ANÁLISIS DE VIDEO PARA LA SEGMENTACIÓN Y CLASIFICACIÓN DE JUGADORES EN JUEGOS DE FÚTBOL

Ángela María Gómez Correa
Luisa Fernanda Trejos Largo

Proyecto de grado presentado como requisito parcial
para aspirar al título de Ingeniera Electrónica

Director
Ing Germán A. Holguín L, M.Sc, Ph.D(C)

UNIVERSIDAD TECNOLÓGICA DE PEREIRA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
PEREIRA
2014

Nota de Aceptación

_____

_____

_____

_____

_____

_____

_____

Firma del Presidente del jurado

_____

Firma del jurado

_____

Firma del jurado

Pereira, 11 de Junio de 2014

Dedicado a nuestras Familias, por el sacrificio, por la confiaza, apoyo incondicional y paciencia al esperar siempre de nosotras los mejores procesos de formación integral y crecimiento ético - profesional en todas nuestras acciones.

Agradecimientos y reconocimiento a Dios por el dicernimiento. A las personas que creyeron y apoyaron el proceso de estudio y construcción de nuestro proyecto y trabajo de grado: a la U.T.P., al ingeniero Arley Bejarano por acompañarnos al inicio, al ingeniero Andrés Calvo por toda su colaboración, al director y asesor, Ph.D. Germán Holquin por compartir su conocimiento; a todos ellos, mil gracias, por el tiempo y paciencia que nos tuvieron en los momento de estudio e indagación necesaria para llegar al resultado obtenido. Muchas gracias.

CONTENIDO

pág.

Índice de figuras

Índice de tablas

1. INTRODUCCIÓN

El ingenio y la innovación del ser humano no conocen límites. El desarrollo social, científico y tecnológico es hoy una consecuencia de aquella característica humana, esta nos permite evolucionar en diferentes aspectos de la sociedad. Para aprovechar esta condición del hombre, se plantea la ingeniería como una herramienta que aporta a la solución de problemas. En este caso, desde la Ingeniería Electrónica, se quiere desarrollar una metodología que permita el análisis de video para avanzar en la detección del fuera de juego en el fútbol.

Actualmente la tecnología es un pilar fundamental dentro de nuestra sociedad. Dentro de los procesos tecnológicos es objeto de estudio el campo de inteligencia artificial que converge otras ciencias como la lógica, la computación y la filosofía, y que busca el diseño y creación de entidades con capacidad de razonar de manera autónoma. En ella encontramos un área conocida como visión por computador, la cual incorpora métodos para la adquisición, el procesamiento y el análisis de imágenes permitiendo extraer información para su comprensión.

Estas herramientas tecnológicas se han venido incorporando en el deporte, con el fin de mejorar no solo el rendimiento de los atletas, sino también el desempeño de las competencias. En el caso del desempeño del deportista encontramos desarrollos como Wimu-Quiko, sistema desarrollado por la empresa RealTrackSystems, que permite al deportista acceder a información sobre sus parámetros como frecuencia cardiaca, velocidad, distancia recorrida o la posición de su cuerpo, ayudando así a que este pueda mejorar su rendimiento [4]. En el caso de desempeño de las competencias encontramos desarrollos como el Ojo de Halcón, aplicado en el tenis de campo como apoyo para el juez en jugadas dudosas [39], el GoalRef, implementado en el fútbol como un sistema de alarma que notifica al juez el evento de gol [6], y el first down line, implementado en el fútbol americano, que permite al espectador visualizar a través de una pantalla el próximo down. Específicamente para el fútbol, es indispensable implementar desarrollos que brinden un análisis claro y objetivo de situaciones controversiales como el fuera de juego.

Para algunas de las aplicaciones mencionadas se resalta el uso de técnicas basadas en visión por computador, las cuales se implementan dentro de este proyecto, permitiendo como primera fase encontrar metodologías para la segmentación y clasificación de jugadores dentro de videos de fútbol, con fin que en trabajos posteriores se logre un desarrollo para la detección del fuera de juego y este sirva de herramienta de apoyo a los jueces de linea en la determinación de este tipo de jugadas, dado que posibilita la disminución de la subjetividad de los juez en la toma de decisiones.

1.1. DEFINICIÓN DEL PROBLEMA

La norma número 11 del reglamento oficial de fútbol, popularmente conocida como fuera de lugar, ha sido una de las más controversiales. Esta norma, indica cuando un jugador se encuentra en fuera de juego, cuando no y como se sanciona [15].

El principio básico del fuera de juego, define que un jugador está en posición de fuera de juego si se encuentra más cerca de la línea de meta contraria que el balón y el penúltimo adversario [15]. Esta norma fue creada con el fin de beneficiar el juego de ataque y el gol, que son los objetivos finales del Fútbol [57], así como desarrollar el juego colectivo, en vez del oportunismo individual [65], evitando que el atacante saque provecho de su posición.

El fuera de juego suele señalarse en jugadas que a menudo finalizan con el balón dentro de la portería o en posibilidad de anotación, es decir, está directamente relacionado con el marcador del partido [30]. Esta norma es centro de constantes cuestionamientos, ya que en muchos casos se presentan errores de apreciación por parte de los árbitros asistentes que favorecen injustamente a un equipo sobre otro, lo cual llega a ser trascendental y definitivo en campeonatos importantes, y que producen una serie de debates entre los mismos jueces, jugadores, directivos, aficionados, medios de comunicación e incluso la FIFA (organismo rector del fútbol mundial).

Estos errores se deben a diferentes factores, entre los que se destacan la falta de atención y concentración, donde aspectos que se encuentran en el terreno de juego como la novedad, la complejidad, el cambio, la sorpresa, el conflicto y la incertidumbre hacen que estos factores (atención y concentración) se dirijan, sin buscarlo, hacia otros estímulos en vez de aquellos que realmente importan, juzgar el fuera de juego [16].

Otro factor que cabe mencionar es el que tiene que ver con la percepción visual, en el que el árbitro asistente deberá observar objetos diferentes de manera simultánea:

- El atacante que realiza el pase
- El balón
- El atacante que recibe el pase
- El penúltimo adversario (defensa)
- El portero

Algo que no es compatible con la función normal del ojo, puesto que no está en capacidad de percibir la posición relativa de los jugadores y del balón a la vez, esto basado en la fisiología de los movimientos oculares [13].

Lo anteriormente mencionado deja en evidencia la necesidad de desarrollar un sistema que permita la detección del fuera de juego, siendo este un macro proyecto que consta de 5 etapas ilustradas en la figura 1.

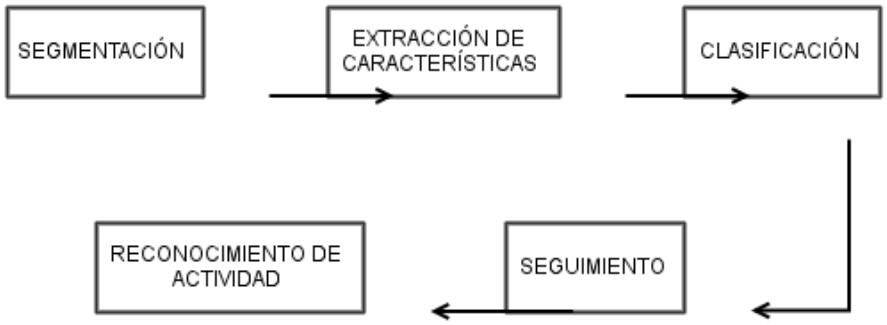


Figura 1: Etapas requeridas en reconocimiento de actividad

1. Segmentación: Resaltar en las imágenes las regiones o los objetos que son más importantes, es decir, separar o aislar del fondo aquellos cuerpos que realmente interesan.
2. Extracción de Características: Tomar y analizar la información más relevantes y significativa de las imágenes segmentadas.
3. Clasificación: Organizar y catalogar en grupos aquellos objetos con características similares.
4. Seguimiento: Estimar una probable siguiente posición de los objetos.
5. Reconocimiento de actividad: Estimar ciertos comportamientos de los objetos.

El objetivo de este proyecto es llevar a cabo el desarrollo de la etapa 3, dando solución a su vez a las etapas 1 y 2, con el fin de concluir si es posible la clasificación de jugadores en la imagen analizada por medio de técnicas de visión por computador.

1.2. JUSTIFICACIÓN

La utilización de herramientas tecnológicas para el control y monitoreo de los diferentes espectáculos deportivos es una tendencia que se ha proliferado en los últimos años. Muchos

deportes de aceptación mundial hoy en día cuentan con tecnologías de ayuda que permiten a los jueces tomar decisiones en tiempo real. Es el caso del tenis de campo con el ojo de halcón, figura 2, el cual es un sistema de red de cámaras que se utilizan para seguir los movimiento de la bola, entregando datos de posición y velocidad de esta; y el fútbol americano con el first down line, figura 3, que consiste en superponer una línea en el campo de juego por medio de un computador que procesa en tiempo real, permitiendo al espectador visualizar el próximo down. Estas tecnologías ayudan a tener claridad en jugadas polémicas.

El fútbol, (football soccer, como es conocido mundialmente) no es la excepción, aunque la penetración de dichas tecnologías en este deporte en particular apenas está en sus primeras fases.

Según el informe de Finanzas de la FIFA, presentado en la isla de Mauricio en Mayo de 2013, el ingreso total, derivado de competiciones, eventos, derechos deportivos, entre otros, asciende a 1166 millones de dólares [28].

Dado que el fútbol es un deporte tan popular y que tiene la capacidad de mover esta cantidad de dinero, tiene sentido trabajar en tecnologías que lo hagan aún mejor. Es por este motivo que vemos importante llevar a cabo la realización de este proyecto, y así cumplir con la finalidad del objetivo del macroproyecto el cual es ayudar al árbitro asistente en la detección del fuera de juego en partidos de fútbol.

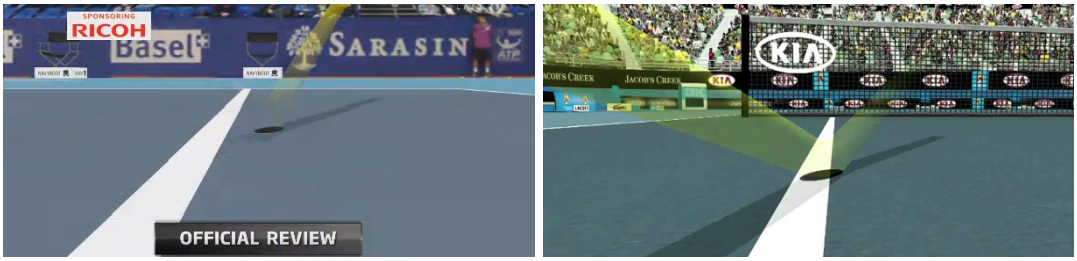


Figura 2: Ojo de Halcón [39]

Figura 3: First down line [36][50]

1.3. OBJETIVOS

1.3.1. Objetivo General

Desarrollar una metodología para la segmentación y clasificación de los jugadores presentes en secuencias de vídeo de partidos de fútbol, con el objetivo posterior de facilitar la detección automática del fuera de juego.

1.3.2. Objetivos Específicos

- Realizar una revisión bibliográfica sobre las diferentes técnicas modernas utilizadas en visión por computador para la extracción de características, segmentación y clasificación de objetos en movimiento.
- Crear una base de datos con escenas de juegos de fútbol que pueda ser utilizada en el entrenamiento y prueba del sistema.
- Determinar una metodología para segmentar los jugadores en los juegos de la base de datos.
- Determinar una metodología para la clasificación de los jugadores en sus respectivos equipos.
- Verificar estadísticamente el funcionamiento de los métodos desarrollados.

2. ESTADO DEL ARTE

Este capítulo presenta una breve recopilación de los trabajos más relevantes que tratan el tema del análisis del video de juegos de fútbol y que en su mayoría fueron publicados en las series de la IEEE.

En 2005, Xu y Shi de la Universidad Jiaotong de Shangai [66], trabajaron en segmentación de jugadores y discriminación de equipos en vídeos de fútbol. Aquí se presentan métodos para la extracción de características de bajo nivel en vídeos de fútbol. Para la segmentación de jugadores se propone la distribución principal de color característico y para discriminar a que equipo pertenece cada jugador se usa el grado de correlación cruzada cromática eliminando la necesidad de extraer plantillas de jugadores con antelación.

En este mismo año Sato y Aggarwal de la Universidad de Texas [59], publicaron su artículo acerca del seguimiento de jugadores de fútbol en imágenes trasmitidas por televisión. En esta publicación se presenta la transformada de distribución espacio-temporal de velocidad (TVS) como método para el seguimiento de varios jugadores simultáneamente. Además se extiende la transformada TVS para aplicarla, bien sea en escala de grises o en imágenes a color, con el fin de realizar el seguimiento de las texturas de los objetos. Para aumentar el rendimiento computacional de la transformada TVS, se utilizó una ventana alrededor objeto que se está siguiendo. Esta operación mantiene la ventana en el centro del objeto a seguir y proporciona un método alternativo de seguimiento cuando los jugadores quedan ocluidos.

Para el 2008, Mazzeo, Spagnolo, Leo, D'orazio del Consejo Nacional Italiano de Investigación [49] publican el artículo detección visual y seguimiento de jugadores en partidos de fútbol. Se propone en este artículo un algoritmo de seguimiento de personas que es capaz de detectar y seguir jugadores de fútbol en situaciones complejas como, variación de condiciones de iluminación, alta frecuencia de imágenes y procesamiento en tiempo real. La segmentación se lleva a cabo mediante un algoritmo basado en la sustracción del fondo. Los objetos detectados se clasifican de un algoritmo de agrupamiento no supervisado, que permite la solución de los problemas de manchas, división y fusión.

En este mismo año Nuñez, Facon y Britto Jr de la Pontificia Universidad Católica de Paraná en Brasil [53], realizaron su investigación sobre segmentación de vídeos de fútbol: detección de refreí y jugadores. En este artículo se presenta una metodología para la segmentación de jugadores en vídeos de partidos de fútbol. Esta metodología incluye algunos algoritmos de procesamiento de alto y bajo nivel, para la detección del campo se utiliza la detección del color dominante en la región de segmentación, y en la identificación de réferi y jugadores el método consiste en binarizar la componente de matiz desde el intervalo máximo y mínimo, el

píxel matiz que tenga un valor dentro del intervalo se pone negro y el resto blanco. El algoritmo utilizado para la detección de color dominante muestra eficiencia a las variaciones de clima, iluminación, de color y la calidad del vídeo. Los resultados para la segmentación de jugadores y réferi muestran que la extracción de estos solo con el matiz dominante es limitada, y que una metodología basada en agrupamiento no supervisado y algunos supuestos es prometedora para la detección de dos jugadores y el réferi.

En el 2008, Huang, Llach, Zhang de la Corporación de Investigación Thompson de Princeton y State Key Lab of Machine Perception de la Universidad de Peking [38], trabajaron en un método para detección y seguimiento de objetos pequeños basados en filtros de partículas. Aquí se presenta un método eficiente para la localización de objetos que integra localización y seguimiento. El sistema inicializa utilizando un detector fuerte, que se crea a partir del análisis de las formas de las manchas de primer plano y se usa para activar el seguidor de objetos, luego un detector débil se construye con las salidas de la probabilidad de detección de primer plano integrado con la probabilidad de la observación del seguidor, por lo tanto con detector débil y un seguidor temporal se localiza el objeto a través del tiempo. En el seguimiento de objetos basado en filtro de partículas, la estimación de movimiento es integrado para generar una mejor distribución propuesta y una mezcla de modelos diseñada para manejar la ambigüedad de la plantilla de juego debido al fondo desordenado. En este artículo el esquema propuesto es aplicado a la detección y seguimiento del balón en vídeos de fútbol.

En el año 2010, Liu y Jian del departamento de ingeniería, ciencias de la computación, física y matemáticas de la Universidad Oral Roberts de Tulsa, Oklahoma y Garner y Vermette de la facultad de ingeniería de la Republic Polytechnic de Singapore [47], realizan una publicación sobre seguimiento del balón a partir de vídeos de fútbol, donde se propone un sistema capaz de localizar y hacer seguimiento al balón durante un partido. Los algoritmos para este sistema logran distinguir con precisión los objetos fijos y los móviles en una serie de tramas de vídeo, son capaces de detectar el balón, determinar sus coordenadas, estimar donde estará ubicado en las próximas tramas de vídeo. Para el logro de lo anterior se realiza inicialmente un algoritmo para el aprendizaje del fondo, este proceso contiene principalmente un bucle que se ejecutará sobre los primeros cientos de tramas para obtener una base buena; el bucle se inicia mediante la lectura de la trama actual del vídeo, después se toman todos los promedios de las tramas leídas para actualizar el cuadro y construir el fondo. Luego se inicia un proceso de umbralización, desarrollando una función umbral basada en la varianza, por su eficiencia y simplicidad en términos de implementación y adaptación a cambios de iluminación. Para retirar el ruido no deseado se desarrolla una función morfológica con una entrada y una salida. El primer plano de la imagen de entrada es la imagen binaria resultante de la umbralización y la salida es (the resulting de-noised image) una imagen clara con distintas regiones que pueden ser reconocidas. Para la detección del balón se crea una matriz U que se encarga de examinar las regiones de

cada trama; cada tamaño de la región es verificado contra el rango de tamaño típico del balón en esa ubicación de la trama, si se encuentra una región con un tamaño aproximado al del balón en dicha posición esa región será marcada como balón. La función también puede dar como salida la ubicación del centro del balón si es necesario.

Para el 2011, Hossein-Khani, Soltanian-Zadeh, Kameri y Staadt del Departamento de Ingeniería Eléctrica y Computación de la Universidad de Tehran en Irán [37], trabajaron en detección del balón con el objetivo de la detección de eventos de tiros de esquina en vídeos de fútbol. En este artículo se propone un sistema que se centra principalmente en la detección del balón. La trayectoria del balón puede ser obtenida por medio de seguimiento y se utiliza usualmente para la identificación y detección de los eventos más importantes en un partido de fútbol. Para la detección del campo de juego, líneas de campo y del balón se realiza procesamiento de imágenes basado en segmentación. Los métodos de espacio HSV y RGB se utilizan para la extracción del color dominante del campo de juego, luego se remueven los jugadores por medio de operaciones morfológicas. Para la detección de las líneas se utiliza Hough Transform. El método de limpieza morfológica se aplica para la detección del balón, es automático y en tiempo real, lo que produce resultados superiores en comparación con otros métodos comunes como Template Matching y Circular Hough Transform (CHT).

3. MARCO TEÓRICO

Este capítulo presenta un resumen de las metodologías del estado del arte que constituyen el marco teórico sobre el cual esta desarrollado el presente trabajo. Existen tres partes muy importantes en el desarrollo de la aplicación de interés, como son, la segmentación, la extracción de características y la clasificación, que por tanto serán abordadas de forma independiente en este capítulo.

3.1. SEGMENTACIÓN

Este método divide la imagen en sus partes hasta un nivel de subdivisión en el que se aíslan las regiones u objetos de interés. Tiene como objetivo resaltar los objetos o regiones de primer plano cambiando la representación de la imagen por una más significativa, ya que se encarga de localizar los objetos y de encontrar los límites de los mismos dentro de la imagen. Al aplicar segmentación a una imagen lo que se obtiene es un conjunto de segmentos o regiones que representan toda la imagen.

Los algoritmos de segmentación se basan en una de estas dos propiedades básicas de los valores de nivel de gris: discontinuidad o similitud entre los niveles de píxeles vecinos.

3.1.1. Discontinuidad

Se divide la imagen basándose en cambios bruscos de nivel de grises. Para la detección de discontinuidades el métodos más utilizado es la correlación de la imagen con una mascara de 3x3:

$$\begin{matrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{matrix}$$

Se multiplica entonces el valor de los píxeles de la imagen que están cubiertos por la mascara con la componente de la mascara correspondiente a cada píxel[26] . La formula para dicha operación es:

$$R = \sum_{i=1}^{9} (x_i * z_i)$$

donde R es la respuesta de la máscara, z_i es el nivel de gris asociado al píxel de la imagen con coeficiente de la mascara x_i.

Detección de puntos aislados (Filtro Laplaciano):

Para este fin se usa la siguiente máscara:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Se mide entonces la diferencia entre el píxel central y sus vecinos, y se dice que un píxel será un punto aislado siempre y cuando la diferencia con sus vecinos sea suficientemente relevante o significativa [26].

Detección de líneas:

En este caso se usan las siguientes máscaras para cada dirección:

$$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix}_{Horizontal} \quad \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix}_{45^\circ} \quad \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}_{Vertical}$$

$$\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}_{-45^\circ}$$

Las direcciones se pueden establecer al observar que para cada dirección las máscaras presentan valores mayores que para otras posibles direcciones [26].

Detección de bordes:

La frontera entre dos regiones con nivel de gris diferente se define como borde. Los métodos de detección de bordes son los más utilizados para la detección de discontinuidades, y la mayoría de estos están basados en el cálculo del gradiente (primera y segunda derivada), ya que el gradiente de una imagen en un punto indica la variación máxima de la función en este.

Métodos basados en gradiente.

El gradiente de una imagen $I(x, y)$ en la posición (x, y) se define como:

$$\nabla I = \begin{bmatrix} I_x \\ I_y \end{bmatrix} = \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix}$$

La derivada es el vector que apunta en la dirección de la máxima variación $I(x, y)$. Se determina si es borde o no, si el valor de la magnitud del gradiente supera un umbral determinado [23][26]. La magnitud del gradiente se calcula mediante:

$$\|\nabla I\| = \sqrt{I_x^2 + I_y^2}$$

En la figura 4, se observa que para cambio de nivel gris oscuro a claro la primera derivada es positiva, para cambio de nivel de gris claro a oscuro es negativa y en las partes donde el nivel de gris no varia es cero. En la segunda derivada en la parte oscura de cada borde el valor es positivo, en la parte claro negativo y cero en la posición de los bordes y donde el gris no varia.

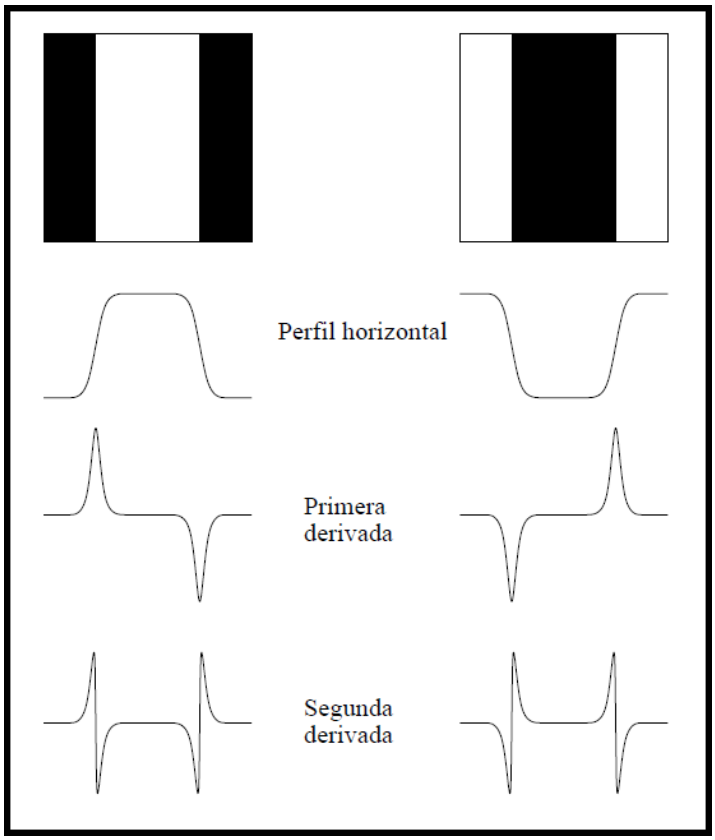


Figura 4: Detección de bordes

Existen varios operadores que hacen parte de los métodos basados en gradiente para la detección de bordes:

- Operador de Roberts:

Utiliza las siguientes máscaras

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Este operador tiene un buen rendimiento con imágenes binarias, pero debido a que tiene en cuenta muy pocos píxeles de entrada para hacer la aproximación es muy sensible al ruido y solo detecta la ubicación de los bordes, más no su orientación.

- Operador Sobel:

Utiliza las siguientes máscaras

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Esta máscara pasa por cada píxel, calculando el valor para cada uno. Teniendo un umbral previamente determinado, se define si es borde o no, tomando el valor calculado del gradiente en función del umbral predefinido. Sobel enfatiza el valor de los píxeles cercanos al centro, al proporcionarles un coeficiente de 2. Además, proporciona una suavizado al efecto de la derivación, lo que se ve reflejado en la disminución del ruido.

- Operador Prewit:

Utiliza las siguientes máscaras

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Se diferencia del operador Sobel en que este no enfatiza en los píxeles cercanos al centro, como se observa en las mascaras, por lo que este operador brinda una mejor detección de los bordes verticales y horizontales. Además para este caso, este operador no solo proporciona la magnitud de los bordes, también entrega la dirección de los mismos.

- Operador Kirsch

Este operador basa su funcionamiento utilizando una máscara simple y rotandola en las 8 direcciones principales de la brújula (Norte, Noroeste, Oeste, Suroeste, Sur, Sureste, Este y Noreste). Las máscaras utilizadas son:

$$k_0 \equiv \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, 0^\circ$$

$$k_1 \equiv \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, 45^\circ$$

$$k_2 \equiv \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, 90°$$

$$k_3 \equiv \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}, 135°$$

Para cada píxel de la imagen se obtienen 4 valores, los cuales son el resultado de la convolución con cada máscara, el valor del modulo del gradiente resulta ser el máximo de esos 4 valores, y la dirección se determina por el ángulo asociado a la máscara que ha generado el valor máximo con el que se definió el gradiente.

- Operador Frei-Chen

Este operador mejora la enfatización que se pretende en Sobel al cambiar los coeficientes de la máscara. Frei-Chen costa de 9 máscaras que se pueden dividir en 3 subespacios: bordes, líneas y el medio. Los dos primeros sirven para la detección de bordes y líneas, y el medio sirve para la detección de regiones de intensidad uniforme.

Las máscaras utilizadas son:

Subespacio de bordes

$$f_1 = \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix}$$

$$f_2 = \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 \\ \sqrt{2} & 0 & \sqrt{-2} \\ 1 & 0 & -1 \end{bmatrix}$$

$$f_3 = \frac{1}{2\sqrt{2}} \begin{bmatrix} 0 & -1 & \sqrt{2} \\ 1 & 0 & -1 \\ -\sqrt{2} & 1 & 0 \end{bmatrix}$$

$$f_4 = \frac{1}{2\sqrt{2}} \begin{bmatrix} \sqrt{2} & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -\sqrt{2} \end{bmatrix}$$

Subespacio de líneas

$$f_5 = \frac{1}{2} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$f_6 = \frac{1}{2} \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}$$

$$f_7 = \frac{1}{6} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

$$f_8 = \frac{1}{6} \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix}$$

Subespacio uniforme

$$f_9 = \frac{1}{6} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Se aplican la máscaras a cada punto de la imagen, se suman sus resultados y a este resultado se le haya la raíz cuadrada y se saca el coseno:

$$Frei - Chen = cos\left(\sqrt{\sum_i} f_i\right)$$

Métodos basados en la segunda derivada.

- Operador Laplaciano

El Laplaciano de una función[23] es:

$$\nabla f = \nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Este operador es capaz de localizar los bordes con bastante exactitud por medio de la determinación del cruce por cero. El Laplaciano vale cero si $f(x, y)$ es constante o cambia de forma lineal. Se dice que hay un cruce por cero si se presenta un cambio de signo en la función resultante y esto hace que se determine la presencia de un borde.

El operador Laplaciano tiene la desventaja de ser bastante sensible al ruido, teniendo en cuenta que esta basado en la segunda derivada, además que se pueden dar casos de detección de bordes dobles o falsos bordes.

Una de las mascara utilizadas para el Laplaciano es [26]:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Las máscaras utilizadas en el Laplaciano son simétricas rotacionalmente, lo que le permite detectar bordes en todas las direcciones del espacio.

Operador Canny.

El operador Canny se basa en los cambios de intensidad para la detección de bordes, los cuales son detectados al observarse un cambio brusco en la primera derivada.

Los criterios básicos utilizados en este método son [23]:

- Criterio de detección. Evitar la detección de bordes inexistentes, a igual que la eliminación o no detección de bordes importantes.

- Criterio de localización. Minimizar la distancia entre la posición real del borde y la posición del borde localizado o detectado.
- Criterio de única respuesta. Entregará un único píxel por cada píxel de borde verdadero, por lo que el detector no deberá encontrar múltiples píxeles de borde donde solo existe uno.

Este método consta de 3 fases que serán descritas a continuación:

1. Obtención de Gradiente:

Inicialmente se debe eliminar el ruido y suavizar la imagen utilizando un filtro gaussiano. Este resultado se obtiene al promediar los valores de intensidad de los píxeles del entorno de vecindad con una máscara como las que se muestran a continuación.

$$\frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}, \frac{1}{115}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

Luego de suavizar la imagen se calcula la magnitud y la orientación del gradiente mediante las formulas:

$$G = \sqrt{G_x^2 + G_y^2}$$

$$\theta = arctan2\left(G_y, G_x\right)$$

donde arctan2, es la función arco tangente con dos argumentos. El ángulo de dirección del borde se aproxima a unos de los siguientes ángulos: 0°, 45°, 90° y 135°.

2. Supresión no máxima al resultado del gradiente:

Esta fase tiene el objetivo del adelgazamiento de los bordes. Tomando la imagen suavizada, para cada píxel de la imagen se encontrará la dirección que más se aproxime a la dirección del ángulo del gradiente hallado. Luego, se verifica si la magnitud del gradiente es menor que al menos uno de sus dos vecinos en la dirección obtenida anteriormente, en caso de que esto se cumpla se asigna el valor de 0 a dicho píxel, en el caso que la magnitud sea mayor que sus vecinos se asigna el valor de la magnitud.

3. Histeresis de umbral:

Para esta fase se deben aplicar dos umbrales, uno alto y uno bajo. Se utiliza el umbral alto en primera instancia, el cual marca los bordes que se puede estar seguro son reales. A partir de esto, se siguen las cadenas de máximos locales para cada punto de la imagen en ambas direcciones perpendiculares a la normal del borde, que sean mayores que el valor del umbral bajo. Con esto se guarda la lista de píxeles que aparecen como componentes conectadas. En este proceso se eliminan píxeles ruidosos que no constituyen una línea pero que se habían tomado en cuenta como tal.

3.1.2. Similitud

Se divide la imagen basándose en la búsqueda de zonas que tengan valores similares conforme a unos criterios prefijados:

Crecimiento de regiones:

Este método consiste en agrupar píxeles en regiones en regiones mayores. Para este procedimiento se definen un conjunto de semillas que marcan un píxel de cada uno de los objetos de interés. A partir de la semilla, se realiza la comparación de los píxeles vecinos que no están marcados como semilla, aquellos píxeles con propiedades similares se añaden a la semilla formando una región. La similitud entre la semilla o región y el píxel vecino se definirá mediante la diferencia entre el valor de intensidad del píxel y la media de la región. Se formarán entonces tantas regiones como semillas se hayan definido y el proceso se detendrá hasta que cada píxel de la imagen sea asignado a una región. El buen resultado de la segmentación depende de la correcta selección de las semillas, ya que estos deben representar adecuadamente las regiones de interés, y la definición de los criterios de similitud que permitirán que se añadan píxeles a la región [26].

Umbralización:

Es una técnica muy implementada en imágenes cuando las diferencias entre el objeto de interés y el fondo son bastante claras. Es método lo que hace básicamente es binarizar la imagen, quedando separado el fondo de las regiones de interés. Este resultado se logra, primero tomando la imagen en escala de grises, luego se define un umbral U y se compara el nivel de gris de cada píxel de la imagen con el umbral establecido, de manera que, si el nivel del píxel es mayor que el umbral se le asigna un valor de 1 y de lo contrario (si es menor) se le asigna un valor de 0, obteniendo así las dos secciones (fondo y objeto de interés).

3.2. SUSTRACCIÓN DE FONDO EN VIDEO

La sustracción de fondo también se conoce como detección de primer plano [56], es una técnica en el campo de procesamiento de imágenes y visión de computador donde las imágenes de primer plano son extraídas luego de haber sido procesado. Generalmente las regiones de interés de una imagen son objetos (humanos, carros, etc) en primer plano. Después de la etapa de pre-procesamiento de la imagen (que puede incluir la eliminación de ruido, etc) se requiere la localización de objetos, para lo que se puede hacer uso de esta técnica. La substracción de fondo es un método ampliamente utilizado para la detección de objetos en movimiento en vídeos de cámara estáticas. Este método se justifica mediante la detección de objetos a partir de la diferencia entre el cuadro actual y un cuadro de referencia, a menudo llamado "imagen de fondo" o "modelo de fondo" como se muestra en la Figura 5.

Un algoritmo robusto de sustracción de fondo debe ser capaz de manejar los cambios de iluminación, movimientos repetitivos de desorden y cambios de escena a largo plazo [64].

Los siguientes análisis hacen uso de la función $V(x, y, t)$ como una secuencia de vídeo donde t es la dimensión de tiempo, x y y son las variables de localización de píxel. Ejemplo $V(1, 2, 3)$ la intensidad del píxel esta en (1,2) y es la localización del píxel en la imagen en t=3 en la secuencia del vídeo.

3.2.1. Diferencia de fotogramas

Diferencia de fotogramas (absoluta) en el tiempo t+1 es:

$$D(t+1) = |V(x,\, y,\, t+1) - V(x,\, y,\, t)|$$

El fondo se asume como el frame en el tiempo t. Esta diferencia de imagen solo mostrará alguna intensidad de los píxeles que cambiaran de ubicación en los dos frames.

Aunque aparentemente se ha removido el fondo, este método solo funcionará para los casos en que los píxeles de primer plano están en movimiento y los píxeles de fondo permanecen estáticos.

Un umbral (threshold) Th se pone en esta diferencia de imágenes para mejorar la resta.

$$|V\,(x,\,y,\,t+1) - V\,(x,\,y,\,t)| > Th$$

Es una de las técnicas más comunes y utilizadas en aplicaciones relacionadas con procesamiento de vídeo e imágenes. Su objetivo es la detección de objetos en movimiento por medio de una mascara de primer plano. Para la obtención de la mascara de primer plano se debe determinar el modelo de fondo, es decir la parte estática de la escena, el cual será el objeto sustractor de fondo. El modelado de fondo se realiza mediante el calculo de un modelo inicial de fondo, el cual se debe estar actualizando constantemente con el fin de adaptarse a los posibles cambios de la escena. La mascara de primer plano se obtiene entonces mediante la resta entre el frame actual y el modelo de fondo como se muestra en la Figura 5.

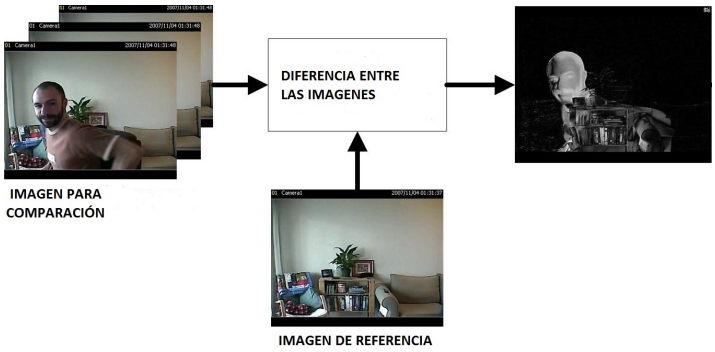


Figura 5: Sustracción de fondo

3.2.2. Mezcla de Gausianas MOG

Es un algoritmo de segmentación de fondo basado en una mezcla gaussiana de Fondo/Primer plano. Este algoritmo utiliza un método de k distribuciones gaussianas (k= 3 a 5) para modelar cada uno de los píxeles que pertenecen al fondo. Los pesos de la mezcla me indica el tiempo

que los colores permanecen en la escena, definiendo entonces que los colores de fondo probables son los que se mantienen más tiempo y más estáticos en la escena [1], ver Figura 6.

Cada píxel en la escena se modela mediante una mezcla de K distribuciones gausianas [62]. La probabilidad de que cierto píxel tenga un valor de x_N en un tiempo N se escribe como:

$$p\left(x_N\right) = \sum_{j=1}^{K} w_i \eta\left(x_N; \theta_j\right)$$

donde $w_k$es el peso del parámetro de la componente gaussiana k^{th}. $\eta\left(x; \theta_k\right)$ es la distribución normal de k^{th}, representada por:

$$\eta\left(x;\, \theta_k\right) = \eta\left(x;\, \mu_k, \sum_k\right) = \frac{1}{(2\pi)^{\frac{D}{2}} \left|\sum_k\right|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_k)^T \sum_k^{-1}\left(x-\mu_k\right)}$$

donde μ_k es la media y $\sum_k = \sigma_k^2 I$ es la covarianza de la componente k^{th}.

Las distribuciones k son ordenadas basados en el valor fitness $\frac{w_k}{\sigma_k}$ y las primeras distribuciones B son utilizadas como modelo de fondo en la escena donde B es estimado como:

$$B = \underset{b}{argmin}\left(\sum_{j=1}^{b} w_j > T\right)$$

El umbral T es la mínima fracción del modelo de fondo. La sustracción de fondo se realiza marcando como píxel de primer plano cualquier píxel que su desviación estándar es mayor que 2.5 en cualquiera de las distribuciones B. La primer componente Gausiana que coincida con el valor de prueba será actualizada de acuerdo a las siguientes ecuaciones:

$$\hat{w}_k^{N+1} = (1-\alpha)\,\hat{w}_k^N + \alpha\hat{\rho}\left(w_k \left| x_{N+1}\right.\right)$$

$$\hat{\mu}_k^{N+1} = (1-\alpha)\,\hat{\mu}_k^N + \rho x_{N+1}$$

$$\hat{\sum}_k^{N+1} = (1-\alpha)\hat{\sum}_k^{N} + \rho\left(x_{N+1} - \hat{\mu}_k^{N+1}\right)\left(x_{N+1} - \hat{\mu}_k^{N+1}\right)^T$$

$$\rho = \alpha\eta\left(x_{N+1}; \hat{\mu}_k^N, \hat{\sum}_k^N\right)$$

$\hat{\rho}\left(\omega_k \left| x_{N+1} \right.\right) = 1$; si $\omega_k$es el primer componente Gausiano que coincide, ó 0 en el caso contrario.

donde ω_k es la componente gaussiana k^{th} y $\frac{1}{\alpha}$ define el tiempo constante que determina el cambio. Si ninguna de las K distribuciones coincide con el píxel de prueba, la componente menos probable se sustituye por una distribución con un valor actual como su media, una varianza inicialmente alta y un parámetro de peso bajo [42].

1. Algoritmo de Maximización de la Esperanza.

La estimación inicial mejora la precisión de la estimación y también el rendimiento del seguidor permitiendo la convergencia rápida de un modelo de fondo estable. Las ecuaciones de actualización de la ventana L da prioridad sobre los datos recientes, por tanto, el seguidor puede adaptarse a los cambios en el medio ambiente [42].

Los algoritmos de EM por estadística de esperanza suficiente en línea son:

$$\hat{w}_k^{N+1} = \hat{w}_k^N + \frac{1}{N+1}\left(\hat{\rho}\left(\omega_k \left| x_{N+1} \right.\right) - \hat{w}_k^N\right)$$

$$\hat{\mu}_k^{N+1} = \hat{\mu}_k^N + \frac{\hat{\rho}\left(\omega_k \left| x_{N+1} \right.\right)}{\sum_{i=1}^{N+1}\hat{\rho}\left(\omega_k \left| x_i \right.\right)}\left(x_{N+1} - \hat{\mu}_k^N\right)$$

$$\hat{\sum}_k^{N+1} = \hat{\sum}_k^{N} + \frac{\hat{\rho}((\omega|x_{N+1})}{\sum_{i=1}^{N+1}\hat{\rho}\left(\omega_k \left| x_i \right.\right)}\left(\left(x_{N+1} - \hat{\mu}_k^N\right)\left(x_{N+1} - \hat{\mu}_k^N\right)^T - \hat{\sum}_k^N\right)$$

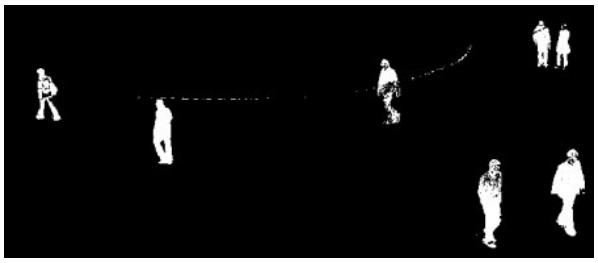


Figura 6: BackgroundSubtractorMOG

3.2.3. Mezcla de Gausianas adaptativa MOG2

Este algoritmo, expuesto por Zivkovic [69] en agosto de 2004 en la conferencia internacional de reconocimiento de patrones en Reino Unido, es una mejora adaptable del modelo de mezcla gaussiana propuesto por Stauffer y Grimso para la sustracción de fondo. La diferencia radica en que el algoritmo de Zivkovic selecciona de manera apropiada el número de distribución gaussina para el modelamiento de cada píxel de fondo, lo que permite adaptación ante las variaciones por cambios de iluminación en la escena. Adicionalmente, este algoritmo realiza también la detección de sombras [1], ver Figura 7.

Figura 7: BackgroundSubtractorMOG2

Zivkovic parte de la premisa de que no se sabe nada acerca de los objetos en primer plano que se pueden ver ni cuándo y con qué frecuencia van a estar presentes, por lo tanto crea lo siguiente:

$$p\left(FG\right) = p\left(BG\right)$$

Asumiendo una distribución uniforme de la apariencia del objeto en primer plano:

$$p\left(\overrightarrow{x}^{(t)}\,|FG\right) = c_{FG}$$

Decide entonces que el píxel pertenece al fondo sí:

$$p\left(\overrightarrow{x}^{(t)}\,|BG\right) > c_{thr}\,(= Rc_{FG})$$

Donde c_{thr} es un valor de umbral y $p\,(\overrightarrow{x}\,|BG)$ es el modelo de fondo.

3.3. EXTRACCIÓN DE CARACTERÍSTICAS

3.3.1. Descriptores

Los descriptores son entidades matemáticas utilizadas para decodificar las características más importantes de un objeto o evento de interés y que permiten asociarlo como miembro de una clase particular de objetos o eventos. En el caso particular de las imágenes, los dos tipos de descriptores más comunes son los basados en intensidad y los morfológicos.

Los píxeles son la forma de codificación digital donde se encuentra el contenido de una imagen, siendo el píxel la unidad mínima de información de una imagen. Se define entonces que los píxeles representan las características propias de una imagen que entendemos como relevantes para estudio. Es por ello que cualquier método de gestión de imágenes basado en su contenido deberá guardar algún tipo de relación o actuar sobre el valor de los mismos[17].

Los descriptores visuales deberán tener las siguientes propiedades preferiblemente:

- Simplicidad: Representación de las características de la imagen de forma clara y sencilla para su fácil interpretación.
- Repetibilidad: El descriptor generado a partir de una imagen debe ser independiente del momento en que se genere.
- Diferenciabilidad: El descriptor debe tener la capacidad de diferenciar una imagen del resto y al mismo tiempo contener la información que le permita establecer una relación entre imagen con características similares.
- Invarianza: Capacidad del descriptor de mantener la relación de las imágenes a pesar de sus deformaciones o transformaciones.
- Eficiencia: Los recursos consumidos para generar el descriptor deben ser aceptables para su correcto funcionamiento en aplicaciones con restricciones críticas de espacio y/o tiempo.

3.3.2. Histogramas de color

Un histograma de color representa la frecuencia de aparición de cada una de las intensidades de color presentes en la imagen, esto se hace por medio de la cuantificación de los píxeles que comparten dichos valores de intensidad de color. Se compone entonces de diferentes rangos o contenedores que representan un valor o conjunto de valores de intensidad de color. Se debe tener en cuenta que la cantidad de intervalos pueden aumentar o disminuir la información representada por el descriptor, cuanto mayor sea el número de intervalos, mayor poder discriminativo tendrá el descriptor, pero aumentará también el costo computacional y presentará más sensibilidad al ruido.

El histograma de color de una imagen esta dado mediante la siguiente formula:

$$h_{R,G,B} = N\ Prob(R = r, G = g, C = c)$$

donde R, G y B representan los tres canales de color y N es el número de píxeles en la imagen.

El espacio de color es un modelo de representación del color con respecto a los valores de intensidad, y puede ser desde una hasta de cuatro dimensiones. El espacio de color RGB se conforma por los colores rojo, verde y azul, y su mezcla arroja como resultado el color deseado. Este espacio utiliza coordenadas cartesianas como se muestra en la Figura 8.

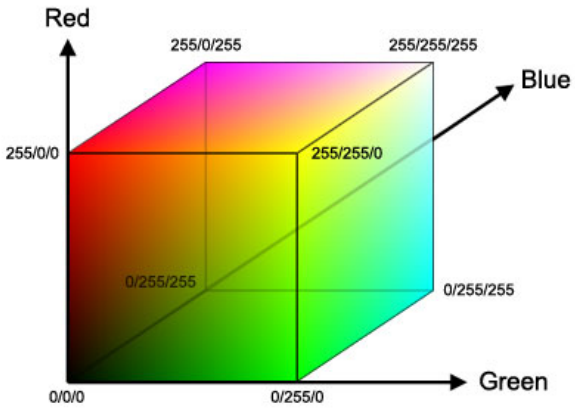


Figura 8: Espacio de Color RGB

Para el espacio de color HSV, la componente Value representa la intensidad de color o brillo, la componente Hue representa la tonalidad y la componente Saturación representa la intensidad dentro del propio color. El espacio de color HSV se representa en la Figura 9

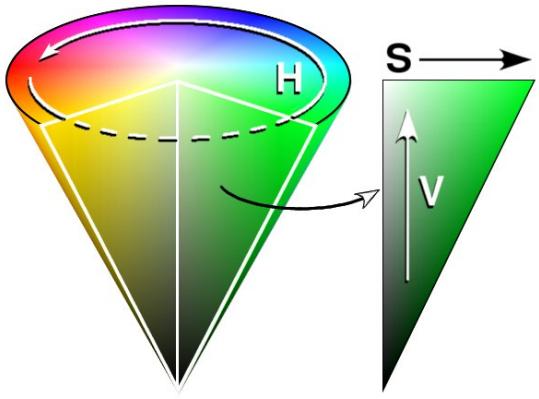


Figura 9: Espacio de color HSV

3.3.3. Descriptores de forma

Una de las principales y más relevantes características de una imagen es la forma teniendo en cuenta que el reconocimiento de objetos para el ser humano se basa en está característica. Esta posee una información semántica muy importante que se extrae únicamente mediante la segmentación, la cual deberá ser muy cercana a la que realiza el sistema visual humano.

La representación de los descriptores de forma se pueden basar en características del contorno y su contenido interno, o únicamente del contorno de acuerdo a la necesidad. Se han definido varias características como, la firma de forma, firma de histograma, invariantes de forma, momentos, curvatura, contexto de forma, características espectrales, entre otros. A continuación se describirán algunos de los descriptores de forma más importantes:

Descriptores de Fourier

Los descriptores de Fourier se han utilizado en muchas aplicaciones de representación de forma con resultados satisfactorios sobre todo para el reconocimiento de caracteres. Para obtener un descriptor de Fourier, se aplica la transformada de Fourier a un vector complejo derivado de las coordenadas de los límites de la forma. El vector complejo $\bar{U}$ esta dado por la diferencia de los límites de los puntos desde el centroide (x_c, y_c) de la forma:

$$x_c = \frac{1}{N}\sum_{n=0}^{N-1} x\left(n\right), \quad y_c = \frac{1}{N}\sum_{n=0}^{N-1} y\left(n\right)$$

$$\bar{U} = \begin{pmatrix} x_0 - x_c & + & i\,(y_0 - y_c) \\ x_1 - x_c & + & i\,(y_1 - y_c) \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ x_n - x_c & + & i\,(y_n - y_c) \end{pmatrix}$$

Es necesario encontrar los coeficientes de Fourier F_k, y al limitar los coeficientes k, se logra la reducción del ruido de alta frecuencia, dejando al mismo tiempo los principales detalles de los patrones. Para obtener los coeficientes de Fourier es necesario aplicar sobre el vector $\bar{U}$ la transformada de Fourier unidimensional:

$$\bar{F}_k = FFT\left[\bar{U}\right]$$

Los descriptores de Fourier adquiridos son traslación, rotación y escala invariante. La aplicación de un número limitado de de descriptores de Fourier tiene un efecto similar al de un filtro pasa-bajo, sin embargo, esta reducción puede producir de información espacial en cuanto a los detalles finos [**?**].

Descriptor de escala de espacio de curvatura (CSS)

El descriptor de escala de espacio de curvatura, toma la forma de la frontera como una señal en una dimensión y la analiza en el espacio de escala. Desde la curvatura se encuentra una medida local que indica que tan rápido un contorno plano esta cambiando, explotando la escala de espacio de curvatura. Por tanto, a través de una suavizado de Gauss, se tiene que los cruces por cero de las curvaturas en diferentes escalas representan las características perspectivas de forma de contorno.

Para la obtención de estos descriptores se debe realizar una normalización a escala un vector complejo derivado de las coordenadas de los límites de la forma así:

$$k\,(t) = \frac{(\dot{x}\,(t)\ \ddot{y}\,(t) - \ddot{x}\,(t)\ \dot{y}\,(t))}{\sqrt{\left(\dot{x}^2\,(t) + \dot{y}\,(t)^2\right)^3}}$$

En los puntos de cruce por cero de la curvatura están ubicados los límites de la forma. Se aplica un suavizado de Gauss aplicando un función de Gauss $g(t, \sigma, c) = e^{\left[-(t-c)^2\right]/2\sigma^2}$ dentro de la ecuación, como se muestra a continuación:

$$x\prime(t) = x(t) * g(t, \sigma, c), \quad y\prime(t) = y(t) * g(t, \sigma, c)$$

con c constante y σ en aumento, la forma en evolución se vuelve más suave. Este proceso se realizar hasta que no hayan más cruces por cero. Los puntos de cruce por cero que se obtienen se representan gráficamente en el plano (t, σ) para crear el mapa de contornos de escala de espacio de curvatura [**?**].

3.4. CLASIFICACIÓN

Es un proceso de inducción del conocimiento y hace parte de las ramas de la inteligencia artificial[8]. Tiene como objetivo el desarrollo y estudio de sistemas que permitan a una máquina aprender a partir de información suministrada [34]. De acuerdo a lo anterior en la Figura 10 se muestra un esquema de aprendizaje de máquina. La base de este aprendizaje, al igual que cualquier otro tipo de aprendizaje es utilizar la evidencia conocida para poder dar una hipótesis y poder dar solución a situaciones no conocidas o nuevas [11].

El aprendizaje de máquinas es un termino que se puede confundir fácilmente con la minería de datos, debido al uso de los mismos métodos y el solapamiento de los mismos. Sin embargo, se aclara que el aprendizaje de maquinas se enfoca en la predicción con base en la información extraída de los datos de entrenamiento, mientras que la minería de datos se centra en el descubrimiento de información desconocida en los datos (antes) por lo que se le llama etapa de análisis de descubrimiento de conocimiento en bases de datos.

Los desarrollos es este campo se enfocan principalmente en dos objetivos: por una parte en la eliminación del conocimiento experto y la intuición que se genera mediante la interacción Hombre-Máquina; otros en cambio tratan de establecer una colaboración entre estos dos elementos. Se tiene en cuenta, sin embargo, que la participación humana y su intuición no puede ser reemplazada en su totalidad por una máquina, ya que es el experto (humano) quien desarrolla y diseña estos sistemas determinando los procesos que debe realizar la máquina [11].

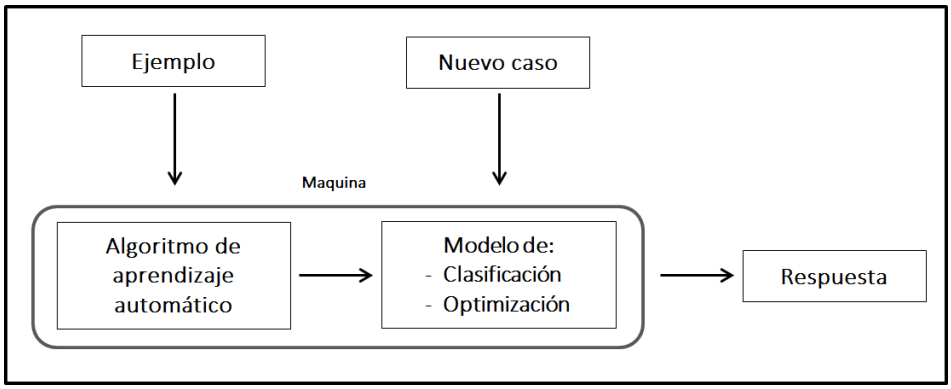


Figura 10: Estructura general de un sistema de aprendizaje

Por medio del aprendizaje automático se generan tres tipos de conocimientos:

- Crecimiento: Se adquiere de lo que nos rodea y guarda información como si dejara huella.
- Reestructuración: Al interpretar los conocimientos se hace un razonamiento y se genera un nuevo conocimiento que se le llama reestructuración.
- Ajuste: Se obtiene al generalizar varios conceptos o generando propios.

En aprendizaje automático, para la generación de conocimiento y mejoramiento en el rendimiento de los sistemas se usan los siguientes algoritmos, que se explicaran en la siguiente sección:

1. Clasificación Supervisada.
2. Clasificación No Supervisada.
3. Aprendizaje por esfuerzo.
4. Transducción.
5. Aprendizaje multi-tarea.

3.4.1. Clasificación Supervisada

Produce una función que establece correspondencia entre las entradas y las salidas deseadas del sistema. Estos algoritmos son entrenados con ejemplos etiquetados, que se toman como entrada, y además se conoce la salida deseada [11] .

Un algoritmo de clasificación supervisada analiza los datos de entrenamiento e infiere una función a partir de dicho análisis que es capaz de predecir el valor correspondiente a cualquier objeto de la entrada.

En general la clasificación supervisada sigue los siguientes pasos para la solución de un problema:

1. Determinar el tipo de ejemplos de entrenamiento: Datos a utilizar para entrenar el modelo.
2. Reunir un conjunto de entrenamiento.
3. Determinar la función de ingreso de la representación de la función aprendido.
4. Determinar la estructura de la función adecuada para resolver y el problema y la técnica de aprendizaje correspondiente.
5. Completar el diseño.

En imágenes, se definen como funciones, técnicas y/o algoritmos que se encargan de asignar una determinada etiqueta a una serie de imágenes entrantes, de acuerdo a ciertos patrones o parámetros que estas presentan, permitiendo ordenarlas por clases o categorías. Estos parámetros pueden ser el tono, la textura, la forma, entre otros [20], [29].

Para este caso, sobre la imagen se delimitan manualmente campos de entrenamiento o áreas piloto que se consideran suficientemente representativas para luego realizar la segmentación automática. En palabras más sencillas, se entrena el algoritmo en el reconocimiento de distintas categorías, a partir de ejemplos de etiquetados anteriores para que este realice luego la clasificación automáticamente [20].

3.4.2. Clasificación No Supervisada

En este tipo de aprendizaje no hay un conocimiento a priori. Los datos de entrada se tratan como un conjunto de variables aleatorias a parir del cual se construye un modelo de densidad para el conjunto de datos.

El aprendizaje no supervisado utiliza técnicas de estimación de densidades estadísticas y métodos de minería de datos que tratan de resumir y explicar las principales características de los datos [41].

Los enfoques del clasificación no supervisada incluyen:

- Agrupamiento [35].
- Modelos de Markov.
- Separación ciega de señales mediante extracción de características para la reducción de dimensionalidad [7].

Para el caso de imágenes, son técnicas de agrupamiento que pretenden segmentar una imagen en clases desconocidas y asignarles una etiqueta, a partir de una búsqueda automática de grupos con valores semejantes que dicha imagen presenta. No requiere de ningún conocimiento previo del área de estudio ni necesita un proceso de muestreo de la imagen, por lo que se hace precisa la intervención humana para la interpretación de resultados. La finalidad de los algoritmos de agrupamiento es ordenar los objetos en conjuntos tales que los que estén en el mismo sean muy semejantes entre sí, pero que cada conjunto sea diferente [20].

3.4.3. Aprendizaje por esfuerzo.

El aprendizaje por esfuerzo es adecuado cuando no se tiene conocimiento a priori del entorno o es demasiado complejo. El aprendizaje por esfuerzo consiste en aprender a decidir, ante una situación determinada, cuales es la acción más adecuada para cumplir un objetivo. Este consta de una componente selectiva, encargada de seleccionar la opción más adecuada entre varias acciones posibles a ejecutar y otra componente asociativa, que se encarga de asociar las alternativas encontradas a situaciones particulares en que se tomaron. Este problema genérico cubre tareas tales como el aprendizaje del control de un robot móvil, aprendizaje de cómo optimizar operaciones en una factoría, y aprendizaje de cómo realizar jugadas en juegos de tableros. Cada vez que un agente realiza una acción en su entorno, un entrenador provee un premio o penalización que indica la bondad del estado resultante. Por ejemplo, cuando se entrena a un agente para jugar un juego, el entrenador debe proveer una recompensa positiva cuando el juego es ganado, negativa si se pierde y cero en los otros estados. La tarea del agente es la de aprender a partir de esta recompensa indirecta y retrasada, a elegir secuencias de acciones que produzcan la mayor acumulación de recompensas. Un ejemplo de estos tipos de algoritmos es el Q-learning, que permite adquirir estrategias de control óptimas a partir de recompensas retrasadas, aun cuando el agente no posee un conocimiento inicial del efecto de las acciones en el entorno. Estos algoritmos están relacionados con la programación dinámica, frecuentemente empleada en la resolución de problemas de optimización [11].

3.4.4. Transducción.

Similar al aprendizaje supervisado, pero no construye de forma explícita una función. Trata de predecir las categorías de los futuros ejemplos basándose en los ejemplos de entrada, sus respectivas categorías y ejemplos nuevos.

3.4.5. Aprendizaje Multitarea.

El Aprendizaje Multitarea (MultiTask Learning - MTL) permite entrenar una tarea junto con un conjunto de tareas relacionadas con ésta, se define una tarea como principal y las demás como sub-tareas. Lo permite que las tareas secundarias ayuden a la principal a mejorar su entrenamiento [18].

3.4.6. Tipos de Clasificadores

Entre los tipos se clasificadores más comunes se encuentran

Clasificador de Bayes.

Hace parte de los métodos de clasificación supervisada, ya que requiere de ejemplos clasificados para su correcto funcionamiento. Este clasificador puede predecir las probabilidades del número de miembros de una clase, y la probabilidad de que una muestra dada pertenezca a una clase en particular. Esta basada en el teorema de probabilidad condicionada, más conocido como el teorema de Bayes.

Sean A y B dos sucesos aleatorios cuyas probabilidades se escriben como $P(A)$ y $P(B)$ respectivamente, verificándose que $P(B) > 0$. Supongamos que se conocen las probabilidades a priori de los sucesos A y B, es decir, $P(A)$ y $P(B)$, así como la probabilidad condicionada del suceso B dado el suceso A, es decir $P(B|A)$. La probabilidad a posteriori del suceso A conocido que se verifica el suceso B, es decir $P(A|B)$, puede calcularse a partir de:

$$P(A|B) = \frac{P(A|B) * P(A)}{P(B)}$$

La formula anterior define la probabilidad de que se de un suceso habiendo sucedido otro que influye en el anterior.

Vecino más Cercano.

Nearest Neighbor (NN). Es un método simple de agrupamiento jerárquico con propiedades estadísticas, que consiste en clasificar cada píxel o patrón en la clase de su vecino más próximo según una medida de distancia (figura 11). Este método se puede extender y generalizar usando los k vecinos más cercanos kNN, lo que permite mejorar la tasa de acierto en la clasificación.

El kNN es un método no paramétrico el cual consiste en tomar k ejemplos de entrenamiento cercanos en el espacio de características [10]. El método sirve para estimar la función de densidad de probabilidad $F(x|C_j)$ a partir de la información entregada por el conjunto de ejemplos luego de saber si un elemento x pertenece a la clase C_j.

Los ejemplos de entrenamiento son vectores en un espacio de características multidimensionales, cada uno con una etiqueta de clase. Para el entrenamiento se almacenan los vectores de características y las etiquetas de la clase de los ejemplos de entrenamiento.

Para la clasificación, el usuario define una constante k. Se tiene un nuevo vector del que no se conoce su clase. Se calcula la distancia entre los vectores almacenados (característicos de las clases) y el nuevo, seleccionando los k ejemplos más cercanos. El nuevo vector es clasificado de acuerdo a la frecuencia con que se repita en los vectores característicos.

Por lo general una métrica de distancia utilizada es la distancia euclidiana:

$$d\left(x_i, x_j\right) = \sqrt{\sum_{r=1}^{p} \left(x_{ir} - x_{jr}\right)^2}$$

En resumen, el método de knn lleva dos procesos:

- El entrenamiento: Para cada ejemplo $< x,\ f\left(x\right) >$, donde $x \epsilon X$, se agrega el ejemplo a una estructura que representa los ejemplos de aprendizaje o entrenamiento.
- Clasificación: Dado un ejemplar x_q que debe ser clasificado, y sean x_1, ..., x_k los k vecinos mas cercanos a x_q en los ejemplos de aprendizaje, entonces se tiene que:

$$\hat{f}\left(x\right) \leftarrow argmax_{v \epsilon V} \sum_{i=1}^{k} \delta\left(v, f\left(x_i\right)\right)$$

donde $\delta\left(a, b\right) = 1$ si $a = b$, y 0 si en cualquier otro caso.

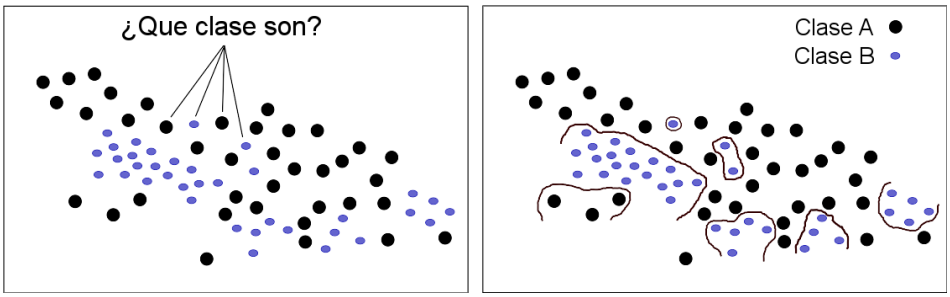


Figura 11: Clasificación por el método del vecino mas cercano

La clasificación entrega el valor de $\hat{f}(x_q)$ que es un estimador de $f(x)$. Este solo es el valor mas común de entre los k vecinos mas cercanos a x_q.

El kNN parte del supuesto de que los vecinos más cercanos dan la mejor clasificación utilizando para ello todos los atributos. El problema de esta suposición esta en que es posible que se tengan muchos atributos irrelevantes que dominen sobre la clasificación: dos atributos relevantes perderían peso entre otros veinte irrelevantes, por ejemplo [22].

Para elegir el valor de k se debe tener en cuenta lo siguiente:

- Si k es pequeño el modelo será muy sensitivo a datos que son atípicos (ruido).
- Si k es muy grande reduce el efecto del ruido pero tienen a asignar a la clase mas grande y crean limites entre clases parecidas

La mejor elección de k dependerá fundamentalmente de los datos. Un buen k puede ser seleccionado heurísticamente [24].

Redes Neuronales. Con el fin de imitar el funcionamiento del cerebro humano, este sistema se basa en la experiencia como principio de aprendizaje.Este sistema es capaz de realizar la clasificación de objetos a partir de la construcción de una red neuronal. Su estructura está compuesta por nodos (neuronas), cada una con una cantidad de memoria local, y conectadas entre ellas por canales de comunicación. Tienen en común algún tipo de regla de aprendizaje (entrenamiento), están en la capacidad de reconocer patrones incompletos, distorsionados o con ruido, y por su implementación paralela, se adaptan a las operaciones en tiempo real. La dificultad de este método de clasificación reside en la obtención de la red neuronal, pues una vez conseguida, el procesamiento es rápido [46], [60].

Máquinas de Soporte Vectorial. Support Vector Machines (SVM). Sistema de clasificación que permite separar dos conjuntos de datos que pertenecen a dos clases distintas. Se compone por una serie de algoritmos, los cuales son los encargados de encontrar la separación entre todo el conjunto de datos. En este sistema se tendrá un conjunto de aprendizaje, que son datos de entrenamiento que se dividen en distintos tipos o clases, y el otro conjunto será el encargado de corroborar el correcto funcionamiento del clasificador [54].

3.4.7. Distancias Estadísticas

El concepto de distancia entre objetos o individuos permite interpretar geométricamente muchas técnicas clásicas del análisis multivariante, equivalentes a representar estos objetos como puntos de un espacio métrico adecuado. Las distancias estadísticas permiten la comparación de distribuciones de frecuencia [2].

Euclidiana La distancia Euclidiana se define como la distancia entre dos puntos medidos de manera ordinaria (metro, regla, etc): Esta viene dada mediante la formula de Pitágoras. Con el uso de esta formula como distancia, el espacio Euclidiano se convierte en un espacio métrico.

La distancia Euclidiana entre los puntos a y b es la longitud del segmento de linea que los conecta ($\bar{PQ}$).

En coordenadas cartesianas, si $a = (a_1\,,\, a_2\,,\, ...\,,\, a_n)$ y $b = (b_1\,,\, b_2\,,\, ...\,,\, b_n)$ son dos puntos en el espacio euclidiano n, entonces la distancia desde a hasta b, o desde b hasta a esta dada por:

$$d_{(a,b)} = d_{(b,a)} = \sqrt{(b_1 - a_1)^2 + (b_2 - a_2)^2 + ... + (b_n - a_n)^2} = \sqrt{\sum_{i=1}^{n} (b_i - a_i)}$$

La posición de un punto en un espacio euclidiano n es un vector euclidiano. Por lo tanto a y b son vectores euclidianos, comenzando desde el origen del espacio, y sus puntas indican dos puntos. La norma euclidiana, o la magnitud del vector mide la longitud de este:

$$\|a\| = \sqrt{a_1^2 + a_2^2 + ... + a_n^2} = \sqrt{a.a}$$

La norma euclidiana es un caso particular de la distancia euclidiana, ya que seria la distancia entre la cola y la punta.

Mahalanobis La distancia de Mahalanobis proporciona una medida relativa de la distancia entre un punto de datos a un punto común. Es una medida sin unidades introducida por P.C Mahalanobis en 1936.

Se utiliza para medir o identificar la similitud entre un conjunto muestra desconocido y uno conocido. Se diferencia de la distancia euclidiana, ya que tiene en cuenta las correlaciones del conjunto de datos y tiene escala invariante.

La distancia de Mahalanobis de una observación $x = (x_1, x_2, x_3, ..., x_n)^T$ a partir de un grupo de observaciones con media $\mu = (\mu_1, \mu_2, \mu_3, ..., \mu_N)^T$ y una matriz e covarianza S se define así:

$$D_M(x) = \sqrt{(x-\mu)^T S^{-1}(x-\mu)}$$

La distancia de Mahalanobis de una observación $x = (x_1, x_2, x_3, ..., x_n)^T$ a partir de un grupo de observaciones con media $\mu = (\mu_1, \mu_2, \mu_3, ..., \mu_N)^T$ y una matriz e covarianza S se define así:

$$D_M\left(x\right) = \sqrt{\left(x-\mu\right)^T S^{-1}\left(x-\mu\right)}$$

La distancia de Mahalanobis también puede ser definida como una medida de disimilitud entre dos vectores aleatorios $\vec{x}$ y $\vec{y}$ en la misma distribución con la matriz de covarianza S:

$$d\left(\vec{x},\vec{y}\right) = \sqrt{\left(\vec{x-y}\right)^T S^{-1}\left(\vec{x-y}\right)}$$

Si la matriz de covarianza es la matriz identidad, la distancia de Mahalanobis se reduce a la distancia Euclidiana. Si la matriz de covarianza es una diagonal, entonces la medida de la distancia resultante se llama distancia euclidiana normalizada:

$d\left(\vec{x}, \vec{y}\right) = \sqrt{\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{S_i^2}}$, donde $S_i$es la desviación estándar de x_i y y_i sobre el conjunto muestra.

Bhattacharyya La distancia de Bhattacharyya mide la similitud de dos distribuciones de probabilidad discretas o continuas. Esta estrechamente relacionado con el coeficiente de Bhattacharyya que es una medida de la cantidad de superposición entre dos muestras estadísticas o poblaciones. El coeficiente se puede utilizar para determinar la cercanía relativa de las dos muestras que se están considerando. Se utiliza para medir la capacidad de separación de clases

en la clasificación, se considera que es mas fiable que la distancia de Mahalanobis, ya que la distancia de Mahalanobis es un caso particular de la distancia de Bhattacharyya cuando las desviaciones estándar de las dos clases son la misma. Por lo tanto, cuando dos clases tienen medias similares pero diferentes desviaciones estándar, la distancia de Mahalanobis tendería a cero, mientras que la distancia de Bhattacharyya crecería en función de la diferencia entre las desviaciones estándar [2].

Para las distribuciones de probabilidad discretas p y q en el mismo dominio de X, se define como:

$$D_B(p,q) = -ln(BC(p,q))$$

donde $BC\left(p,q\right) = \sum_{x \epsilon X} \sqrt{p\left(x\right) q\left(x\right)}$ es el coeficiente de Bhattacharyya.

Para distribuciones de probabilidad continua, el coeficiente de Bhattacharyya se define como:

$$BC\left(p,q\right) = \int \sqrt{p\left(x\right) q\left(x\right)} dx$$

en cualquiera de los casos, $0 \leq BC \leq 1$ y $0 \leq D_B \leq \infty$. D_B no obedece a la desigualdad de triangulo, pero la distancia Hellinger $\sqrt{1 - BC}$ hace obedecer a la desigualdad triangular.

En su formulación más simple, la distancia de Bhattacharyya entre dos clases bajo la distribución normal se puede calcular por la extracción de la media y la varianza de dos distribuciones o clases separadas:

$$D_B\left(p,q\right) = \frac{1}{4} ln\left(\frac{1}{4}\left(\frac{\sigma_p^2}{\sigma_q^2} + \frac{\sigma_q^2}{\sigma_p^2} + 2\right)\right) + \frac{1}{4}\left(\frac{\left(\mu_p - \mu_q\right)^2}{\sigma_p^2 + \sigma_q^2}\right)$$

donde $D_B(p,q)$ es la distancia de Bhattacharyya entre p y q distribuciones o clases,

σ_p es la varianza de distribución p-th

μ_p es la media de la distribución p-th, y

p, q son dos distribuciones diferentes.

El coeficiente de Bhattacharyya es una medida aproximada de la cantidad de superposición entre dos muestras estadísticas. El coeficiente puede ser utilizado para determinar la proximidad relativa de las dos muestras que se están considerando.

El cálculo del coeficiente de Bhattacharyya es una formula rudimentaria de integración de la superposición de las dos muestras. El intervalo de las dos muestras se divide en un número elegido de particiones, y el número de miembros de cada muestra en cada partición se utiliza en la siguiente formula [21],

$$Bhattacharyya = \sum_{i=1}^{n} \sqrt{\left(\sum a_i \sum b_i\right)}$$

Considerando las muestras a y b, n es el número de particiones, y $\sum_{a_i}$, $\sum_{b_i}$son el número de miembros de las muestras a y b en la i-ésima partición. Esta formula, por tanto, es más grande con cada partición que tiene los miembros de cada muestra, y más grande con cada partición que tiene una gran superposición de los miembros de las dos muestras. La elección del numero de particiones depende del número de miembros en cada muestra; pocas particiones perderán la precisión al sobre estimar la región de superposición, y demasiadas particiones perderá precisión mediante la creación de particiones individuales con ningún miembro, a pesar de estar en un espacio muestral envolmentemente poblado.

El coeficiente de Bhattacharyya será 0 si no hay solapamiento en absoluto debido a la multiplicación por cero en cada partición. Esto significa que la distancia entre las muestras completamente separadas no estará expuesta solo por este coeficiente.

Movimiento de Tierra (EMD) Este método evalúa la diferencia entre dos distribuciones de probabilidad multidimensionales, hallando la distancia entre ellas sobre una región determinada.

Dadas dos distribuciones, una puede ser vista como una masa de tierra repartida adecuadamente en un espacio, y el otro como una colección de agujeros en el mismo espacio. La menor cantidad de trabajo necesario para llenar los agujeros de tierra es el EMD[67].

El EMD puede ser aplicado en histogramas, teniendo en cuenta que los histogramas pueden ser vistos como un tipo especial de firma, donde cada bin del histograma corresponde a un elemento en la firma. El EMD en histogramas de acuerdo a Rubner, se define como el mínimo costo que se debe pagar para transformar un histograma P en otro Q [55][67].

$$EMD^{D}(P,Q) = \left(\underset{\{F_{ij}\}}{min}\sum_{i,j} F_{ij}D_{ij}\right) / \left(\sum_{i,j} F_{ij}\right) \text{ para } F_{ij} \geq 0$$

$$\sum_i F_{ij} \leq P_i \sum_i F_{ij} \leq Q_j \sum_{i,j} F_{i,j} = min\left(\sum_i P_i, \sum_j Q_j\right)$$

donde $\{F_{ij}\}$ denota el flujo. Cada F_{ij} representa la cantidad transportada de la oferta de i por la demanda de jth. Llamamos D_{ij} a la distancia de suelo entre el bin i y el bin j. Si D_{ij} es una métrica, el EMD es una medida solo para histogramas normalizados. Recientemente Pele y Werman sugirieron $\widehat{EMD}$.

$$\widehat{EMD}_\alpha^D(P,Q) = \left(\min_{\{F_{ij}\}} \sum_{ij} F_{ij} D_{ij}\right) + \left|\sum_i P_i - \sum_j Q_j\right| \alpha \max_{ij} D_{ij}$$

Si D_{ij}es una métrica y $\alpha \geq \frac{1}{2}$, $\widehat{EMD}$ es una métrica para todos los histogramas. Para histogramas normalizados $\widehat{EMD}$ y EMD son iguales.

3.5. EVALUACIÓN DE DESEMPEÑO

3.5.1. Validación cruzada

Técnica utilizada para evaluar los resultados de un análisis estadístico y garantizar que son independientes de la partición entre datos de entrenamiento y prueba. Consiste en repetir y calcular la media aritmética obtenida de las medidas de evaluación sobre diferentes particiones. Se utiliza en entornos donde el objetivo principal es la predicción y se quiere estimar como es de preciso un modelo que se llevará a cabo en la práctica.

La validación cruzada es la mejora del método de retención (holdout method). Este consiste en dividir en dos conjuntos los datos de muestra, se realiza el análisis de cada subconjunto de forma independiente, de forma que la función de aproximación solo se ajusta con el conjunto de datos de entrenamiento y a partir de aquí calcula los valores de salida para el conjunto de datos de prueba (valores que no se han analizado antes). Este es un método fácil de computar, pero no es demasiado preciso debido a la variación de resultados obtenidos para diferentes datos de entrenamiento.

- Objetivo de la validación cruzada: La validación cruzada es una manera de predecir el ajuste de un modelo a un hipotético conjunto de datos de prueba cuando no disponemos del conjunto explicito de datos prueba.
- Tipos de validación cruzada [40]

- Validación cruzada de K iteraciones. En esta validación se dividen los datos de muestra en k subconjuntos. Uno de los subconjuntos se utiliza como datos de prueba y el resto como datos de entrenamiento. El proceso de validación se repite durante k iteraciones, con cada uno de los posibles subconjuntos prueba. Finalmente se calcula la media aritmética de los resultados de cada iteración para obtener en un único resultado. La desventaja que tiene es que es lento desde punto de vista computacional .
- Validación cruzada aleatoria. Este método divide aleatoriamente el conjunto de datos de entrenamiento y el conjunto de datos de prueba. Para cada división la función de aproximación se ajusta a partir de los datos de entrenamiento y calcula los valores de salida para el conjunto de datos prueba. El resultado final es la media aritmética de los datos obtenidos para las diferentes divisiones. Su desventaja es que pueden quedar muestras sin evaluar y otras que se evalúan más de una vez.
- Validación cruzada dejando uno afuera. Esta validación implica separar los datos que para cada iteración tengamos una sola muestra para los datos de prueba y el resto de datos sea para entrenamiento. En este tipo de validación el error es muy bajo, pero el costo computacional se eleva puesto que se deben realizar tantas iteraciones como n muestras se tengan y para cada una analizar datos de entrenamiento y datos prueba.

3.5.2. Método de Monte Carlo

Su nombre se debe al famoso casino de Montercarlo, ya que el método presenta una similitud con los juego de ruleta de los casinos. Es un método estadístico utilizado para calcular probabilidades y otras cantidades relacionadas utilizando secuencias de números aleatorios. Este se utiliza para la aproximación de expresiones matemáticas complejas y costosas de evaluar con exactitud [3][52].

Este método permite la solución de problemas difíciles de resolver por métodos exclusivamente analíticos o numéricos, y que dependen de factores aleatorios o se pueden asociar a un modelo probabilístico artificial (resolución de integrales de muchas variables, minimización de funciones, etc).

Este método consta de una clase amplia de algoritmos computacionales que se basan en repetir el muestreo aleatorio para obtener resultados numéricos, típicamente se corren las simulaciones muchas veces con el fin de obtener la distribución de una entidad probabilística desconocida.

El error absoluto de la estimación decrece como [52][3]

$$errorAbs = \frac{1}{\sqrt{N}}$$

en virtud del teorema de límite central. N es el número de prueba, por tanto, obtener una cifra decimal precisa implica el aumento de N en 100 veces.

La base probabilística del método de Montecarlo es la generación de una secuencia de números aleatorios.

Los algoritmos de Montecarlo varían, pero tienden a seguir un mismo patrón:

1. Definir un dominio de las posibles entradas.
2. Generar las entradas al azar de una distribución de probabilidad sobre el dominio.
3. Realizar el cálculo determinista de las entradas.
4. Agregar los resultados.

Generación de números aleatorios

- Distribución uniforme: Para obtener una secuencia aleatoria a partir de una distribución uniforme [11]. La formula recursiva que genera la secuencia es:

$$x_{n+1} = (ax_n + c)\, mod$$

$$m$$

Donde a es el multiplicador, c es el incremento, m el modulo y x_o el valor inicial. Todos estos parámetros deben ser mayores que cero y cumplir cada uno con ciertas reglas para asegurar una secuencia aleatoria. Si los parámetros son escogidos adecuadamente, el periodo máximo de repetición que puede alcanzar la secuencia es igual a m.

- Distribuciones continuas: El método general para cualquier distribución continua se basa en las funciones acumuladas de distribución de probabilidad (cdf), $F(x)$. Se tiene entonces que:

$$y = F(x) \Leftrightarrow x = F^{-1}(y)$$

Si se tiene $F^{-1}(y)$ se puede obtener una variable aleatoria con una distribución continua, X, a partir de una variable aleatoria de distribución uniforme Y, tal que:

$$X = F^{-1}(Y)$$

3.5.3. Indicadores de desempeño

Porcentaje de Acierto Se define como:

$$\%Aciertos = \left(\frac{TotalAciertos}{TotalMuestras}\right) * 100$$

Porcentaje de Error Se define como:

$$\%Error = \left(\frac{TotalErrores}{TotalMuestras}\right) * 100$$

Sensibilidad y Especificidad Son las medidas básicas y tradicionales del valor del diagnóstico de la prueba. Miden la discriminación diagnóstica de una prueba en relación a un criterio de referencia que se considera verdad[68][27][25].

Sensibilidad Es la capacidad de la prueba de identificar como de la clase A los datos que realmente pertenecen a ella. Por lo tanto arroja como resultado la probabilidad de que datos de la clase A sean identificados de manera correcta [68][27][25].

La sensibilidad se define como:

$$Sensibilidad = \frac{VP}{VP + FN}$$

donde VP son los verdaderos positivos, es decir, los datos de la clase A que fueron clasificados como de esa clase y FN falsos negativos, es decir, los datos de la clase A que fueron clasificados como pertenecientes a la clase B.

Especificidad La especificidad es una prueba que indica la probabilidad de que los datos de la clase B sean clasificados como de esta clase. También se puede expresar como la capacidad de la prueba de identificar que los datos de la clase B no pertenecen a la clase A [68][27][25].

La especificidad se define como:

$$Especificidad = \frac{VN}{VN + FP}$$

donde VN son los verdaderos negativos, es decir, los datos de la clase B que fueron clasificados como de esa clase; y FP los falsos positivos, es decir, los datos de la clase B que fueron clasificados como pertenecientes a la clase A.

PROCESAMIENTO DE VIDEO

Basándose en que una señal de video es una secuencia de imágenes estáticas se determina que es posible realizar un análisis de una señal de video al procesar cierta cantidad de imágenes estáticas en un intervalo de tiempo.

Actualmente, la realidad aumentada viene siendo utilizada en diversas áreas de aplicación, como, marketing y publicidad, medicina, entretenimiento, arquitectura, robótica entre otros.

Es una tecnología que aparece con sus primeros trabajos investigativos en los años 90's. La realidad aumentada permite la mezcla de imágenes virtuales con imágenes reales, por lo que esta, permite a observador continuar en contacto con el mundo real y al mismo tiempo interactuar con objetos o imágenes virtuales.

De acuerdo al grado de incidencia de realidad o virtualidad se definen cuatro espacios: mundo real, realidad aumentada, virtualidad aumentada y mundo virtual [gráfica de realidad aumentada].

Los primeros conceptos de realidad aumentada estaban vinculados de manera directa con dispositivos de visualización, puntualmente a los dispositivos HMD (Head Mounted Display). Estos dispositivos ópticos le permiten al observador ver el mundo físico (su entorno) con información gráfica virtual que se superpone.

Luego, Bared y Hendrix en 1995 [12] definen la realidad aumentada como la ampliación de un mundo real con imágenes sintéticas. En este caso, la imagen sintética se utiliza como complemento a la escena real. Estos autores además, amplían la idea de aumentar otros sentidos (no solo el visual) con información táctil o auditiva.

Finalmente, en el año 2005, Bimber y Raskar [14] señalan que es un vínculo espacial el que establece la relación entre el mundo real y el mundo virtual. Esto implica el uso de elementos adicionales como sensores o marcadores que se instalan en el entorno y que actúan como referencia espacial para situar los objetos virtuales.

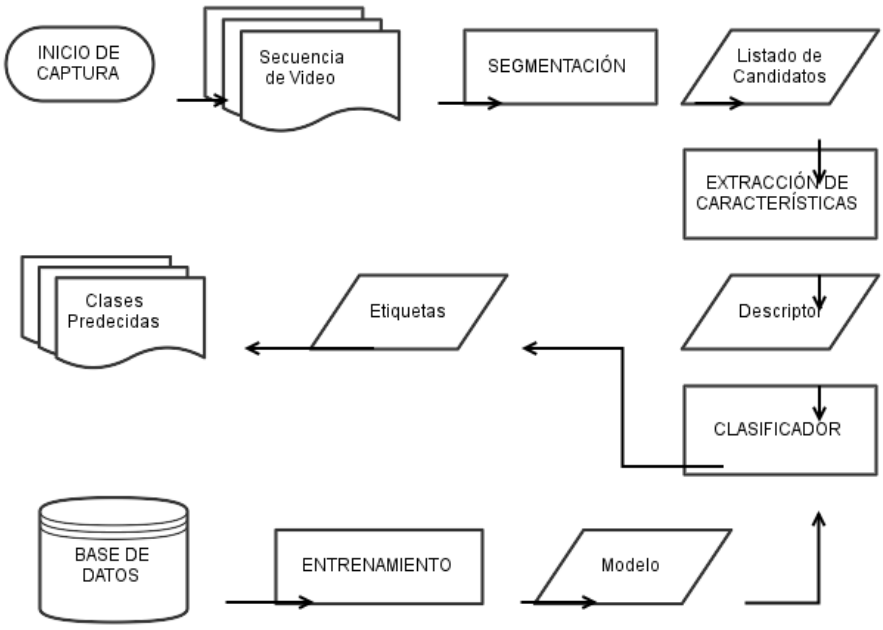


Figura 12: Diagrama de P

4. METODOLOGÍA

En este capítulo se describirán las etapas involucradas en el alcance de este proyecto. Para el desarrollo general, se dividió el problema en sus tres tareas fundamentales, segmentación, extracción de características y clasificación. Para cada una de estas etapas, se seleccionaron y probaron metodologías del estado del arte, como las presentadas en la sección 3, hasta obtener un balance adecuado entre desempeño y costo computacional. Para poder validar el funcionamiento de las tres etapas fundamentales, fue necesario obtener videos de juegos de fútbol donde se pudieran probar convenientemente las técnicas implementadas y analizar el desempeño de los algoritmos implementados con el mayor rigor estadístico posible.

Por esta razón, el procedimiento que se ilustra a continuación en el diagrama de flujo, Figura 12 también incluye la construcción de una base de datos, y un procedimiento de validación cruzada posterior al entrenamiento del sistema inteligente encargado de clasificar los jugadores. En este orden de ideas, el resto del presente capítulo está dividido en, la construcción de la base de datos, el diseño de la etapa de segmentación, el diseño de la etapa de extracción de características, el diseño de la etapa de clasificación y el procedimiento de validación de los resultados.

4.1. CONSTRUCCIÓN DE LA BASE DE DATOS

Con base en los requerimientos del procesamiento que se pretende hacer, se decidió construir la base de datos de videos de fútbol utilizando videos disponibles en el portal web YouTube©, de la compañía Google Inc. Se escogieron videos base cuyas características de grabación fueran típicas de una transmisión de fútbol por televisión. Entre las características técnicas que se prefirieron para estos videos iniciales se consideró que se pudieran descargar con resolución Standard HD de 720p (1280×720) píxeles en formato H.264 y relación de aspecto $16 : 9$. Entre las características de escena que se prefirieron, estuvo por ejemplo, que contuvieran muchas escenas de ataque donde uno de los equipos actuaba a la ofensiva mientras el otro se defendía, que existieran diversas condiciones de iluminación natural y artificial y que existieran escenas donde el público también se alcance a observar en la imagen, entre otras. Los videos escogidos contienen además gran cantidad de escenas no relevantes. Por tal razón, de cada video se escogieron algunos clips, o fragmentos, que contuvieran las escenas más interesantes para el presente trabajo. La duración promedio de cada clip es de 10 segundos.

Para la utilización de la base de datos fue necesario extraer cada uno de los fotogramas de cada video. De cada fotograma, o frame, se determinaron manualmente las coordenadas del centro de masa aproximado de cada uno de los actores de interés en la escena y se etiquetaron con una de las cuatro categorías posibles

- Equipo A
- Equipo B
- Juez
- Portero

Al rededor de cada actor en la escena, se establece una región de interés ROI, que es un rectángulo de tamaño 49×79 píxeles, y que es guardado en la subcarpeta correspondiente a su clase. La figura 13 muestra un ejemplo de ROIs para cada clase de algún frame de algún clip particular.

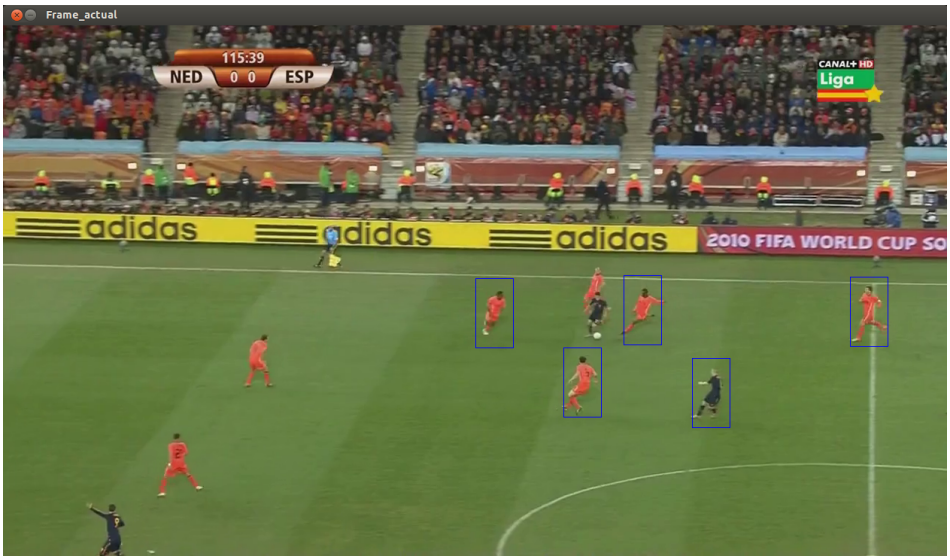
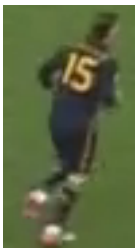


Figura 14: Regiones de interés vistas desde el clip


España, ejemplo 113


Holanda, ejemplo 385


Juez, ejemplo 226


Portero Holanda, ejemplo 167

Figura 13: Regiones de interés para el clip 16

Con el fin de visualizar en el clip las ROIs que están siendo seleccionadas, se dibuja un rectángulo con dimensión de 50×80 píxeles, ver Figura 14.

4.2. ENTRENAMIENTO

La fase de entrenamiento consiste en encontrar un vector de características de referencia por cada clase y para cada clip.

Para el presente desarrollo, se eligió el histograma de color en el espacio RGB como descriptor de la región de interés. Esta decisión se tomó fundamentalmente por dos razones. En primer lugar, porque la información más relevante para diferenciar el equipo al que pertenece el jugador es el color de su uniforme, y en segundo lugar, porque en esta aplicación se han

separado totalmente las etapas de detección e identificación, lo que permite asumir que la ROI ya contiene un jugador.

La figura 15 muestra la ROI para 4 ejemplos de jugadores de diferente clase a la que se le calculó el histogramas RGB, y que se representa en el espacio 3D mediante un cubo dividido en pequeñas celdas tridimensionales.

El procedimiento de entrenamiento consiste en realizar la extracción de características de cada una de las ROI de cada clase, para obtener el histograma promedio, asumiendo que la distribución del ruido que afecta a cada clase es de naturaleza Gausiana.

Si H_i es el histograma de la ROI i, entonces el histograma de referencia H_{ref_K} para la clase particular K, vendrá dado por

$$H_{ref_K} = \frac{1}{M} \sum_{i=1}^{M} H_i$$

siendo M el número total de ROIs disponibles para el entrenamiento de la clase K.

4.3. SEGMENTACIÓN

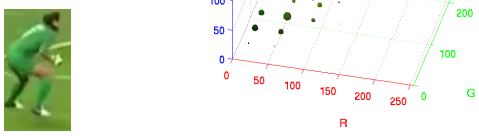
Para la obtención de la regiones de interés, y algunas de sus propiedades se realiza el procedimiento mostrado en el diagrama de flujo de la Figura 16:

4.3.1. Sustracción de fondo

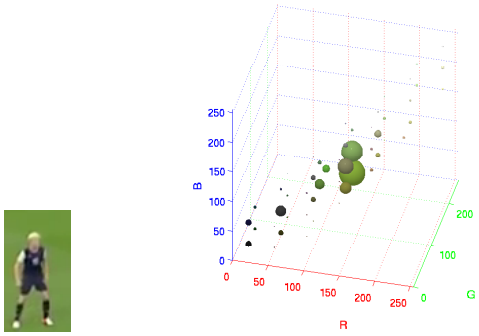
La segmentación se realiza por medio del método de sustracción de fondo utilizando Mezclas de Gausiana. Este método se basa en la presunción que los píxeles que pertenecen al fondo y los que pertenecen a los objetos de interés, se distribuyen probabilisticamente de alguna forma multimodal, que puede ser representada mediante la combinación de un número finito de distribuciones gausianas, obteniendo finalmente una máscara de primer plano. Se debe tener en cuenta que los parámetros de una Gaussiana están dados por su media μ y varianza σ.

Para encontrar los parámetros de cada una de las K Gausianas, se utiliza el método de Maximización de la Esperanza, cuyas ecuaciones de actualización vienen dadas por [62]:

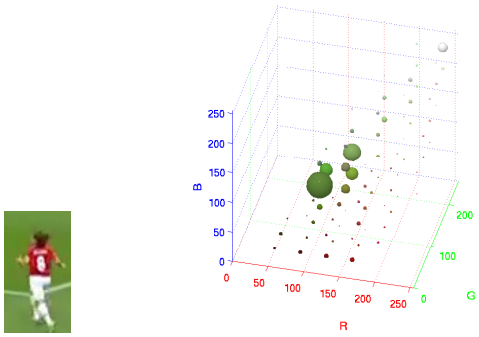
$$\hat{w}_k^{N+1} = \hat{w}_k^N + \frac{1}{N+1} \left(\hat{\rho} \left(\omega_k \left| x_{N+1} \right. \right) - \hat{w}_k^N \right)$$



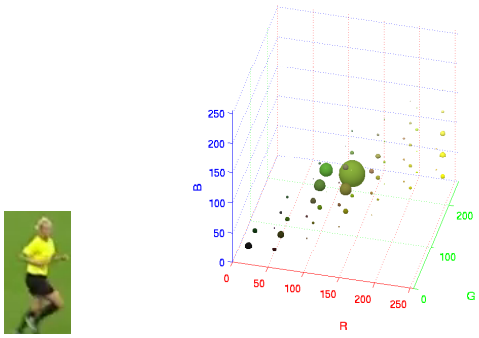
(a) Clase I



(b) Clase II



(c) Clase III



(d) Clase IV

Figura 15: Histogramas 3D RGB para 4 actores diferentes de un mismo fotograma.

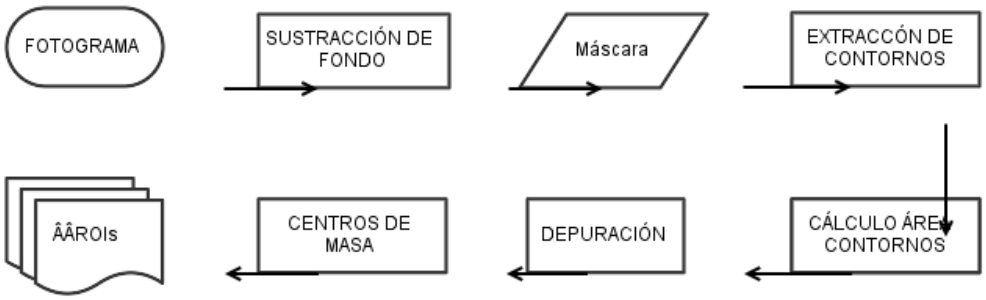
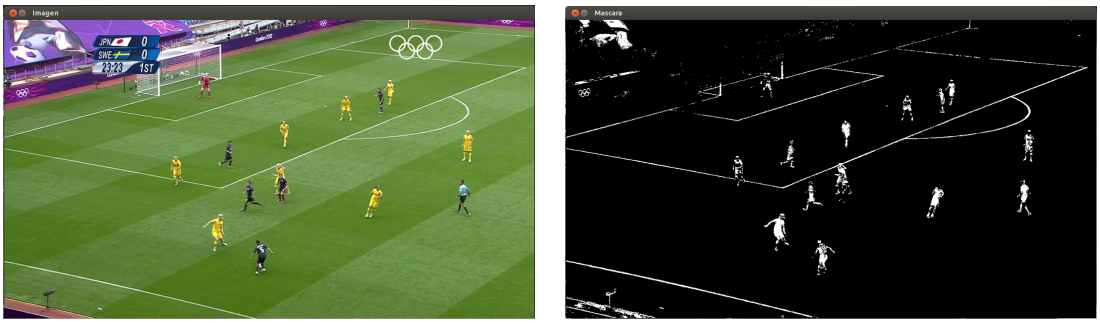


Figura 16: Diagrama de Flujo de la Segmentación

$$\hat{\mu}_k^{N+1} = \hat{\mu}_k^N + \frac{\hat{\rho}\left(\omega_k \left| x_{N+1} \right.\right)}{\sum_{i=1}^{N+1} \hat{\rho}\left(\omega_k \left| x_i \right.\right)} \left(x_{N+1} - \hat{\mu}_k^N\right)$$

$$\hat{\sum}_k^{N+1} = \hat{\sum}_k^N + \frac{\hat{\rho}((\omega | x_{N+1})}{\sum_{i=1}^{N+1} \hat{\rho}\left(\omega_k \left| x_i \right.\right)} \left(\left(x_{N+1} - \hat{\mu}_k^N\right) \left(x_{N+1} - \hat{\mu}_k^N\right)^T - \hat{\sum}_k^N \right)$$

Se define entonces un modelo de fondo estable que se adapta a los cambios de medio ambiente y que se actualiza de acuerdo a estos cambios, por medio del cual se obtiene la mascara de primer plano destacando los objetos en movimiento que para el caso serán los actores en el clip (ver Figura 12 17), quienes se definen como nuestras regiones de interés.



Clip Original Clip Mascara

Figura 17: Implementación del método MOG para sustraccion de fondo

4.3.2. Extracción de contornos

Se procede entonces, a encontrar los contornos que proporcionan la silueta de los objetos de primer plano, permitiendo identificar sus formas.

Es preciso recordar que la máscara está compuesta por píxeles blancos de valor uno que indican los objetos de primer plano (FO) y por píxeles negros de valor cero que representan el fondo (FB) lo que implica que FO>FB. Todo el conjunto de píxeles negros y el conjunto de píxeles blancos conforman lo que se conoce como componente conexa, la cual puede ser 0-componente para píxeles conectados de valor 0 y 1-componente para píxeles conectados de valor 1.

Para identificar las formas se recorre la imagen píxel por píxel y se realiza un proceso de etiquetado que consiste en asignar una rotulo (0 o 1) a cada uno de estos píxeles de acuerdo a la condición: si dos píxeles consecutivos tiene tienen el mismo valor se marca como cero; si dos píxeles consecutivos cambian de valor se marca como 1.

Basándose en el algoritmo de Suzuki [63] se hallan los contornos siguiendo la frontera entre **0-componentes** y **1-componentes**. Se inicia la búsqueda recorriendo la imagen binarizada píxel por píxel hasta encontrar un píxel con valor igual a uno, se almacena la coordenada de este en un vector de puntos, y luego se analizan sus vecinos (8-conectados) para encontrar otro píxel con el mismo valor. La acción la repite hasta llegar al píxel de partida logrado cerrar la figura y recorrer la imagen completamente, ver Figura 13 18.

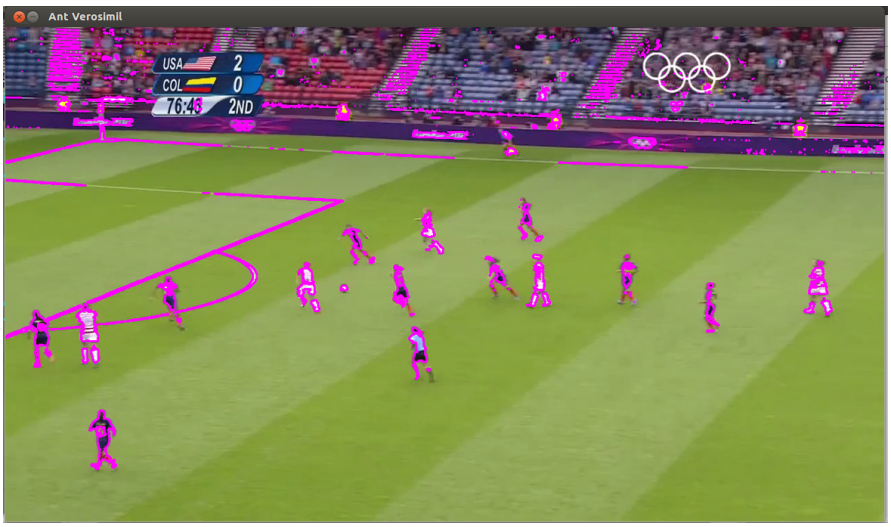


Figura 18: Obtención de los contornos

4.3.3. Calculo de áreas de contornos

Una vez obtenidos los contornos se calculan el área de estos mediante la aplicación del teorema de Green que relaciona una integral de línea a lo largo de una curva cerrada simple en un plano con una integral doble en la región encerrada:

Dado que σ es una curva cerrada simple, y $A(i(\sigma))$ es el área interior de σ que se define como:

$$A\left(i\left(\sigma\right)\right) = \iint_{i(\sigma)} \left(1\right) dxdy$$

Para cualquier curva se tiene el teorema:

$$\oint_{\sigma} \left(P\left(x,y\right) dx + Q\left(x,y\right) dy\right) = \iint_{i(\sigma)} \left(\frac{\partial Q\left(x,y\right)}{\partial x} - \frac{\partial P\left(x,y\right)}{\partial y}\right) dxdy$$

Donde i es la parte interna de σ y P y Q son funciones definidas en un conjunto Ω (conjunto conexo que contiene al interior de σ) en $\mathbb{R}$ de clase C^i $(P, Q : \Omega \to \mathbb{R}\ C^1)$.

Si se eligen P y Q convenientemente tal que

$$\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} = 1$$

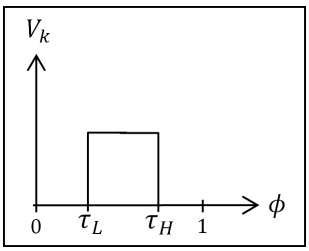
Entonces, el área vendrá dada simplemente por

$$A = \oint_{\sigma} xdy = -\oint_{\sigma} ydx = \frac{1}{2}\oint_{\sigma} \left(-ydx + xdy\right)$$

4.3.4. Depuración

MOG hace un buen trabajo al detectar los objetos de primer plano, pero no es perfecto, ya que alcanza detectar regiones que no son de interés para este trabajo, por lo cual se requiere depurar la segmentación.

Se escogió para esta etapa un filtrado morfológico basado en una función de verosimilitud, ver Figura 19, entre el objeto segmentado por MOG con lo esperado o la referencia teniendo en cuenta las áreas de los contornos obtenidos.



$$V_k = \begin{cases} 1, & \tau_L < \phi < \tau_K \\ \\ 0, & En\,otro\,caso \end{cases}$$

Figura 19: Función de Verosimilitud

La función de verosimilitud escogida es una función rectangular, debido a su bajo costo computacional.

como se puede visualizar en la Figura 14 20.

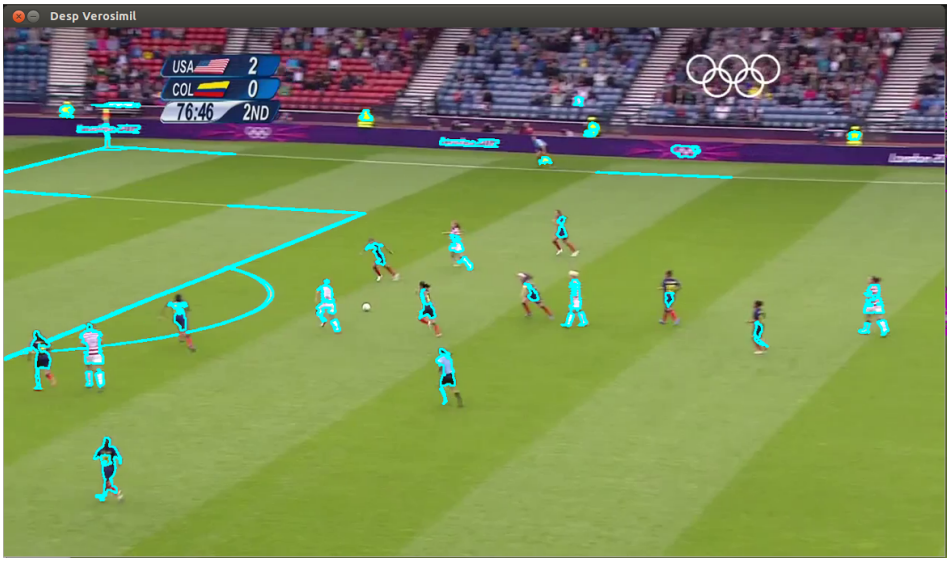


Figura 20: Aplicando Función de verosimilitud

4.3.5. Centro de Masa

Para calcular los centros de masa en primera instancia es necesario hallar los momentos centrales los cuales son un promedio ponderado de las intensidades de los píxeles de la imagen y están definidos como:

$$mu_{ji} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \bar{x})^j (y - \bar{y})^i f(x, y) dx dy$$

Como es el caso de una imagen digital, la ecuación anterior se convierte en:

$$mu_{ji} = \sum_{x} \sum_{y} (x - \bar{x})^j (y - \bar{y})^i f(x, y)$$

Donde las componentes del centro de masa son:

$$\bar{x} = \frac{m_{10}}{m_{00}}$$

$$\bar{y} = \frac{m_{01}}{m_{00}}$$

A partir del centro de masa se definirá un rectángulo con una dimensión de 49 píxeles de ancho por 91 píxeles de largo, este rectángulo contendrá la región de interés (ROI), ver Figura 1521.

Figura 21: ROI's a partir de los centros de masa

4.4. CLASIFICACIÓN

El proceso de clasificación se realiza utilizando técnicas de clasificación supervisada. La clasificación se realiza por medio de la comparación de histogramas de las imágenes, la distancia entre el histograma de referencia o histograma promedio de la clase y el histograma de la imagen que se quiere clasificar. Las etapas para la realización de este proceso se listan a continuación y serán descritas detenidamente:

1. Extracción de características.
2. Comparación de características vs referencia (Distancia de Bhattacharyya).
3. Etiquetado y filtrado.

4.4.1. Extracción de características:

Para esta etapa se toman las imágenes definidas como regiones de interés y se calcula el histograma de color en espacio RGB a cada una de las imágenes, lo que permite representar la cantidad de píxeles que tienen colores dentro de los rangos R, G y B.

El histograma de color de una imagen esta dado mediante la siguiente formula:

$$h_{R,G,B} = N\ Prob(R = r, G = g, C = c)$$

donde R, G y B representan los tres canales de color y N es el número de píxeles en la imagen.

4.4.2. Comparación.

Esta etapa consiste en la comparación de las características extraídas de los candidatos resultantes contra la referencia. Para este caso las características son los histogramas de cada candidato y la referencia el histograma promedio de la clase.

Como medida de similaridad, se utiliza la distancia estadística de Bhattacharyya d_B, que permite establecer la separabilidad entre dos distribuciones de probabilidad. El resultado es un valor entre 0 y 1, donde los valores cercanos a 0 indican alta similitud entre las distribuciones, y los valores cercanos a 1 indican que dichas distribuciones tienen una alta separabilidad.

La Distancia de Bhattacharyya esta dada por:

$$d_{Bhattacharyya}(H_1, H_2) = \sqrt{1 - \sum_i \frac{\sqrt{H_1(i).H_2(i)}}{\sqrt{\sum_i H_1(i). \sum_i H_2(i)}}}$$

Se toman los histogramas promedio de las clases de un clip y se hayan las distancias entre estos y cada histograma de los candidatos o las imágenes definidas como regiones de interés, que serán las imágenes a clasificar.

Para convertir esta distancia en una probabilidad de pertenencia, basta con hacer

$$p\left(H_i, H_{REF}\,|k\right) = -log\left(d_B\right)$$

lo cual, es básicamente una función de verosimilitud para el histograma respecto a cada clase.

4.4.3. Clasificador Bayesiano

Un clasificador Bayesiano, definido de acuerdo con la formula de Bayes

$$P\left(Clase|Datos\right) = \frac{P\left(Clase\right)P\left(Datos|Clase\right)}{P\left(Datos\right)}$$

puede interpretarse, de forma simple como

$$posterior = \frac{prior \times verosimilitud}{evidencia}$$

Este clasificador asigna los datos a aquella clase con posterior mayor. Es decir, aquella clase con mayor probabilidad de pertenencia.Si la probabilidad de pertenencia a todas las clases es muy baja, entonces se asume que es un dato atípico, probablemente resultado de una falla en la segmentación.

El las Figuras 16 22 y 17 23 se puede observar el etiquetado de los jugadores en la escena. Para ello se utilizan rectángulos de colores diferentes (azul, rojo,verde,amarillo), uno para cada clase.

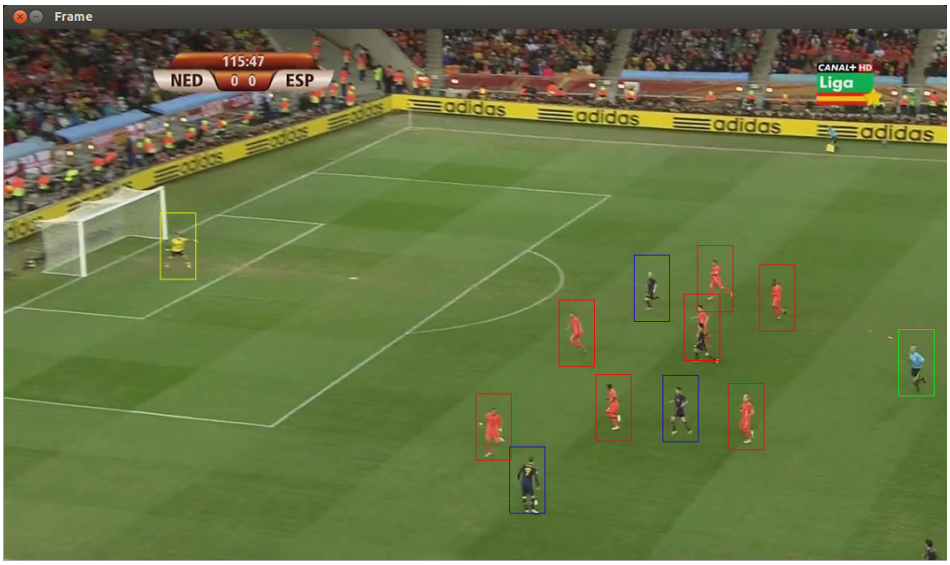


Figura 22: Etiquetado clip 16 frame 209

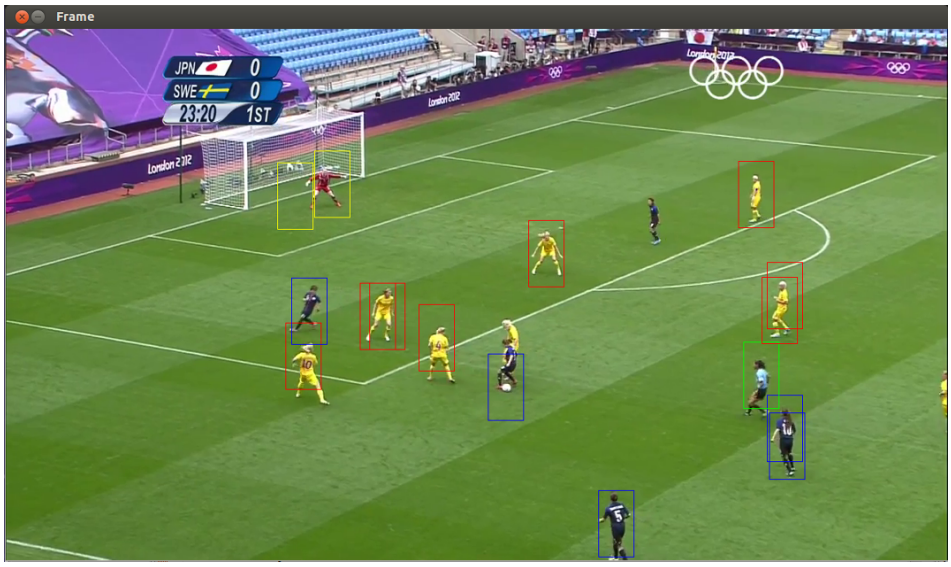


Figura 23: Etiquetado clip 46 frame 71

5. EXPERIMENTOS Y RESULTADOS

En este capítulo se presentan los resultados obtenidos al implementar la metodología descrita. Se describen algunos aspectos básicos de la implementación y de la evaluación de desempeño de la metodología implementada. Además se discuten los resultados obtenidos al aplicar la metodología a un total de 7 video clips extraídos de la base de datos.

5.1. IMPLEMENTACIÓN

Para el desarrollo de la aplicación se utilizó el Ambiente de Desarrollo Integrado IDE de Eclipse CDT sobre el sistema operativo Linux. Para la implementación de la metodología se utilizó el lenguaje de programación C++ con la librería de código abierto OpenCV© de WillowGarage©. Estas herramientas son estándares en la industria y la comunidad científica en Visión por Computador.

Para la realización de los experimentos se implementaron las siguientes etapas, para los cuales fue necesario utilizar los módulos principales de OpenCV como core, highgui, e imgproc y las funciones asociadas a estos módulos.

Etapa de entrenamiento.

El siguiente diagrama de bloques (figura 24) muestra el esquema de la etapa de entrenamiento:

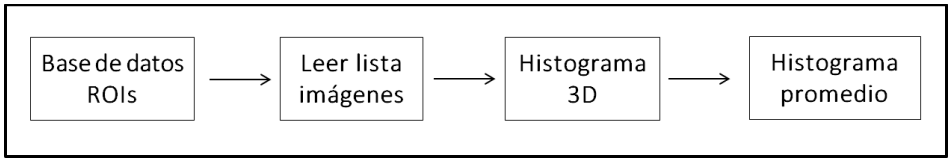


Figura 24: Etapa de entrenamiento

El objetivo para la etapa de entrenamiento es encontrar un vector de características de referencia en el espacio RGB como descriptor que determine el comportamiento de la distribución de color de las ROIs de cada una de las clases que se quieren clasificar.

Para esta etapa de entrenamiento se generaron las siguientes cuatro funciones:

- Función “leerlista”. Una función vector de tipo string, que tiene como parámetro de entrada un vector de tipo char. Esta función se encarga de leer un archivo de extensión .txt el cual contiene los nombres de las ROIs de la base de datos de cada clase. Este archivo fue generado previamente mediante la línea de comando ls -R > nombrearchivo.txt en la terminal de Linux.

- Función “hitogramaColor”. Una función matriz n dimensional que tiene como parámetros de entrada una variable tipo Mat y tres variables tipo int, una para el número de bins, y las otras dos para los valores de rango máximo y mínimo de las intensidades de los píxeles. Esta función calcula el histograma 3D para cada ROI con ayuda de la función calcHist del módulo de OpenCV, imgproc.

- Función “constM”. Esta función tiene como con parámetros entrada una variable tipo Mat ND para el histograma 3D calculado previamente, y una variable tipo int que indica el numero de bins utilizado. Esta función vectoriza el histograma 3D, es decir transforma la matriz en un vector de una sola fila y de n columnas, que contiene las características asociadas a cada espacio de color (Rojo, Verde, Azul). Este proceso lo hace a cada histograma 3D calculado de las ROIs de la base de datos, y los va almacenando en un vector de vectores de tipo Mat.

- Función “promedio”. Su parámetro de entrada es una variable de tipo Mat. Esta función obtiene el histograma promedio de la clase. Este promedio se obtiene sumando los valores que se encuentran en cada columna del vector de vectores, uno por uno y los divide por el número de filas (cantidad de ROIs de la base de datos de una clase).

Al final, la etapa de entrenamiento arroja un vector de características promedio en el espacio de color RGB (histograma 3D vectorizado) de las ROIs de la base de datos de cada clase, el cual es guardado en un archivo con extensión .yml con el nombre de la clase a la que pertenece y el numero del clip.

Parte de el código implementado se muestra a continuación:

```
cv::MatND histogramaColor(cv::Mat im, int bins, int min, int max)
{
    int histSize[] = {bins,bins,bins};
    float rranges[] = {min, max};
    float granges[] = {min, max};
    float branges[] = {min, max};
    int channels[] = {0,1,2};
    const float* ranges[] = { rranges, granges, branges};
    cv::MatND hist;
    cv::calcHist(&im, 1, channels, cv::Mat(),hist, 3, histSize, ranges, true, false);
    return hist;
}
```

```
Mat constM(cv::MatND HistC, int bins)
{   Mat g=Mat::zeros(1,1,CV_32F);
     for(int i=0; i<bins; i++)
     {
         for(int j=0; j<bins; j++)
         {
             for(int k=0; k<bins; k++)
             {
                 g.push_back(HistC.at<float>(i,j,k));
             }
         }
     }
return g;
}
```

```
Mat promedio (Mat RGB)
{
    int fil=RGB.rows;
    Mat prom;
    Mat ant, nuevo;
    ant=Mat::zeros(1,RGB.cols,CV_32F);
    for(int i=0; i<fil; i++)
    {
        nuevo=RGB.row(i);
        ant += nuevo;
    }
    prom=ant/((float)fil);
    return prom;
}
```

Etapa de Segmentación.

En el siguiente diagrama de bloque (figura 25) se muestra el esquema de la etapa de segmentación:

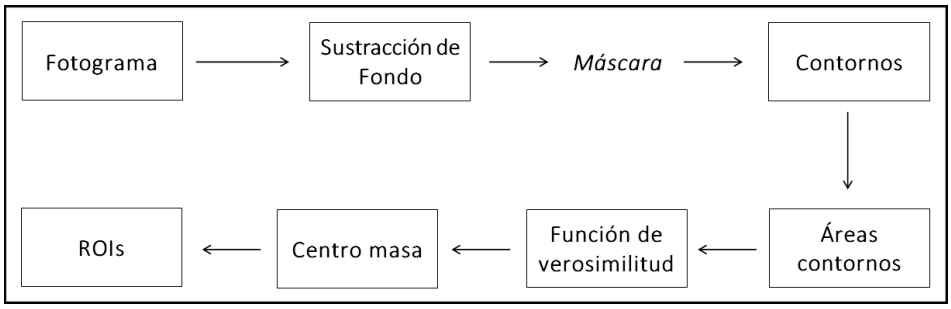


Figura 25: Etapa de segmentación

El objetivo para la esta etapa es segmentar el fondo y el primer plano en cada frame para obtener los objetos de interés, que en este caso son los jugadores que se encuentran en la escena, incluyendo al portero y al juez.

Para la etapa de segmentación fue necesario incluir la librería background_segm.hpp del módulo de video de OpenCV.

En esta etapa se utiliza la clase VideoCapture del modulo highgui para capturar los clips de la base de datos. Para cada frame se obtiene una máscara de primer plano aplicando la función de getBackgroundImage. Posteriormente, al tener esta máscara se encuentran los contornos de los objetos segmentados en dicha imagen aplicando para ello la función findContours del módulo imgproc.

Adquiridos los contornos, se procede a encontrar sus las áreas con la función contourArea del módulo imgproc. Una vez obtenidos los valores de las áreas se aplica una función de verosimilitud para descartar los objetos en el frame que no son de interés como por ejemplo el público y mantener los objetos que son posibles jugadores, de los cuales se les hallan los momentos con la función moments del modulo imgporc, permitiendo hallar posteriormente los centros de masa.

Se utiliza la función "ROII" creada previamente para generar las regiones de interés ROIs en cada frame a partir de las coordenadas los centros de masa.

Parte del código implementado se muestra a continuación:

```
VideoCapture cap ("clip16.mp4");
bool update_bg_model = true;

if( !cap.isOpened() )
{
    printf("No se puede abrir el archivo\n");
    return -1;
}
```

```
BackgroundSubtractorMOG bg_model;
                .
                .
                .
Mat bgimg;
bg_model.getBackgroundImage(bgimg);
```

```
//Momentos
vector<Moments> mu(compC.size() );
for( int i = 0; i < compC.size(); i++ )
{
    mu[i] = moments(compC[i], false );
}

//Centro de Masa
vector<Point> mc(compC.size());
for( int i = 0; i < compC.size(); i++ )
{
    mc[i] = Point2f( mu[i].m10/mu[i].m00 , mu[i].m01/mu[i].m00);
}
```

Etapa de Clasificación.

El siguiente diagrama de bloques muestra el esquema de la etapa de clasificación, ver Figura 26:

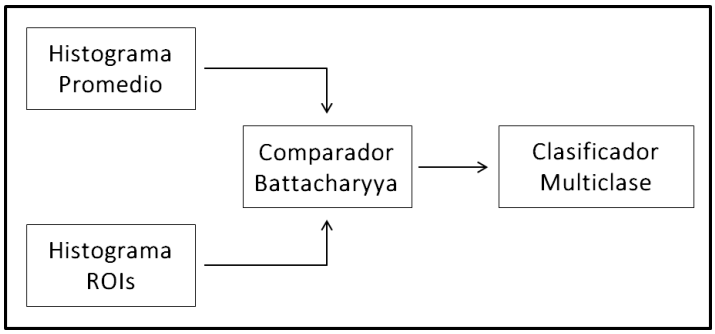


Figura 26: Etapa de clasificación

Los archivos de extensión .yml que contienen los histogramas promedio de las cuatro clases de un clip (equipo A, equipo B, juez, portero X), son cargados en el main del programa.

En esta etapa se calcula para cada una de las ROIs halladas en la etapa anterior, los histogramas 3D y se vectorizan como se hizo en la etapa de entrenamiento, haciendo uso de las funciones "hitogramaColor" y "constM".

Se hace una comparación mediante la función compareHist del módulo imgproc, entre cada una de las referencias de las clases (histogramas promedio) con cada región de interés utilizando el método de Bhattacharyya. El valor obtenido de esta comparación se almacena en un vector de cuatro posiciones. La primera posición guarda la comparación con el Equipo A, la segunda con el Equipo B, la tercera con el Juez y la cuarta posición almacena la comparación con el Portero X.

Se hace uso de la función minMaxIdx perteneciente al modulo core, la cual encuentra los valores máximo y mínimo del vector de comparaciones y su posición en él.

La función compareHist arroja un valor entre 0 y 1, siendo este el coeficiente de Bhattacharyya, donde los valores cercanos a cero indican que ambos histogramas tienen una distribución de probabilidad similares, y que valores cercanos a uno indican que dichas distribuciones son muy diferentes. Teniendo en cuenta este criterio se establece entonces la función de máxima verosimilitud la cual solo permite pasar los valores menores a cierto umbral o threshold.

La función de máxima verosimilitud indica que se podría tratar de un jugador y quien determina a que clase pertenece es el clasificador.

Parte del código implementado se muestra a continuación:

```
hb_A=compareHist(promed_A,RGB,CV_COMP_BHATTACHARYYA);
hb_B=compareHist(promed_B,RGB,CV_COMP_BHATTACHARYYA);
hb_C=compareHist(promed_C,RGB,CV_COMP_BHATTACHARYYA);
hb_D=compareHist(promed_D,RGB,CV_COMP_BHATTACHARYYA);

distancias.at<double>(0,0)=hb_A;
distancias.at<double>(1,0)=hb_B;
distancias.at<double>(2,0)=hb_C;
distancias.at<double>(3,0)=hb_D;

minMaxIdx(distancias, &Min, &Max, &idmin, &idmax);
```

```
if(idmin==0){drawRectt(mc.at(j).x,mc.at(j).y,255,0,0);} //Azul
if(idmin==1){drawRectt(mc.at(j).x,mc.at(j).y,0,0,255);}//Rojo
if(idmin==2){drawRectt(mc.at(j).x,mc.at(j).y,0,255,0);}//Verde
if(idmin==3){drawRectt(mc.at(j).x,mc.at(j).y,0,255,255);}//Amarillo
```

5.2. EVALUACIÓN DE DESEMPEÑO

Para analizar los video clips seleccionados, se utilizó la técnica de análisis cuadro por cuadro, donde los valores verdaderos de referencia fueron capturados y etiquetados de forma manual. Luego, se comparan con los resultados obtenidos mediante la metodología automática de detección de jugadores, para por último hacer una comparación estadística determinando indicadores de desempeño como error, sensibilidad y especificidad de la detección.

Para consolidar esta información, se utilizó la llamada matriz de confusión, que permite fácilmente determinar la relación de clasificación por cada clase. Entre los clips analizados, se encuentran fundamentalmente los siguientes:

5.2.1. Clip 16

Este clip hace parte del partido de fútbol entre España y Holanda por la final del mundial de Sudafrica 2010. Para este clip se tomaron las siguientes cuatro clases:

- **C1:** Jugadores de España (Uniforme azul oscuro)
- **C2:** Jugadores de Holanda (Uniforme naranja)
- **C3:** Juez (Camiseta Azul clara)
- **C4:** Portero de Holanda (Camiseta amarilla)

De cada clase se obtuvieron, durante la duración del clip, el siguiente número de ejemplos.

- **C1:** 1.098 muestras
- **C2:** 2.121 muestras
- **C3:** 864 muestras
- **C4:** 573 muestras

Utilizando cada uno de estos ejemplos, se obtiene un vector de características, como por ejemplo el histograma RGB, con los cuales se obtienen los histogramas promedio que servirán como la referencia de cada clase.

Las siguientes definiciones deben ser tenidas en cuenta para la interpretación de la tabla de confusión de cada clase:

Verdaderos positivos: Imágenes de la clase Cx que fueron clasificadas correctamente como de la clase Cx

Falsos positivos: Imágenes de otras clases que fueron clasificadas incorrectamente como clase Cx (Suma Vertical)

Falsos_negativos: Imágenes de la clase Cx que fueron clasificadas incorrectamente como de otra clase (Suma Horizontal)

Verdaderos_negativos: Imágenes de las clases restantes correctamente clasificadas como no pertenecientes a Cx

Los resultados para este clip tras aplicar la metodología se muestran en la matriz de confusión, ver Tabla 1:

	C1	**C2**	**C3**	**C4**
C1	**962**	64	37	0
C2	3	**1917**	0	0
C3	0	0	**303**	0
C4	0	0	0	**65**

	C1	**C2**	**C3**	**C4**
C1	**90 %**	6 %	4 %	0 %
C2	0 %	**100 %**	0 %	0 %
C3	0 %	0 %	**100 %**	0 %
C4	0 %	0 %	0 %	**100 %**

Tabla 1: Matriz de confusión para el clip 16

Por los resultados arrojados por la matriz de confusión para el clip 16, se puede decir lo siguiente:

- Para la clase C1 se detectan 1.063 jugadores, de los cuales el 90 % fueron clasificados correctamente como C1, el 6 % fueron clasificados como C2 y 4 % jugadores fueron clasificados como C3. Lo que arroja un porcentaje de error en la clasificación de jugadores de la clase C1 del 10 % y un 90 % de acierto.

- Para la clase C2 se detectan 1.920 jugadores, de los cuales el 100 % fueron clasificados correctamente dentro de C2. Por lo que la clasificación para este caso tiene un acierto del 100 %.

- Para la clase C3 se detectan 303 jugadores, y se obtiene un acierto en la clasificación del 100 % al clasificar todos los jugadores de esta clase de manera correcta.

- Para la clase C4 se detectan 65 jugadores, con un resultado del 100 % de acierto en la clasificación para esta clase, ya que todos los jugadores fueron clasificados como pertenecientes a C4.

  Para el análisis se obtiene la siguiente tabla de confusión, ver Tabla 2:

	VP	**FN**	**FP**	**VN**
C1	962	101	3	2285
C2	1917	3	64	1367
C3	303	0	37	3011
C4	65	0	0	3286

Tabla 2: Tabla de desempeño por cada clase

Donde **VP** son Verdaderos Positivos, **FN** son falsos negativos, **FP** son falsos positivos y **VN** verdaderos negativos. Teniendo en cuenta esta información se calcula la Sensibilidad y Especificidad como indicadores de desempeño para cada clase de acuerdo a las siguientes formulas:

$$Sensibilidad = \frac{VP}{VP + FN}$$

$$Especificidad = \frac{VN}{VN + FP}$$

	C1	**C2**	**C3**	**C4**
Sensibilidad	90 %	100 %	100 %	100 %
Especificidad	100 %	96 %	99 %	100 %

Tabla 3: Indicadores de desempeño Clip 16

5.2.2. Clip 45.

Este clip de video hace parte del partido de fútbol femenino de los Olímpicos de Londres 2012 entre la selección de Japón y la selección de Suecia en fase de grupos. Para este clip se tomaron las siguientes cuatro clases:

- **C1:** Jugadoras de Colombia (Uniforme azul oscuro)
- **C2:** Jugadoras de Korea (Uniforme blanco)
- **C3:** Juez (Uniforme negro)
- **C4:** Portera de Colombia (Uniforme verde)

Para generar el histograma promedio de cada clase se tomo la siguiente cantidad de muestras:

- **C1:** 787 muestras
- **C2:** 654 muestras
- **C3:** 133 muestras
- **C4:** 184 muestras

Los resultados para este clip tras aplicar la metodología se muestran en la matriz de confusión, ver Tabla 4:

	C1	**C2**	**C3**	**C4**
C1	**550**	0	14	0
C2	0	**564**	0	2
C3	0	0	**14**	0
C4	0	0	0	**108**

	C1	**C2**	**C3**	**C4**
C1	**98 %**	0 %	2 %	0 %
C2	0 %	**100 %**	0 %	0 %
C3	0 %	0 %	**100 %**	0 %
C4	3 %	0 %	0 %	**97 %**

Tabla 4: Matriz de confusión clip 45

- Para la clase C1 se detectan 564 jugadores, de los cuales el 98 % fueron clasificados correctamente como C1, y solo el 2 % fue erróneamente clasificado en C3, observando un buen comportamiento del clasificador con respecto a la clasificación de los jugadores detectados de esta clase.
- Para la clase C2 se detectan 566 jugadores, de los cuales el 100 % fueron clasificados correctamente dentro de C2. Por lo que la clasificación para este caso tiene un acierto del 100 %.

- Para la clase C3 se detectan 14 jugadores, y se obtiene un acierto en la clasificación del 100 % al clasificar todos los jugadores de esta clase de manera correcta.
- Para la clase C4 se detectan 111 jugadores, de los cuales el 3 % son clasificados de manera errada en C1, y el 98 % restante se clasificada correctamente como pertenecientes a la clase C4.

  Para el análisis se obtiene la siguiente tabla de confusión, ver Tabla 5:

	VP	**FN**	**FP**	**VN**
C1	550	14	3	688
C2	564	2	0	689
C3	14	0	14	1227
C4	108	3	2	1142

Tabla 5: Tabla de Confusión Clip45

Se calcula la Sensibilidad y Especificidad para cada clase 6:

	C1	**C2**	**C3**	**C4**
Sensibilidad	98 %	100 %	100 %	97 %
Especificidad	100 %	100 %	99 %	100 %

Tabla 6: Indicadores de desempeño Clip 45

5.2.3. Clip 46

Este clip de video hace parte del partido de fútbol femenino de los Olímpicos de Londres 2012 entre la selección de Japón y la selección de Suecia en fase de grupos. Para este clip se tomaron las siguientes cuatro clases:

- **C1:** Jugadoras de Japón (Uniforme azul)
- **C2:** Jugadoras de Suecia (Uniforme amarillo)
- **C3:** Juez (Camiseta azul clara)
- **C4:** Portera de Suecia (Uniforme rojo)

Para generar el histograma promedio de cada clase se tomo la siguiente cantidad de muestras:

- **C1:** 589 muestras

- **C2:** 773 muestras
- **C3:** 390 muestras
- **C4:** 325 muestras

Tras aplicar la metodología los resultados para este clip se muestran en la matriz de confusión, ver Tabla 7:

	C1	**C2**	**C3**	**C4**
C1	**247**	0	0	0
C2	0	**271**	0	0
C3	0	0	**66**	0
C4	0	0	0	**5**

	C1	**C2**	**C3**	**C4**
C1	**100 %**	0 %	0 %	0 %
C2	0 %	**100 %**	0 %	0 %
C3	0 %	0 %	**100 %**	0 %
C4	0 %	0 %	0 %	**100 %**

Tabla 7: Matriz de confusión clip 46

- Se detectan 466 jugadores pertenecientes a la clase C1, de los cuales el 92 % fueron clasificados de forma acertada de la clase C1, y el 8 % restante fue clasificado incorrectamente como de la clase C3, incurriendo en un error en clasificación del 8 %.
- Se detectan 1224 jugadores pertenecientes a la clase C2, de los cuales el 100 % fueron clasificados correctamente dentro de C2, obteniendo un acierto del 100 %.
- Se detectan 147 jugadores pertenecientes a la clase C3, donde solo el 3 % de estos fueron clasificados como pertenecientes a las clase C1 de forma errada y el 97 % son clasificados acertadamente dentro de C3.
- Para la clase C4 se detectan 145 jugadores, donde el 100 % se clasifican de manera correcta dentro de dicha clase.

Para el análisis se obtiene la siguiente tabla de confusión, ver Tabla 8

	VP	**FN**	**FP**	**VN**
C1	428	36	4	1512
C2	1224	0	1	757
C3	143	4	37	1798
C4	145	0	0	1837

Tabla 8: Tabla de Confusión Clip 46

Se calcula la Sensibilidad y Especificidad para cada clase, ver Tabla 9:

	C1	C2	C3	C4
Sensibilidad	92 %	100 %	97 %	100 %
Especificidad	100 %	100 %	98 %	100 %

Tabla 9: Indicadores de desempeño Clip 46

5.2.4. Clip 68.

Este clip de video hace parte del partido de fútbol femenino de los Olímpicos de Londres 2012 entre la selección de Colombia y la selección de Estados Unidos en fase de grupos. Para este clip se tomaron las siguientes cuatro clases:

- **C1:** Jugadoras de Colombia (Uniforme azul oscuro)
- **C2:** Jugadoras de Estados Unidos (Uniforme blanco rayas rojas)
- **C3:** Juez (Camiseta Azul clara)
- **C4:** Portera de Colombia (Uniforme verde)

Para generar el histograma promedio de cada clase se tomo la siguiente cantidad de muestras:

- **C1:** 579 muestras
- **C2:** 439 muestras
- **C3:** 262 muestras
- **C4:** 65 muestras

Los resultados para este clip tras aplicar la metodología se muestran en la matriz de confusión 10:

	C1	C2	C3	C4
C1	**247**	0	0	0
C2	0	**271**	0	0
C3	0	0	**66**	0
C4	0	0	0	**5**

	C1	C2	C3	C4
C1	**100 %**	0 %	0 %	0 %
C2	0 %	**100 %**	0 %	0 %
C3	0 %	0 %	**100 %**	0 %
C4	0 %	0 %	0 %	**100 %**

Tabla 10: Matriz de confusión clip 68

- De 247 detecciones de jugadores pertenecientes a la clase, fueron clasificados correctamente como de la clase C1 el 100 %, obtienen do una clasificación acertada del 100 %.
- Para la clase C2 se detectan 271 jugadores, y se obtiene un acierto en clasificación del 100 % al clasificar todos los jugadores correctamente.
- Se detectan para la clase C3 66 jugadores, donde al realizar la clasificación no se presenta ningún error al clasificar el 100 % de los jugadores como pertenecientes a esta clase.
- Para la clase C4 se detectan 5 jugadores, de los cuales todos se clasifican correctamente dentro de la clase C4, obteniendo un acierto del 100 %.

  Para el análisis se obtiene la siguiente tabla de confusión, ver Tabla 11:

	VP	**FN**	**FP**	**VN**
C1	247	0	0	342
C2	271	0	0	318
C3	66	0	0	523
C4	5	0	0	584

Tabla 11: Tabla de confusión

Se calcula la Sensibilidad y Especificidad para cada clase, obteniendo la siguiente tabla. Ver Tabla 12:

	C1	**C2**	**C3**	**C4**
Sensibilidad	100 %	100 %	100 %	100 %
Especificidad	100 %	100 %	100 %	100 %

Tabla 12: Indicadores de desempeño Clip 68

5.2.5. Clip 51.

Este clip de video hace parte del partido de fútbol femenino de los Olímpicos de Londres 2012 entre la selección de Sudafrica y la selección de Suecia en fase de grupos. Para este clip se tomaron las siguientes cuatro clases:

- **C1:** Jugadoras de Sudafrica (Uniforme blanco)
- **C2:** Jugadoras de Suecia (Uniforme amarillo)
- **C3:** Juez (Camiseta azul clara)

- **C4:** Portera de Suecia (Camiseta negra)

Para generar el histograma promedio de cada clase se tomo la siguiente cantidad de muestras:

- **C1:** 1374 muestras
- **C2:** 1228 muestras
- **C3:** 330 muestras
- **C4:** 41 muestras

Los resultados para este clip tras aplicar la metodología se muestran en la matriz de confusión, ver Tabla 13:

	C1	**C2**	**C3**	**C4**
C1	**606**	1	0	0
C2	8	**507**	0	0
C3	4	0	**101**	0
C4	0	0	0	**24**

	C1	**C2**	**C3**	**C4**
C1	**100 %**	0 %	0 %	0 %
C2	2 %	**98 %**	0 %	0 %
C3	4 %	0 %	**96 %**	0 %
C4	0 %	0 %	0 %	**100 %**

Tabla 13: Matriz de confusión Clip 51

- Se obtienen 607 detecciones de jugadores de la clase C1, donde se clasifican como pertenecientes a esta clase el 100 % de estos jugadores, arrojando como resultado un acierto en clasificación del 100 %.
- Se obtienen 515 detecciones de jugadores de la clase C2, de los cuales el 98 % es clasificado de forma acertada, y se tienen un error del 2 % al clasificar este porcentaje de jugadores como pertenecientes a clases diferentes a C2, para este caso fueron clasificados como clase C1.
- Para la clase C3 se detectan 105 jugadores, de los cuales el 4 % fueron clasificados como de la clase C1 y el 98 % restante como de clase C3, obteniendo por tanto un acierto del 98 %.
- Para la clase C4 se detectan 44 jugadores, para este caso se obtiene un acierto del 100 % ya que todos los jugadores detectados para esta clase fueron clasificados correctamente como pertenecientes a esta.

  Para el análisis se obtiene la siguiente tabla de confusión, ver Tabla 14:

	VP	FN	FP	VN
C1	606	1	12	632
C2	507	8	1	735
C3	101	4	0	1146
C4	24	0	0	1227

Tabla 14: Tabla de confusión Clip 51

Se calcula la Sensibilidad y Especificidad para cada clase. Ver Tabla 15:

	C1	C2	C3	C4
Sensibilidad	100 %	98 %	96 %	100 %
Especificidad	98 %	100 %	100 %	100 %

Tabla 15: Indicadores de desempeño Clip 51

5.2.6. Clip 40.

Este clip de video hace parte del partido de fútbol entre Alemania y Suecia en la fase de octavos de final del Mundial Almenia 2006. Para este clip se tomaron las siguientes cuatro clases:

- **C1:** Jugadores de Alemania (Uniforme blanco)
- **C2:** Jugadores de Suecia (Uniforme amarillo)
- **C3:** Juez (Uniforme negro)
- **C4:** Portero de Suecia (Uniforme rojo)

Para generar el histograma promedio de cada clase se tomo la siguiente cantidad de muestras:

- **C1:** 650 muestras
- **C2:** 1230 muestras
- **C3:** 257 muestras
- **C4:** 66 muestras

Los resultados para este clip tras aplicar la metodología se muestran en la matriz de confusión, ver Tabla 16:

	C1	**C2**	**C3**	**C4**
C1	**383**	0	39	2
C2	0	**461**	0	0
C3	0	0	**166**	0
C4	0	0	0	**21**

	C1	**C2**	**C3**	**C4**
C1	**90 %**	0 %	9 %	0 %
C2	0 %	**100 %**	0 %	0 %
C3	0 %	0 %	**100 %**	0 %
C4	0 %	0 %	0 %	**100 %**

Tabla 16: Matriz de confusión Clip 40

- Para la clase C1 se obtienen 424 detecciones de jugadores, donde el 90 % se clasifica de manera acertada como pertenecientes a esta clase, y se obtiene un error de clasificación del 10 % al clasificar los jugadores correspondientes a este porcentaje como pertenecientes a la clase C3.

- Se detectan 461 jugadores de la clase C2, y se obtiene un acierto en la clasificación del 100 % al clasificar correctamente todos los jugadores detectados de esta clase.

- Para la clase C3 se detectan 166 jugadores, de los cuales el 100 % de estos son clasificados correctamente como de la clase C3, con un porcentaje del error del 0 %.

- Se obtienen 21 detecciones de jugadores de la clase C4, donde el 100 % se clasificados de manera acertada.

  Para el análisis se obtiene la siguiente tabla de confusión, ver Tabla 17:

	VP	**FN**	**FP**	**VN**
C1	383	41	0	648
C2	461	0	0	609
C3	166	0	39	867
C4	21	0	2	1049

Tabla 17: Tabla de confusión Clip 40

Se calcula la Sensibilidad y Especificidad para cada clase. Ver Tabla 18:

	C1	**C2**	**C3**	**C4**
Sensibilidad	100 %	98 %	96 %	100 %
Especificidad	98 %	100 %	100 %	100 %

Tabla 18: Indicadores de desempeño Clip 40

5.2.7. Clip 54.

Este clip de video hace parte del partido de fútbol femenino entre Estados Unidos y Japón disputando la final de los Juegos Olímpicos Londres 2012. Para este clip se tomaron las siguientes cuatro clases:

- **C1:** Jugadores de Estados Unidos (Uniforme azul oscuro)
- **C2:** Jugadores de Japón (Uniforme rojo)
- **C3:** Juez (Camiseta amarilla)
- **C4:** Portero de Estados Unidos (Uniforme verde)

Para generar el histograma promedio de cada clase se tomo la siguiente cantidad de muestras:

- **C1:** 852 muestras
- **C2:** 838 muestras
- **C3:** 221 muestras
- **C4:** 132 muestras

Los resultados para este clip tras aplicar la metodología se muestran en la matriz de confusión, ver Tabla 19:

	C1	**C2**	**C3**	**C4**
C1	**974**	0	0	1
C2	0	**395**	0	0
C3	0	0	**39**	0
C4	0	0	0	**11**

	C1	**C2**	**C3**	**C4**
C1	**100 %**	0 %	0 %	0 %
C2	0 %	**100 %**	0 %	0 %
C3	0 %	0 %	**100 %**	0 %
C4	0 %	0 %	0 %	**100 %**

Tabla 19: Matriz de confusión Clip 54

- Para la clase C1 se detectan 975 jugadores, de los cuales el 100 % fueron clasificados correctamente dentro de C1. Por lo que la clasificación para este caso tiene un acierto del 100 %.
- Para la clase C2 se detectan 395 jugadores, de los cuales el 100 % fueron clasificados correctamente dentro de C2. Por lo que la clasificación para este caso tiene un acierto del 100 %.

- Para la clase C3 se detectan 39 jugadores, de los cuales el 100 % fueron clasificados correctamente dentro de C3. Por lo que la clasificación para este caso tiene un acierto del 100 %.

- Para la clase C4 se detectan 11 jugadores, de los cuales el 100 % fueron clasificados correctamente dentro de C4. Por lo que la clasificación para este caso tiene un acierto del 100 %.

  Para el análisis se obtiene la siguiente tabla de confusión, ver Tabla 20:

	VP	**FN**	**FP**	**VN**
C1	974	1	0	445
C2	395	0	0	1025
C3	39	0	0	1380
C4	11	0	1	1408

Tabla 20: Tabla de confusión Clip 54

Se calcula la Sensibilidad y Especificidad para cada clase. Ver Tabla 21:

	C1	**C2**	**C3**	**C4**
Sensibilidad	100 %	100 %	100 %	100 %
Especificidad	100 %	100 %	100 %	100 %

Tabla 21: Indicadores de desempeño Clip 54

5.3. DISCUSIÓN DE RESULTADOS

En los resultados obtenidos se observa que el porcentaje de acierto en la clasificación en general esta por superior al 90 % para todos los clips en cada una de las clases evaluadas. Para los clips 16 y clip 40 en las clases definidas como C1 se presenta un porcentaje de acierto igual 90 %, siendo este el más bajo en la evaluación.

El clasificador presenta un comportamiento ideal con los clips 54 y 68, los cuales no presentan error en clasificación de los jugadores detectados para ninguna de sus clases.

Se observa un desempeño satisfactorio del método implementado para la clasificación, teniendo en cuenta que la sensibilidad y especificidad para los clips es mayor o igual a 90 % en todos los casos, por lo que se puede inferir que el método tiene una alta capacidad para identificar correctamente los jugadores pertenecientes a una clase y los que no pertenecen a ella.

6. CONCLUSIONES Y RECOMENDACIONES

6.1. CONCLUSIONES

La metodología desarrollada para la segmentación y clasificación de jugadores presentes en secuencias de video, de juegos de fútbol, presentó resultados satisfactorios con alto porcentaje de desempeño, lo que permitirá su utilización en etapas posteriores de un sistema de detección automática de fuera de juego.

El desempeño de la distancia de Bhattacharyya como función de verosimilitud para el clasificador Bayesiano, fue evaluado independientemente mediante validación cruzada, utilizando la base de datos construida y aleatorizando la entrada mediante 100 iteraciones de Monte Carlo.

En cada etapa del proceso existieron diferentes alternativas, que fueron evaluadas para seleccionar la que ofreciera un buen desempeño con un costo computacional razonable. La metodología presentada fue diseñada en particular para minimizar el número de parámetros libres asociados.

Para el método de la mezcla de Gausianas, MOG, se observó que aumentar indiscriminadamente el número de Gausianas a ajustar, no necesariamente mejora la salida deseada. En particular, se realizaron experimentos con un número fijo de Gausianas, (MOG en OpenCV) o con un número adaptativo de Gausianas (MOG2), siendo el primer método el que ofreció mejores resultados.

El problema más importante que tiene la utilización de MOG, es que utiliza el movimiento para determinar cual es el cluster de fondo en la mezcla de Gausianas. Por esta razón, cuando un objeto de interés permanece inmóvil durante algunos fotogramas, este pasa a pertenecer al cluster de fondo, incluso cuando el color del objeto sea distintivo.

Teniendo en cuenta que para la aplicación particular desarrollada, la forma más importante para diferenciar jugadores de equipos diferentes, es el color de la camiseta, entonces, se definió como descriptor la distribución de colores en el espacio RGB.

Dado que el video no se encuentra en un ambiente controlado, las escenas presentan cambios súbitos de iluminación y las tomas no son hechas con cámaras estáticas, es posible que algunos de los actores presentes en la escena queden por fuera de foco, lo que ocasiona cambios en su distribución de color dificultando su clasificación.

En algunas de las escenas de los clips se presentan oclusiones de un actor perteneciente a una clase sobre otro de otra clase distinta. Estas oclusiones se ven reflejadas en el histograma 3D

en las ROIs al momento de ser comparado con el histograma de referencia, ocasionando un error de clasificación.

Las técnicas mostradas en este trabajo, no son excluyentes de otras disponibles en el estado del arte de las comunidades en aprendizaje de máquina y visión por computador. Por ejemplo, es necesario profundizar el estudio presentado explorando otras técnicas de segmentación y clasificación.

BIBLIOGRAFíA

[1] Background subtraction, opencv documentation. 3.2.2, 3.2.3

[2] Distancias estadisticas y escalado multidimensional. 3.4.7, 3.4.7

[3] Método de montecarlo. 3.5.2

[4] Sports science, health science, rehabilitation, biomechanics investigation, robotics and engineering, training and simulation. 1

[5] *Detección de Bordes*, 2012.

[6] FÚtbol | la tecnologÍa llega al balompiÉ ¿qué son y cómo funcionan el goalref y el ojo de halcón?, 2012. 1

[7] Ranjan Acharyya. *A New Approach for Blind Source Separation of Convolutive Sources*. 2008. 3.4.2

[8] Rostamizadeh Ameet Talwalkar Afshin. *Foundations of Machine Learning*. The MIT Press, 2012. 3.4

[9] M Alcañiz Raya, V Grau Colomer, M. C. Juan Lizandra, C. Monserrat Aranda, J. M. Navarro Jover, and E. Moltó García. *Procesamiento digital de imagen*. Universidad Politécnica de Valencia, 1999.

[10] N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992. 3.4.6

[11] John Robert Anderson, Ryszard S Michalski, Ryszard Stanisław Michalski, Jaime Guillermo Carbonell, and Tom Michael Mitchell. *Machine learning: An artificial intelligence approach*, volume 1. Morgan Kaufmann, 1986. 3.4, 3.4.1, 3.4.3

[12] W Bareld and C Hendrix. The eect of update rate on the sense of presence within virtual environments, virtual reality. *The Journal of the Virtual Reality Society*, pages pp. 3–16, 1995. 3.5.3

[13] Francisco Belda Maruenda. Can the human eye detect an offside position during a football match? *BMJ*, 329(7480):1470–1472, 12 2004. 1.1

[14] O Bimber and R. Raskar. Spatial augmented reality: Merging real and virtual worlds. *A K Peters*, 2005. 3.5.3

[15] International Football Association Board. *Reglas de Juego 2012 / 2013.* FIFA, FIFA-Strasse 20. Zúrich. Suiza, Junio 2012. 1.1

[16] Juan Botella Ausina. La atención. In *La percepción visual*, pages 499–531. Madrid: Biblioteca Nueva, 2 edition, 2008. 1.1

[17] Oscar Boullosa G. Estudio comparativo de descriptores visuales para la detección de escenas cuasi-duplicadas. Master's thesis, Universidad Autnoma de Madrid, 2011. 3.3.1

[18] José Luis Bueno Crespo, Andrés; Sancho Gómez. Aprendizaje multitarea en problemas con un número reducido de datos. *Universidad Politécnica de Valencia*, 2005. 3.4.5

[19] Carlos Cerrada. Introducción a la visión por computador. In *Avances en robótica y visión por computador*, pages 29–60. Ediciones de la Universidad de Castilla-La Mancha, 1 edition, 2002.

[20] Emilio Chuvieco Salinero. *Fundamentos de teledetección espacial.* RIALP S.A., Alcalá, Madrid, 2 edition, Julio 1995. 3.4.1, 3.4.2

[21] Dorin Ramesh Visvanatan Meer Peter Comaniciu. Real-time tracking of non-rigid objects using mean shift. *Imaging & Visualization Department Siemens Corporate Research 755 College Road East, Princeton, NJ 08540 Electrical & Computer Engineering Department Rutgers University 94 Brett Road, Piscataway, NJ 08855*, 2000. 3.4.7

[22] D. Coomans and D.L. Massart. Alternative k-nearest neighbour rules in supervised pattern recognition : Part 1. k-nearest neighbour classification by using alternative voting rules. *Analytica Chimica Acta*, 136(0):15 – 27, 1982. 3.4.6

[23] Lola Curras Martinez, Manuel Traba Martinez. Detección de bordes. 2012. 3.1.1, 3.1.1, 3.1.1

[24] Brian S Everitt, Sabine Landau, Morven Leese, and Daniel Stahl. Miscellaneous clustering methods. *Cluster Analysis, 5th Edition*, pages 215–255. 3.4.6

[25] Alvan R. Feinstein et al. On the sensitivity, specificity and discrimination of diagnostic tests. In *Clinical biostatistics*, chapter XXXI. C V. Mosby Company, 11830 Westline Industrial Drive, St. Louis, Missouri 63141, USA, 1977. 3.5.3, 3.5.3, 3.5.3

[26] Martin Fernandez. Técnicas clásicas de segmentación de imágenes. *Harvard*, 2004. 3.1.1, 3.1.1, 3.1.1, 3.1.1, 3.1.1, 3.1.2

[27] Ricardo Horacio Fescina and Rubén Belitzky. Evaluación de los procedimientos diagnósticos: Aspectos metodológicos. In *Tecnologías perinatales*, pages 69–90. Centro Latinoamericano de Perinatología y Desarrollo Humano, 1988. 3.5.3, 3.5.3, 3.5.3

[28] FIFA. Informe de finanzas 2012. Technical Report 63, Fédération Internationale de Football Association, Isla Mauricio, Mayo 2012. 1.2

[29] José Luis Gil Rodríguez, Edel B. García Reyes, Damasco R. Ponvert Delisles, Reinaldo Sanchez Alvarez, and Marina B. Vega Carreño. Enfoques para la clasificación digital de imágenes mono y multiespectrales y su implementación en el software cubano tn estudio v2.0. *AET*, (20):35–52, Diciembre 2003. 3.4.1

[30] María Teresa Gómez López. *Limitaciones cognitivas del árbitro asistente en la aplicación del"fuera de juego": efecto del feedback.* PhD thesis, Ciencias, Madrid, 2004. 1.1

[31] R. C. Gonzalez and R. E. Woods. *Digital Image Processing.* Prentice Hall, 2 edition, 2002.

[32] Rafael Gonzalez and Richard Woods. *Tratamiento Digital de Imágenes.* Addison-Wesley Iberoamericana, 1996.

[33] R. M. Haralick and L. G. Shapiro. *Computer and robot vision*, volume 1. Addison-Wesley, New York, 1991.

[34] Stevan Harnad. The annotation game: On turing (1950) on computing, machinery, and intelligence. In Robert Epstein and Grace Peters, editors, *Philosophical and Methodological Issues in the Quest for the Thinking Computer.* Kluwer, 2006. Address: Amsterdam. 3.4

[35] Robert Tibshirani Friedman Jerome Hastie, Trevor. *The Elements of Statistical Learning: Data mining,Inference,and Prediction.* New York: Springer, 2009. 3.4.2

[36] Annie Hill. Information design goes to the game, Enero 2013. (document), 3

[37] J. Hossein-Khani, H. Soltanian-Zadeh, M. Kamarei, and O. Staadt. Ball detection with the aim of corner event detection in soccer video. In *Parallel and Distributed Processing with Applications Workshops (ISPAW), 2011 Ninth IEEE International Symposium on*, pages 147–152, 2011. 2

[38] T.S. Huang, J. Llach, and Chao Zhang. A method of small object detection and tracking based on particle filters. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4, 2008. 2

[39] Andy Ingham. Statistical data design/ 3d modelling, Mayo 2011. (document), 1, 2

[40] Anil K Jain, Robert P. W. Duin, and Jianchang Mao. Statistical pattern recognition: A review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):4–37, 2000. 3.5.1

[41] Christopher M. Jordan, Michael I.; Bishop. *Neural Networks.* Computer Science Handbook, 2004. 3.4.2

[42] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. *Department of Systems Engineering, Brunel University, Middlesex, UB8 3PH, UK*, 2001. 3.2.2, 3.2.2

[43] Pakorn KaewTraKulPong and Richard Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Video-Based Surveillance Systems*, pages 135–144. Springer, 2002.

[44] S. Kotsiantis. Aprendizaje automático: Una revisión de la clasificación de técnicas de informatica. *Diario 31*, 2007.

[45] Iñaki Moujahid Abdelmalik Larraña, Pedro Inza. Clasificadores bayesianos.

[46] Cornelius T. Leondes. *Neural Network Systems Techniques and Applications*, volume 7. Academic Press, San Diego, California, USA, 4 edition, 1998. 3.4.6

[47] S.X. Liu, Lijun Jiang, J. Garner, and S. Vermette. Video based soccer ball tracking. In *Image Analysis Interpretation (SSIAI), 2010 IEEE Southwest Symposium on*, pages 53–56, 2010. 2

[48] D.J.C Mackay. Introduction to monte carlo methods. *Department of Physics, Cambridge University Cavendish Laboratory, Madingley Road*, 2013.

[49] P.L. Mazzeo, P. Spagnolo, M. Leo, and T. D'Orazio. Visual players detection and tracking in soccer matches. In *Advanced Video and Signal Based Surveillance, 2008. AVSS '08. IEEE Fifth International Conference on*, pages 326–333, 2008. 2

[50] Collin McCollough. Big blue breakdown: The comprehensive third down edition, Noviembre 2010. (document), 3

[51] Ramakant Nevatia. *Machine perception.* Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

[52] Harald Niederreiter. *Random number generation and quasi-Monte Carlo methods*, volume 63. SIAM, 1992. 3.5.2

[53] J.R. Nunez, J. Facon, and A. de Souza Brito. Soccer video segmentation: Referee and player detection. In *Systems, Signals and Image Processing, 2008. IWSSIP 2008. 15th International Conference on*, pages 279–282, 2008. 2

[54] Eddie Ángel Ortega Pérez. Multiclasificación discriminativa por partes mediante cóedigos correctores de erroreso. Master's thesis, Universdiad de Barcelona, Barcelona, España, Junio 2012. 3.4.6

[55] O?r Pele and Michael Werman. The quadratic-chi histogram distance family. *European Conference on Computer Vision*, 2010. 3.4.7

[56] Massimo Piccardi. Background subtraction techniques: a review. In *International Conference on Systems, Man and Cybernetics*, 2004. 3.2

[57] J Pino Ortega and J. Cimarro Urbano. *Análisis tactico del fuera de juego en el fútbol.* Editorial Wanceulen, S.L., 1998. 1.1

[58] Cecilia Sanz. *Razonamiento Evidencial Dinámico Un método de clasificación aplicado al analisis de imágenes.* PhD thesis, Universidad Nacional de La Plata Facultad de Ciencias Exactas, 2002.

[59] K. Sato and J.K. Aggarwal. Tracking soccer players using broadcast tv images. In *Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference on*, pages 546–551, 2005. 2

[60] Eddie Angel Sobrado Malpartida. Sistema de visión artificial para el reconocimiento y manupulación de objetos utilizando un brazo robot. Master's thesis, Pontificia Universidad Católica del Peru, Lima, Peru, 2003. 3.4.6

[61] Chris Stauffer and W Eric L Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE, 1999.

[62] W.E.L Stauffer, Chris Grimson. Adaptive background mixture models for real-time tracking. *The Artificial Intelligence Laboratory Massachusetts Institute of Technology, Cambrige*, 1999. 3.2.2, 4.3.1

[63] S. Suzuki and K. Abe. Topological structural analysis of digitized binary images by border following. *CVGIP 30 - 1*, pages pp 32–46, 1985. 4.3.2

[64] Birgi Tamersoy. Background substration. Technical report, University of Texas, 2009. 3.2

[65] F. Tresaco García. *Manual para árbitros asistentes.* Escuela nacional de árbitros de futbol, 2003. 1.1

[66] Zhifei Xu and Pengfei Shi. Segmentation of players and team discrimination in soccer videos. In *VLSI Design and Video Technology, 2005. Proceedings of 2005 IEEE International Workshop on*, pages 369–372, 2005. 2

[67] C. Tomasi Y. Rubner and L. J. Guibas. Ieee international conference on computer vision. In *A metric for distributions with applications to image databases*, pages 59–66, 1998. 3.4.7

[68] Jacob Yerushalmy. Statistical problems in assessing methods of medical diagnosis, with special reference to x-ray techniques. *Public Health Reports (1896-1970)*, pages 1432–1449, 1947. 3.5.3, 3.5.3, 3.5.3

[69] Zoran Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 28–31. IEEE, 2004. 3.2.3