoretical facet, this new model offers special features due to the combination of the advantages of the self-organizing maps (in that there are no dead neurons) and standard competitive learning (because there is no topology that keeps the neurons connected). Regarding to the practical part, this new pixel-level method yields a higher or similar performance than other state-of-art methods of the same kind. It must be highlighted that the new model obtains the best result in the Baseline category from the ChangeDetection.net dataset. These videos have widely been employed to study the foreground object detection problem.

For its part, Chapter 4 has addressed the development of the model with the improvement of an existing method instead of a new one. In addition, the problem has been treated from a more specific point of view. However, this apparent simplicity has served with the introduction of the learning and employment of the fuzzy logic, a methodology which can provide an easy and elegant solution. In the particular case of this chapter, it has been focused to treat the illumination changes in order to improve the output of the method, but it could be applied to other surveillance topics. Another fact to be highlighted is the features extraction that it carries out. This extra information affects how to make a decision to improve the performance of the method.

Both methods that have been commented work with any hardware type presented in a personal computer; however, for example, they do not work properly with a low cost hardware system. In order to address this case, in Chapter 5 a movement detection system with this kind of hardware has been developed. While it is true that the movement detection problem is simpler than the foreground object detection one, it is also true that it is not required a sophisticated solution on certain occasions, where the main limitation is the economical cost of the system to set up.

Regarding Chapter 6, it exhibits a specific segmentation case, in particular it belongs to the sanitary field. Due to the analysis of the features that present the objects to segment and the scenario where they are, it can develop alternatives to the generic traditional ones. This is the case of the red blood cells in the blood, where it is observed that they have a shape similar to the circle one, so that, by the combined use of geometrical techniques and computational learning, it can obtain the result of the segmentation of

those cells quickly. It must be highlighted that the application of mathematical principles, like the Hough Circle Transform in this case, can reduce and simplify problems that, a priori, can be complex. It must also be outlined the fact that, although the methods based on traditional solutions offer a very good performance, new proposals that improve the previous ones can appear, as the techniques based on deep learning. A slight improvement of the performance may not be a great advance in many of the research fields, but in others, like medicine, it can mark an inflection point.

All this models have a common feature: they are pixel-level based. This means that, as happens to most of state-of-art methods, they are greatly affected by the size of the images that are going to be processed. In this way, the current dimensions of the frames suppose an additional complexity to its execution in real time. It is true that the power of the current computers mitigates this difficulty, but not enough. That is why, apart from developing new alternatives that treat this problem, solutions should be found that allow to re-use and adapt the existing methods to the new context. This is the idea that inspires the work presented in Chapter 8, where a framework is elaborated, based on the image downsampling, in order to be applied to all foreground object detection methods in order to improve their computational time and to achieve the real time execution. Image downsampling is a solution which is applied to alleviate the computational load in several proposals. In fact, in Chapter 6 this strategy is applied to obtain a preliminary segmentation of the red blood cells.

From the beginning, the general context has been that surveillance systems have been formed by a static camera. However, the evolution of technology has allowed to employ other devices which offer additional features, like PTZ cameras. That is why in this thesis the second part is formed by Chapters 9 and 10, focused on surveillance systems which use this specific kind of camera, which is desired to have an automatic intelligent movement to cover the scenario in which it is installed. Furthermore, continuing with the stages of which the video surveillance systems are composed, we can say that both chapters belong to the second phase, where a tracking of the detected objects is done. So, Chapter 9 presents a system that performs a tracking of the object of the scene that considers the most anomalous. The system determines whether it is anomalous or not due to a non-parametric model trained with data only from the same scene, without any human intervention in any of its processes. This system uses a graphics processing unit with capacity to train deep learning models, so that, this device is higher than the standard. However, Chapter 10 offers a method for the PTZ camera to cover those areas where there is a greater presence of objects in the foreground, less intense activity than that performed by method from the previous case, but that can be executed without the need of a powerful graphics processing unit.

It could be said that Chapter 7 combines the main features presented in Chapters 3 and 9. This can be said since it follows the work line with a high theoretical load, like Chapter 3, with the design of a new self-organizing neural gas model based on Bregman Divergences. On the other hand, it is studied the application of this model to the anomaly detection, problem treated in Chapter 9.

On the other side, the third part of this thesis has considered the study of the application of solutions to intelligent transport by using traffic surveillance systems. Chapters 11 and 12 are similar because they address the problem of the classification of the vehicles that travel on a road, by proposing the first one an effective solution but complex due to the required hardware, while the second one makes use of techniques that do not require so much computer processing but that offer worse performance. In Chapter 12, as happens in Chapters 4 and 6, the extraction of certain features allows to address the problem with higher performance; on the other hand, in Chapter 11 the feature extraction is produced automatically due to the deep learning use.

Following with the applications in the systems of traffic video surveillance, and thanks to works like those that have just commented, it is possible to consider other aims or objectives that had not been considered before. An example of this may be the estimation of pollution produced by vehicles traveling on a road using only the information provided by a static video surveillance camera, for which the method presented in Chapter 13 is proposed. It must be highlighted the need of relevant data to undertake the implementation and to contrast the solution that provides. In the case of this raised problem, it is vital to have certain information such as the number of vehicles registered per fuel type, the existing pollution at each moment of time or even the traffic video itself. The publication of reports and works that offer quality open data is essential for the advancement of this research.

Finally, the milestone of greatest impact during the realization of this thesis has been the emergence of deep learning in the middle of its development. This new technique has meant a revolution in the processing of images and in the case of this thesis it has motivated that its study and subsequent application to the work that was being executed be considered. Example of this are Chapters 6, 7, 9, 11 and 13, which employ methodologies based on deep learning.

As it can be seen, the carried out studies are very varied but always within the field of image processing, a fact that indicates the richness and breadth of this area within Computer Science.

14.2.2. Future lines of research

After presenting the general conclusions, several future lines of research can be commented with the aim of improving and / or completing the studies that form this thesis. Thus, some of these ideas would be the following:

- To design new algorithms for background scene modeling and foreground object detection in specific context scenarios, especially those in which there are numerous difficulties to take into consideration such as dynamic backgrounds or adverse weather conditions, since in many of these cases the output of the algorithms is not the desired one.
- To design new object detection algorithms that can adapt to the hardware needs that each situation requires. For example, low cost hardware devices have cheapened the technology so that now a system can be implemented where previously it would have been unthinkable, so it would be desirable to have specific object detection methods for this type of architecture, so that new challenges can be tackled.
- To study the (early) detection of diseases through the analysis of medical images. In the same way that it has been done with the study of the detection of red blood cells in this thesis, studies could be done that address the detection of other types of cells, or deep learning techniques could be applied for the early detection of diseases.
- To analyze and predict the traffic status of a road according to the behavior of the vehicles that are driving along this road.
- To improve the estimation of the pollution by making a more exhaustive calculation of the trajectory of the vehicles, which implies obtaining the speed with higher precision and therefore the estimation of the contamination. Pollution estimation models can also be incorporated for each class of vehicle.
- To design new anomaly detection methods. It would be desirable to have systems that interpret a scene and detect anomalies automatically. In the case of a static camera in a known context it can be quite simple, but this work is complicated when the context in which the camera is located is unknown, being more difficult when it has a PTZ camera since it is desirable to have the highest possible coverage of those areas where anomalies are detected.

Parte V

Apéndices

En esta parte se incluye información de interés sobre las publicaciones obtenidas en la comunidad científica gracias a la realización de los trabajos que forman la presente tesis.

Apéndice A

Resumen de publicaciones obtenidas

Resumen: En esta sección se muestran unas tablas que resumen la información asociada a las publicaciones obtenidas con los trabajos presentados en esta tesis. Para las revistas se ha utilizado el ranking *Journal Citation Reports (JCR)*, mientras que para los congresos se ha empleado el *GII-GRIN-SCIE (GGS) Conference Rating* publicado el 30 de mayo de 2018 y el *CORE2018*.

Título	Smart motion detection sensor based on video processing using self-organizing maps
Autores	Francisco Ortega-Zamorano, Miguel A. Molina-Cabello, Ezequiel López-Rubio, Esteban J. Palomo
Revista	Expert Systems with Applications
Año	2016
Factor de impacto	3,928
Categorías JCR	COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE (18/133 (Q1)) ENGINEERING, ELECTRICAL & ELECTRONIC (37/262 (Q1)) OPERATIONS RESEARCH & MANAGEMENT SCIENCE (3/83 (Q1))
Estado	Publicado
DOI	`https://doi.org/10.1016/j.eswa.2016.08.010`
Referencia	Ortega-Zamorano et al. 2016

Título	Foreground Detection by Competitive Learning for Varying Input Distributions
Autores	Ezequiel López-Rubio, Miguel A. Molina-Cabello, Rafael Marcos Luque-Baena, Enrique Domínguez
Revista	International Journal of Neural Systems
Año	2018
Factor de impacto	4,580
Categorías JCR	COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE (13/132 (Q1))
Estado	Publicado
DOI	https://doi.org/10.1142/S0129065717500563
Referencia	López-Rubio et al. 2018a

Título	Vehicle type detection by ensembles of Convolutional Neural Networks operating on super resolved images
Autores	Miguel A. Molina-Cabello, Rafael Marcos Luque-Baena, Ezequiel López-Rubio, Karl Thurnhofer-Hemsi
Revista	Integrated Computer-Aided Engineering
Año	2018
Factor de impacto	3,667
Categorías JCR	COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE (21/132 (Q1)) ENGINEERING, MULTIDISCIPLINARY (7/86 (Q1)) COMPUTER SCIENCE, INTERDISCIPLINARY APPLICATIONS (17/105 (Q1))
Estado	Aceptado
DOI	-
Referencia	-

Título	Frame Size Reduction for Foreground Detection in Video Sequences
Autores	Miguel A. Molina-Cabello, Ezequiel López-Rubio, Rafael Marcos Luque Baena, Esteban J. Palomo, Enrique Domínguez
Congreso	Conference of the Spanish Association for Artificial Intelligence (CAEPIA)
Año	2016
GGS Rating	-
CORE Rating	-
Estado	Publicado
DOI	https://doi.org/10.1007/978-3-319-44636-3_1
Referencia	Molina-Cabello et al. 2016b

Título	Pixel Features for Self-organizing Map Based Detection of Foreground Objects in Dynamic Environments
Autores	Miguel A. Molina-Cabello, Ezequiel López-Rubio, Rafael Marcos Luque Baena, Enrique Domínguez, Esteban J. Palomo
Congreso	International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO)
Año	2016
GGS Rating	Work in Progress
CORE Rating	-
Estado	Publicado
DOI	https://doi.org/10.1007/978-3-319-47364-2_24
Referencia	Molina-Cabello et al. 2016a

Título	Vehicle Type Detection by Convolutional Neural Networks
Autores	Miguel A. Molina-Cabello, Rafael Marcos Luque Baena, Ezequiel López-Rubio, Karl Thurnhofer-Hemsi
Congreso	International Work-Conference on the Interplay Between Natural and Artificial Computation (IWINAC)
Año	2017
GGS Rating	Work in Progress
CORE Rating	National:Spain
Estado	Publicado
DOI	https://doi.org/10.1007/978-3-319-59773-7_28
Referencia	Molina-Cabello et al. 2017c

Título	Vehicle Classification in Traffic Environments Using the Growing Neural Gas
Autores	Miguel A. Molina-Cabello, Rafael Marcos Luque Baena, Ezequiel López-Rubio, Juan Miguel Ortiz-de-Lazcano-Lobato, Enrique Domínguez, José Muñoz-Pérez
Congreso	International Work-Conference on Artificial and Natural Neural Networks (IWANN)
Año	2017
GGS Rating	B-
CORE Rating	B
Estado	Publicado
DOI	https://doi.org/10.1007/978-3-319-59147-6_20
Referencia	Molina-Cabello et al. 2017b

Título	Neural controller for PTZ cameras based on nonpanoramic foreground detection
Autores	Miguel A. Molina-Cabello, Ezequiel López-Rubio, Rafael Marcos Luque Baena, Enrique Domínguez, Karl Thurnhofer-Hemsi
Congreso	IEEE International Joint Conference on Neural Networks (IJCNN)
Año	2017
GGS Rating	B
CORE Rating	A
Estado	Publicado
DOI	https://doi.org/10.1109/IJCNN.2017.7965882
Referencia	Molina-Cabello et al. 2017a

Título	Blood Cell Classification Using the Hough Transform and Convolutional Neural Networks
Autores	Miguel A. Molina-Cabello, Ezequiel López-Rubio, Rafael M. Luque-Baena, María Jesús Rodríguez-Espinosa, Karl Thurnhofer-Hemsi
Congreso	World Conference on Information Systems and Technologies (WORLDCIST)
Año	2018
GGS Rating	Work in Progress
CORE Rating	C
Estado	Publicado
DOI	https://doi.org/10.1007/978-3-319-77712-2_62
Referencia	Molina-Cabello et al. 2018

Título	Road pollution estimation using static cameras and neural networks
Autores	Miguel A. Molina-Cabello, Rafael Marcos Luque-Baena, Ezequiel López-Rubio, Lipika Deka, Karl Thurnhofer-Hemsi
Congreso	IEEE International Joint Conference on Neural Networks (IJCNN)
Año	2018
GGS Rating	B
CORE Rating	A
Estado	Aceptado
DOI	-
Referencia	-

Título	A New Self-Organizing Neural Gas Model based on Bregman Divergences
Autores	Esteban J. Palomo, Miguel A. Molina-Cabello, Ezequiel López-Rubio, Rafael Marcos Luque-Baena
Congreso	IEEE International Joint Conference on Neural Networks (IJCNN)
Año	2018
GGS Rating	B
CORE Rating	A
Estado	Aceptado
DOI	-
Referencia	-

Apéndice B

Publicaciones obtenidas

Resumen: En esta sección se encuentra la primera página de cada uno de los trabajos publicados.

International Journal of Neural Systems, Vol. 28, No. 5 (2018) 1750056 (16 pages)

DOI: 10.1142/S0129065717500563

Foreground Detection by Competitive Learning for Varying Input Distributions

Ezequiel López-Rubio* and Miguel A. Molina-Cabello†
Department of Computer Languages and Computer Science
University of Málaga
Bulevar Louis Pasteur, 35
29071 Málaga, Spain
**ezeqlr@lcc.uma.es*
†*miguelangel@lcc.uma.es*

Rafael Marcos Luque-Baena
Department of Computer Systems and Telematics Engineering
University of Extremadura
Calle Sta. Teresa Jornet, 38
06800 Mérida (Badajoz), Spain
rmluque@unex.es

Enrique Domínguez‡
Department of Computer Languages and Computer Science
University of Málaga, Bulevar Louis Pasteur
35, 29071 Málaga, Spain
‡*enriqued@lcc.uma.es*

Accepted 9 November 2017
Published Online 3 January 2018

One of the most important challenges in computer vision applications is the background modeling, especially when the background is dynamic and the input distribution might not be stationary, i.e. the distribution of the input data could change with time (e.g. changing illuminations, waving trees, water, etc.). In this work, an unsupervised learning neural network is proposed which is able to cope with progressive changes in the input distribution. It is based on a dual learning mechanism which manages the changes of the input distribution separately from the cluster detection. The proposal is adequate for scenes where the background varies slowly. The performance of the method is tested against several state-of-the-art foreground detectors both quantitatively and qualitatively, with favorable results.

Keywords: Computer vision; foreground detection; competitive learning; stationary distribution.

1. Introduction

In computer vision applications, such as video surveillance or human motion analysis, the capability of extracting the objects of interest from a video sequence is a crucial preliminary task. Video surveillance is indeed a key technology for public safety (e.g. in transport networks, town centers, schools and hospitals), efficient management of transport networks and public facilities (e.g. traffic lights, railroad crossings), recognition of human behavior patterns, etc.[1] The video frames are commonly captured from a scene using a static camera which compresses the video information. In this setting, detecting intruding objects is an essential step in analyzing

‡Corresponding author.

Expert Systems With Applications 64 (2016) 476–489

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Smart motion detection sensor based on video processing using self-organizing maps

Francisco Ortega-Zamorano[a,b], Miguel A. Molina-Cabello[a], Ezequiel López-Rubio[a,*], Esteban J. Palomo[a,b]

[a] Department of Computer Languages and Computer Science, University of Málaga, Málaga, Spain
[b] School of Mathematics and Computer Science, University of Yachay Tech, San Miguel de Urcuquí, Ecuador

ARTICLE INFO

Article history:
Received 8 March 2016
Revised 13 July 2016
Accepted 2 August 2016
Available online 3 August 2016

Keywords:
Self-organizing map
Microcontroller
Arduino
Image processing
Block processing

ABSTRACT

Most current approaches to computer vision are based on expensive, high performance hardware to meet the heavy computational requirements of the employed algorithms. These system architectures are severely limited in their practical application due to financial and technical limitations. In this work a different strategy is used, namely the development of an inexpensive and easy to deploy computer vision system for motion detection. This is achieved by three means. First of all, an affordable and flexible hardware platform is employed. Secondly, the motion detection algorithm is specifically tailored to involve a very small computational load. Thirdly, a fixed point programming paradigm is followed in implementing the system so as to further reduce the computational requirements. The proposed system is experimentally compared to the standard motion detector for a wide range of benchmark videos. The reported results indicate that our proposal attains substantially better performance, while it remains affordable and easy to install in practice.

1. Introduction

Motion detection is the process of detecting a change in the position of an object relative to its surroundings or a change in the surroundings relative to an object. Motion detection can be achieved by either mechanical or electronic methods, but it is most usually implemented by electronic sensors.

Motion sensors can be passive or active. Passive sensors do not emit any energy to the environment and they are the most common kind of electronic sensors. They are sensitive to a person's skin temperature through emitted blackbody radiation at mid-infrared wavelengths, in contrast to background objects at room temperature. On the other hand, active sensors emit some type of signal like light, microwave or sound into the environment and they detect some change in the behavior of the responses.

Currently new techniques are being introduced in motion detection systems with the proliferation of digital cameras capable of shooting video. Nowadays it is possible to use the output of such a camera to detect motion in its field of view using software. Motion detection is usually carried out by a software-based monitoring algorithm. When the algorithm detects motions it signals the surveillance camera to begin capturing the event. This is also called activity detection. An advanced motion detection surveillance system can analyze the type of motion to see if it warrants an alarm (García, García, Ponz, de la Escalera, & Armingol, 2014; Gómez, García, Martín, de la Escalera, & Armingol, 2015).

The Self-Organizing Map (SOM) is a kind of artificial neural network which is capable of unsupervised learning (Kohonen, 1982). Since its proposal, the SOM has been applied to knowledge discovery, data mining, detection of inherent structures in high-dimensional data and mapping these data into a two-dimensional representation space (Kohonen, 2013; Yin, 2008). This mapping retains the relationships among input data and preserves their topology. Hence this artificial neural network has had a wide range of application fields over the decades (Oja, Kaski, & Kohonen, 2003; Kaski, Kangas, & Kohonen, 1998). In particular, it has been applied to several areas of computer vision, such as color quantization (Dekker, 1994; Palomo & Domínguez, 2014; Papamarkos, 1999; Xiao, Leung, Lam, & Ho, 2012), and image segmentation (Bhandarkar, Koh, & Suk, 1997; Dong & Xie, 2005; Lacerda & Mello, 2013; Maddalena & Petrosino, 2008a). The SOM is based on an incremental (online) learning process, which has better ability to escape from local minima than batch learning (Bermejo & Cabestany, 2002) and consumes less computational time in

* Corresponding author. Fax: +34 952 13 13 97.
E-mail addresses: fortega@lcc.uma.es (F. Ortega-Zamorano), miguelangel@lcc.uma.es (M.A. Molina-Cabello), ezeqlr@lcc.uma.es (E. López-Rubio), ejpalomo@lcc.uma.es (E.J. Palomo).

http://dx.doi.org/10.1016/j.eswa.2016.08.010
0957-4174/

Frame Size Reduction for Foreground Detection in Video Sequences

Miguel A. Molina-Cabello[1( )], Ezequiel López-Rubio[1], Rafael Marcos Luque-Baena[2], Esteban J. Palomo[1,3], and Enrique Domínguez[1]

[1] Department of Computer Languages and Computer Science, University of Málaga, Bulevar Louis Pasteur, 35, 29071 Málaga, Spain
{miguelangel,ezeqlr,ejpalomo,enriqued}@lcc.uma.es
[2] Department of Computer Systems and Telematics Engineering, University of Extremadura, University Centre of Mérida, 06800 Mérida, Spain
rmluque@unex.es
[3] School of Mathematical Science and Information Technology, University of Yachay Tech, Hacienda San José s/n, San Miguel de Urcuquí, Ecuador
epalomo@yachaytech.edu.ec

Abstract. A frame resolution reduction framework to reduce the computational load and improve the foreground detection in video sequences is presented in this work. The proposed framework consists of three different stages. Firstly, the original video frame is downsampled using a specific interpolation function. Secondly, a foreground detection of the reduced video frame is performed by a probabilistic background model called MFBM. Finally, the class probabilities for the reduced video frame are upsampled using a bicubic interpolation to estimate the class probabilities of the original frame. Experimental results applied to standard benchmark video sequences demonstrate the goodness of our proposal.

Keywords: Foreground detection · Video size reduction · Interpolation techniques

1 Introduction

Within the field of artificial vision, the research on video surveillance systems mainly focuses on detecting, recognizing and tracking the movement of the foreground objects in a sequence of images. Any video surveillance system begins its activity by detecting moving objects in the scene. However, this process is more complex than subtracting the current frame and the background image previously calculated, which is considered a naive approach, but there are several problems to be solved which increase its complexity. Unfavorable factors such as illumination changes both abrupt as continuous, casting shadows of objects on the background or repetitive motions of stationary objects such as tree branches, should be taken into account by the developed methods.

There are several proposals which try to manage the problem. In [2] a temporal average of the sequence is used to obtain a background image. The Kalman

O. Luaces et al. (Eds.): CAEPIA 2016, LNAI 9868, pp. 3–12, 2016.
DOI: 10.1007/978-3-319-44636-3_1

Pixel Features for Self-organizing Map Based Detection of Foreground Objects in Dynamic Environments

Miguel A. Molina-Cabello[1( )], Ezequiel López-Rubio[1], Rafael Marcos Luque-Baena[2], Enrique Domínguez[1], and Esteban J. Palomo[1,3]

[1] Department of Computer Languages and Computer Science, University of Málaga, Bulevar Louis Pasteur, 35, 29071 Málaga, Spain
{miguelangel,ezeqlr,enriqued}@lcc.uma.es

[2] Department of Computer Systems and Telematics Engineering, University of Extremadura, University Centre of Mérida, 06800 Mérida, Spain
rmluque@unex.es

[3] School of Mathematical Science and Information Technology, University of Yachay Tech., Hacienda San José s/n., San Miguel de Urcuquí, Ecuador
epalomo@yachaytech.edu.ec

Abstract. Among current foreground detection algorithms for video sequences, methods based on self-organizing maps are obtaining a greater relevance. In this work we propose a probabilistic self-organising map based model, which uses a uniform distribution to represent the foreground. A suitable set of characteristic pixel features is chosen to train the probabilistic model. Our approach has been compared to some competing methods on a test set of benchmark videos, with favorable results.

Keywords: Foreground detection · Background modeling · Probabilistic self-organising maps · Background features

1 Introduction

Foreground object detection is a key problem in the design of computer vision systems. Algorithms to solve this problem must handle many difficulties which arise in real life videos. These inconveniences include illumination changes, shadow appearances in the foreground because of object lighting in the background or repetitive motions of background objects from the scene (waves of the sea, branches of the trees), among many others.

There are several approaches in the literature to model the background of a video sequence, employing different techniques like mixtures of Gaussians or probabilistic neural networks. In this paper we present a model based on probabilistic self-organising maps, with a suitable choice of characteristic pixel features.

The rest of the paper is structured as follows. The methodology from our proposal is described in Sect. 2. The experimental results are shown in Sect. 3. Finally we present our conclusions in Sect. 4.

M. Graña et al. (eds.), *International Joint Conference SOCO'16-CISIS'16-ICEUTE'16*, Advances in Intelligent Systems and Computing 527, DOI 10.1007/978-3-319-47364-2_24

Vehicle Type Detection by Convolutional Neural Networks

Miguel A. Molina-Cabello( ), Rafael Marcos Luque-Baena, Ezequiel López-Rubio, and Karl Thurnhofer-Hemsi

Department of Computer Languages and Computer Science, University of Málaga, Bulevar Louis Pasteur, 35, 29071 Málaga, Spain
{miguelangel,rmluque,ezeqlr,karlkhader}@lcc.uma.es

Abstract. In this work a new vehicle type detection procedure for traffic surveillance videos is proposed. A Convolutional Neural Network is integrated into a vehicle tracking system in order to accomplish this task. Solutions for vehicle overlapping, differing vehicle sizes and poor spatial resolution are presented. The system is tested on well known benchmarks, and multiclass recognition performance results are reported. Our proposal is shown to attain good results over a wide range of difficult situations.

Keywords: Foreground detection · Background modeling · Convolutional neural networks · Probabilistic self-organizing maps · Background features

1 Introduction

Nowadays, research on video surveillance systems is considered a prolific area due to mainly the great amount of available data obtained from any corner of the world. Concretely, the automatic analysis of traffic scenes is particularly relevant since it is possible to detect and avoid traffic congestions, incident and some breaches of road worthiness requirements [11]. Thus, a high-level description of the road sequences which involves the position, speed and class of the vehicles is sufficient to provide useful information about road traffic [9].

Foreground detection is the first step in any generic traffic video surveillance system. There are many algorithms which can model the background. For example, the background of a general scene can be modeled by using a single Gaussian distribution [15] or with a self-organizing neural network [8]. Furthermore, if the scenario is well-known, different techniques can be applied in order to improve the performance. For example, in this particular case of a traffic sequence, there are techniques which consider several facets like foggy or snow conditions [13].

Once an object is detected and tracked along the sequence, a simple labeling task, which identify the type of the object in motion, could be carried out. However, this process is not as straightforward as it seems to be, because a

J.M. Ferrández Vicente et al. (Eds.): IWINAC 2017, Part II, LNCS 10338, pp. 268–278, 2017.
DOI: 10.1007/978-3-319-59773-7_28

Vehicle Classification in Traffic Environments Using the Growing Neural Gas

Miguel A. Molina-Cabello( ), Rafael Marcos Luque-Baena, Ezequiel López-Rubio, Juan Miguel Ortiz-de-Lazcano-Lobato, Enrique Domínguez, and José Muñoz Pérez

Department of Computer Languages and Computer Science, University of Málaga, Bulevar Louis Pasteur, 35, 29071 Málaga, Spain
{miguelangel,rmluque,ezeqlr,jmortiz,enriqued,munozp}@lcc.uma.es
http://www.lcc.uma.es/~ezeqlr/index-en.html

Abstract. Traffic monitoring is one of the most popular applications of automated video surveillance. Classification of the vehicles into types is important in order to provide the human traffic controllers with updated information about the characteristics of the traffic flow, which facilitates their decision making process. In this work, a video surveillance system is proposed to carry out such classification. First of all, a feature extraction process is carried out to obtain the most significant features of the detected vehicles. After that, a set of Growing Neural Gas neural networks is employed to determine their types. A qualitative and quantitative assessment of the proposal is carried out on a set of benchmark traffic video sequences, with favorable results.

Keywords: Foreground detection · Background modeling · Probabilistic self-organising maps · Background features

1 Introduction

The field of traffic monitoring has generated great excitement in recent years within the intelligent transport systems community due to the increase of hardware development, the low cost sensor technologies and the improvement in the development and optimization of data processing algorithms. Specifically, the video detection and monitoring solutions for traffic applications can help to improve the performance in traffic management [3,10,13]. Thus, for example, if a high frequency of heavy vehicles is detected in one of the analyzed road sections, it is possible to redirect the traffic in a previous point with the aim of avoiding traffic congestion.

Automatic video surveillance systems can be divided into several phases [1,2]. A first step involves the detection of moving objects within the scene; a second stage performs monitoring tasks to associate the same vehicle detected in all frames of the sequence in which it appears; and finally a feature detection phase to extract relevant knowledge of the movement of these objects, their behavior and appearance.

I. Rojas et al. (Eds.): IWANN 2017, Part II, LNCS 10306, pp. 225–234, 2017.
DOI: 10.1007/978-3-319-59147-6_20

Neural Controller for PTZ cameras based on nonpanoramic foreground detection

Miguel A. Molina-Cabello*, Ezequiel López-Rubio*, Rafael Marcos Luque-Baena*,
Enrique Domínguez* and Karl Thurnhofer-Hemsi*
*Department of Computer Languages and Computer Science
University of Málaga, Bulevar Louis Pasteur, 35, 29071 Málaga, Spain
Emails: {miguelangel,ezeqlr,rmluque,enriqued,karlkhader}@lcc.uma.es

***Abstract*—In this paper a controller for PTZ cameras based on an unsupervised neural network model is presented. It takes advantage of the foreground mask generated by a non-parametric foreground detection subsystem. Thus, our aim is to optimize the movements of the PTZ camera to attain the maximum coverage of the observed scene in presence of moving objects. A growing neural gas (GNG) is applied to enhance the representation of the foreground objects. Both qualitative and quantitative results are reported using several widely used datasets, which demonstrate the suitability of our approach.**

1. Introduction

Most of the former surveillance systems were built with a single stationary camera for many years. However, nowadays it is possible to find different types of cameras and any surveillance system is frequently composed by multiple devices which try to cover the largest possible area. [1]. Among other types or criteria, two of the most used types of camera are the omnidirectional and the pan-tilt-zoom (PTZ) cameras. Conventional surveillance systems usually comprise at least one omnidirectional camera and one PTZ camera.

Surveillance systems are capable of monitoring the entire scene by using a single omnidirectional camera but, due to the limited resolution of these panoramic cameras, detailed information of the objects might not be acquired. As a result of that, a PTZ camera is used for those tasks, which require close-up views at high resolution.

PTZ cameras are well suited for object identification and recognition in far-field scenes. However, the practical use of PTZ cameras in real world scenarios is complicated due to several reasons [2]. A continuous online camera calibration is needed since the absolute pan, tilt and zoom positional values provided by the camera actuators are not synchronized with the video stream in most cases. Moreover, some adaptive background representation becomes necessary to make target tracking, since the scene background is continuously changing due to the camera operation [3].

Conventional camera systems are usually easily customizable and let users deploy the sensing infrastructures according to their needs, by adjusting the various camera parameters such as field of view, resolution, operating mode (night/day vision, indoor/outdoor) [4]. In most cases the design of the infrastructure of a surveillance system is performed manually, although the wide range of configurations and parameter settings leads to suboptimal solutions, which imply an incomplete coverage of the monitored area or, conversely, higher deployment costs to achieve a satisfactory result [5]. As a general rule, we can assume that the goal of a camera planner is to guarantee the maximum coverage of the observed space, minimizing occlusions and obtaining the best visibility of the objects of interest [6].

In this paper, we propose a method for PTZ cameras based on Growing Neural Gas (GNG) models in order to automatically determine the position and pan-tilt-zoom settings to optimize the coverage of the foreground.

Traditional foreground algorithms identify foreground pixels because their features are different from those of the background, but this leads to false detection for moving cameras. Apart of other proposals based on building a panoramic model of the scene, we are focusing on non-panoramic methods [7], [8] which are suitable for free moving cameras. The use of the neural approach filters the noise and spurious objects obtained in the foreground mask. Furthermore the moving objects in the scene are represented with higher accuracy and robustness. These are the key elements to design an effective PTZ controller.

GNG models are a type of self-organizing neural networks and one of the most successful example of unsupervised learning in a graph. The original GNG model was proposed by Fritzke [9] and has become a standard of applications in computer vision [10] and robotics [11], as well as other self-organizing models for foreground detection [12] or object tracking [13] in video sequences.

The rest of the paper is organized as follows. In section 2, the proposed control system for PTZ cameras is presented. In section 3, we report the results achieved with the proposed neural controller. Finally, section 4 includes some concluding remarks.

2. System architecture

In this section the architecture of the proposed PTZ camera control system is described. The system is made of three modules, namely a foreground detection procedure (Subsection 2.1), an unsupervised learning model to learn

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

Blood Cell Classification Using the Hough Transform and Convolutional Neural Networks

Miguel A. Molina-Cabello( ), Ezequiel López-Rubio, Rafael M. Luque-Baena, María Jesús Rodríguez-Espinosa, and Karl Thurnhofer-Hemsi

Department of Computer Languages and Computer Science, University of Málaga, Bulevar Louis Pasteur, 35, 29071 Málaga, Spain
{miguelangel,ezeqlr,rmluque,karlkhader}@lcc.uma.es, mjesus.rodriguez.espinosa@hotmail.com

Abstract. The detection of red blood cells in blood samples can be crucial for the disease detection in its early stages. The use of image processing techniques can accelerate and improve the effectiveness and efficiency of this detection. In this work, the use of the Circle Hough transform for cell detection and artificial neural networks for their identification as a red blood cell is proposed. Specifically, the application of neural networks (MLP) as a standard classification technique with (MLP) is compared with new proposals related to deep learning such as convolutional neural networks (CNNs). The different experiments carried out reveal the high classification ratio and show promising results after the application of the CNNs.

Keywords: Blood cell detection · Blood cell classification
Circle hough transform · Convolutional neural networks

1 Introduction

Digital image processing is of paramount importance in various medicine fields, from images which are obtained in medical tests such as X-ray image, computed tomography, magnetic resonance imaging, ultrasound image and nuclear medicine image, to images which are obtained in laboratory by microscopy. In this way, digital image processing allows to process the image to obtain a better visibility, to emphasize the required parts or to make analysis and predictions. In addition, image segmentation through image processing is essential for pathology detection [4,12].

Nowadays, most of images obtained in laboratory by microscopy are digitalized afterwards and processed by computers to make easier and faster the image analysis process. Among other fields, blood optical microscopy provides image samples whose study can supply very useful information about patient health. The techniques applied in hematology to count blood cells in blood samples,

Á. Rocha et al. (Eds.): WorldCIST'18 2018, AISC 746, pp. 669–678, 2018.
https://doi.org/10.1007/978-3-319-77712-2_62

Bibliografía

La mejor vida no es la más larga, sino la más rica en buenas acciones.

Marie Curie

Abuassba, A. O., Zhang, D., Luo, X., Shaheryar, A. y Ali, H. Improving classification performance through an advanced ensemble based heterogeneous extreme learning machines. *Computational intelligence and neuroscience*, vol. 2017, 2017.

Adeli, H. y Hung, S. A concurrent adaptive conjugate gradient learning algorithm on mimd shared-memory machines. *Int. J. High Perform. Comput. Appl.*, vol. 7(2), páginas 155–166, 1993. ISSN 1094-3420.

Adeli, H. y Hung, S. An adaptive conjugate gradient learning algorithm for efficient training of neural networks. *Applied Mathematics and Computation*, vol. 62(1), páginas 81 – 102, 1994.

Adeli, H. y Hung, S. *Machine Learning - Neural Networks, Genetic Algorithms, and Fuzzy Systems*. John Wiley and Sons, New York, 1995. ISBN 0471016330.

Adnan, L., Yussoff, Y., Johar, H. y Baki, S. Energy-saving street lighting system based on the waspmote mote. *Jurnal Teknologi*, vol. 76(4), páginas 55–58, 2015.

Aggarwal, J. y Ryoo, M. Human activity analysis: A review. *ACM Computing Surveys*, vol. 43(3), 2011.

Aleksendrić, D., Jakovljević, I. y Irović, V. Intelligent control of braking process. *Expert Systems with Applications*, vol. 39(14), 2012.

Ali, M., Khan, M., Tung, N. T. et al. Segmentation of dental x-ray images in medical imaging using neutrosophic orthogonal matrices. *Expert Systems with Applications*, vol. 91, páginas 434–441, 2018.

AMATO, G., CARRARA, F., FALCHI, F., GENNARO, C., MEGHINI, C. y VAIRO, C. Deep learning for decentralized parking lot occupancy detection. *Expert Systems with Applications*, vol. 72, páginas 327–334, 2017.

ARTHUR, D. y VASSILVITSKII, S. k-means++: The advantages of careful seeding. En *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, páginas 1027–1035. Society for Industrial and Applied Mathematics, 2007.

ATMEL. DataSheet Atmel SAM3X8E ARM Cortex-M3 CPU. `http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-11057-32-bit-Cortex-M3-Microcontroller-SAM3X-SAM3A_Datasheet.pdf`, 2016. En línea; accedida el 29 de junio de 2018.

BAF, F., BOUWMANS, T. y VACHON, B. Type-2 fuzzy mixture of gaussians model: Application to background modeling. En *Proceedings of the 4th International Symposium on Advances in Visual Computing (ISVC)*, páginas 772–781. 2008.

BAGUI, O. K. y ZOUEU, J. T. Red blood cells counting by circular hough transform using multispectral images. *Journal of Applied Sciences*, vol. 14(24), páginas 3591–3594, 2014.

BALAFAR, M. Gaussian mixture model based segmentation methods for brain mri images. *Artificial Intelligence Review*, vol. 41(3), páginas 429–439, 2014.

BANERJEE, A., MERUGU, S., DHILLON, I. S. y GHOSH, J. Clustering with Bregman divergences. *Journal of Machine Learning Research*, vol. 6, páginas 1705–1749, 2005.

BAR-SHALOM, Y. *Tracking and Data Association*. 1987. ISBN 0-120-79760-7.

BAUMANN, A., BOLTZ, M., EBLING, J., KOENIG, M., LOOS, H., MERKEL, M., NIEM, W., WARZELHAN, J. y YU, J. A review and comparison of measures for automatic video surveillance systems. *Eurasip Journal on Image and Video Processing*, vol. 2008, 2008.

BAUMGARTNER, J., FLESIA, A. G., GIMENEZ, J. y PUCHETA, J. A new image segmentation framework based on two-dimensional hidden markov models. *Integrated Computer-Aided Engineering*, vol. 23(1), páginas 1–13, 2016.

BAXT, W. G. Application of artificial neural networks to clinical medicine. *The lancet*, vol. 346(8983), páginas 1135–1138, 1995.

BEATON, D., VALOVA, I. y MACLEAN, D. CQoCO: A measure for comparative quality of coverage and organization for self-organizing maps. *Neurocomputing*, vol. 73(10-12), páginas 2147–2159, 2010.

BENGIO, Y., GOODFELLOW, I. J. y COURVILLE, A. Deep learning, 2015. Book in preparation for MIT Press.

BERMEJO, S. y CABESTANY, J. The effect of finite sample size on on-line k-means. *Neurocomputing*, vol. 48(1), páginas 511–539, 2002.

BHANDARKAR, S., KOH, J. y SUK, M. Multiscale image segmentation using a hierarchical self-organizing map. *Neurocomputing*, vol. 14(3), páginas 241–272, 1997.

BIANCO, S., CIOCCA, G. y SCHETTINI, R. How far can you get by combining change detection algorithms? En *International Conference on Image Analysis and Processing*, páginas 96–107. 2017.

BOUWMANS, T. *Background Modeling and Foreground Detection for Video Surveillance*, capítulo Traditional Approaches in Background Modeling for Static Cameras, páginas 1–54. Chapman and Hall/CRC, 2014a.

BOUWMANS, T. Traditional and recent approaches in background modeling for foreground detection: An overview. *Computer Science Review*, vol. 11-12, páginas 31–66, 2014b.

BRAHAM, M. y DROOGENBROECK, M. V. Deep background subtraction with scene-specific convolutional neural networks. En *International Conference on Systems, Signals and Image Processing (IWSSIP)*, páginas 1–4. Bratislava, Slovakia, 2016.

BREGMAN, L. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, vol. 7(3), páginas 200–217, 1967.

BREIMAN, L. Random forests. *Machine Learning*, vol. 45(1), páginas 5–32, 2001.

BUCH, N., VELASTIN, S. y ORWELL, J. A review of computer vision techniques for the analysis of urban traffic. *IEEE Transactions on Intelligent Transportation Systems*, vol. 12(3), páginas 920–939, 2011.

CASANOVA, C., FRANCO, A., LUMINI, A. y MAIO, D. Smartvisionapp: A framework for computer vision applications on mobile devices. *Expert Systems with Applications*, vol. 40(15), páginas 5884 – 5894, 2013.

CELA, A., YEBES, J. J., ARROYO, R., BERGASA, L. M., BAREA, R. y LÓPEZ, E. Complete low-cost implementation of a teleoperated control system for a humanoid robot. *Sensors*, vol. 13(2), páginas 1385–1401, 2013.

CENSOR, Y. y ZENIOS, S. *Parallel Optimization: Theory, Algorithms, and Applications*. Oxford University Press, 1998.

CHA, Y.-J., CHOI, W. y BÜYÜKÖZTÜRK, O. Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, vol. 32(5), páginas 361–378, 2017.

CHAKRABARTTY, S., SHAGA, R. y AONO, K. Noise-shaping gradient descent-based online adaptation algorithms for digital calibration of analog circuits. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24(4), páginas 554–565, 2013.

CHANG, C.-H., PENGFEI, X., XIAO, R. y SRIKANTHAN, T. New adaptive color quantization method based on self-organizing maps. *IEEE Transactions on Neural Networks*, vol. 16(1), páginas 237–249, 2005.

CHEN, D., HAN, X., CHENG, R. y YANG, L. Position calculation models by neural computing and online learning methods for high-speed train. *Neural Computing and Applications*, vol. 27(6), páginas 1617–1628, 2016a.

CHEN, G., ST-CHARLES, P.-L., BOUACHIR, W., BILODEAU, G.-A. y BERGEVIN, R. Reproducible evaluation of pan-tilt-zoom tracking. En *Image Processing (ICIP), 2015 IEEE International Conference on*, páginas 2055–2059. IEEE, 2015.

CHEN, X., HENRICKSON, K. y WANG, Y. Kinect-based pedestrian detection for crowded scenes. *Computer-Aided Civil and Infrastructure Engineering*, vol. 31(3), páginas 229–240, 2016b.

CHENG, H.-Y. y HSU, S.-H. Intelligent highway traffic surveillance with self-diagnosis abilities. *IEEE Transactions on Intelligent Transportation Systems*, vol. 12(4), páginas 1462–1472, 2011.

CHIRA, C., SEDANO, J., CAMARA, M., PRIETO, C., VILLAR, J. R. y CORCHADO, E. A cluster merging method for time series microarray with production values. *International Journal of Neural Systems*, vol. 24(6), página 1450018, 2014.

CLARK, J. Neural network modelling. *Physics in Medicine and Biology*, vol. 36(10), página 1259, 1991.

CORNBLEET, J. Spurious results from automated hematology cell counters. *Laboratory Medicine*, vol. 14(8), páginas 509–514, 2016.

CORTES, C. y VAPNIK, V. Support-vector networks. *Machine Learning*, vol. 20(3), páginas 273–297, 1995.

CROUZIL, A., KHOUDOUR, L., VALIERE, P. y TRUONG CONG, D. Automatic vehicle counting system for traffic monitoring. *Journal of Electronic Imaging*, vol. 25(5), 2016.

CUCCHIARA, G. y PICCARDI, P. Detecting moving objects, ghosts and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25(10), páginas 1337–1342, 2003.

DAVIS, R. y BOYERS, S. The role of digital image analysis in reproductive biology and medicine. *Archives of pathology & laboratory medicine*, vol. 116(4), páginas 351–363, 1992.

DEKKER, A. Kohonen neural networks for optimal color quantization. *Network: Computation in Neural Systems*, vol. 5, páginas 351–367, 1994.

DELYON, B., LAVIELLE, M. y MOULINES, E. Convergence of a stochastic approximation version of the EM algorithm. *Annals of Statistics*, vol. 27(1), páginas 94–128, 1999.

DLUGOSZ, R., TALASKA, T., PEDRYCZ, W. y WOJTYNA, R. Realization of the conscience mechanism in CMOS implementation of winner-takes-all self-organizing neural networks. *IEEE Transactions on Neural Networks*, vol. 21(6), páginas 961–971, 2010.

DOBRZYNSKI, M., PERICET-CAMARA, R. y FLOREANO, D. Vision tape-a flexible compound vision sensor for motion detection and proximity estimation. *IEEE Sensors Journal*, vol. 12(5), páginas 1131–1139, 2012.

DONG, G. y XIE, M. Color clustering and learning for image segmentation based on neural networks. *IEEE Transactions on Neural Networks*, vol. 16(4), páginas 925–936, 2005.

ECABERT, O. y THIRAN, J.-P. Adaptive hough transform for the detection of natural shapes under weak affine transformations. *Pattern Recognition Letters*, vol. 25(12), páginas 1411–1419, 2004.

EGBERT, D. D., KABURLASOS, V. G. y GOODMAN, P. H. Neural network discrimination of subtle image patterns. En *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*, páginas 517–524. IEEE, 1990.

EL BAF, F., BOUWMANS, T. y VACHON, B. Fuzzy integral for moving object detection. En *IEEE International Conference on Fuzzy Systems*, páginas 1729–1736. Hong Kong, 2008.

ELGAMMAL, A., DURAISWAMI, R., HARWOOD, D. y DAVIS, L. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. En *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, páginas 1151–1163. 2002.

ELGAMMAL, A., HARWOOD, D. y DAVIS, L. Non-parametric model for background subtraction. En *Computer Vision (ECCV)*, páginas 751–767. Springer, 2000.

EVERINGHAM, M., VAN GOOL, L., WILLIAMS, C. K. I., WINN, J. y ZISSERMAN, A. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://host.robots.ox.ac.uk/pascal/VOC/voc2007/, 2018. En línea; accedida el 29 de junio de 2018.

FEDOSOV, D. A., CASWELL, B. y KARNIADAKIS, G. E. A multiscale red blood cell model with accurate mechanics, rheology, and dynamics. *Biophysical journal*, vol. 98(10), páginas 2215–2225, 2010.

FERNÁNDEZ, A., CARMONA, C. J., DEL JESUS, M. J. y HERRERA, F. A pareto based ensemble with feature and instance selection for learning from multi-class imbalanced datasets. *International Journal of Neural Systems*, 2017.

FINK, O., ZIO, E. y WEIDMANN, U. Novelty detection by multivariate kernel density estimation and growing neural gas algorithm. *Mechanical Systems and Signal Processing*, vol. 50–51, páginas 427 – 436, 2015.

FRIEDMAN, N. y RUSSELL, S. Image segmentation in video sequences: A probabilistic approach. En *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, páginas 175–181. 1997. ISBN 1-55860-485-5.

FRITZKE, B. A growing neural gas network learns topologies. *Advances in Neural Information Processing Systems*, vol. 7, páginas 625–632, 1995.

FUNG, V., BOSCH, J., ROBERTS, S. y KLEISSL, J. Cloud shadow speed sensor. *Atmospheric Measurement Techniques*, vol. 7(6), páginas 1693–1700, 2014.

GARCÍA, F., GARCÍA, J., PONZ, A., DE LA ESCALERA, A. y ARMINGOL, J. M. Context aided pedestrian detection for danger estimation based on laser scanner and computer vision. *Expert Systems with Applications*, vol. 41(15), páginas 6646 – 6661, 2014.

GARCÍA, J. F., TOMÁS, V. R., GARCÍA, L. A. y MARTÍNEZ, J. J. A negotiation protocol to improve long distance truck parking. *Integrated Computer-Aided Engineering*, vol. 24(2), páginas 157–170, 2017.

GERONIMO, D., LOPEZ, A. M., SAPPA, A. D. y GRAF, T. Survey of pedestrian detection for advanced driver assistance systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32(7), páginas 1239–1258, 2010. ISSN 0162-8828.

GIL WHOAN CHU y MYUNG JIN CHUNG. Selection of an optimal camera position using visibility and manipulability measures for an active camera system. En *Proceedings International Conference on Intelligent Robots and Systems (IROS 2000)*, vol. 1, páginas 429–434. IEEE, 2000. ISBN 0-7803-6348-5.

GIRSHICK, R., DONAHUE, J., DARRELL, T. y MALIK, J. Rich feature hierarchies for accurate object detection and semantic segmentation. páginas 580–587. 2014.

GIRSHICK, R. B. Fast R-CNN. *CoRR*, vol. abs/1504.08083, 2015.

GÓMEZ, M. J., GARCÍA, F., MARTÍN, D., DE LA ESCALERA, A. y ARMINGOL, J. M. Intelligent surveillance of indoor environments based on computer vision and 3D point cloud fusion. *Expert Systems with Applications*, vol. 42(21), páginas 8156 – 8171, 2015.

GOYETTE, N., JODOIN, P.-M., PORIKLI, F., KONRAD, J. y ISHWAR, P. Changedetection. net: A new change detection benchmark dataset. En *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, páginas 1–8. IEEE, 2012.

GRIMSON, W., STAUFFER, C., ROMANO, R. y LEE, L. Using adaptive tracking to classify and monitor activities in a site. En *Conference on Computer Vision and Pattern Recognition (CVPR)*, páginas 22–29. 1998.

GURUBEL, K. J., ALANIS, A. Y., SANCHEZ, E. N. y CARLOS-HERNANDEZ, S. A neural observer with time-varying learning rate: analysis and applications. *International Journal of Neural Systems*, vol. 24(1), página 1450011, 2014.

HARITAOGLU, I., HARWOOD, D. y DAVIS, L. W4: real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22(8), páginas 809–830, 2000.

HE, J., TAN, A.-H., TAN, C.-L. y SUNG, S.-Y. On quantitative evaluation of clustering systems. En *Clustering and information retrieval*, páginas 105–133. Springer, 2004.

HE, K., ZHANG, X., REN, S. y SUN, J. Deep residual learning for image recognition. En *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 770–778. 2016.

HÖRSTER, E. y LIENHART, R. On the optimal placement of multiple visual sensors. En *Proceedings of the 4th ACM International Workshop on Video Surveillance and Sensor Networks*, VSSN '06, páginas 111–120. ACM, New York, NY, USA, 2006. ISBN 1-59593-496-0.

HSU, A. y HALGAMUGE, S. Enhancement of topology preservation and hierarchical dynamic self-organising maps for data visualisation. *International Journal of Approximate Reasoning*, vol. 32(2-3), páginas 259–279, 2003.

HUANG, D.-Y., CHEN, C.-H., CHEN, T.-Y., HU, W.-C. y LIN, Y.-L. A vehicle flow counting system in rainy environment based on vehicle feature analysis. *Journal of Information Hiding and Multimedia Signal Processing*, vol. 7(1), páginas 101–114, 2016.

HUNG, S. y ADELI, H. Parallel backpropagation learning algorithms on {CRAY} y-mp8/864 supercomputer. *Neurocomputing*, vol. 5(6), páginas 287 – 302, 1993.

HUNG, S. y ADELI, H. A parallel genetic/neural network learning algorithm for MIMD shared memory machines. *IEEE Transactions on Neural Networks*, vol. 5(6), páginas 900–909, 1994.

IACCA, G., CARAFFINI, F. y NERI, F. Continuous parameter pools in ensemble differential evolution. En *2015 IEEE Symposium Series on Computational Intelligence*, páginas 1529–1536. 2015.

IACCA, G., NERI, F., CARAFFINI, F. y SUGANTHAN, P. N. A differential evolution framework with ensemble of parameters and strategies and pool of local search algorithms. En *Applications of Evolutionary Computation* (editado por A. I. Esparcia-Alcázar y A. M. Mora), páginas 615–626. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

IMERI, F., HERKLOTZ, R., RISCH, L., ARBETSLEITNER, C., ZERLAUTH, M., RISCH, G. M. y HUBER, A. R. Stability of hematological analytes depends on the hematology analyser used: a stability study with bayer advia 120, beckman coulter lh 750 and sysmex xe 2100. *Clinica chimica acta*, vol. 397(1), páginas 68–71, 2008.

ISLAM, M. M. y YAO, X. *Evolving Artificial Neural Network Ensembles*, páginas 851–880. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-78293-3.

JAIN, A. K. y DUBES, R. C. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

KAMIJO, S., MATSUSHITA, Y., IKEUCHI, K. y SAKAUCHI, M. Traffic monitoring and accident detection at intersections. *IEEE Transactions on Intelligent Transportation Systems*, vol. 1(2), páginas 108–117, 2000.

KATO, N., FADLULLAH, Z. M., MAO, B., TANG, F., AKASHI, O., INOUE, T. y MIZUTANI, K. The deep learning vision for heterogeneous network traffic control: Proposal, challenges, and future perspective. *IEEE Wireless Communications*, 2016.

KIM, K., CHALIDABHONGSE, T. H., HARWOOD, D. y DAVIS, L. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, vol. 11(3), páginas 172 – 185, 2005.

KIM, S., YUN, K., YI, K., KIM, S. y CHOI, J. Detection of moving objects with a moving camera using non-panoramic background model. *Machine Vision and Applications*, vol. 24(5), páginas 1015–1028, 2013.

KIM, W. y KIM, Y. Background subtraction using illumination-invariant structural complexity. *IEEE Signal Processing Letters*, vol. 23(5), páginas 634–638, 2016.

KOHONEN, T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, vol. 43(1), páginas 59–69, 1982. ISSN 0340-1200.

KOHONEN, T. The self-organizing map. *Proceedings of the IEEE*, vol. 78(9), páginas 1464–1480, 1990.

KOHONEN, T. Essentials of the self-organizing map. *Neural Networks*, vol. 37, páginas 52 – 65, 2013.

KOHONEN, T. y HONKELA, T. Kohonen network. *Scholarpedia*, vol. 2(1), página 1568, 2007.

KONDA, K., CONCI, N. y DE NATALE, F. Global coverage maximization in ptz-camera networks based on visual quality assessment. *IEEE Sensors Journal*, vol. 16(16), páginas 6317–6332, 2016.

KOPETZ, H. *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Kluwer Academic Publishers, Norwell, MA, USA, 1st edición, 1997. ISBN 0792398947.

KORNUTA, J. A., NIPPER, M. E. y BRANDON DIXON, J. Low-cost microcontroller platform for studying lymphatic biomechanics in vitro. *Journal of Biomechanics*, vol. 46(1), páginas 183–186, 2012.

KOZIARSKI, M. y CYGANEK, B. Image recognition with deep neural networks in presence of noise–dealing with and taking advantage of distortions. *Integrated Computer-Aided Engineering*, vol. 24(4), páginas 337–349, 2017.

KRAUSE, J. Automated differentials in the hematology laboratory. *American journal of clinical pathology*, vol. 93(4 Suppl 1), páginas S11–6, 1990.

KRIZHEVSKY, A., SUTSKEVER, I. y HINTON, G. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, vol. 25, páginas 1097–1105, 2012.

KUSHNER, H. J. y YIN, G. G. *Stochastic approximation and Recursive Algorithms and Applications*. Springer-Verlag, New York, NY, USA, 2003.

LACABEX, B., CUESTA-INFANTE, A., MONTEMAYOR, A. S. y PANTRIGO, J. J. Lightweight tracking-by-detection system for multiple pedestrian targets. *Integrated computer-aided engineering*, vol. 23(3), páginas 299–311, 2016.

LACERDA, E. B. y MELLO, C. A. Segmentation of connected handwritten digits using self-organizing maps. *Expert Systems with Applications*, vol. 40(15), páginas 5867 – 5877, 2013.

LAI, T. Stochastic approximation. *Annals of Statistics*, vol. 31(2), páginas 391–406, 2003.

LANTIS, K. L., HARRIS, R. J., DAVIS, G., RENNER, N. y FINN, W. G. Elimination of instrument-driven reflex manual differential leukocyte counts: optimization of manual blood smear review criteria in a high-volume automated hematology laboratory. *American journal of clinical pathology*, vol. 119(5), páginas 656–662, 2003.

LECUN, Y., BENGIO, Y. y HINTON, G. Deep learning. *Nature*, vol. 521, páginas 436–444, 2015.

LI, L., HUANG, W., GU, I. Y.-H. y TIAN, Q. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing*, vol. 13(11), páginas 1459–1472, 2004.

LIAN, K.-Y., HSIAO, S.-J. y SUNG, W.-T. Intelligent multi-sensor control system based on innovative technology integration via ZigBee and Wi-Fi networks. *Journal of Network and Computer Applications*, vol. 36(2), páginas 756–767, 2013.

LIANG, M., HUANG, X., CHEN, C.-H., CHEN, X. y TOKUTA, A. Counting and classification of highway vehicles by regression analysis. *IEEE Transactions on Intelligent Transportation Systems*, vol. 16(5), páginas 2878–2888, 2015.

Lin, Y.-z., Nie, Z.-h. y Ma, H.-w. Structural damage detection with automatic feature-extraction through deep learning. *Computer-Aided Civil and Infrastructure Engineering*, vol. 32(12), páginas 1025–1046, 2017.

Lisanti, G., Masi, I., Pernici, F. y Del Bimbo, A. Continuous localization and mapping of a panâ€“tiltâ€“zoom camera for wide area tracking. *Machine Vision and Applications*, vol. 27(7), páginas 1071–1085, 2016.

Lloyd, S. Least squares quantization in pcm. *IEEE transactions on information theory*, vol. 28(2), páginas 129–137, 1982.

López-Rubio, E. Superresolution from a single noisy image by the median filter transform. *SIAM Journal on Imaging Sciences*, vol. 9(1), páginas 82–115, 2016.

López-Rubio, E., Luque-Baena, R. y Domínguez, E. Foreground detection in video sequences with probabilistic self-organizing maps. *International Journal of Neural Systems*, vol. 21(3), páginas 225–246, 2011a.

López-Rubio, E. y Luque-Baena, R. M. Stochastic approximation for background modelling. *Computer Vision and Image Understanding*, vol. 115(6), páginas 735 – 749, 2011.

López-Rubio, E., Luque-Baena, R. M. y Domínguez, E. Foreground detection in video sequences with probabilistic self-organizing maps. *International Journal of Neural Systems*, vol. 21(3), páginas 225–246, 2011b.

López-Rubio, E., Molina-Cabello, M. A., Luque-Baena, R. M. y Domínguez, E. Foreground detection by competitive learning for varying input distributions. *International journal of neural systems*, vol. 28(05), página 1750056, 2018a.

López-Rubio, E., Ortiz-de-Lazcano-Lobato, J. M. y López-Rodríguez, D. Probabilistic PCA self-organizing maps. *IEEE Transactions on Neural Networks*, vol. 20(9), páginas 1474–1489, 2009. ISSN 1045-9227.

López-Rubio, E., Palomo, E. J. y Domínguez, E. Bregman divergences for growing hierarchical self-organizing networks. *International Journal of Neural Systems*, vol. 24(4), página 1450016, 2014.

López-Rubio, E., Palomo-Ferrer, E. J., Ortiz-de Lazcano-Lobato, J. M. y Vargas-González, M. C. Dynamic topology learning with the probabilistic self-organizing graph. *Neurocomputing*, vol. 74(16), páginas 2633–2648, 2011.

López-Rubio, F. J. y López-Rubio, E. Features for stochastic approximation based foreground detection. *Computer Vision and Image Understanding*, vol. 133, páginas 30 – 50, 2015.

LÓPEZ-RUBIO, F. J. y LÓPEZ-RUBIO, E. Foreground detection for moving cameras with stochastic approximation. *Pattern Recognition Letters*, vol. 68, páginas 161–168, 2015.

LÓPEZ-RUBIO, F. J. y LÓPEZ-RUBIO, E. Local color transformation analysis for sudden illumination change detection. *Image Vision Comput.*, vol. 37, páginas 31–47, 2015.

LÓPEZ-RUBIO, F. J., LÓPEZ-RUBIO, E., MOLINA-CABELLO, M. A., LUQUE-BAENA, R. M., PALOMO, E. J. y DOMÍNGUEZ, E. The effect of noise on foreground detection algorithms. *Artificial Intelligence Review*, vol. 49(3), páginas 407–438, 2018b.

LOZHKINA, O. V. y LOZHKIN, V. N. Estimation of road transport related air pollution in saint petersburg using european and russian calculation models. *Transportation Research Part D: Transport and Environment*, vol. 36, páginas 178–189, 2015.

LUQUE, R., DOMÍNGUEZ, E., MUÑOZ, J. y PALOMO, E. Un modelo neuronal de agrupamiento basado en regiones para segmentación de vídeo. En *XIII Conference of the Spanish Association for Artificial Intelligence (CAEPIA)*, páginas 243 – 252. 2009.

LUQUE, R. M., DOMÍNGUEZ, E., PALOMO, E. J. y MUÑOZ, J. A neural network approach for video object segmentation in traffic surveillance. En *International Conference of Image Analysis and Recognition (ICIAR)*, páginas 151 – 158. Póvoa de Varzim, Portugal, 2008a. ISBN 978-3-540-69812-8.

LUQUE, R. M., DOMÍNGUEZ, E., PALOMO, E. J. y MUÑOZ, J. An ART-type network approach for video object detection. En *European Symposium on Artificial Neural Network (ESANN)*, páginas 423 – 428. Bruges, Belgium, 2010.

LUQUE, R. M., LÓPEZ-RODRÍGUEZ, D., MÉRIDA-CASERMEIRO, E. y PALOMO, E. J. Video object segmentation with multivalued neural networks. En *Eighth International Conference on Hybrid Intelligent Systems, HIS*, páginas 613–618. 2008b.

LUQUE-BAENA, R., LÓPEZ-RUBIO, E., DOMÍNGUEZ, E., PALOMO, E. y JEREZ, J. A self-organizing map to improve vehicle detection in flow monitoring systems. *Soft Computing*, vol. 19(9), páginas 2499–2509, 2015a.

LUQUE BAENA, R. M., DOMÍNGUEZ, E., LÓPEZ-RODRÍGUEZ, D. y PALOMO, E. J. A neighborhood-based competitive network for video segmentation and object detection. En *International Conference on Artificial Neural Networks (ICANN)*, páginas 877–886. Prague, Czech Republic, 2008. ISBN 978-3-540-87536-9.

Luque-Baena, R. M., Ortiz-de Lazcano-Lobato, J. M., López-Rubio, E., Domínguez, E. y Palomo, E. J. A competitive neural network for multiple object tracking in video sequence analysis. *Neural Processing Letters*, vol. 37(1), páginas 47–67, 2013.

Luque-Baena, R. M., López-Rubio, E., Domínguez, E., Palomo, E. J. y Jerez, J. M. A self-organizing map to improve vehicle detection in flow monitoring systems. *Soft Computing*, vol. 19(9), páginas 2499–2509, 2015b.

Maddalena, L. y Petrosino, A. A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing*, vol. 17(7), páginas 1168–1177, 2008.

Maddalena, L. y Petrosino, A. The sobs algorithm: what are the limits? En *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, páginas 21–26. IEEE, 2012.

Mahmoud, S., Lotfi, A. y Langensiepen, C. Behavioural pattern identification and prediction in intelligent environments. *Applied Soft Computing*, vol. 13(4), páginas 1813–1822, 2013.

Maiseli, B. J., Elisha, O. A. y Gao, H. A multi-frame super-resolution method based on the variable-exponent nonlinear diffusion regularizer. *EURASIP Journal on Image and Video Processing*, vol. 2015(1), página 22, 2015.

Mamdoohi, G., Fauzi Abas, A., Samsudin, K., Ibrahim, N. H., Hidayat, A. y Mahdi, M. A. Implementation of genetic algorithm in an embedded microcontroller-based polarization control system. *Eng. Appl. Artif. Intell.*, vol. 25(4), páginas 869–873, 2012.

Martinez, F., Pissaloux, E. y Carbone, A. Towards activity recognition from eye-movements using contextual temporal learning. *Integrated Computer-Aided Engineering*, vol. 24(1), páginas 1–16, 2017.

Marwedel, P. *Embedded System Design*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 1402076908.

Mazalan, S. M., Mahmood, N. H. y Razak, M. A. A. Automated red blood cells counting in peripheral blood smear image using circular hough transform. En *Artificial Intelligence, Modelling and Simulation (AIMS), 2013 1st International Conference on*, páginas 320–324. IEEE, 2013.

McAuliffe, M. J., Lalonde, F. M., McGarry, D., Gandler, W., Csaky, K. y Trus, B. L. Medical image processing, analysis and visualization in clinical research. En *Computer-Based Medical Systems, 2001.*

CBMS 2001. Proceedings. 14th IEEE Symposium on, páginas 381–386. IEEE, 2001.

MENÉNDEZ, H. D., BARRERO, D. F. y CAMACHO, D. A genetic graph-based approach for partitional clustering. *International Journal of Neural Systems*, vol. 24(3), página 1430008, 2014.

MITHUN, N., HOWLADER, T. y RAHMAN, S. Video-based tracking of vehicles using multiple time-spatial images. *Expert Systems with Applications*, vol. 62, páginas 17–31, 2016.

MOLINA-CABELLO, M. A., LÓPEZ-RUBIO, E., LUQUE-BAENA, R. M., DOMÍNGUEZ, E. y PALOMO, E. J. Pixel features for self-organizing map based detection of foreground objects in dynamic environments. En *International Joint Conference SOCO'16-CISIS'16-ICEUTE'16*, páginas 247–255. Springer, 2016a.

MOLINA-CABELLO, M. A., LÓPEZ-RUBIO, E., LUQUE-BAENA, R. M., DOMÍNGUEZ, E. y THURNHOFER-HEMSI, K. Neural controller for ptz cameras based on nonpanoramic foreground detection. En *Neural Networks (IJCNN), 2017 International Joint Conference on*, páginas 404–411. IEEE, 2017a.

MOLINA-CABELLO, M. A., LÓPEZ-RUBIO, E., LUQUE-BAENA, R. M., PALOMO, E. J. y DOMÍNGUEZ, E. Frame size reduction for foreground detection in video sequences. En *Conference of the Spanish Association for Artificial Intelligence*, páginas 3–12. Springer, 2016b.

MOLINA-CABELLO, M. A., LÓPEZ-RUBIO, E., LUQUE-BAENA, R. M., RODRÍGUEZ-ESPINOSA, M. J. y THURNHOFER-HEMSI, K. Blood cell classification using the hough transform and convolutional neural networks. En *World Conference on Information Systems and Technologies*, páginas 669–678. Springer, 2018.

MOLINA-CABELLO, M. A., LUQUE-BAENA, R. M., LÓPEZ-RUBIO, E., ORTIZ-DE LAZCANO-LOBATO, J. M., DOMÍNGUEZ, E. y PÉREZ, J. M. Vehicle classification in traffic environments using the growing neural gas. En *International Work-Conference on Artificial Neural Networks*, páginas 225–234. Springer, 2017b.

MOLINA-CABELLO, M. A., LUQUE-BAENA, R. M., LÓPEZ-RUBIO, E. y THURNHOFER-HEMSI, K. Vehicle type detection by convolutional neural networks. En *International Work-Conference on the Interplay Between Natural and Artificial Computation*, páginas 268–278. Springer, 2017c.

MOSCHOU, V., VERVERIDIS, D. y KOTROPOULOS, C. Assessment of self-organizing map variants for clustering with application to redistribution

of emotional speech patterns. *Neurocomputing*, vol. 71(1-3), páginas 147 – 156, 2007.

Mwebaze, E., Schneider, P., Schleif, F. M., Aduwo, J. R., Quinn, J. A., Haase, S., Villmann, T. y Biehl, M. Divergence-based classification in learning vector quantization. *Neurocomputing*, vol. 74(9), páginas 1429–1435, 2011. ISSN 0925-2312.

Naghiyev, E., Gillott, M. y Wilson, R. Three unobtrusive domestic occupancy measurement technologies under qualitative review. *Energy and Buildings*, vol. 69, páginas 507 – 514, 2014.

Ngocho, B. M. y Mwangi, E. Single image super resolution with guided back-projection and log sharpening. En *Electrotechnical Conference (MELECON), 2016 18th Mediterranean*, páginas 1–6. IEEE, 2016.

Nogueira, P. A. y Teófilo, L. F. A multi-layered segmentation method for nucleus detection in highly clustered microscopy imaging: a practical application and validation using human u2os cytoplasm–nucleus translocation images. *Artificial Intelligence Review*, vol. 42(3), páginas 331–346, 2014.

Oja, M., K., S. y Kohonen, T. Bibliography of self-organizing map (som) papers: 1998-2001 addendum. *Neural Computing Surveys*, vol. 3(1), páginas 1–156, 2003.

Oliver, N., Rosario, B. y Pentland, A. A Bayesian Computer Vision System for Modeling Human Interactions. En *Computer Vision Systems. ICVS 1999*, vol. 22, páginas 255–272. Springer, 1999.

Oliver, N., Rosario, B. y Pentland, A. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22(8), páginas 831–843, 2000.

Ortega-Zamorano, F., Jerez, J., Juarez, G., Perez, J. y Franco, L. High precision FPGA implementation of neural network activation functions. En *2014 IEEE Symposium on Intelligent Embedded Systems (IES)*, páginas 55–60. 2014a.

Ortega-Zamorano, F., Jerez, J., Urda Munoz, D., Luque-Baena, R. y Franco, L. Efficient implementation of the backpropagation algorithm in fpgas and microcontrollers. *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP(99), páginas 1–1, 2015.

Ortega-Zamorano, F., Jerez, J. M., Gómez, I. y Franco, L. Layer multiplexing fpga implementation for deep back-propagation learning. *Integrated Computer-Aided Engineering*, vol. 24(2), páginas 171–185, 2017.

ORTEGA-ZAMORANO, F., JEREZ, J. M., SUBIRATS, J. L., MOLINA, I. y FRANCO, L. Smart sensor/actuator node reprogramming in changing environments using a neural network model. *Engineering Applications of Artificial Intelligence*, vol. 30(0), páginas 179 – 188, 2014b.

ORTEGA-ZAMORANO, F., MOLINA-CABELLO, M. A., LÓPEZ-RUBIO, E. y PALOMO, E. J. Smart motion detection sensor based on video processing using self-organizing maps. *Expert Systems with Applications*, vol. 64, páginas 476–489, 2016.

ORTIZ, A., MUNILLA, J., GORRIZ, J. M. y RAMIREZ, J. Ensembles of deep learning architectures for the early diagnosis of the alzheimer's disease. *International journal of neural systems*, vol. 26(07), página 1650025, 2016.

OXER, J. y BLEMINGS, H. *Practical Arduino: Cool Projects for Open Source Hardware*. Apress, Berkely, CA, USA, 2009. ISBN 1430224770, 9781430224778.

PALOMO, E. J. y DOMÍNGUEZ, E. Hierarchical color quantization based on self-organization. *Journal of Mathematical Imaging and Vision*, vol. 49(1), páginas 1–19, 2014.

PALOMO, E. J. y LÓPEZ-RUBIO, E. Learning topologies with the growing neural forest. *International Journal of Neural Systems*, vol. 26(4), página 1650019, 2016.

PALOMO, E. J. y LÓPEZ-RUBIO, E. The growing hierarchical neural gas self-organizing neural network. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28(9), páginas 2000–2009, 2017.

PAPADIMITRIOU, K., DOLLAS, A. y SOTIROPOULOS, S. Low-cost real-time 2-D motion detection based on reconfigurable computing. *IEEE Transactions on Instrumentation and Measurement*, vol. 55(6), páginas 2234–2243, 2006.

PAPAMARKOS, N. Color reduction using local features and a sofm neural network. *Journal of Imaging Systems and Technology*, vol. 10(5), páginas 404–409, 1999.

PARIS, P. C. D., PEDRINO, E. C. y NICOLETTI, M. C. Automatic learning of image filters using cartesian genetic programming. *Integr. Comput.-Aided Eng.*, vol. 22(2), páginas 135–151, 2015. ISSN 1069-2509.

PARK, H., PARK, J., KIM, H., JUN, J., SON, S. H., PARK, T. y KO, J. ReLiSCE: Utilizing resource-limited sensors for office activity context extraction. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45(8), páginas 1151–1164, 2015.

PARKS, D. y FELS, S. Evaluation of background subtraction algorithms with post-processing. páginas 192–199. 2008.

PHILIP, K. P., DOVE, E. L., MCPHERSON, D. D., GOTTEINER, N. L., STANFORD, W. y CHANDRAN, K. B. The fuzzy hough transform-feature extraction in medical images. *IEEE Transactions on Medical Imaging*, vol. 13(2), páginas 235–240, 1994.

PODOLAK, I. y JASTRZEBSKI, S. Density invariant detection of osteoporosis using growing neural gas. *Advances in Intelligent Systems and Computing*, vol. 226, páginas 629–638, 2013.

PULLI, K., BAKSHEEV, A., KORNYAKOV, K. y ERUHIMOV, V. Real-time computer vision with OpenCV. *Communications of the ACM*, vol. 55(6), páginas 61–69, 2012.

QUINLAN, J. Induction of decision trees. *Machine Learning*, vol. 1(1), páginas 81–106, 1986.

RAD, R. y JAMZAD, M. Real time classification and tracking of multiple vehicles in highways. *Pattern Recognition Letters*, vol. 26(10), páginas 1597–1607, 2005.

RAFIEI, M. H. y ADELI, H. A novel machine learning model for estimation of sale prices of real estate units. *Journal of Construction Engineering and Management*, vol. 142(2), página 04015066, 2015.

RAFIEI, M. H. y ADELI, H. A novel unsupervised deep learning model for global and local health condition assessment of structures. *Engineering Structures*, vol. 156, páginas 598–607, 2018.

RAFIEI, M. H., KHUSHEFATI, W. H., DEMIRBOGA, R. y ADELI, H. Supervised deep restricted boltzmann machine for estimation of concrete. *ACI Materials Journal*, vol. 114(2), 2017.

RÄTY, T. D. Survey on contemporary remote surveillance systems for public safety. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40(5), páginas 493–515, 2010. ISSN 1094-6977.

RAUBER, A., MERKL, D. y DITTENBACH, M. The growing hierarchical self-organizing map: Exploratory analysis of high-dimensional data. *IEEE Transactions on Neural Networks*, vol. 13(6), páginas 1331–1341, 2002.

REDDY, K. K. y SHAH, M. Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, vol. 24(5), páginas 971–981, 2013. ISSN 09328092.

REID, D. An algorithm for tracking multiple targets. *IEEE transactions on Automatic Control*, vol. 24(6), páginas 843–854, 1979.

REN, J., CHEN, Y., XIN, L., SHI, J., LI, B. y LIU, Y. Detecting and positioning of traffic incidents via video-based analysis of traffic states in a road segment. *IET Intelligent Transport Systems*, vol. 10(6), páginas 428–437, 2016.

REN, S., HE, K., GIRSHICK, R. y SUN, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39(6), páginas 1137–1149, 2017.

RIDDER, C., MUNKELT, O. y KIRCHNER, H. Adaptive background estimation and foreground detection using kalman-filtering. En *Proc. Int. Conf. Recent Advances in Mechatronics*, páginas 193–199. 1995.

ROBBINS, H. y MONRO, S. A stochastic approximation method. *The Annals of Mathematical Statistics*, vol. 22(3), páginas 400–407, 1951.

ROSTAMI, S. y NERI, F. Covariance matrix adaptation Pareto archived evolution strategy with hyper volume-sorted adaptive grid algorithm. *Integrated Computer-Aided Engineering*, vol. 23(4), páginas 313–329, 2016.

ROSTAMI, S., NERI, F. y EPITROPAKIS, M. Progressive preference articulation for decision making in multi-objective optimisation problems. *Integrated Computer-Aided Engineering*, vol. 24(4), páginas 315–335, 2017.

RUMELHART, D. E., HINTON, G. E. y WILLIAMS, R. J. Learning Representations by Back-propagating Errors. *Nature*, vol. 323, páginas 533–536, 1986.

SAJID, H., CHEUNG, S.-C. y JACOBS, N. Appearance based background subtraction for ptz cameras. *Signal Processing: Image Communication*, vol. 47, páginas 417–425, 2016.

SAMUEL KASKI, J. y KOHONEN, T. Bibliography of self-organizing map (som) papers: 1981-1997. *Neural Computing Surveys*, vol. 1, páginas 102–350, 1998.

SANCHEZ, A., MORENO, A. B., VELEZ, D. y VÉLEZ, J. F. Analyzing the influence of contrast in large-scale recognition of natural images. *Integrated Computer-Aided Engineering*, vol. 23(3), páginas 221–235, 2016.

SATO, M. y ISHII, S. On-line EM algorithm for the normalized Gaussian network. *Neural Computation*, vol. 12(2), páginas 407–432, 2000.

SCHMITT, J. M., ZHOU, G.-X. y MILLER, J. Measurement of blood hematocrit by dual-wavelength near-ir photoplethysmography. En *Physiological monitoring and early detection diagnostic methods*, vol. 1641, páginas 150–162. International Society for Optics and Photonics, 1992.

SEN-CHING, S. C. y KAMATH, C. Robust techniques for background subtraction in urban traffic video. En *Electronic Imaging 2004*, páginas 881–892. International Society for Optics and Photonics, 2004.

SENGUPTA, S., DAS, S., NASIR, M. y PANIGRAHI, B. K. Multi-objective node deployment in WSNs: In search of an optimal trade-off among coverage, lifetime, energy consumption, and connectivity. *Engineering Applications of Artificial Intelligence*, vol. 26(1), páginas 405–416, 2013.

SIMONYAN, K. y ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *CoRR*, vol. abs/1409.1556, 2014.

SINGH, Y., GUPTA, P. y YADAV, V. S. Implementation of a Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications. *International Journal of Computer Science and Network Security*, vol. 10(3), páginas 136 – 143, 2010.

SOBRAL, A. Bgslibrary: An opencv c++ background subtraction library. En *IX Workshop de Visao Computacional*, vol. 2, página 7. 2013.

SOBRAL, A. y VACAVANT, A. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding*, vol. 122, páginas 4–21, 2014. ISSN 10773142.

ST-CHARLES, P.-L., BILODEAU, G.-A. y BERGEVIN, R. A self-adjusting approach to change detection based on background word consensus. En *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, páginas 990–997. IEEE, 2015a.

ST-CHARLES, P.-L., BILODEAU, G.-A. y BERGEVIN, R. Subsense: A universal change detection method with local adaptive sensitivity. *IEEE Transactions on Image Processing*, vol. 24(1), páginas 359–373, 2015b.

STAUFFER, C. y GRIMSON, W. Adaptive background mixture models for real-time tracking. En *Proc. IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, páginas 246–252. 1999.

SUBUDHI, B. N., GHOSH, S. y GHOSH, A. Application of gibbs–markov random field and hopfield-type neural networks for detecting moving objects from video sequences captured by static camera. *Soft Computing*, vol. 19(10), páginas 2769–2781, 2015.

SZEGEDY, C., LIU, W., JIA, Y., SERMANET, P., REED, S., ANGUELOV, D., ERHAN, D., VANHOUCKE, V., RABINOVICH, A. ET AL. Going deeper with convolutions. Cvpr, 2015.

TANG, Z., LEE, J. H., LOUIE, R. F. y KOST, G. J. Effects of different hematocrit levels on glucose measurements with handheld meters for point-of-care testing. *Archives of pathology & laboratory medicine*, vol. 124(8), páginas 1135–1140, 2000.

THURNHOFER-HEMSI, K., LÓPEZ-RUBIO, E., DOMÍNGUEZ, E., LUQUE-BAENA, R. M. y MOLINA-CABELLO, M. A. Panorama construction for ptz camera surveillance with the neural gas network. *Expert Systems*, 2018.

TIAN, J. y MA, K.-K. A survey on super-resolution imaging. *Signal, Image and Video Processing*, vol. 5(3), páginas 329–342, 2011.

TOYAMA, K., KRUMM, J., BRUMITT, B. y MEYERS, B. Wallflower: Principles and practice of background maintenance. En *IEEE International Conference on Computer Vision, ICCV*, páginas 255–261. Kerkyra, Greece, 1999.

TURAGA, P., CHELLAPPA, R., SUBRAHMANIAN, V. S. y UDREA, O. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18(11), páginas 1473–1488, 2008. ISSN 1051-8215.

UEDA, N. y NAKANO, R. A new competitive learning approach based on an equidistortion principle for designing optimal vector quantizers. *Neural Networks*, vol. 7(8), páginas 1211 – 1227, 1994.

VILLMANN, T. y HAASE, S. Divergence-based vector quantization. *Neural Computation*, vol. 23, páginas 1343–1392, 2011.

VISHNUVARTHANAN, A., RAJASEKARAN, M. P., GOVINDARAJ, V., ZHANG, Y. y THIYAGARAJAN, A. Development of a combinational framework to concurrently perform tissue segmentation and tumor identification in t1-w, t2-w, flair and mpr type magnetic resonance brain images. *Expert Systems with Applications*, vol. 95, páginas 280–311, 2018.

WANG, H., ZHANG, Y., NIE, R., YANG, Y., PENG, B. y LI, T. Bayesian image segmentation fusion. *Knowledge-Based Systems*, vol. 71, páginas 162–168, 2014a.

WANG, J., XU, W. y GONG, Y. Real-time driving danger-level prediction. *Eng. Appl. Artif. Intell.*, vol. 23(8), páginas 1247–1254, 2010.

WANG, K., LIU, Y., GOU, C. y WANG, F.-Y. A multi-view learning approach to foreground detection for traffic surveillance applications. *IEEE Transactions on Vehicular Technology*, vol. 65(6), páginas 4144–4158, 2016.

WANG, X. Deep learning in object recognition, detection, and segmentation. *Foundations and Trends in Signal Processing*, vol. 8(4), páginas 217–382, 2016.

WANG, X.-J. y SHEN, H. Improved growing learning vector quantification for text classification. *Jisuanji Xuebao/Chinese Journal of Computers*, vol. 30(8), páginas 1277–1285, 2007.

WANG, Y., JODOIN, P.-M., PORIKLI, F., KONRAD, J., BENEZETH, Y. y ISHWAR, P. Cdnet 2014: An expanded change detection benchmark dataset. En *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, páginas 393–400. 2014b.

WANG, Z., LU, F., LU, Q.-C., WANG, D., PENG, Z.-R. ET AL. Fine-scale estimation of carbon monoxide and fine particulate matter concentrations in proximity to a road intersection by using wavelet neural network with genetic algorithm. *Atmospheric Environment*, vol. 104, páginas 264–272, 2015.

WENNECKE, G. Hematocrit-a review of different analytical methods. *Radiometer Medical ApS*, 2004.

WREN, C., AZARBAYEJANI, A., DARRELL, T. y PENTL, A. Pfinder: Real-time tracking of the human body. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19(7), páginas 780–785, 1997.

WSHAH, S., XU, B., BULAN, O., KUMAR, J. y PAUL, P. Deep learning architectures for domain adaptation in HOV/HOT lane enforcement. En *IEEE Winter Conference on Applications of Computer Vision (WACV)*, páginas 1–7. 2016.

XIAO, Y., LEUNG, C.-S., LAM, P.-M. y HO, T.-Y. Self-organizing map-based color palette for high-dynamic range texture compression. *Neural Computing and Applications*, vol. 21(4), páginas 639–647, 2012.

XU, Y. y SONG, D. Systems and algorithms for autonomous and scalable crowd surveillance using robotic ptz cameras assisted by a wide-angle camera. *Autonomous Robots*, vol. 29(1), páginas 53–66, 2010.

XUE, H., LIU, Y., CAI, D. y HE, X. Tracking people in rgbd videos using deep learning and motion clues. *Neurocomputing*, vol. 204, páginas 70–76, 2016.

YANG, C.-Y., MA, C. y YANG, M.-H. *Single-Image Super-Resolution: A Benchmark*, páginas 372–386. Springer International Publishing, Cham, 2014a. ISBN 978-3-319-10593-2.

YANG, S., WU, Y.-J. y WOOLSCHLAGER, J. Integrated modeling framework for highway traffic pollution estimation and dispersion. *American Journal of Environmental Sciences*, vol. 12(3), páginas 140–151, 2016.

YANG, Y.-B., LI, Y.-N., GAO, Y., YIN, H. y TANG, Y. Structurally enhanced incremental neural learning for image classification with subgraph extraction. *International Journal of Neural Systems*, vol. 24(7), página 1450024, 2014b.

YANIK, P. M., MANGANELLI, J., MERINO, J., THREATT, A. L., BROOKS, J. O., GREEN, K. E. y WALKER, I. D. A Gesture Learning Interface for Simulated Robot Path Shaping With a Human Teacher. *IEEE Transactions on Human-Machine Systems*, vol. 44(1), páginas 41–54, 2014. ISSN 2168-2291.

YEUM, C. M. y DYKE, S. J. Vision-Based Automated Crack Detection for Bridge Inspection. *Computer-Aided Civil and Infrastructure Engineering*, vol. 30(10), páginas 759–770, 2015.

YICK, J., MUKHERJEE, B. y GHOSAL, D. Wireless sensor network survey. *Computer Networks*, vol. 52(12), páginas 2292–2330, 2008.

YILMAZ, A., JAVED, O. y SHAH, M. Object tracking: A survey. *ACM Computing Surveys*, vol. 38(4), 2006.

YIN, H. Visom - a novel method for multivariate data projection and structure visualization. *IEEE Transactions on Neural Networks*, vol. 13(1), páginas 237–243, 2002.

YIN, H. The self-organizing maps: Background, theories, extensions and applications. *Studies in Computational Intelligence*, vol. 115, páginas 715–762, 2008.

ZHANG, A., WANG, K. C., LI, B., YANG, E., DAI, X., PENG, Y., FEI, Y., LIU, Y., LI, J. Q. y CHEN, C. Automated pixel-level pavement crack detection on 3d asphalt surfaces using a deep-learning network. *Computer-Aided Civil and Infrastructure Engineering*, vol. 32(10), páginas 805–819, 2017.

ZHANG, C., LI, H., WANG, X. y YANG, X. Cross-scene crowd counting via deep convolutional neural networks. En *IEEE Conf. on Computer Vision and Pattern Recognition*, páginas 833–841. Boston, USA, 2015. ISSN 1063-6919.

ZHAO, Z., BOUWMANS, T., ZHANG, X. y FANG, Y. A fuzzy background modeling approach for motion detection in dynamic backgrounds. En *Multimedia and Signal Processing* (editado por F. L. Wang, J. Lei, R. W. H. Lau y J. Zhang), páginas 177–185. 2012. ISBN 978-3-642-35286-7.

ZHAO, Z., ZHANG, X. y FANG, Y. Stacked multilayer self-organizing map for background modeling. *IEEE Transactions on Image Processing*, vol. 24(9), páginas 2841–2850, 2015.

ZHENG, J., WANG, Y., NIHAN, N. y HALLENBECK, M. Extracting roadway background image: Mode-based approach. *Transp. Res. Record: Journal of the Transportation Research Board*, vol. 1944, páginas 82–88, 2006.

ZIVKOVIC, Z. Improved adaptive gaussian mixture model for background subtraction. En *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 2 - Volume 02*, ICPR '04, páginas 28–31. IEEE Computer Society, Washington, DC, USA, 2004. ISBN 0-7695-2128-2.

ZIVKOVIC, Z. y VAN DER HEIJDEN, F. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, vol. 27(7), páginas 773–780, 2006.

Nadie sabe el potencial que encierra este poderoso sistema;
algún día podrá llegar a ejecutar música,
componer sinfonías y complejos diseños gráficos.

Ada Byron