



UNIVERSIDAD CATÓLICA
de Colombia

**MÉTODO DE DETECCIÓN AUTOMÁTICA DE ARMAS DE MANO EN VIDEO
USANDO APRENDIZAJE PROFUNDO**

**BRAYAN ALEJANDRO CRIOLLO LEAL - 625619
NICOLÁS DÍAZ RONDÓN - 625631**

**UNIVERSIDAD CATÓLICA DE COLOMBIA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
MODALIDAD TRABAJO DE INVESTIGACIÓN TECNOLÓGICA
BOGOTÁ D.C.
2019**

MÉTODO DE DETECCIÓN AUTOMÁTICA DE ARMAS DE MANO EN VIDEO
USANDO APRENDIZAJE PROFUNDO

BRAYAN ALEJANDRO CRIOLLO LEAL - 625619
NICOLÁS DÍAZ RONDÓN - 625631

ESTE TRABAJO DE GRADO ES PRESENTADO COMO REQUISITO PARA
OPTAR AL TÍTULO DE: INGENIERO DE SISTEMAS

ASESOR:
ROGER ENRIQUE GUZMÁN AVENDAÑO
MSC. INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

ALTERNATIVA:
TRABAJO DE INVESTIGACIÓN TECNOLÓGICA
GRUPO DE INVESTIGACIÓN:
GISIC
SEMILLERO DE INVESTIGACIÓN:
MAILAB

UNIVERSIDAD CATÓLICA DE COLOMBIA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
MODALIDAD TRABAJO DE INVESTIGACIÓN TECNOLÓGICA
BOGOTÁ D.C.
2019



Atribución-NoComercial 2.5 Colombia (CC BY-NC 2.5)

La presente obra está bajo una licencia:
Atribución-NoComercial 2.5 Colombia (CC BY-NC 2.5)

Para leer el texto completo de la licencia, visita:
<http://creativecommons.org/licenses/by-nc/2.5/co/>

Usted es libre de:



Compartir - copiar, distribuir, ejecutar y comunicar públicamente la obra
hacer obras derivadas

Bajo las condiciones siguientes:



Atribución — Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciante (pero no de una manera que sugiera que tiene su apoyo o que apoyan el uso que hace de su obra).



No Comercial — No puede utilizar esta obra para fines comerciales.

Nota de Aceptación

Aprobado por el comité de grado en cumplimiento de los requisitos exigidos por la Facultad de Ingeniería y la Universidad Católica de Colombia para optar al título de ingeniero de sistemas.

Alexandra María López Sevillano
Jurado 1

Sandra Milena Bernate Bautista
Jurado 2

Roger Enrique Guzmán Avendaño, Msc.
Asesor

BOGOTÁ D.C. DICIEMBRE 2 DE 2019.

DEDICATORIA

A Dios, por estar siempre en cada momento de nuestras vidas guiando nuestros pasos y brindándonos diariamente lo necesario para alcanzar nuestros objetivos.

AGRADECIMIENTOS

A nuestros padres: Nubia Leal Camelo y Wilson Criollo Ruiz; y, Victoria Rondón Carvajal y Farid Díaz Torres por ser los principales promotores de nuestros sueños, por cada día confiar y brindándonos su apoyo, por creer en nosotros y en nuestras expectativas.

A nuestro tutor, Roger Guzmán por guiarnos a lo largo del proyecto aportando valiosos consejos para lograr tan anhelado objetivo, además de brindarnos enseñanzas a nivel profesional y personal.

A nuestros compañeros: Juan Montañez y David Rubiano por su amabilidad y apoyo que permitieron que este proyecto saliera adelante.

TABLA DE CONTENIDO

RESUMEN.....	11
INTRODUCCIÓN.....	13
1. GENERALIDADES.....	15
1.1. LÍNEA DE INVESTIGACIÓN.....	15
1.2. PLANTEAMIENTO DEL PROBLEMA.....	16
1.2.1. Descripción del problema.....	16
1.2.2. Formulación del Problema.....	19
1.3. OBJETIVOS.....	20
1.3.1. Objetivo General.....	20
1.3.2. Objetivos Específicos.....	20
1.4. JUSTIFICACIÓN.....	21
1.5. DELIMITACIÓN.....	23
1.5.1. Limitaciones.....	23
1.5.2. Alcances.....	23
2. MARCO DE REFERENCIA.....	24
2.1. Marco Teórico.....	24
2.1.1. Aprendizaje de maquina.....	24
2.1.2. Aprendizaje profundo.....	24
2.1.3. Aprendizaje supervisado.....	24
2.1.4. Inteligencia artificial.....	25
2.1.5. Faster-RCNN.....	25
2.2. Marco Conceptual.....	27
2.2.1. Armas de cañón corto.....	27
2.2.2. Conjunto de datos.....	28
2.2.3. Preprocesamiento.....	29
2.2.4. Extracción de Características.....	34
2.2.5. Reconocimiento de objetos.....	42
2.2.6. Medidas de desempeño.....	44
2.3. Estado del arte.....	50
3. METODOLOGÍA.....	58
3.1. Fases del Proyecto.....	58
4. DISEÑO METODOLÓGICO.....	61
4.1. INSTALACIONES Y EQUIPO REQUERIDO.....	75
4.2. ESTRATEGIAS DE COMUNICACIÓN Y DIVULGACIÓN.....	76
5. DISCUSIÓN DE RESULTADOS.....	77
6. CONCLUSIONES.....	80
7. RECOMENDACIONES.....	81
8. ANEXOS.....	82
9. BIBLIOGRAFÍA.....	85

ÍNDICE DE FIGURAS

Figura 1: Aprendizaje Profundo Vs Aprendizaje de máquina.....	21
Figura 2: Red Neuronal.....	26
Figura 3: Armas de cañón corto.....	27
Figura 4: Conjunto de datos armas de cañón corto.	28
Figura 5: Ejemplo redimensionamiento.....	30
Figura 6: Categorías de las técnicas de muestreo.....	31
Figura 7: Muestreo Aleatorio Simple.....	33
Figura 8: Red Neuronal.....	35
Figura 9: Filtros de capa de convolución.	37
Figura 10: Operación ReLU.	38
Figura 11: Ejemplo Max Pooling.	39
Figura 12: Salida RPN	41
Figura 13: Reconocimiento de Objetos.....	42
Figura 14: Agrupación regiones de interés.	42
Figura 15: Detección frente a área prevista del objeto.....	44
Figura 16: Matriz de confusión.....	45
Figura 17: Ecuación de la intersección sobre la unión.....	49
Figura 18: Imagen de rayos X con reconocimiento de pistola.	50
Figura 19: Reconocimiento contorno de pistola.....	51
Figura 20: Conjunto de datos COCO.....	52
Figura 21: Detector de puntos de interés (Harris).....	53
Figura 22: SSD + Mobilenet.....	54
Figura 23: FRCNN + Resnet 100.....	54
Figura 24: RFCN + Resnet 10.	55
Figura 25: Ejemplo de falsos negativos, con las armas del fondo.	55
Figura 26: Pruebas de detección de objetos.....	57
Figura 27: Metodología del proyecto.....	58
Figura 28: Conjunto de videos.	61
Figura 29: Frames de revolver y pistola.....	62
Figura 30: Frames de 1 segundo.....	63
Figura 31: Redimensionamiento de frames.	64
Figura 32: Frames etiquetados.	64
Figura 33: XML de la imagen 500000.jpg	65
Figura 34: Muestreo del conjunto de datos.....	66
Figura 37: Regiones propuestas.	69
Figura 38: Función de pérdida total.	70
Figura 39: Detección de armas de mano 1.....	71
Figura 40: Detección de armas de mano 2.....	72
Figura 35: Descarga de controladores NVIDIA.....	82
Figura 36: Instalación de CUDA Versión 9.0	83

ÍNDICE DE TABLAS

Tabla 1: Cantidad de frames.....	62
Tabla 2: Cantidad de frames a utilizar.	62
Tabla 3: Muestreo en número de frames.	67
Tabla 4: Contenido del archivo csv.	67
Tabla 7: Precision en el muestreo 70-30.	73
Tabla 8: Recall en el muestreo 70-30.	73
Tabla 9: Precision en el muestreo 75-25.	74
Tabla 10: Recall en el muestreo 75-25.	74
Tabla 11: Precision en el muestreo 80-20.	74
Tabla 12: Recall en el muestreo 80-20.	75
Tabla 13: Mejores resultados según el área del arma de mano.	78
Tabla 5: Versiones de Controladores y CUDA.....	82
Tabla 6: Versiones compatibles para Tensorflow-gpu.	83

ANEXOS

Anexo A: Manual de Configuración GPU.....	83
Anexo B: Conjunto de Datos.....	85
Anexo C: Repositorio de los Códigos del Proyecto.....	85

RESUMEN

Este documento presenta un experimento de un método basado en aprendizaje profundo para el apoyo del proceso de monitoreo y detección de armas de mano (revólveres y pistolas), de forma automática con el fin de proponer una solución a la problemática de seguridad existente en la ciudad de Bogotá, donde según estadísticas, la mayoría de los delitos en la ciudad se está dando con algún tipo de arma. Por parte de las autoridades se han propuesto soluciones tecnológicas haciendo uso de las cámaras instaladas en la ciudad, pero no atienden al proceso de detección de armas.

El método utiliza diferentes algoritmos que permiten identificar objetos existentes en un video, compuesto por frames (imágenes), posteriormente pasa a la etapa de clasificación y detección de armas de mano.

La finalidad del presente experimento es implementar un modelo de detección que optimice las labores de monitoreo, específicamente en la detección de armas, proporcionando un método automatizado, apoyado en el uso de técnicas de aprendizaje profundo.

Se propone una metodología que tiene siete (7) pasos importantes, los cuales son: construcción del conjunto de datos, pre procesamiento de los videos, extracción de características de los frames, muestreo del conjunto de datos, red de regiones propuestas, clasificación y detección de armas de mano, y rendimiento del modelo de detección de armas de mano en video. Al llegar a la etapa final, se obtiene un método de detección automática de armas de mano, con valores superiores al sesenta por ciento (60%), en las métricas de desempeño (mAP COCO).

Palabras clave: Inteligencia artificial, aprendizaje profundo, armas de mano, detección de objetos, Faster-RCNN, Visión Artificial.

ABSTRACT

This document presents an experiment of a method based on deep learning to support the process of monitoring and detection of handguns (revolvers and pistols), automatically in order to propose a solution to the security problem existing in the city of Bogotá where according to statistics, most of the crimes in the city are being given with some kind of weapon. On the part of the authorities, technological solutions have been proposed using the cameras installed in the city, but no weapons detection process is implemented.

The method uses different algorithms that allow the identification of objects in a video, composed of frames (images), then enter to the stage of classification and detection of handguns.

The purpose of this experiment is to implement a detection model that optimizes monitoring tasks, specifically in the detection of handguns, providing an automated method, supported by the use of deep learning techniques.

A methodology is proposed that has seven (7) important steps, which are: data set construction, videos preprocessing, feature extraction of the frames, dataset sampling, region proposal network, classification and detection of handguns, and performance of the handgun detection model in video. At the final stage, an automatic handgun detection method is obtained, with values greater than sixty percent (60%), in the performance metrics (mAP COCO).

Keywords: Artificial intelligence, deep learning, hand weapons, object detection, Faster-RCNN, Computer vision.

INTRODUCCIÓN

Durante los últimos años la ciudad de Bogotá presenta reportes deficientes en materia de seguridad, estos han dejado un panorama inquietante para los ciudadanos y las autoridades.¹ Actualmente la policía nacional hace uso de las más de 4.000 cámaras instaladas en la ciudad, que se encuentran conectadas a un centro de monitoreo, desde el cual se realiza el seguimiento a cualquier tipo de suceso que altere el orden público, pero esta tarea es llevada a cabo por agentes de la policía, generando limitaciones al ser llevado a cabo por seres humanos.²

Este proyecto de investigación tecnológica propone un experimento para la detección de armas de mano (pistolas y revólveres), utilizando aprendizaje profundo con el fin de optimizar el monitoreo y detección de armas de mano ya que en los últimos años el aprendizaje profundo ha tenido un gran desempeño en estos ámbitos y ha ganado mucha trascendencia por generar conocimiento sin necesidad de una supervisión humana.³

A nivel de impacto, el proyecto busca generar un avance tecnológico en el área del monitoreo de Bogotá utilizando métodos más rápidos que a futuro pueden llegar a utilizarse en tiempo real. En la parte social se puede lograr que la gente sienta mayor seguridad al salir de sus hogares y que la policía pueda actuar lo antes posible en situaciones de delitos donde se identifique algún tipo de armamento.

Dentro del contenido de este documento se presenta en primera instancia el planteamiento del problema, que se enfoca en la ineficiencia en el monitoreo en la ciudad de Bogotá. A partir del problema se generan unos objetivos que buscan la realización del proyecto junto con el alcance y limitaciones. Además, se propone una estrategia metodológica compuesta por siete fases como se muestra en la figura 13, esta metodología se va a realizar en un periodo de tiempo separado por entregables que se plasman en el cronograma, y en la tabla de entregables.

¹ Murillo Mojica. Oscar. Seguridad, el tema del fogueo entre Policía y ciudadanía. [en línea]. En: *El tiempo* [Citado el 29 de junio, 2019] Disponible en internet: <<https://www.eltiempo.com/bogota/como-esta-la-seguridad-en-bogota-segun-los-ciudadanos-y-la-policia-348276>>

² Secretaría Distrital de Seguridad, Convivencia y Justicia, 2018. Bogotá ya Cuenta con más de 1.600 Cámaras De Vigilancia. [en línea]. En: *Secretaría Distrital de Seguridad, Convivencia y Justicia*. [Citado el 20 de noviembre, 2019]. Disponible en internet: <<https://scj.gov.co/es/noticias/bogot%C3%A1-cuenta-m%C3%A1s-1600-c%C3%A1maras-vigilancia>>

³ Zhao, Zhong-Qiu & Zheng, Peng & Xu, Shou-Tao & Wu, Xindong, 2019. Object Detection with Deep Learning: A Review. *IEEE. Transactions on Neural Networks and Learning Systems*. [Citado el 16 marzo, 2019]; pp. 1-21.

También se evidencia el equipo requerido y el presupuesto que se necesita para el proyecto.

1. GENERALIDADES

1.1. LÍNEA DE INVESTIGACIÓN

El proyecto pertenece al grupo de investigación en Software Inteligente y Convergencia Tecnológica – GISIC, semillero MAILAB. Debido a que esta línea de investigación aborda proyectos en el campo de inteligencia artificial, que compone diferentes ramas como: aprendizaje de máquina, que busca simular procesos inteligentes por medio de programas, pero estos algoritmos no son óptimos para desempeñar cierto tipo de tareas y por esto se aborda una subcategoría llamada aprendizaje profundo que utiliza métodos basados en el comportamiento del cerebro humano, específicamente las neuronas y sus interconexiones, el experimento que se realiza en este proyecto utiliza algoritmos basados en técnicas de este tipo de aprendizaje por el alto desempeño que han logrado en los últimos años en tareas de detección y reconocimiento de objetos.

1.2. PLANTEAMIENTO DEL PROBLEMA

1.2.1. Descripción del problema

A lo largo de la historia se han presentado altos niveles de delincuencia en Bogotá, este tema tiene preocupados a los capitalinos por la cantidad de crímenes recientes en los que incluso hay víctimas mortales.⁴ Hoy en día el delito más frecuente es el hurto a personas, con un aumento en el primer semestre del 2019 del 17% en comparación con las cifras del primer semestre del 2018.⁵ En la mayoría de delitos se ha incrementado el uso de armas de fuego, que en algunos casos ha llegado a terminar con la vida de las víctimas.⁶ De acuerdo a los informes de la policía metropolitana de Bogotá las armas más confiscadas son las armas de mano (revólveres y pistolas), ya que presentan una facilidad para portarlas y se pueden accionar con una sola mano, hasta mayo del 2019, se evidencio que el mayor número de incautaciones corresponde a este tipo de armas con revólveres (360) y pistolas (149).⁷

Los bajos índices de seguridad en la ciudad de Bogotá, han hecho necesario la puesta en marcha de planes de choque y el uso herramientas que ayuden a mitigar las falencias por parte de las autoridades encargadas de la seguridad en la ciudad. En 2018 la policía nacional y la Alcaldía de Bogotá ejecutaron el plan Metrópoli donde se instalaron cámaras de vigilancia en puntos estratégicos, las cuales están conectadas a un centro de inteligencia de la policía, permitiendo el monitoreo y la toma de evidencia en caso que se presente cualquier suceso que altere el orden.⁸

⁴ El Espectador. Preocupa incremento de hurtos violentos en Bogotá. [en línea]. En: *El Espectador*. [Citado el 04 de julio, 2019]. Disponible en internet: <<https://www.elespectador.com/noticias/bogota/preocupa-incremento-de-hurtos-violentos-en-bogota-articulo-846973>>

⁵ Velásquez Loaiza, Melissa. Bogotá, ¿cada vez más insegura?: estas son las cifras de los crímenes en la capital colombiana. [en línea]. En: *CNN*. [Citado el 14 de marzo, 2019] Disponible en internet: <<https://cnnespanol.cnn.com/2018/02/06/bogota-cada-vez-mas-insegura-estas-son-las-cifras-de-los-crimenes-en-la-capital-colombiana/>>

⁶ El tiempo. En Bogotá sube uso de armas de fuego en hurtos y baja en homicidios. [en línea]. En: *El tiempo*. [Citado el 14 de marzo, 2019]. Disponible en internet: <<https://www.eltiempo.com/bogota/entre-enero-y-febrero-subio-el-uso-de-armas-de-fuego-en-atracos-en-bogota-208274>>

⁷ El Espectador. Advierten uso de armas largas para perpetrar delitos en Bogotá. [en línea]. En: *El Espectador*. [Citado el 14 de marzo, 2019]. Disponible en internet: <<https://www.elespectador.com/noticias/bogota/advierten-creciente-uso-de-armas-largas-para-perpetrar-delitos-en-bogota-articulo-869118>>

⁸ El tiempo. ¿Qué tanto impactan en la seguridad las cámaras de vigilancia? [en línea]. En: *El tiempo*. [Citado el 14 de marzo, 2019]. Disponible en internet: <<https://www.eltiempo.com/bogota/impacto-de-las-cameras-de-vigilancia-en-la-seguridad-de-bogota-170988>>

Este monitoreo se apoya en el análisis de video enfocado en los procesos de judicialización e identificación de delincuentes. Esta herramienta tecnológica depende de agentes de policía designados para esta labor, por tal motivo se presenta una baja eficiencia ocasionada por la pérdida de concentración por parte del personal, ya que en este tipo de tareas durante turnos de máximo 4 horas se mantiene la concentración. Cuando el tiempo que dedica el personal es mayor, se producen errores de apreciación y atención, debido al cansancio, la subjetividad de cada persona y factores ambientales que influyen en la detección de objetos como armas de mano, según estudios de la empresa SIE (Seguridad Integral Empresarial).⁹ Con base a los estudios presentados, se evidencia que la ejecución de acciones posteriores a los actos delictivos con presencia de armas de mano son de tipo reactivo y toman cierta cantidad de tiempo en ser ejecutadas, a diferencia, el uso de procesos en tiempo real proporcionan una detección inmediata.¹⁰

En los últimos años, desde el campo de inteligencia artificial se han propuesto nuevas formas de abarcar las tareas de detección de objetos, haciendo uso de software especializado para poder representar, manipular y transformar las señales desde un computador. Inicialmente se llegaron a plantear soluciones desde el área de aprendizaje de máquina utilizando algoritmos para la detección o clasificación de objetos que aporten al reconocimiento de armas, pero estos no han tenido una efectividad muy alta, debido a que los algoritmos presentaban un bajo rendimiento y las métricas de desempeño presentaban valores muy bajos. Al obtener estos resultados, este tema del reconocimiento de objetos se trasladó a las técnicas de aprendizaje profundo donde han existido mejores resultados en las pruebas y donde en los últimos años estos algoritmos han presentado un alto rendimiento.¹¹

La implementación de un algoritmo de aprendizaje profundo para la detección de armas de mano en video depende de la visualización del

⁹ SIE. El importante rol del sistema de CCTV en la Seguridad Privada. [en línea]. En: *SIE*. [Citado el 14 de marzo, 2019]. Disponible en internet: <<https://siesa.com.ar/el-importante-rol-del-sistema-de-cctv-en-la-seguridad-privada/>>

¹⁰ Rodríguez Gómez, José David. Refuerzan videovigilancia en Bogotá con nuevos centros de monitoreo. [en línea]. [Citado el 30 de marzo, 2019] En: *RCN radio*. Disponible en internet: <https://www.rcnradio.com/bogota/refuerzan-videovigilancia-en-bogota-con-nuevos-centros-de-monitoreo>

¹¹ Kumar Tiwari, Rohit. K. Verma, Gyanendra, 2015. A Computer Vision based Framework for Visual Gun Detection using Harris Interest Point Detector. *ScienceDirect*. Procedia Computer Science. Vol.54, (06, 2015); pp. 703-712.

arma, para lograr la detección de un revolver o pistola es necesario que en la captura del video se encuentren visibles en su totalidad preferiblemente. además, se pueden presentar armas de fuego falsas, pero con las mismas características morfológicas que se pueden detectar como reales.

1.2.2. Formulación del Problema

De acuerdo al problema planteado anteriormente, surge la pregunta de investigación:

¿Cómo se puede apoyar el proceso de monitoreo y detección de armas de mano, por medio de la implementación de un algoritmo automático de reconocimiento, a través de video usando aprendizaje profundo?

1.3. OBJETIVOS

1.3.1. Objetivo General

Implementar un método, utilizando algoritmos de aprendizaje profundo, para la detección automática de armas de mano en videos.

1.3.2. Objetivos Específicos

- Construir un conjunto de datos de videos que incluyan revólveres y pistolas, recopilando videos de plataformas en línea (YouTube), para modelar el algoritmo de detección automática de armas de mano.
- Diseñar una estrategia metodológica, tomando como base los procesos que se encuentran en la literatura, para la detección automática de armas de mano en videos utilizando aprendizaje profundo.
- Implementar un algoritmo de redes convolucionales (Faster-RCNN), tomando un tipo de arquitectura existente y adaptándola al modelo del experimento, para la detección automática de armas de mano en videos.
- Evaluar el rendimiento del algoritmo de detección, por medio de un conjunto de métricas, para verificar su desempeño.

1.4. JUSTIFICACIÓN

La detección de objetos tiene aplicaciones en muchas áreas, una de las principales es el monitoreo en cámaras de vigilancia. Actualmente se comercializan sistemas de vigilancia que se apoyan en el uso de la visión artificial, esto mejora evidentemente la eficiencia de estos sistemas, debido a que tiempo atrás se asignaba todo el proceso de detección a los operadores del sistema de cámaras.

El procesamiento, análisis y compresión de las imágenes que recolecta un sistema de visión artificial requieren algoritmos que cuenten con la capacidad para realizar ese tipo de tareas, en el campo de inteligencia artificial se manejan este tipo de algoritmos. Inicialmente se trabajaron con el uso de modelos basados en aprendizaje de máquina, pero al medir su rendimiento con las diferentes métricas de desempeño existentes se evidenciaban niveles muy bajos a la hora de detectar objetos, además, estos algoritmos tenían un alto costo computacional que dificultaba llevar a la detección en tiempo real.

Las diversas investigaciones y la aparición de nuevas ramas del aprendizaje de máquina, marcaron la tendencia del aprendizaje profundo para este tipo de tareas, ya que presentaban los valores más altos en rendimiento y eficiencia. Las técnicas que se implementan son referentes al campo del aprendizaje no supervisado, estas permiten una mayor abstracción de características de los diferentes objetos que se logran identificar frame y mejoran el rendimiento del algoritmo.¹²

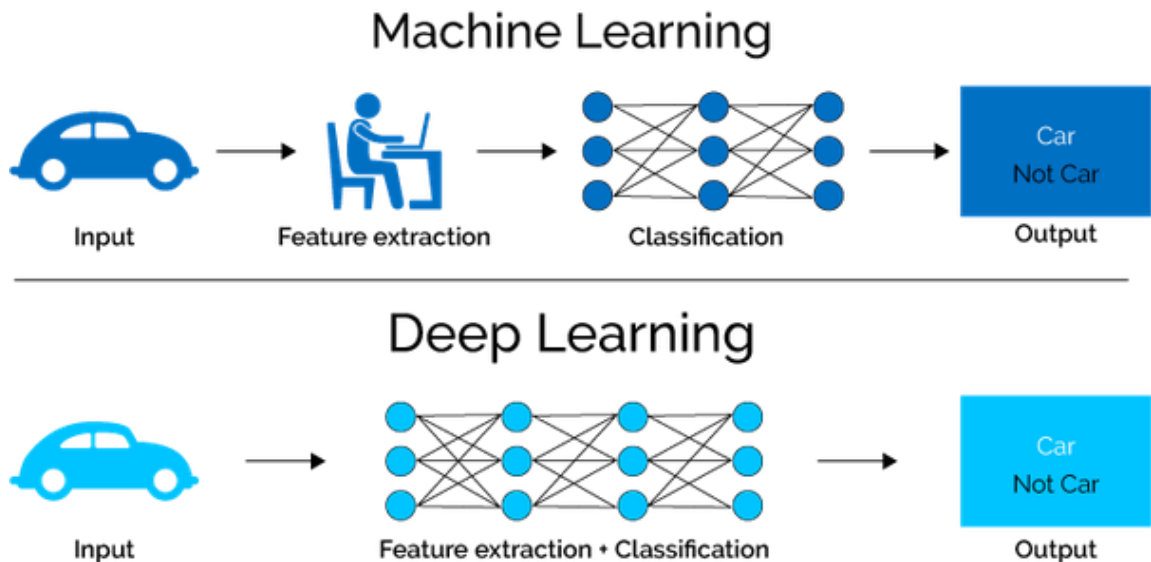


Figura 1: Aprendizaje Profundo Vs Aprendizaje de máquina.

¹²MJ, Bahmani [april, 2019]. Disponible en internet: <<https://medium.com/datadriveninvestor/ai-vs-machine-learning-vs-deep-learning-ba3b3c58c32>>

Fuente: Quora. What is the difference between deep learning and usual machine learning? [en línea]. En: Quora, C. [Citado el 23 abril, 2019]. Disponible en: <https://www.quora.com/What-is-the-difference-between-deep-learning-and-usual-machine-learning>.

La realización de este proyecto representa un logro significativo, puesto que muestra el desarrollo de una propuesta innovadora, dando una solución a una problemática real, donde se aplican los conocimientos adquiridos a lo largo de la carrera universitaria. Para la Universidad Católica de Colombia es una evidencia de los procesos de investigación, dejando un precedente para procesos futuros que quieran continuar con la propuesta.

Los beneficios a nivel macro inicialmente serían para las empresas de seguridad, proporcionando la automatización de procesos, optimización de tiempos y un aumento en la eficiencia de los procesos de detección. Consecuencia del uso de este tipo de sistemas trae un beneficio para la sociedad, alertando tempranamente la aparición de armas de mano y permitiendo una alerta inmediata para las autoridades.

1.5. DELIMITACIÓN

1.5.1. Limitaciones

Para lograr los objetivos del proyecto, se tienen en cuenta los siguientes aspectos: el experimento a realizar solo tiene en cuenta las armas de mano (revólveres y pistolas), el proceso de detección es en video y no en tiempo real, la exactitud de la detección dependerá del ángulo en que se visualice el arma de mano y solo será para distancias cercanas.

1.5.2. Alcances

Se desarrolla un experimento de un método para la detección automática de armas de mano en video siguiendo una metodología que consta de 7 fases: conjunto de datos, pre procesamiento, extracción de características, muestreo, regiones propuestas, clasificación y detección, y rendimiento.

2. MARCO DE REFERENCIA

2.1. Marco Teórico

2.1.1. Aprendizaje de maquina

El aprendizaje de maquina es una rama de la inteligencia artificial que le proporciona a las maquinas la capacidad de aprender y mejorar por medio de un conjunto de experiencias que son datos e información que provienen de interacciones con el mundo real.¹³

Este trabajo busca por medio de la experiencia, en este caso un conjunto de videos de armas de mano conseguir que en un video se detecte pistolas y revólveres usando algoritmos de aprendizaje profundo.

2.1.2. Aprendizaje profundo

El aprendizaje profundo es una subcategoría del aprendizaje de máquina, que permite a las computadoras aprender. Este aprendizaje sigue teorías acerca de cómo funciona el cerebro humano, específicamente las neuronas, buscando fundamentarse en ideas básicas al inicio para después basarse en ellas para abstraer ideas más amplias.¹⁴

En el proyecto gracias al uso de aprendizaje profundo y por medio de una arquitectura de red neuronal (Faster-RCNN), se ingresan todos los frames del dataset a un modelo pre-entrenado, en la etapa de entrenamiento este modelo comienza a aprender de los datos y al ser una red profunda (deep network) son muchas las capas en las que la señal va modificándose y pasando sucesivamente, permitiendo al algoritmo utilizar múltiples capas de procesamiento, compuestas de múltiples transformaciones lineales y no lineales. Con la finalización del entrenamiento se obtiene un modelo para la detección de armas de mano.

2.1.3. Aprendizaje supervisado

Es un conjunto de algoritmos de inteligencia artificial que hace uso de un conjunto de datos, que se conoce como conjunto de entrenamiento, con el cual realiza predicciones. El conjunto de datos de entrenamiento incluye datos de entrada y valores de respuesta (etiquetas). Todo esto con el fin de

¹³ ExpertSystem. What is Machine Learning? A definition. [en línea]. En: ExpertSystem. [Citado el 10 de julio, 2019]. Disponible en internet: <<https://www.expertsystem.com/machine-learning-definition/>>

¹⁴ Colciencias. ¿qué es el aprendizaje profundo? [en línea]. En: Colciencias. [Citado el 21 de marzo, 2019]. Disponible en internet: <<http://www.todoesciencia.gov.co/aprende/innovacion/que-es-el-aprendizaje-profundo>>

obtener un modelo que permita realizar predicciones en conjuntos de datos nuevos.

El aprendizaje profundo contiene dos categorías de algoritmos, los algoritmos de clasificación que dan como respuesta valores categóricos; los algoritmos de regresión que trabajan con predicciones continuas, según los datos de entrada.

2.1.4. Inteligencia artificial.

Es una rama de la informática con bases en la lógica y las ciencias cognitivas. Esta rama estudia y desarrolla sistemas capaces de realizar tareas que normalmente se atribuyen a la inteligencia humana, como por ejemplo traducir un documento o reconocer a una persona por su cara. También sistemas que aprenden a comportarse de manera autónoma, como vehículos sin conductor o robots.¹⁵

La inteligencia artificial se utiliza por medio de algoritmos que permiten que una maquina pueda ser programada para simular comportamientos inteligentes, en este caso, la detección automática de armas de cañón corto.

2.1.5. Faster-RCNN

(Faster-Recurrent Convolutional Neural Network) por sus siglas en inglés, es una arquitectura de red neuronal del campo de aprendizaje profundo por su tipo de estructura, presenta unos tiempos de procesamiento muy eficientes apoyándose en el uso de la red de regiones propuestas (Region Proposal Network), reduciendo el costo computacional del proceso de detección, permite obtener sólo aquellas regiones donde se tiene la mayor tasa de probabilidad de la presencia de un objeto, posteriormente la red de detección se enfoca en las regiones denotadas por la RNP y evalúa si en la región se encuentra el objeto, para dibujar un cuadro delimitador y etiquetarlo a la clase que pertenezca.¹⁶

¹⁵ BBVA. ¿Qué es la inteligencia artificial? [en línea]. En: BBVA. [Citado el 21 de marzo, 2019]. Disponible en internet: <<https://www.bbva.com/es/que-es-la-inteligencia-artificial-2/>>

¹⁶ Shaoqing, Ren. Kaiming, He. Ross, Girshick. Jian, Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. En: Cornell University (arXiv). Volume 3. 2015. Pages 1-14.

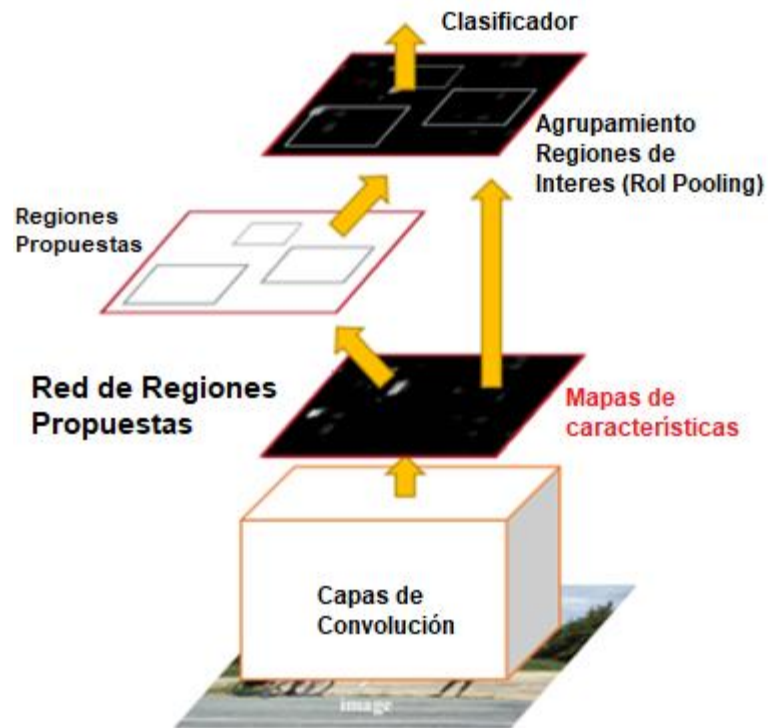


Figura 2: Red Neuronal.

Fuente: Shaoqing, Ren. Kaiming, He. Ross, Girshick. Jian, Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. En: Cornell University (arXiv). Volume 3. 2015. Pages 1-14.

2.2. Marco Conceptual

2.2.1. Armas de cañón corto.

Un arma de longitud de cañón corto es un arma que se puede empuñar con una sola mano, este tipo de armas son las pistolas y revólveres; su dimensión de cañón no puede exceder los 30 cm. La longitud total del arma, no puede exceder de 60 cm. A diferencia de las armas largas, no poseen una culata la cual apoyar sobre el hombro.¹⁷

Este concepto se emplea para la construcción del conjunto de datos, además, el algoritmo se va a enfocar en la detección de este tipo de armas, por lo tanto, se va a abarcar a lo largo del proyecto.



Figura 3: Armas de cañón corto.

Fuente: InfoArmas. PISTOLA Y REVOLVER EN ACERO. En: *InfoArmas* [en línea]. En: *InfoArmas*. [Citado el 11 de julio, 2019]. Disponible en internet: <<https://indumil.wordpress.com/2014/04/22/pistola-beretta-92f-y-revolver-ruger-38l-en-acero/pistola-y-revolver-en-acero/>>

¹⁷ Gobierno de España – Ministerio de interior. Clasificación de armas. [En línea]. En: Gobierno de España – Ministerio de interior. [Citado el 21 de marzo, 2019]. Disponible en internet: <<http://www.interior.gob.es/web/servicios-al-ciudadano/seguridad/armas-y-explosivos/clasificacion-de-armas>>

2.2.2. Conjunto de datos

Es un conjunto de datos recopilados, ya que se correlacionan con los resultados que se buscan llegar a predecir. En general en los campos de aprendizaje profundo y aprendizaje de máquina, se necesitan conjuntos de datos con buen contenido para obtener un funcionamiento correcto y lograr a obtener una tasa de predicción más alta.¹⁸

La construcción del conjunto de datos radica como el punto de referencia sobre el cual se va a llegar a entrenar el modelo, por lo tanto, es una tarea que requiere dedicación ya que los datos se deben alinear correctamente con el problema a solucionar. Para el proyecto propuesto se recopilan videos que contengan armas de mano (revólveres y pistolas) de cualquier tipo, es decir, escenas de películas, entrenamiento con armas, defensa con

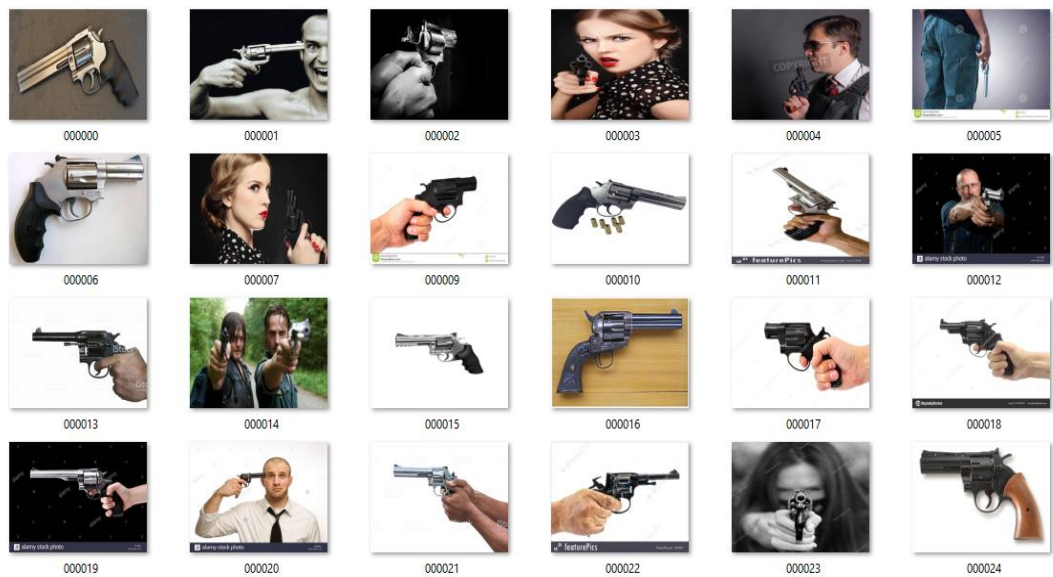


Figura 4: Conjunto de datos armas de cañón corto.

armamento, etc.

Fuente: Los autores.

¹⁸ Skymind. Datasets and Machine Learning. [En línea]. En: *Skymind*. [Citado el 11 de julio, 2019] Disponible en internet: <<https://skymind.ai/wiki/datasets-ml>>

2.2.3. Preprocesamiento

Los datos deben normalizarse, estandarizarse y limpiarse para aumentar su utilidad. Por lo general esto se lleva a cabo en la etapa de preprocesamiento, donde se aplican transformaciones a los datos antes de ser incluidos al algoritmo, esto permite convertir los datos sin procesar en un conjunto de datos limpio.¹⁹

Los algoritmos de aprendizaje profundo tienen una serie de parámetros de entrada para las redes neuronales, dentro de la arquitectura Faster-RCNN se deben segmentar los videos en frames, y luego a estos frames aplicarles un redimensionamiento.

Para realizar estos procesos existen una serie de algoritmos que permiten realizar las técnicas de manera automatizada. Para el preprocesamiento del conjunto de datos se aplicaron los siguientes:

➤ Escalado de Imagen con Vecinos Cercanos (Nearest Neighbor)

Esta técnica es ampliamente utilizada en el procesamiento de imágenes, ya que es una técnica muy eficaz y rápida. Esta técnica utiliza la interpolación de pixeles vecinos, dando como resultado una imagen más uniforme.

El principio de escalado de imagen es tener una imagen de referencia y usar esta imagen como base para construir una nueva imagen escalada. La imagen construida será más pequeña, más grande o de igual tamaño dependiendo de la relación de escala. Cuando se amplía una imagen, se están introduciendo espacios vacíos en la imagen de base original, la reducción por otro lado implica la reducción lo que significa la perdida de información irrecuperable. En este caso el algoritmo encontrará los pixeles correctos para eliminar de la imagen.²⁰

A continuación, se presenta un ejemplo de una imagen de dimensiones 2x2 y se quiere dimensionar a un tamaño de 3x3, se ilustras unos recuadros en negro que representan vacíos, Que necesitan interpolación; finalmente se muestra la imagen resultante de la interpolación de vecinos cercanos.

¹⁹ GeeksforGeeks. Data Preprocessing for Machine learning in Python. [En línea]. En: *GeeksforGeeks*. [Citado el 11 de julio, 2019]. Disponible en internet: <<https://www.geeksforgeeks.org/data-preprocessing-machine-learning-python/>>

²⁰ Virginia Tech. Better Image Scaling. [En línea]. En: VTCourses. [Citado el 11 de julio, 2019] Disponible en internet: <<http://courses.cs.vt.edu/~masc1044/L17-Rotation/ScalingNN.html>>

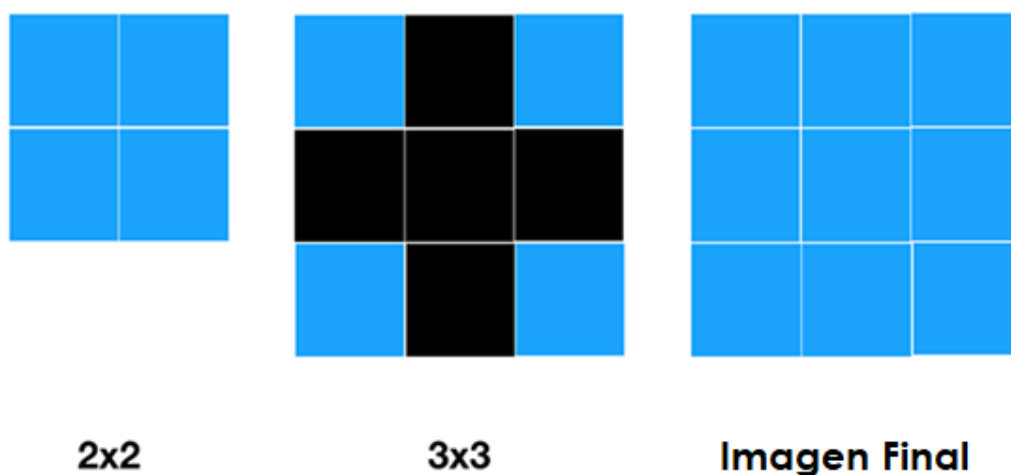


Figura 5: Ejemplo redimensionamiento.

Fuente: Los autores

El color de un pixel en la imagen resultante es el color del pixel más cercano de la imagen original. Si se amplía una imagen al doble de sus dimensiones, un pixel se ampliará a un área de 2x2 con el mismo color. Si se reduce una imagen a la mitad (1/2), solo se conservará un pixel sobre un área de 2x2 pixeles en la imagen resultante.

➤ Muestreo

El muestreo define un conjunto de métodos que permiten seleccionar subconjuntos de observaciones o elementos de datos individuales que conforman una población. El objetivo de esto es llegar a estimar una proporción estadística que llegue a estimar las características de toda la población. Existen una gran cantidad de métodos, pero depende del tipo de tarea en la que se usará para determinar su escogencia.²¹

Dentro de la terminología básica para comprender estos métodos, se encuentran tres términos esenciales, previamente se definió el más importante (muestreo), pero también se encuentran la población, que representa una colección de elementos que tienen una o varias características en común, el número de elementos en la población

²¹ Katharopoulos, Angelos. Fleure, François. Not All Samples Are Created Equal: Deep Learning with Importance Sampling. En: Arxiv. Volume 2. 2018. Pages 1-13.

representa su tamaño. Además, la muestra, que representa un subconjunto de la población.²²

Las técnicas de muestreo se agrupan en dos categorías, el muestreo probabilístico y el muestreo no probabilístico. La diferencia entre estos dos grupos radica en si la selección de la muestra se basa en la aleatoriedad o no. Con la probabilidad, cada elemento tiene la misma oportunidad para ser escogido y hacer parte de la muestra.

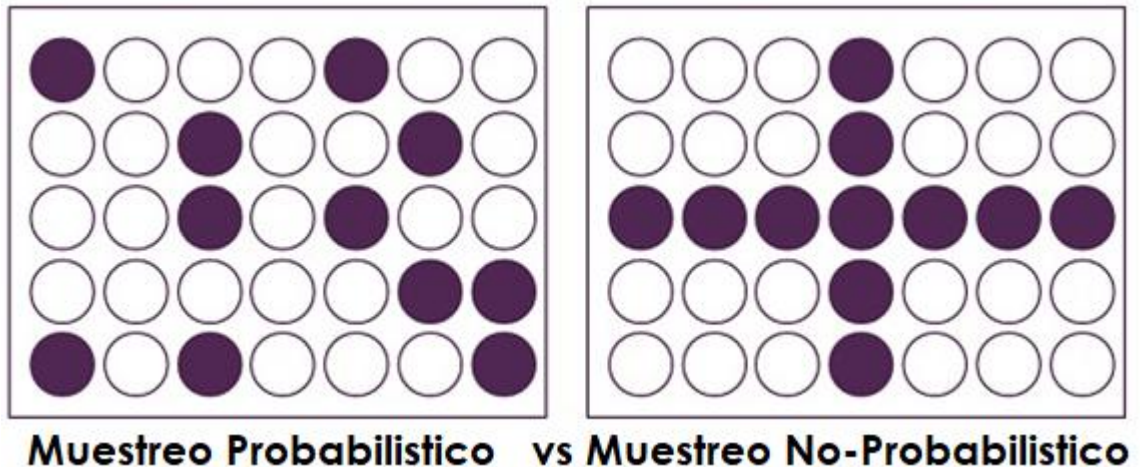


Figura 6: Categorías de las técnicas de muestreo.

Fuente: Towards Data Science, 2019. Sampling Techniques. En: Towards Data Science [en línea]. En: *Towards Data Science*. [Citado el 11 de julio, 2019]. Disponible en internet: <<https://towardsdatascience.com/sampling-techniques-a4e34111d808>>

En el experimento se optó por una técnica probabilística, ya que permiten la obtención de una muestra que si resulta significativa ya que es más probable la obtención de un subconjunto que posea todos los tipos de elementos que conforman el marco muestral.

Específicamente se aplicó una técnica de muestreo aleatorio simple, la cual en términos simple es como realizar un sorteo justo entre todos los elementos que conforman el marco muestral, a cada elemento se le asigna un número y este número entra en una urna, se empiezan a extraer los

²² Towards Data Science, 2019. Sampling Techniques. [En línea]. En: Towards Data Science. En: *Towards Data Science*. [Citado el 11 de julio, 2019]. Disponible en internet: <<https://towardsdatascience.com/sampling-techniques-a4e34111d808>>

números y todos los elementos que tengan el número extraído formarán parte de la muestra.

Si los elementos pueden ser seleccionados más de una vez se habla de Muestreo Aleatorio Simple, en el cual el número de combinaciones posibles se obtiene con N^n , donde N representa el número total de la población y n el número de elementos que serán usados para la muestra.

Si por el contrario se usa un Muestreo Aleatorio Simple sin reposición, un elemento seleccionado solo podrá ser seleccionado una vez y ya no entrará en el sorteo. El número posible de muestras que se pueden obtener está dado por la siguiente fórmula $\frac{N!}{n!(N-n)!}$, donde N representa el número total de la población y n el número de elementos que serán usados para la muestra, (!) es la notación factorial para la multiplicación secuencial de un número por un número menos, continuando hasta llegar a 1.

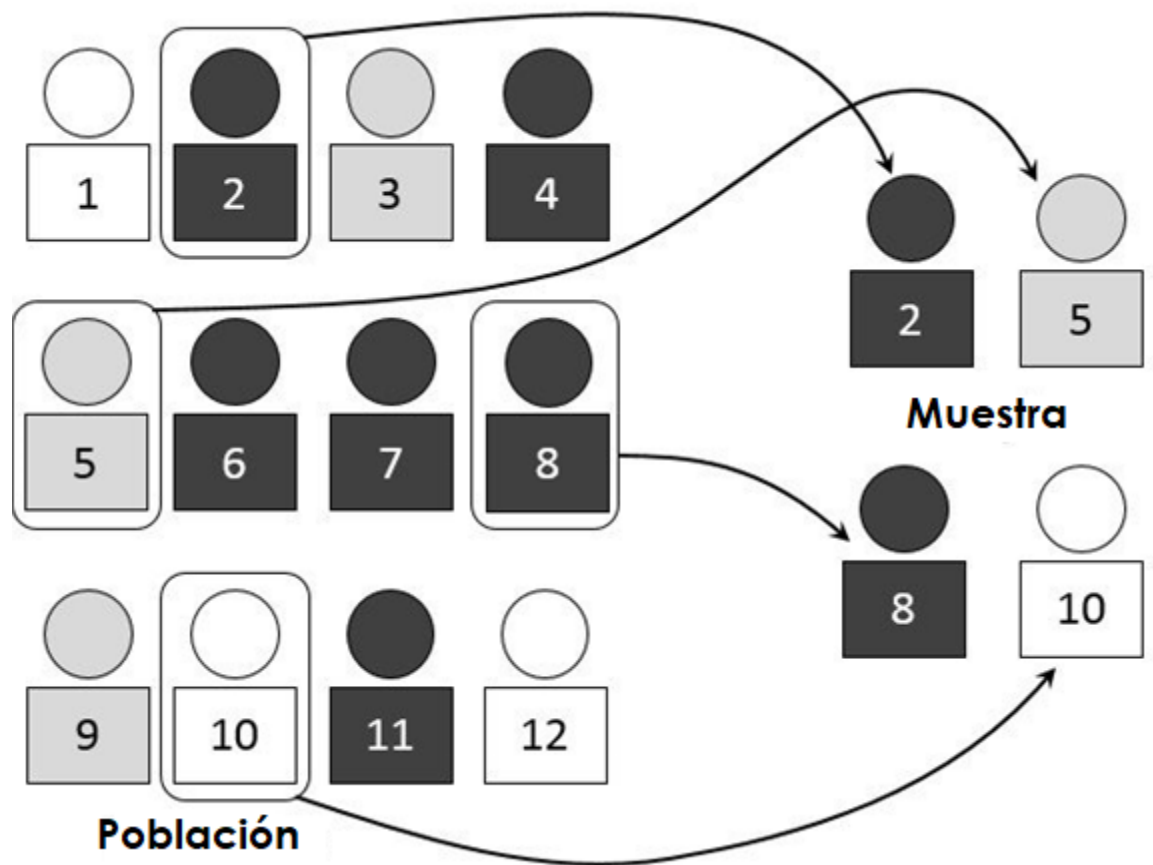


Figura 7: Muestreo Aleatorio Simple.

Fuente: Gestipolis, 2019. Muestreo Probabilístico y no Probabilístico. [En línea]. En: *Gestipolis*. [Citado el 11 de julio, 2019]. Disponible en internet: <<https://www.gestipolis.com/muestreo-probabilistico-no-probabilistico/a4e34111d808>>

2.2.4. Extracción de Características

Es un proceso de reducción de dimensionalidad sobre un conjunto de datos, estos poseen cientos, o incluso miles; de características. Si la cantidad de características se vuelve similar que la cantidad de observaciones almacenadas en un conjunto de datos, entonces puede llevar a un modelo a un sobreajuste. Para evitar este tipo de problemas, es necesario aplicar esta técnica.²³

Dado un conjunto de características $F = \{f_1, \dots, f_i, \dots, f_n\}$, surge el problema de encontrar un subconjunto con alguna de estas características que pueda lograr resumir la mayor parte de la información contenida en el conjunto original de características y que maximice la capacidad del modelo para clasificar los patrones.

Existen diferentes técnicas para la extracción de características, las redes neuronales son herramientas del aprendizaje profundo debido a su capacidad de destilar grandes cantidades de información de diferentes tipos de datos, extrayendo características complejas.

➤ Redes Neuronales

Son modelos que simulan el comportamiento de las neuronas de los seres humanos y su interconexión, cuentan con unidades básicas llamadas neuronas que se encuentran distribuidas en diferentes capas como se ilustra en la siguiente imagen.

Las Redes Neuronales Profundas han tenido éxito en una amplia variedad de aplicaciones en tecnología moderna e investigación. Estas generalmente se componen de varias capas, donde se tiene tres tipos, la capa de entrada que es la encargada de recibir los parámetros que entraran en la red neuronal, seguido de las capas ocultas que son las responsables de la extracción de características, finalmente la capa de salida que hace la clasificación real.

²³ DeepAI. Sampling Techniques. [En línea]. En: DeepAI. [Citado el 11 de julio, 2019]. Disponible en internet: <<https://deepai.org/machine-learning-glossary-and-terms/feature-extraction>>

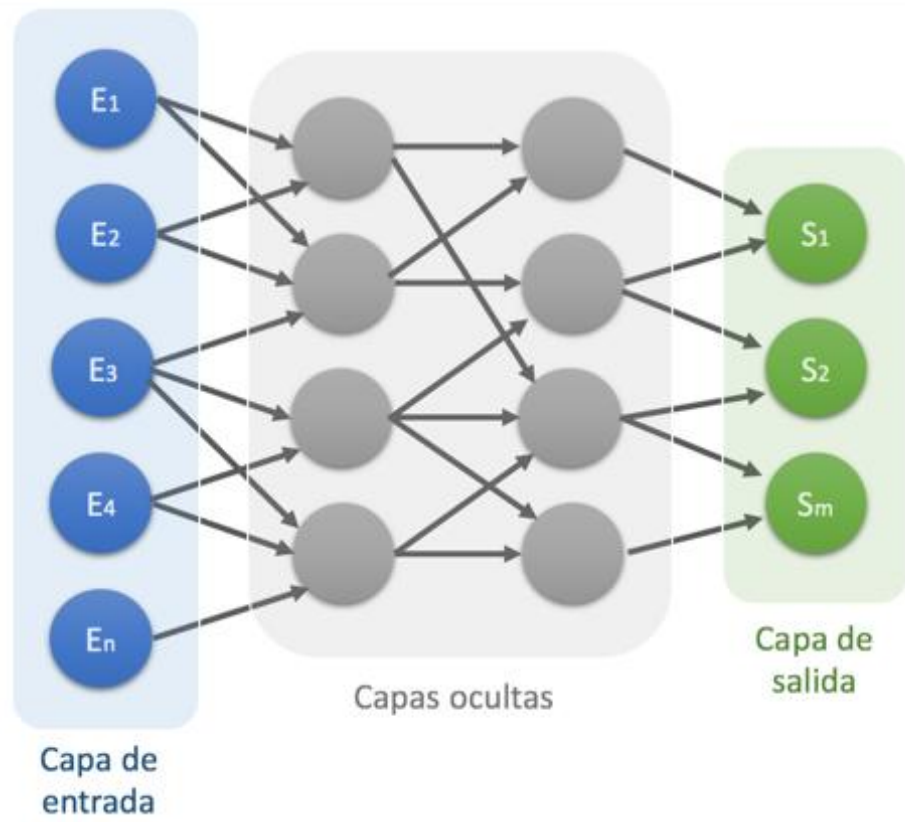


Figura 8: Red Neuronal.

Fuente: Calvo, Diego. Definición de red neuronal artificial. [En línea]. En:

DC. [Citado el 23 de abril, 2019]. Disponible en internet: <<http://www.diegocalvo.es/definicion-de-red-neuronal/>. >

En el experimento se implementó una Red Neuronal Convolutiva (CNN por sus siglas en inglés), exactamente una Faster RCNN (Faster Region Convolutional Neural Network), este tipo de Redes son altamente estructuradas por capas, la mayoría tienen las mismas capas de funciones, incluida la capa de convolución. Las CNN pueden diferir entre sí por cómo se encuentran instaladas las capas y también el método de entrenamiento de la red.

En este tipo de redes se encuentra una capa particular denominada, la capa convolutiva que se encarga de extraer características de la imagen de entrada. La convolución preserva la relación espacial entre los píxeles al aprender las características de la imagen utilizando pequeños cuadrados que representan conjuntos de píxeles.

La CNN aplica un 'filtro', también llamado 'detector de características' que recorrerá la imagen tomando ciertos conjuntos de píxeles y formará una matriz al aplicar todo el filtro sobre la imagen y calcular el producto punto, finalmente obteniendo el mapa de características.

Para una imagen de entrada se pueden realizar operaciones como Detección de bordes, Enfocar y Desenfocar simplemente cambiando los valores numéricos de nuestra matriz de filtros antes de la operación de convolución; esto significa que diferentes filtros pueden detectar diferentes características de una imagen, por ejemplo, bordes, curvas, etc. A continuación, se presenta un ejemplo con algunos de los filtros que se implementan.²⁴

²⁴ Liu, Han Yu Feature Extraction and Image Recognition with Convolutional Neural Networks. En: IOP Conf. Series: Journal of Physics: Conf. Series 1087. 2018. Pages 1-7.



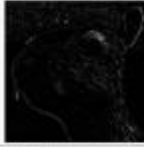
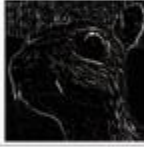



Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Figura 9: Filtros de capa de convolución.

Fuente: Data Science Blog. [En línea]. En: Data Science Blog. [Citado el 23 de septiembre, 2019] Disponible en internet: <<https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>>

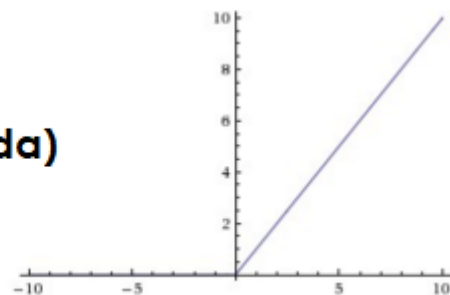
Una CNN en el proceso de entrenamiento va aprendiendo los valores de estos filtros, mientras la arquitectura de la red presente más filtros mejor será la red para llegar a reconocer algún tipo de patrón presente en las imágenes.

El mapa de características que reúne las características obtenidas por los filtros aplicados a lo largo de la Red Neuronal, está controlado por 3 parámetros.

La profundidad que corresponde al número de filtros usados para las operaciones de Convolución, cada filtro representa una matriz 2d con las características extraídas por cada filtro. El siguiente parámetro se conoce como el paso (stride) que representa el número de pixeles por el cual se deslizará la matriz que aplica el filtro, el tener pasos más grandes produce mapas de características más pequeños. Por último, se encuentra el relleno de ceros que consiste en rellenar la matriz de entrada con ceros alrededor del borde, esto para controlar el tamaño de los mapas de características.²⁵

Existe otro tipo de operación llama Unidad Lineal Rectificada (ReLU por sus siglas en ingles), es una operación basada en elementos y reemplaza todos los valores de pixeles que se encuentren negativos en el mapa de características por cero. El objetivo de ReLU es introducir la no linealidad en la CNN.²⁶

$$\text{Salida} = \text{Max}(\text{cero}, \text{entrada})$$



²⁵ Tryo Labs. Faster R-CNN: Down the rabbit hole of modern object detection. [En línea]. En: *Tryo Labs*. [Citado el 11 de julio, 2019]. Disponible en internet: <<https://tryolabs.com/blog/2018/07/faster-rcnn-down-the-rabbit-hole-of-modern-object-detection/>>

²⁶ Towards data science. Activation Functions in Neural Networks. [En línea]. En: *Towards data science*. [Citado el 11 de julio, 2019]. Disponible en internet: <<https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>>

Fuente: Los Autores

Existe otra capa llamada Pooling que se encarga de reducir la dimensionalidad de cada mapa de características, pero no su profundidad.

Por lo general se aplica el método de agrupación de Max Pooling, en donde dependiendo del paso se toma un conjunto de elementos, de este conjunto se toma el elemento más grande dentro de la región y se traslada a un nuevo mapa que contendrá las características más prominentes.

La operación de agrupación implica deslizar un filtro bidimensional sobre cada capa del mapa de características y resumir las características de la región cubierta por el filtro.

Para un mapa de características que tiene dimensiones $n_h * n_w * n_c$, las dimensiones de salida de salida que se obtienen después de una capa de agrupación son dadas por $\frac{n_h - f + 1}{s * n_c}$ donde, n_h representa la altura del mapa de características, n_w el ancho del mapa de características, n_c la profundidad o el número de canales, f el tamaño del filtro y s la longitud del paso.²⁷

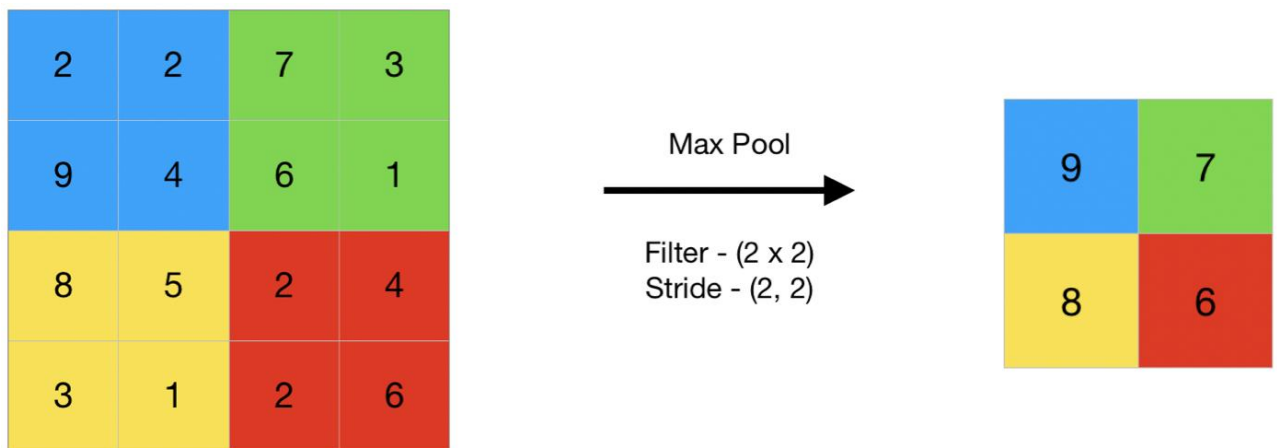


Figura 11: Ejemplo Max Pooling.

²⁷ Geeksforgeeks. CNN | Introduction to Pooling Layer. [En línea]. En: Geeksforgeeks. [Citado el 23 de septiembre, 2019] Disponible en internet: <<https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/>>

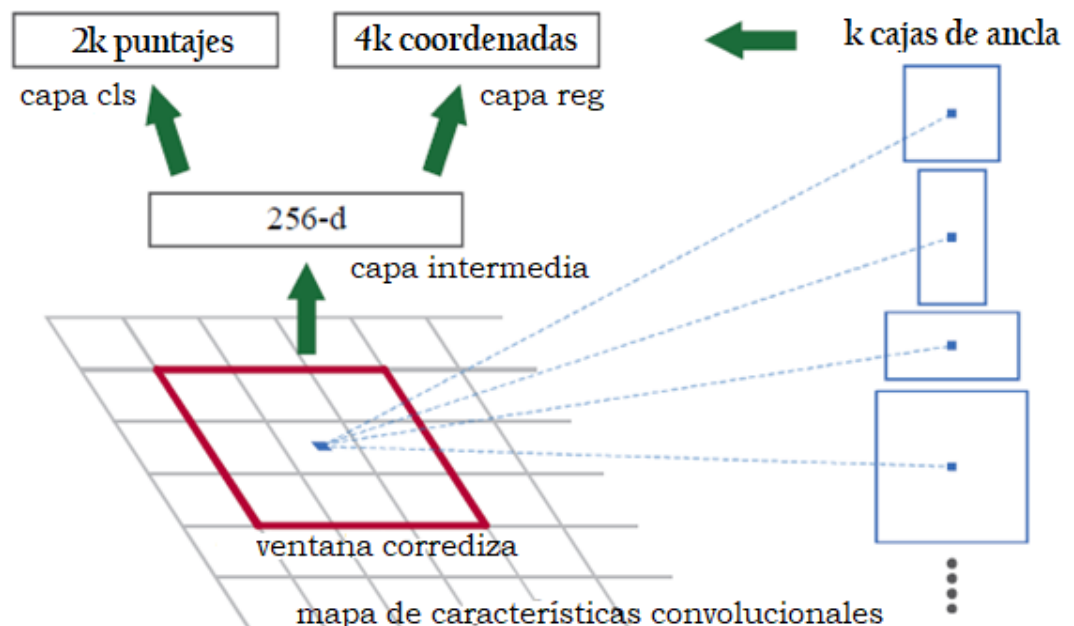
Fuente: Geeksforgeeks. CNN | Introduction to Pooling Layer. [En línea]. En: Geeksforgeeks. [Citado el 23 de septiembre, 2019] Disponible en internet: <<https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/>>

En la figura 13 se ilustra un ejemplo de la agrupación máxima para un mapa de características, con un paso 2 (stride) y un filtro para regiones 2x2. Cabe resaltar que esta operación de aplicación se aplica por separado a cada mapa de características.

➤ Red de Regiones Propuestas

Al finalizar la CNN tiene como salida un mapa de características, el cual tiene representa la entrada de la red de regiones propuestas RPN (por sus siglas en ingles), esta red tiene un clasificador y un regresor. El clasificador determina la probabilidad de que una región propuesta tenga el objeto de destino. La regresión trae las coordenadas de las regiones propuestas.

El proceso comienza por una ventana deslizante para cada ubicación sobre el mapa de características. Para cada ubicación, se utilizan k cuadros de anclaje (3 escalas de 128, 256 y 512, y 3 relaciones de aspecto 1:1, 1:2, 2:1) para generar las propuestas de región. Además, una capa **cls** genera $2k$ puntajes, ya sea que haya un objeto o no para k cajas. Una capa de registro genera $4k$ para las coordenadas (coordenadas del centro del



cuadrado, ancho y alto) de k cuadros.²⁸

Ancla	128², 2:1	128², 1:1	128², 1:2	256², 2:1	256², 1:1	256², 1:2	512², 2:1	512², 1:1	512², 1:2
Propuesta	188X111	113X114	70X90	416X229	261X284	174X332	768X437	499X501	355X715

Figura 12: Salida RPN

Fuente: Towards data science. [En línea]. En: *Towards data science*. [Citado el 23 de septiembre, 2019]. Disponible en internet: <<https://towardsdatascience.com/review-faster-r-cnn-object-detection-f5685cb30202>>

La figura representa la salida de la RPN y muestra el tamaño promedio de las regiones propuestas para 3 escalas de 128, 256, 512, y las 3 relaciones de aspecto 1:1, 1:2, 2:1.

Se aplica la siguiente función de pérdida:

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*)$$

El primer término es la pérdida de clasificación en dos clases (hay o no objeto). El segundo término es la pérdida de regresión de los cuadros delimitadores solo cuando hay un objeto (es decir, $p_i^* = 1$).

Por lo tanto, la red RPN debe verificar previamente qué ubicación contiene el objeto. Las ubicaciones correspondientes y los cuadros delimitadores pasarán a la red de detección para detectar la clase de objeto y devolver el cuadro delimitador de ese objeto.

²⁸ Towards Data Science. Review: Faster R-CNN (Object Detection). [En línea]. En: *Towards Data Science*. [Citado el 11 de julio, 2019]. Disponible en internet: <<https://towardsdatascience.com/review-faster-r-cnn-object-detection-f5685cb30202>>

2.2.5. Reconocimiento de objetos

Es una técnica de visión artificial que permite la identificación de los objetos presentes en una imagen o video. Esta tarea combina la clasificación e identificación para obtener como resultado la etiqueta correspondiente al objeto. Esta técnica se aplica con diversos tipos de aprendizaje de máquina y aprendizaje profundo, pero en cada uno de estos campos se presenta una

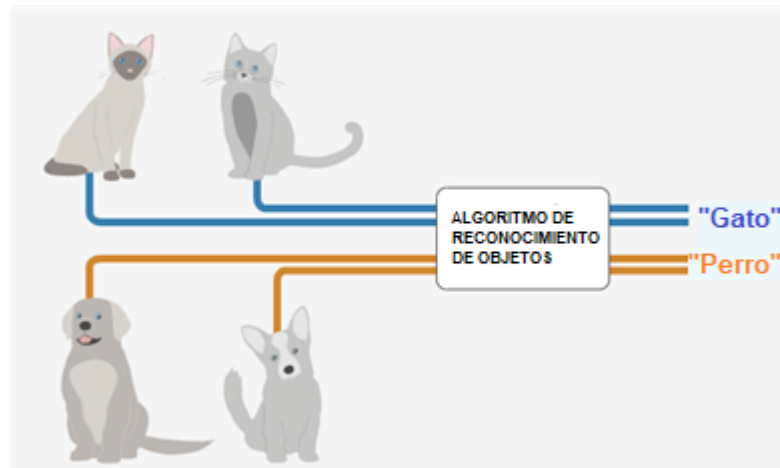


Figura 13: Reconocimiento de Objetos.

diferente ejecución.²⁹

Fuente: Mathworks. ¿Qué es el reconocimiento de objetos? [En línea]. En: *Mathworks*. [Citado el 23 de marzo, 2019]. Disponible en internet: <https://la.mathworks.com/solutions/deep-learning/object-recognition.html>

Esta representa la última fase de la red neuronal Faster-RCNN, después del paso de la RPN, se tiene varias regiones propuestas que no se encuentran asignadas a alguna clase. Este problema se intenta resolver reutilizando el mapa de características convolucional. Se realiza la extracción de mapas de características de tamaño fijo para cada región propuesta utilizando la agrupación de las regiones de interés. Estos mapas de características de tamaño fijo permiten clasificar las regiones en un número fijo de clases.

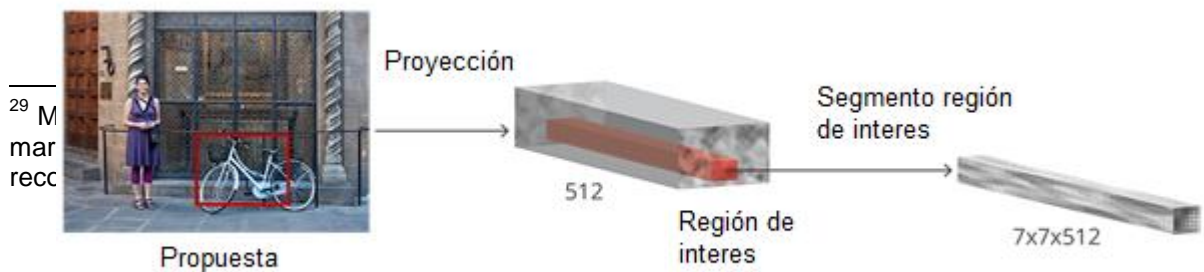


Figura 14: Agrupación regiones de interés.

Fuente: Tryo Labs. Faster R-CNN: Down the rabbit hole of modern object detection. [En línea]. En: Tryo Labs. [Citado 10 de octubre, 2019]. Disponible en internet: <<https://tryolabs.com/blog/2018/01/18/faster-r-cnn-down-the-rabbit-hole-of-modern-object-detection/>>

El paso final dentro del flujo de Faster-RCNN, es utilizar las regiones propuestas para la clasificación, toma cada una de las regiones propuestas y la clasifica en alguna de las clases, además una clase de fondo (para eliminar las propuestas incorrectas). Luego, realiza un ajuste al cuadro delimitador para la propuesta de acuerdo con la clase prevista.

R-CNN toma el mapa de características para cada región propuesta, reduce la dimensionalidad y usa dos capas completamente conectadas con función de activación ReLU. Luego, usa dos capas completamente conectadas diferentes para cada uno de los diferentes objetos:

Una capa completamente conectada con $N+1$ unidades donde N es el número total de clases y una más para la clase de fondo.

Una capa completamente conectada con $4 N$ unidades. Se quiere tener una predicción de regresión, por lo tanto, se necesita $\Delta_{center_x}, \Delta_{center_y}, \Delta_{width}, \Delta_{height}$ para cada una de las N clases posibles.

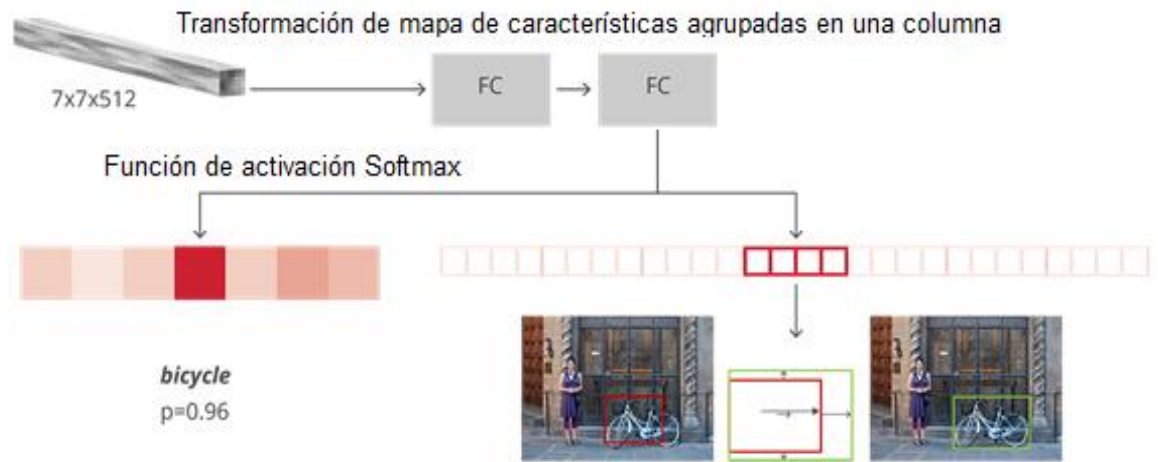


Figura 15: Detección frente a área prevista del objeto.

Fuente: Tryo Labs. Faster R-CNN: Down the rabbit hole of modern object detection. [En línea]. En: Tryo Labs. [Citado 10 de octubre, 2019]. Disponible en internet: <<https://tryolabs.com/blog/2018/01/18/faster-r-cnn-down-the-rabbit-hole-of-modern-object-detection/>>

2.2.6. Medidas de desempeño

Las medidas de desempeño permiten evaluar que tan efectivo es un modelo, basadas en el uso de un conjunto de métricas que permitan medir de forma adecuada el desempeño del algoritmo, sobre un conjunto de datos destinados para la prueba³⁰

En el proyecto se abarca un problema de clasificación, por lo tanto, se hace uso de las medidas de desempeño en dos etapas, la primera es la etapa de entrenamiento (proceso de aprendizaje) y la segunda es la etapa de pruebas. En la etapa de entrenamiento, las métricas son implementadas para seleccionar el modelo más óptimo para la fase de clasificación. Es decir, se usan las medidas de evaluación para discriminar y seleccionar la solución óptima que llegue a dar la predicción más precisa de la evaluación

³⁰ R. Caruana and A. Niculescu-Mizil, "Data mining in metric space: an empirical analysis of supervised learning performance criteria", in Proc. of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04), New York, NY, USA, ACM 2004, pp. 69-78.

futura de un clasificador particular. En la etapa de prueba se utilizan las métricas de evaluación para medir la efectividad que produce el clasificador, cuando se prueba con los datos destinados para ese fin.

➤ **Matriz de confusión**

Consiste una de las métricas más sencillas e intuitivas, que se utiliza para encontrar la exactitud del modelo. Se utiliza para el problema de clasificación donde la salida puede ser dos o más tipos de clases.

La exactitud de una clasificación puede evaluarse calculando el número de objetos de la clase que han sido reconocidos correctamente (verdaderos positivos), la cantidad de objetos reconocidos correctamente, pero que no pertenecen a la clase (verdaderos negativos) y los objetos que no fueron asignados incorrectamente a la clase (falsos positivos) o los que no fueron reconocidos como objetos de la clase (falsos negativos). Estos cuatro

		Predicción	
		Positivos	Negativos
Observación	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Figura 16: Matriz de confusión.

cálculos constituyen una matriz de confusión.³¹

Fuente: Zelada, Carlos. Evaluación de modelos de clasificación. [En línea]. En: *Rpubs*. [Citado el 10 de octubre, 2019]. Disponible en internet: <<https://rpubs.com/chzelada/275494>>

³¹ Sunasra, Mohammed, 2017. Performance Metrics for Classification problems in Machine Learning. [En línea]. En: *Medium*. [Citado el 10 de octubre, 2019]. Disponible en internet: <<https://medium.com/thalus-ai/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b>>

➤ Precision

Precision es una medida que dice qué de la proporción de objetos que se dejaron en una clase, cuales en realidad pertenecen a esta.

$$Precision\ binaria = \frac{VP}{VP + FP}$$

El enfoque de evaluación para el modelo binario en términos de las etiquetas de objetos que son positivas.

$$Precision_{\mu}\ multivariado = \frac{\sum_{i=1}^l vp_i}{\sum_{i=1}^l (vp_i + fp_i)}$$

$$Precision_M\ multivariado = \frac{\sum_{i=1}^l \frac{vp_i}{(vp_i + fp_i)}}{l}$$

El enfoque de evaluación para el modelo multiclase en términos de precisión es evaluado de dos maneras: una medida es el promedio de las mismas medidas para cada clase (promedio macro con un subíndice M), o la suma de conteos para obtener una vp; fn; vp; fp acumulativas (promedio micro con un subíndice μ).³²

➤ Recall

El recall es la medida que dice la proporción de objetos que realmente fueron clasificados y pertenecen a la clase.

$$Recall\ binaria = \frac{VP}{VP + FN}$$

El enfoque de evaluación para el modelo binario es la efectividad de un clasificador para identificar las etiquetas positivas.

$$Recall_{\mu}\ multivariado = \frac{\sum_{i=1}^l vp_i}{\sum_{i=1}^l (vp_i + fn_i)}$$

³² Sunasra, Mohammed, 2017. Performance Metrics for Classification problems in Machine Learning. [En línea]. En: *Medium*. [Citado el 10 de octubre, 2019]. Disponible en internet: <<https://medium.com/thalus-ai/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b>>

$$Precision_M \text{ multivariado} = \frac{\sum_{i=1}^l \frac{vp_i}{(vp_i + fn_i)}}{l}$$

El enfoque de evaluación para el modelo multiclase en términos de recall es evaluado de dos maneras: una medida es el promedio de las mismas medidas para cada clase (promedio macro con un subíndice M), o la suma de conteos para obtener una vp; fn; vn; fp acumulativas (promedio micro con un subíndice μ).³³

➤ F_β Score

La puntuación F_β es la media armónica entre precision y recall Indica que tan preciso es un clasificador, así como que tan robusto es.

$$F_\beta \text{ binaria} = \frac{(\beta^2 + 1)VP}{(\beta^2 + 1)VP + \beta^2 * FN + FP}$$

El enfoque de evaluación para el modelo binario es la relación entre las etiquetas positivas de los datos y las dadas por el clasificador.

$$F_\beta \text{ score}_\mu = \frac{(\beta^2 + 1)Precision_\mu Recall_\mu}{\beta^2 Precision_\mu + Recall_\mu}$$

$$F_\beta \text{ score}_M = \frac{(\beta^2 + 1)Precision_M Recall_M}{\beta^2 Precision_M + Recall_M}$$

El enfoque de evaluación para el modelo multiclase en términos de recall es evaluado de dos maneras: una medida es el promedio de las mismas medidas para cada clase (promedio macro con un subíndice M), o la suma de conteos para obtener una VP; FN; VN; FP acumulativas (promedio micro con un subíndice μ).

➤ Intersección Sobre la Unión

Es una métrica que mide la superposición entre dos límites. Se implementa para medir cuanto se superpone en límite de la predicción con el límite del objeto real (ground truth). La intersección sobre la unión permite evaluar

³³ Sunasra, Mohammed, 2017. Performance Metrics for Classification problems in Machine Learning. [En línea]. En: *Medium*. [Citado el 10 de octubre, 2019]. Disponible en internet: <<https://medium.com/thalus-ai/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b>>

cualquier algoritmo que proporcione cuadros delimitadores predichos como salida.³⁴

Para aplicar esta métrica es necesario contar con los cuadros delimitadores de los límites reales del objeto (son los cuadros etiquetados a mano del conjunto de testeo que indican en que parte de la imagen se encuentra el objeto). Además, los cuadros delimitadores que han sido predichos por el modelo.

³⁴ PyImageSearch. Intersection over Union (IoU) for object detection. [En línea]. En: *PyImageSearch*. [Citado el 10 de octubre, 2019]. Disponible en internet: <<https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>>


$$\text{Intersección sobre Unión} = \frac{\text{Área de Superposición}}{\text{Área de Unión}}$$


Figura 17: Ecuación de la intersección sobre la unión.

Fuente: PyImageSearch. Intersection over Union (IoU) for object detection. [En línea]. En: *PyImageSearch*. [Citado el 10 de octubre, 2019]. Disponible en internet: <<https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>>

La ecuación presentada en el numerador calcula el área de superposición entre el cuadro delimitador predicho y el cuadro delimitador real del objeto, en el denominador se calcula el área que abarca tanto el cuadro delimitador predicho como por el cuadro delimitador real del objeto.

2.3. Estado del arte

➤ **Detección de Pistolas Ocultas En Rayos X O en Imágenes de Ondas Milimétricas**

Las tecnologías de rayos X se han implementado en controles de seguridad como por ejemplo los aeropuertos, ya que por medio de esta herramienta es posible la detección de artefactos ocultos. Este método se aplicó para la detección de pistolas en imágenes de rayos X y de ondas milimétricas por medio de la visión computacional a través de histogramas de región y extensiones tridimensionales. Estas imágenes presentan un grado de complejidad alto, dado que contienen bastante ruido y otros artefactos que dificultan los procesos de clasificación.

Este enfoque en la tarea de reconocimiento presenta un bajo rendimiento ya que el ruido en las imágenes es bastante alto y ocasiona que el descriptor no tenga un grado de confianza alto, además, armas como la pistola Browning presentan diferentes reconocimientos.³⁵



Figura 18: Imagen de rayos X con reconocimiento de pistola.

Fuente: Greg, Flitton. Toby, P. Breckon. Najla, Megherbi. A comparison of 3D interest point descriptors with application to airport

³⁵ Greg, Flitton. Toby, P. Breckon. Najla, Megherbi. A comparison of 3D interest point descriptors with application to airport baggage object detection in complex CT imagery. En: *El Sevier*. Volume 46. Sept 2013. Pages 2420-2436.

baggage object detection in complex CT imagery. En: el sevier. Volume 46. Sept 2013. Pages 2420-2436.

➤ **Proceso Automático De Análisis De Imagen Para La Detección De Armas Ocultas**

La investigación plantea utilizar el hardware de imágenes y sin intervención de personas busca alertar una buena detección de armas ocultas junto con la ubicación en la imagen. Aunque en el momento todavía se utilizan operadores capacitados para observar la imagen resultante y sacar las conclusiones correctas. El primer acercamiento es por el borde de las armas que utiliza el gatillo ya que es bastante consistente en dimensiones.

Los procesos fueron correctos para reconocer pistolas, en las imágenes, excepto en las más simples, el tiempo de cómputo fue excesivo y se generó un número sustancial de falsos positivos. El segundo enfoque empleado transformaciones wavelet Daubechie, pero los resultados hasta ahora no han sido concluyentes. Un tercer enfoque que involucra un algoritmo basado en la transformación de la característica invariante de escala (SIFT) esta propuesto.³⁶

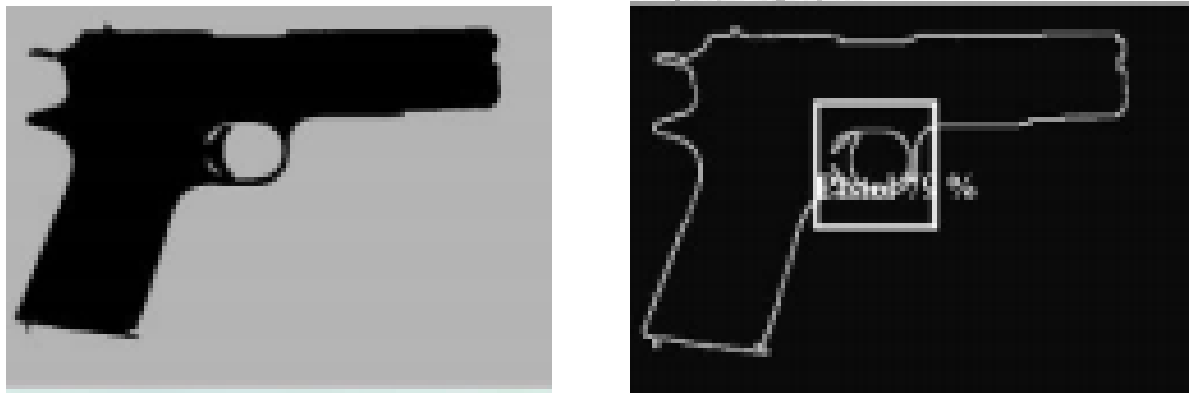


Figura 19: Reconocimiento contorno de pistola.

³⁶ Gesick, Richard. Saritac, Caner. Hung, Chih-Cheng. Automatic Image Analysis Process for the Detection of Concealed Weapons. En: Stanford. ACM International Conference Proceeding Series. [Citado el 10 de octubre, 2019].

Fuente: Gesick, Richard. Saritac, Caner. Hung, Chih-Cheng. Automatic Image Analysis Process for the Detection of Concealed Weapons. En: Stanford. ACM International Conference Proceeding Series. [Citado el 10 de octubre, 2019].

➤ **Reto De Reconocimiento Visual De Gran Escala De Imagenet**

Por medio de una larga investigación y colaboración de varias instituciones se ha logrado obtener un conjunto de datos de referencia que ha logrado diversos avances en el campo de detección y mostrando los avances claves en el reconocimiento de objetos categóricos, proporcionando un análisis del área de la clasificación en imágenes a gran escala y la detección de objetos.³⁷

El contar con este conjunto de datos permite mejorar la precisión de un clasificador, ya que si se entrena inicialmente con este conjunto va a



Figura 20: Conjunto de datos COCO.

³⁷ Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro perona, Deva Ramanan, Piotr Dollár, C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. En: SpringerLink. 2014. Pages 740-755. [Citado el 10 de octubre, 2019].

tener un ajuste inicial más efectivo.

Fuente: Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro perona, Deva Ramanan, Piotr Dollár, C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. En: SpringerLink. 2014. Pages 740-755. [Citado el 10 de octubre, 2019].

➤ **Un Marco Basado en la Visión Artificial Para la Detección Visual de Armas Utilizando el Detector de Puntos de Interés Harris**

En el artículo se entra a trabajar por primera vez la vigilancia automática por medio de la visión computacional, es decir que en una foto se pueda identificar un arma que se encuentre visible. El marco propuesto explora la segmentación basada en el color y la forma de los objetos presentes en una imagen, para dejar únicamente el objeto de interés en ese caso el arma. Al tener la imagen segmentada se aplica un detector de puntos de interés (Harris) que permite definir el contorno del arma, si este contorno tiene una similitud mayor al 50% se dibuja un rectángulo

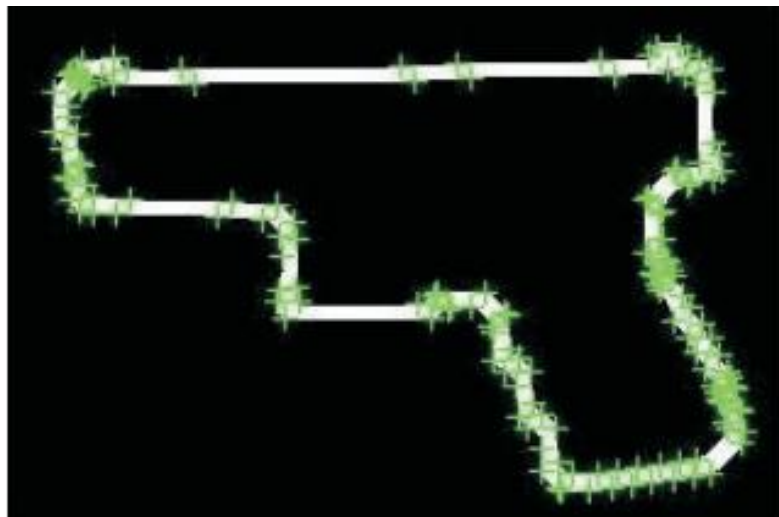


Figura 21: Detector de puntos de interés (Harris).

alrededor del arma.

Fuente: Tiwari, Rohit Kumar. K. Verma, Gyanendra. A Computer Vision based Framework for Visual Gun Detection Using Harris Interest Point Detector. En: ScienceDirect. Volume 54. 2015. Pages 703-712. [Citado el 10 de octubre, 2019].

El método cuenta con un enfoque robusto que es capaz de detectar múltiples armas y en diferentes ángulos de visión, sin embargo, solo aplica para la detección de armas en imágenes, pero aplica como una incursión de modelo de detección en video.³⁸

➤ **Compensación De Velocidad / Precisión Para Detectores De Objetos Convolucionales Modernos**

El documento de investigación se enfoca en varias pruebas de rendimiento en los detectores de objetos más utilizados últimamente para encontrar cual tiene una mejor compensación entre velocidad y precisión. Aunque también aclaran que son difíciles de comparar debido a las características de los extractores (Resnet, Inception, VGG, etc.), la resolución de las imágenes y las especificaciones de hardware. Se presentan las pruebas de desempeño a las redes Faster-RCNN, R-FCN, SSD y se muestra la curva de compensación de velocidad / precisión. En la investigación se presenta un detector que logra velocidades de tiempo real que se puede implementar en dispositivos móviles, así mismo se encuentra el detector con la peor precisión, y por último un detector que logra un mejor rendimiento con el conjunto de

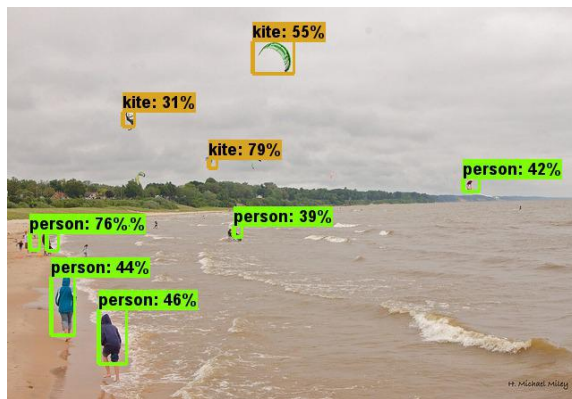


Figura 22: SSD + Mobilenet. datos de COCO.³⁹



Figura 23: FRCNN + Resnet 100.

³⁸ Tiwari, Rohit Kumar. K. Verma, Gyanendra. A Computer Vision based Framework for Visual Gun Detection Using Harris Interest Point Detector. En: ScienceDirect. Volume 54. 2015. Pages 703-712. [Citado el 10 de octubre, 2019].

³⁹ Fuente: Huang, Jonathan. Vivek, Rathod. Chen, Sun. Menglong, Zhu. Anoop, Korattikara. Fathi, Alireza. Ian, Fischer. Zbigniew, Wojna. Yang, Song. Guadarrama, Sergio. Murphy, Kevin. ArXiv: 1611.10012v3, 2017. Review: Speed/accuracy trade-offs for modern convolutional object detectors. [en línea]. [Citado el 10 de octubre, 2019].



Figura 24: RFCN + Resnet 10.

Fuente: Huang, Jonathan. Vivek, Rathod. Chen, Sun. Menglong, Zhu. Anoop, Korattikara. Fathi, Alireza. Ian, Fischer. Zbigniew, Wojna. Yang, Song. Guadarrama, Sergio. Murphy, Kevin. ArXiv: 1611.10012v3, 2017. Review: Speed/accuracy trade-offs for modern convolutional object detectors. [en línea]. [Citado el 10 de octubre, 2019].

➤ Alarma De Detección De Armas En Videos Con Aprendizaje Profundo

La investigación realizada expone la problemática que existe en varios países del mundo donde la compra de armas es legal y se evidencian el uso de diferentes armas. Al utilizar técnicas de aprendizaje profundo se busca minimizar el número de falsos positivos y se evalúan varios algoritmos con las tecinas de ventana deslizante y el de regiones propuestas.

Los mejores resultados son obtenidos por la red Faster RCNN con el conjunto de datos que se utilizó en el artículo, este detector muestra un alto potencial incluso en videos de YouTube de baja calidad y tiene buenos resultados en el sistema de alarma matic. También se define



Figura 25: Ejemplo de falsos negativos, con las armas del fondo.

una nueva forma para la activación de alarma por intervalo (AApl), para evaluar el rendimiento de un modelo de detección con alarma en videos.⁴⁰

Fuente: Olmos, Roberto. Tabik, Siham. Herrera, Francisco. [En línea]. ArXiv: 1702.05147v1, 2017. Automatic Handgun Detection Alarm in Videos Using Deep Learning. [Citado el 10 de octubre, 2019]. Disponible en internet: <<https://arxiv.org/abs/1702.05147>>

Los resultados de la investigación ayudo a elegir la red neuronal convolucional más rápida (Faster RCNN) al demostrar que el método de regiones propuestas presenta los mejores resultados en estos algoritmos, pero también se decide elegir otro tipo de extractores de características para tener mejores resultados con el conjunto de datos obtenido en este proyecto expuesto en el apartado de Metodología.

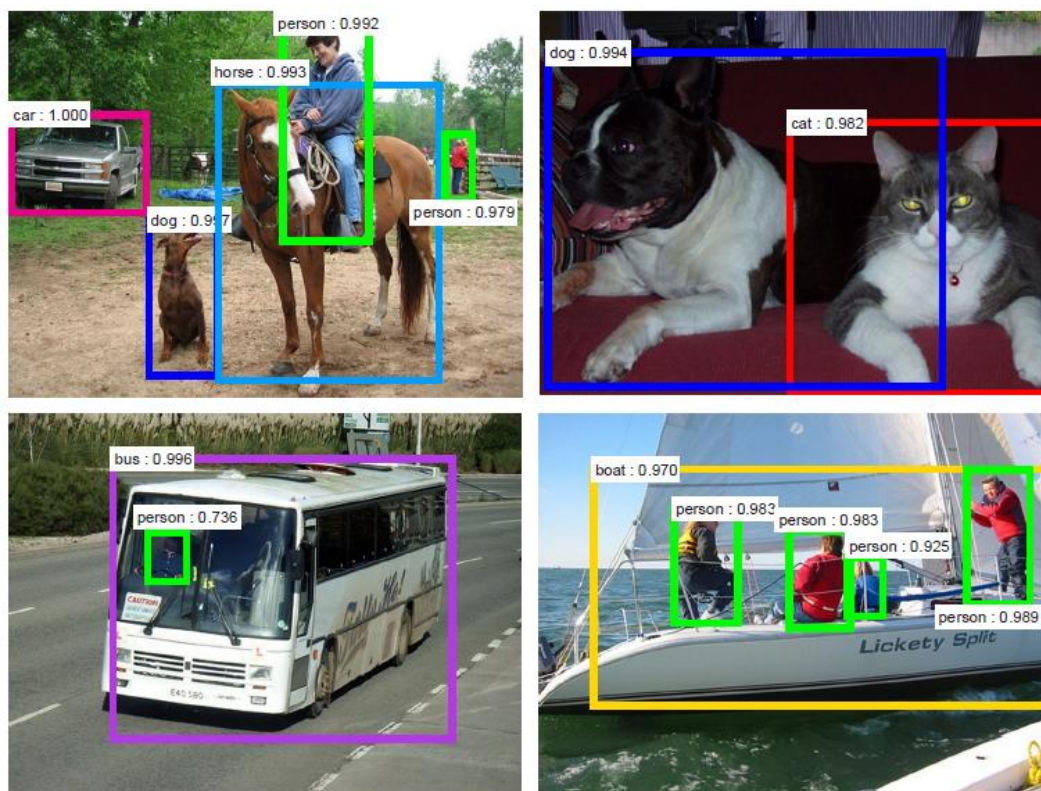
➤ **Faster R-CNN: Para Detección De Objetos En Tiempo Real Con La Red De Regiones Propuestas**

En este documento de investigación se encuentra la arquitectura de la Faster RCNN donde la pieza más importante es la red de regiones propuestas que comparte capas de convolución con la red de detección por lo que no tiene un costo de tiempo relevante.

La red RPN está entrenada para generar regiones de interés donde es muy probable que exista un objeto. Por ello esta red se mezcló con la Fast RCNN y dio como resultado la red Faster RCNN.

⁴⁰ Olmos, Roberto. Tabik, Siham. Herrera, Francisco. [En línea]. ArXiv: 1702.05147v1, 2017. Automatic Handgun Detection Alarm in Videos Using Deep Learning. [Citado el 10 de octubre, 2019]. Disponible en internet: <<https://arxiv.org/abs/1702.05147>>

El artículo implementa una red Faster RCNN y obtiene resultados de velocidad de fotogramas de 5 fps en una GPU, además obtiene buenos



resultados en las detecciones realizadas con conjuntos de datos de COCO y PASCAL VOC.⁴¹

Figura 26: Pruebas de detección de objetos.

Fuente: Ren, Shaoqing. He, Kaiming. Girshick, Ross. Sun, Jian. [En línea]. ArXiv: 1506.01497v3, 2016. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. [Citado el 10 de octubre, 2019] Disponible en internet: <<https://arxiv.org/pdf/1506.01497.pdf>>

⁴¹ Ren, Shaoqing. He, Kaiming. Girshick, Ross. Sun, Jian. [En línea]. ArXiv: 1506.01497v3, 2016. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. [Citado el 10 de octubre, 2019] Disponible en internet: <<https://arxiv.org/pdf/1506.01497.pdf>>

3. METODOLOGÍA

3.1. Fases del Proyecto

Para realizar el trabajo de detección automática de armas de mano en video es necesario realizar una serie de etapas las cuales se definirán de la siguiente manera.

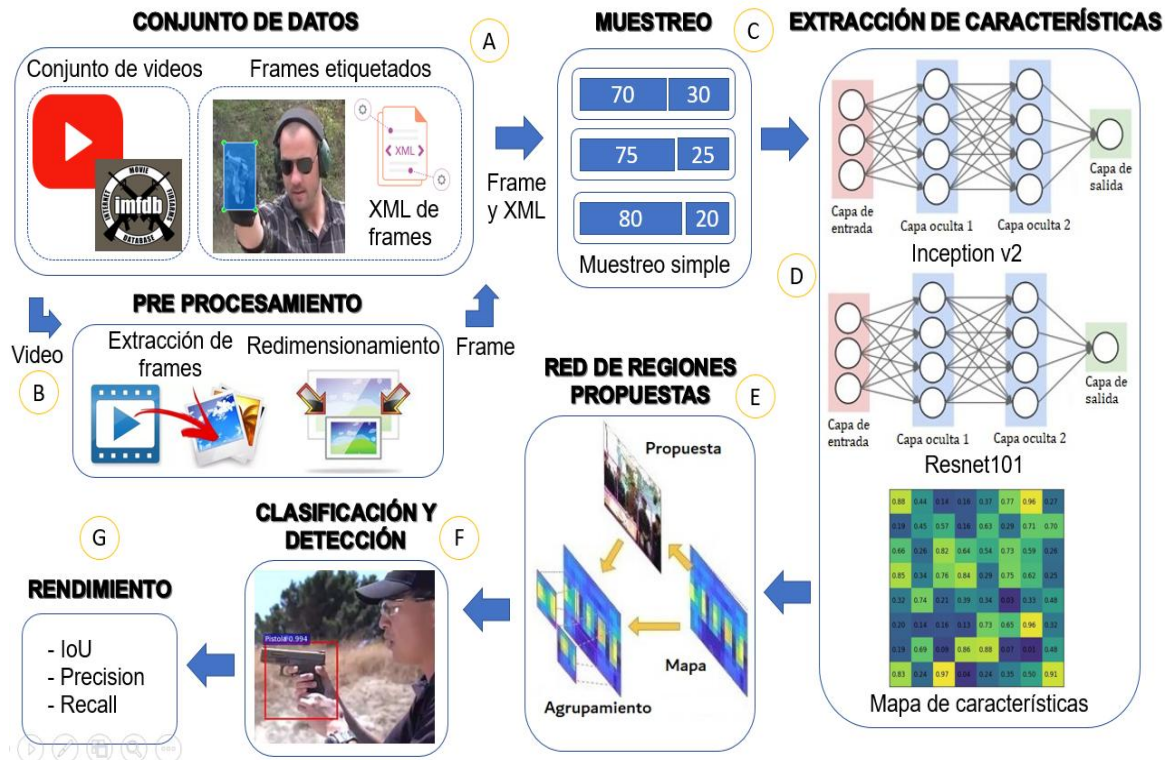


Figura 27: Metodología del proyecto

Conjunto de datos:

En esta etapa se buscan videos que contengan armas de mano (revólveres y pistolas) de cualquier tipo, es decir, escenas de películas, entrenamiento con armas, defensa con armamento, etc. Que se obtienen de diferentes páginas web como YouTube y una plataforma de películas como se ve en la figura 18 en la parte A.

Pre procesamiento:

- En la figura 18 en la parte B se realiza una extracción de frames a todos los videos obtenidos para el conjunto de datos.

- Se realiza un redimensionamiento a cada frame para que la red neuronal cuente con mayor velocidad al tener una resolución baja.

Muestreo:

Como se observa en la figura 18 en la parte C, se utiliza el método de muestreo simple con el fin de tener solamente un conjunto de entrenamiento, y otro de testeo.

Extracción de características:

Para la extracción de características se utilizan dos extractores diferentes, el primer extractor es 'inception v2' el cual tiene una arquitectura sencilla para buscar más velocidad que exactitud, el segundo es el extractor 'resnet101' que es más robusto, pero más exacto.

Como salida de esta red neuronal convolucional se tiene un mapa de características por cada parche de los frames que entran a la red como se muestra en la figura 18 en la parte D.

Regiones propuestas:

En esta fase se busca obtener las regiones que tienen más probabilidad de tener un arma de mano. Esto se logra a partir de los mapas de características los cuales entran a una red de regiones propuestas y tiene como salida una o varias regiones, estas regiones luego se unen si pertenecen al mismo objeto para lograr el área completa y poder pasar esta región de interés al clasificador como se ve en la figura 18 en la parte E.

Clasificación y Detección:

En esta etapa se va a modelar nuestro prototipo de detección de armas de mano por medio de la Faster RCNN enfocándonos en la etapa F la cual mostrara si la región de interés es un arma de mano o es otro objeto. En el caso de ser un arma de mano en la detección se buscan las coordenadas donde se localiza en la imagen original y se realiza una caja delimitadora como se muestra en la figura 18 en la parte F.

Rendimiento:

Como en la figura 18 en la parte G, en esta etapa se aplicarán técnicas de medidas de desempeño (métricas de detección de COCO) esto se realiza para poder detectar y comparar qué tan eficiente es la red de detección obtenida. Finalmente, se tendrá como salida los porcentajes de cada medida de desempeño que se aplicaron por cada conjunto de datos y por

cada extractor, en este caso las medidas AP (Average Precision), AR (Average Recall) con la ayuda de IoU (Intersection over union).

4. DISEÑO METODOLÓGICO

Conjunto de datos:

En esta etapa se buscan videos de YouTube que contengan armas de mano (revólveres y pistolas) con duración de mínimo 29 segundos y máximo 15 minutos por video, además estos se descargan con una calidad de 480p (854 x 480 pixeles) para que sean de fácil obtención en tiempos de descargas, pero al mismo tiempo tengan los datos de pixeles suficientes para el modelo. Este conjunto de datos está conformado por 55 videos donde se busca que sea un conjunto de datos balanceado en sus clases y por ello se obtienen un número igual de videos por revólveres y pistolas como se muestra en la siguiente

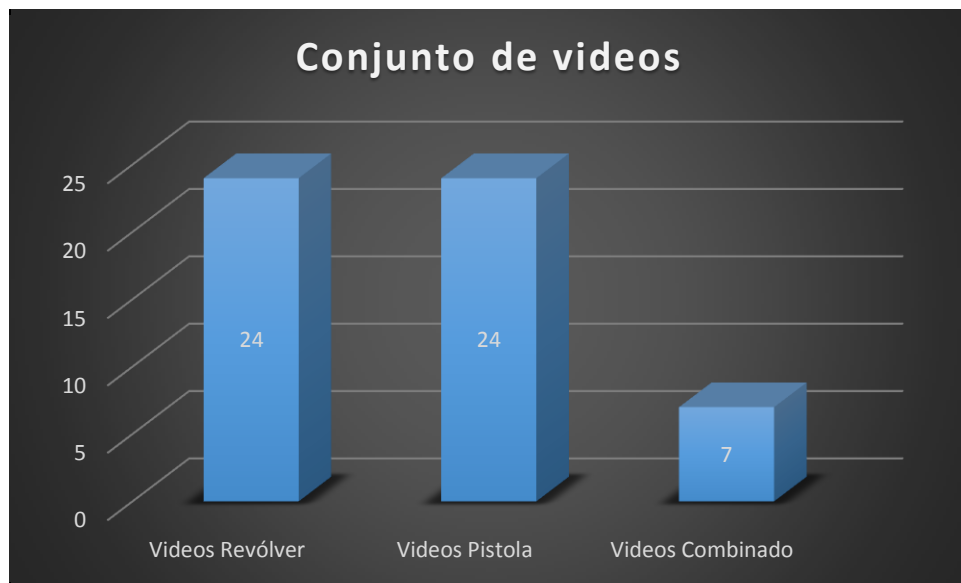


Figura 28: Conjunto de videos.

tabla.

Como parte del conjunto de datos también se encuentran los frames de estos videos dividido en dos partes, el primer conjunto de datos con frames con un periodo de 0,5 segundos y el segundo tiene frames con periodo de 1 segundo, esto con la finalidad de saber con qué conjunto es más preciso el modelo. El número de frames en cada conjunto de datos se muestra en la siguiente tabla.

Conjunto de datos	Cantidad de frames
0,5 segundos	16.271
1 segundo	8.130

Tabla 1: Cantidad de frames.

En estos conjuntos de datos se realiza una limpieza manual de los frames que no tienen armas. Al conjunto de datos de 1 segundo al no tener una buena cantidad de frames se le agregaron las imágenes del conjunto de datos de películas y contiene 100 frames más, por tanto, la tabla 1 queda de la siguiente forma.

Conjunto de datos	Cantidad de frames
0,5 segundos	9.524
1 segundo	2.590

Tabla 2: Cantidad de frames a utilizar.

Unos de los frames que se incluyen en la tabla 2 son mostrados a



Figura 29: Frames de revolver y pistola

continuación en las figuras 20.

En estos frames se evidencia que la distancia del arma con respecto a la cámara no es lejana y que se tiene una buena visión a pesar de que son frames obtenidos de YouTube con calidad de 480p.

Pre procesamiento:

- Luego de la recolección de videos de YouTube se realiza una extracción de frames a cada video, en este paso se decide extraer los frames con periodos de 0,02 segundos, 0,5 segundos y 1 segundo. Al obtener los frames se evidencia que para el conjunto de 0,02 segundos se adquirieron más de 244.000 frames por lo que se decide solo enfocarse en los frames de 0,5 segundos y los de 1 segundo esto con la finalidad de saber con qué conjunto es más preciso el modelo. En la figura 21 se evidencian algunos de los frames.



Figura 30: Frames de 1 segundo.

- Cada frame obtenido en el anterior paso se redimensiona con un tamaño de 800x600. Se escogen estas dimensiones con la finalidad de que estos frames sean de rápido procesamiento y no se pierda información de los pixeles. La figura 22 muestra de mejor manera el proceso de redimensionamiento.

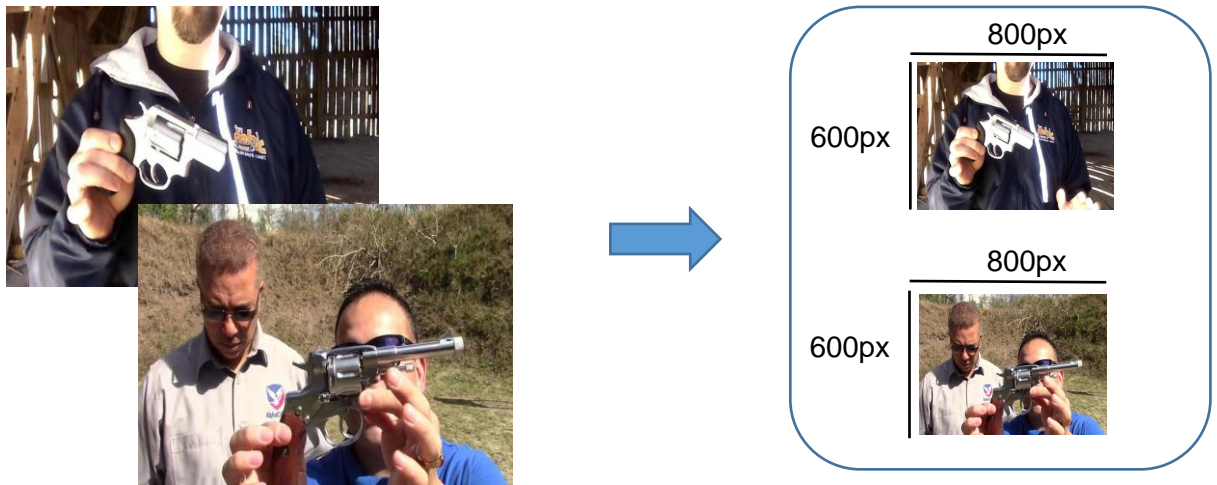


Figura 31: Redimensionamiento de frames.

- El siguiente paso a realizar es el etiquetado manual de los 12.114 frames del conjunto de datos por medio de cajas delimitadoras. La herramienta que se utilizó fue Labellmg y las etiquetas utilizadas fueron 'Revolver' y 'Pistola' como se muestra a continuación.

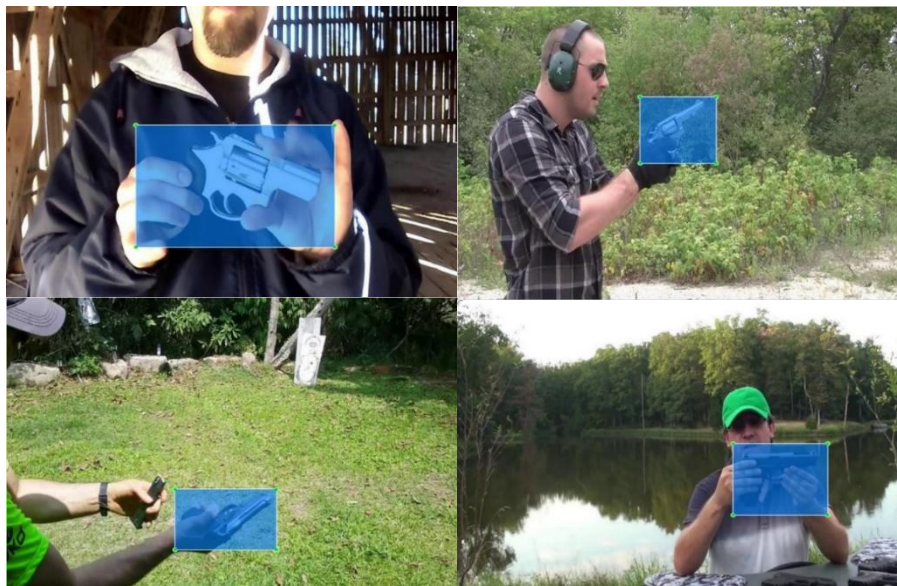


Figura 32: Frames etiquetados.

A partir de estos frames etiquetados por medio de la opción PASCAL/VOC que nos ofrece esta herramienta se toman las coordenadas de las armas de mano (revolver y pistola), y estos se guardan en un XML por cada imagen como se muestra en la figura 24. Estas coordenadas se obtienen con un X mínimo, X máximo, Y mínimo, Y máximo junto con el nombre de la etiqueta.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<annotation verified="yes">
  <folder>Frame 1-25 x 0.5seg</folder>
  <filename>500000.jpg</filename>
  <path>C:\Users\braya\Fram 1-25 x 0.5seg\500000.jpg</path>
  - <source>
    <database>Unknown</database>
  </source>
  - <size>
    <width>800</width>
    <height>600</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  - <object>
    <name>Revolver</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    - <bndbox>
      <xmin>408</xmin>
      <ymin>276</ymin>
      <xmax>468</xmax>
      <ymin>361</ymin>
    </bndbox>
  </object>
</annotation>
```

Figura 33: XML de la imagen 500000.jpg

Muestreo:

Para realizar el muestreo se utiliza el método de muestreo simple para obtener solamente un conjunto de entrenamiento, y otro de prueba. Aun así, para no obtener solamente un modelo, se decide realizar tres muestreos de 80%-20%, 75%-25% y 70%-20% con la finalidad de encontrar el mejor modelo para el conjunto de datos. Este proceso se evidencia en la figura 25.

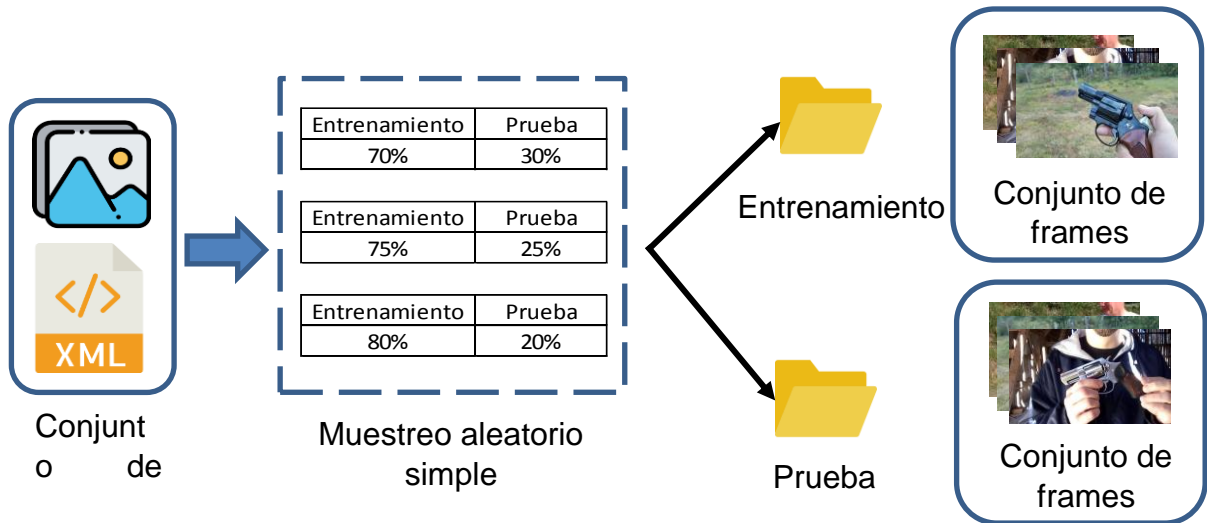


Figura 34: Muestreo del conjunto de datos.

Los resultados del muestreo en el número de frames se muestran en la siguiente tabla.

Muestreo	Conjunto de datos	Tipo	Número de frames
70-30	0,5 segundos	Entrenamiento	6666
		Prueba	2858
	1 segundo	Entrenamiento	1812
		Prueba	778
75-25	0,5 segundos	Entrenamiento	7143
		Prueba	2381
	1 segundo	Entrenamiento	1942
		Prueba	648
80-20	0,5 segundos	Entrenamiento	7619
		Prueba	1905
	1 segundo	Entrenamiento	2072

		Prueba	518
--	--	--------	-----

Tabla 3: Muestreo en número de frames.

Extracción de características:

Antes de realizar esta etapa es necesario conseguir los archivos .csv y .record, ya que este último es un parámetro para la Faster RCNN que es la red de regiones a utilizar en este proyecto, esta red incluye las etapas D, E y F, pero esta sección se enfocara en la fase D. El archivo csv recolecta la siguiente información de los XML:

Nombre del archivo	Ancho del frame	Altura del frame	Clase	Xmin	Ymin	Xmax	Ymax
500005.jpg	800	600	Revolver	370	43	432	148
500006.jpg	800	600	Revolver	378	30	436	114
500007.jpg	800	600	Pistola	402	91	459	169
500011.jpg	800	600	Pistola	461	360	536	467
500027.jpg	800	600	Revolver	137	31	721	485

Tabla 4: Contenido del archivo csv.

En este proyecto se implementa la red neuronal Faster RCNN por medio de la API de detección de objetos que proporciona Tensorflow, la cual apoya este proceso de detección por medio de las librerías y recursos que se encuentran en esta API ⁴². Para esto es necesario previamente realizar los siguientes pasos.

Primero se realiza la instalación de esta API donde se debe instalar todas las dependencias necesarias para su uso, las cuales son las siguientes.

- Protobuf 3.0.0
- Pillow 1.0

⁴² GITHUB TENSORFLOW. Tensorflow Object Detection API. [en línea]. En: *GITHUB*. [Citado el 10 de octubre de, 2019]. Disponible en internet: <https://github.com/tensorflow/models/tree/master/research/object_detection>

- Lxml
- Jupyter notebook
- Matplotlib
- Tensorflow>=1.12.0
- Cython
- Contextlib2
- Cocoapi

En este caso específico se trabaja con Tensorflow-gpu>=1.12.0 para que los procesos se ejecuten en un menor tiempo. La GPU utilizada en este proyecto fue la NVIDIA GeForce RTX 2080 la cual tiene una capacidad de cómputo de 7.5GB.⁴³ Para que la GPU se pueda utilizar es necesario realizar la configuración que se encuentra en el apartado de anexos.

Para terminar la instalación de la API de detección de objetos es necesario convertir los archivos de la carpeta Protos en archivos .py para que sean ejecutables más adelante.

Por último, se debe configurar las variables de entorno creando la variable PYTHONPATH en el sistema y colocando las referencias de las carpetas *object detection*, *slim* y *research*.

⁴³ NVIDIA. Recommended GPU for Developers. [En línea]. En: *NVIDIA*. [Citado el 30 de agosto, 2019]. Disponible en: <https://developer.nvidia.com/cuda-gpus>

Regiones propuestas:

Esta fase se encuentra totalmente oculta dentro de la red neuronal (Faster RCNN), pero se evidencia en las pruebas del algoritmo de detección ya que se proponen diferentes regiones en las que puede existir un arma de mano, la siguiente figura muestra un ejemplo de esta etapa.

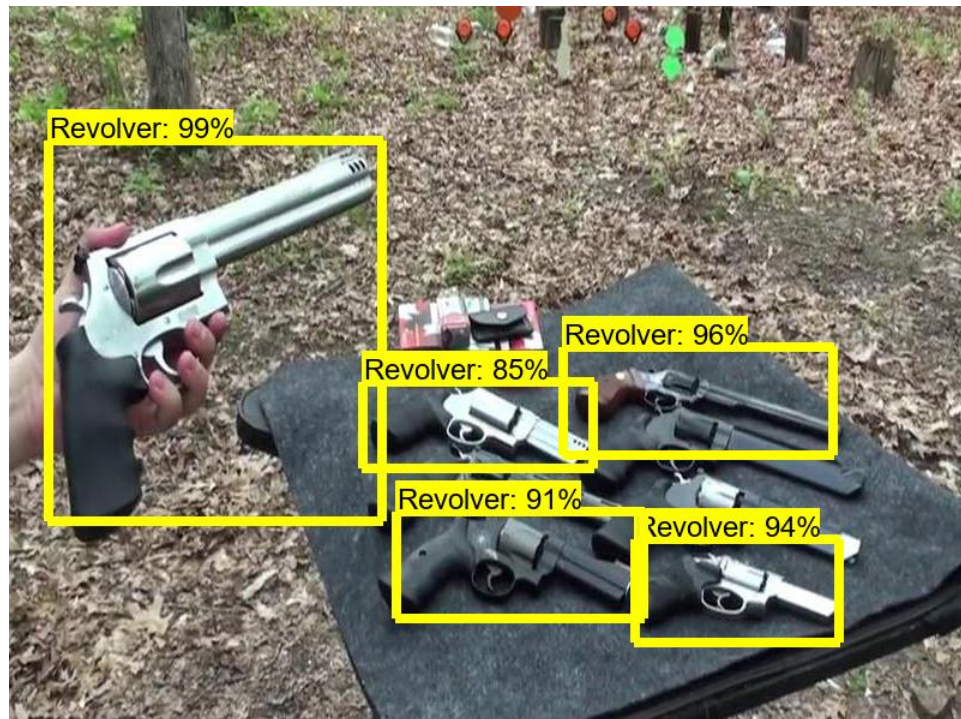


Figura 35: Regiones propuestas.

Clasificación y Detección:

Para entrenar los modelos es necesario realizar configuraciones previas, una de estas configuraciones es crear un mapa de etiquetas en el cual se coloca cada una de las clases junto con el valor numérico que la red neuronal de regiones más rápida (Faster RCNN) va a reconocer, el contenido de este archivo es el siguiente:

```
item {
  id: 1
  name: 'Pistola'
}
item {
  id: 2
  name: 'Revolver'
}
```

También es necesario descargar los modelos de detección de objetos que ofrece Tensorflow, donde los dos modelos que se utilizan en este proyecto son: `faster_rcnn_inception_v2_coco` y `faster_rcnn_resnet101_coco` los cuales necesitan de un archivo de configuración propio donde se define el número de clases, que en este caso es de 2 clases, también se debe colocar la ubicación de los archivos `.record` que previamente se habían construido y una vez esto terminado se pone a entrenar con el archivo `train.py`.

El modelo está totalmente entrenado cuando su función de pérdida total alcanza el valor de 0.1 como se muestra en la siguiente imagen:

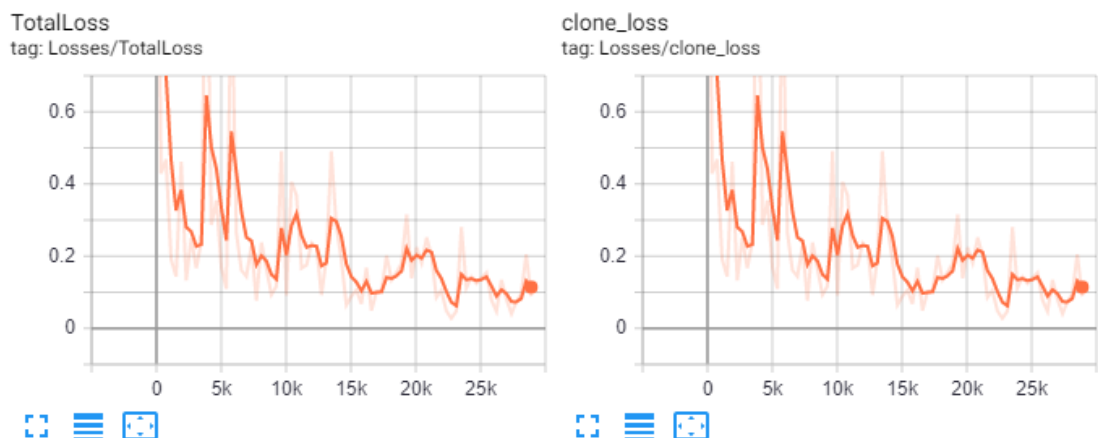


Figura 36: Función de pérdida total.

Cuando se termina el entrenamiento de la red neuronal Faster RCNN es necesario generar el grafico de inferencia el cual permite llamar al modelo entrenado en la etapa de prueba, estas pruebas se pueden realizar en imágenes o videos como se muestra a continuación en la figura 30 y figura 31.

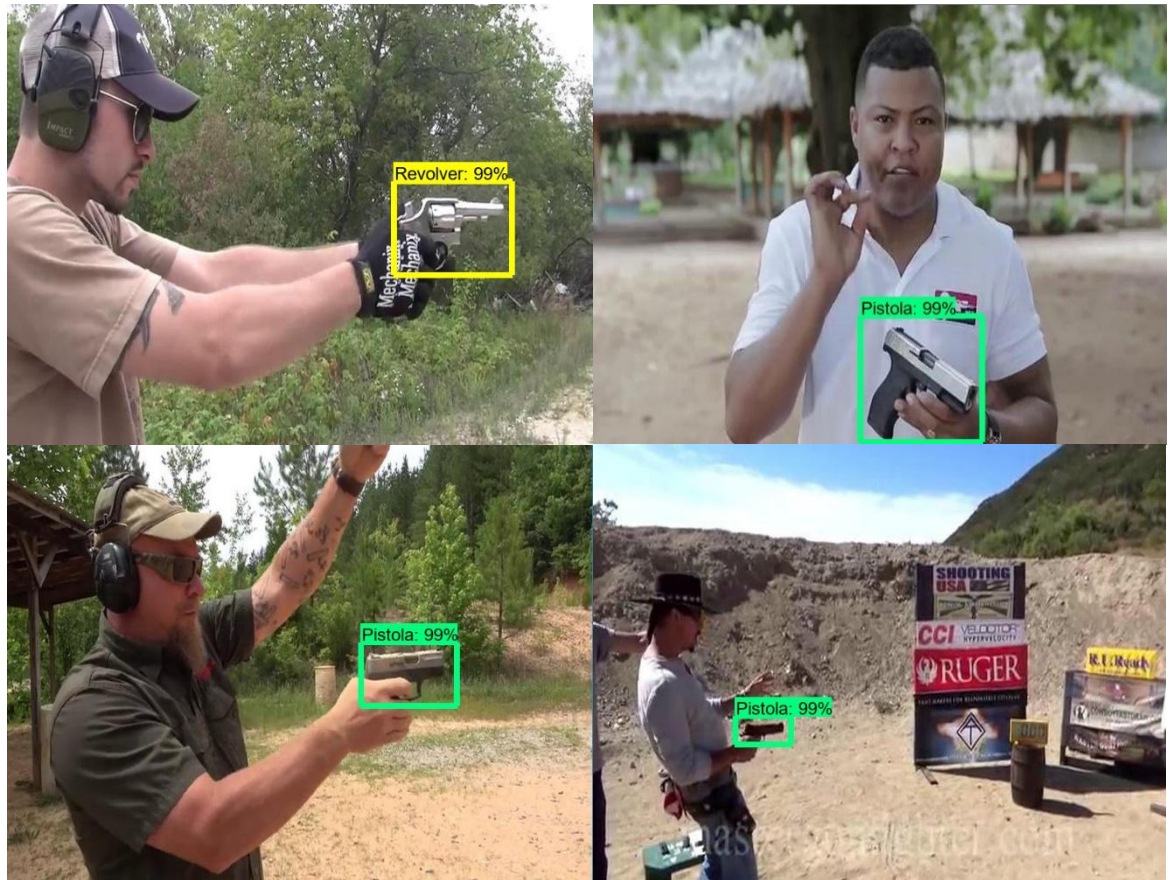


Figura 37: Detección de armas de mano 1.

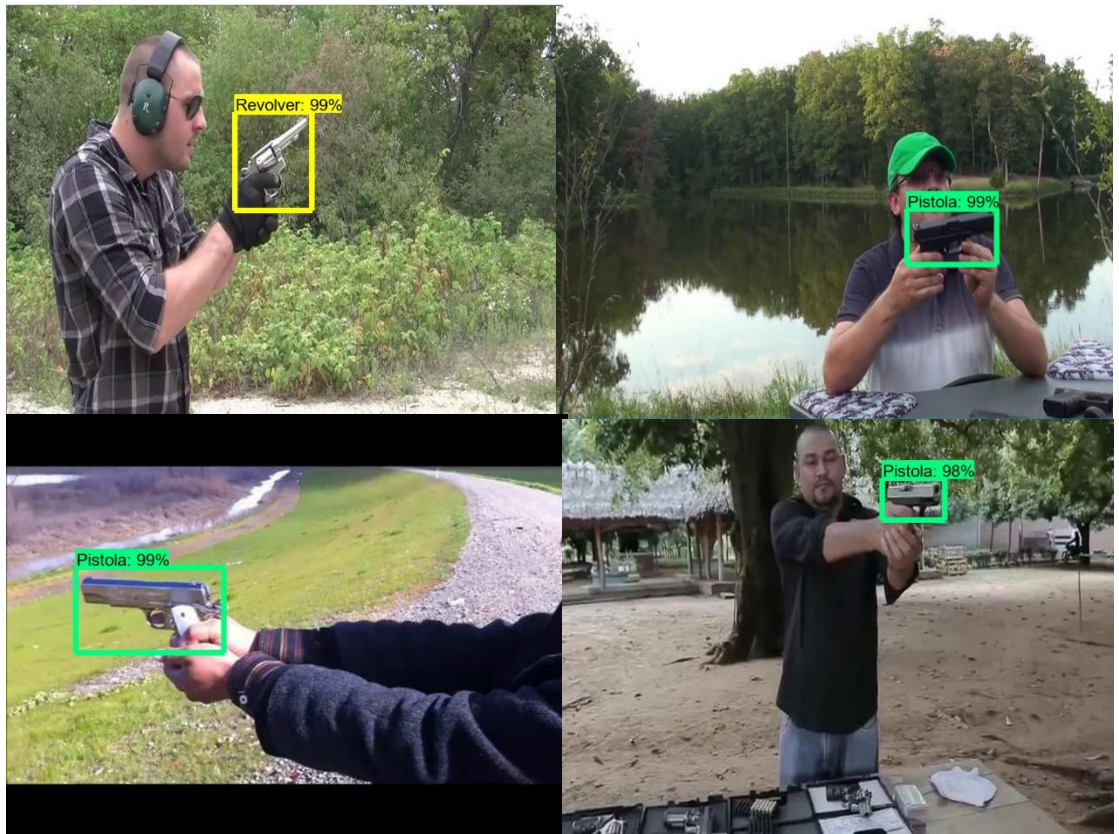


Figura 38: Detección de armas de mano 2.

Rendimiento:

Para medir el rendimiento del algoritmo de detección de armas de mano se utilizaron las métricas de detección COCO, las cuales muestran las medidas de mAP (mean Average Precision) y AR (Average Recall), los resultados obtenidos después de entrenar cada modelo, están reflejados en las tablas siguientes.

		PRECISION						Muestreo
Conjunto de datos	Faster RCNN	Promedio Precision/mAP	Precision (pequeño)	Precision (mediano)	Precision (grande)	Precision IoU=0.5	Precision IoU=0.75	70% - 30%
0,5 seg	Inception v2	0,61509	-1	0,432173	0,63475	0,97036	0,69223	
	Resnet 101	0,62714	-1	0,454428	0,64404	0,97495	0,71247	
1 seg	Inception v2	0,61278	-1	0,482663	0,62302	0,97421	0,69906	
	Resnet 101	0,6465	-1	0,537955	0,65395	0,98353	0,77168	

Tabla 5: Precision en el muestreo 70-30.

		RECALL						Muestreo
Conjunto de datos	Faster RCNN	Promedio Recall/mAP	Recall (pequeño)	Recall (mediano)	Recall (grande)	Recall AR@10	Recall AR@100	70% - 30%
0,5 seg	Inception v2	0,64496	-1	0,561187	0,719150	0,69886	0,70188	
0,5 seg	Resnet 101	0,64863	-1	0,585090	0,7217	0,7056	0,70673	
1 seg	Inception v2	0,63839	-1	0,593184	0,713810	0,70471	0,70738	
1 seg	Resnet 101	0,65296	-1	0,628010	0,72088	0,71346	0,7149	

Tabla 6: Recall en el muestreo 70-30.

		PRECISION						Muestreo
Conjunto de datos	Faster RCNN	Promedio Precision/mAP	Precision (pequeño)	Precision (mediano)	Precision (grande)	Precision IoU=0.5	Precision IoU=0.75	75% - 25%
0,5 seg	Inception v2	0,618180	0	0,439329	0,6381	0,97014	0,69391	
	Resnet 101	0,63168	0	0,459638	0,64949	0,97411	0,71819	

1 seg	Inception v2	0,63064	-1	0,582463	0,6342	0,98	0,7457
	Resnet 101	0,63999	-1	0,558510	0,64533	0,98123	0,76283

Tabla 7: Precision en el muestreo 75-25.

		RECALL						Muestreo
Conjunto de datos	Faster RCNN	Promedio Recall/mAP	Recall (pequeño)	Recall (mediano)	Recall (grande)	Recall AR@10	Recall AR@100	75 % - 25 %
0,5 seg	Inception v2	0,64615	0	0,549164	0,72022	0,700650	0,702060	
	Resnet 101	0,65379	0	0,561535	0,7255	0,70595	0,70759	
1 seg	Inception v2	0,6469	-1	0,655409	0,72133	0,71435	0,71767	
	Resnet 101	0,65461	-1	0,653216	0,72422	0,72045	0,721750	

Tabla 8: Recall en el muestreo 75-25.

		PRECISION						Muestreo
Conjunto de datos	Faster RCNN	Promedio Precision/mAP	Precision (pequeño)	Precision (mediano)	Precision (grande)	Precision IoU=0.5	Precision IoU=0.75	80 % - 20 %
0,5 seg	Inception v2	0,59954	-1	0,431806	0,61686	0,96555	0,684900	
	Resnet 101	0,633470	-1	0,461996	0,651100	0,97436	0,72935	
1 seg	Inception v2	0,62274	-1	0,470563	0,636990	0,98143	0,70125	
	Resnet 101	0,64826	-1	0,52960	0,65782	0,98747	0,78012	

Tabla 9: Precision en el muestreo 80-20.

		RECALL						Muestreo
Conjunto de datos	Faster RCNN	Promedio Recall/mAP	Recall (pequeño)	Recall (mediano)	Recall (grande)	Recall AR@10	Recall AR@100	80 % - 20 %
0,5 seg	Inception v2	0,62457	-1	0,561505	0,69879	0,68149	0,683900	
	Resnet 101	0,6484	-1	0,577016	0,72201	0,70523	0,70618	
1 seg	Inception v2	0,65229	-1	0,573272	0,72064	0,70807	0,70945	

	Resnet 101	0,67375	-1	0,644715	0,73053	0,724240	0,72441
--	------------	---------	----	----------	---------	----------	---------

Tabla 10: Recall en el muestreo 80-20.

4.1. INSTALACIONES Y EQUIPO REQUERIDO

Las herramientas utilizadas para la realización del proyecto son las que se describen a continuación:

- Equipos presentes en las salas de informática de la Universidad católica de Colombia.
- 1 computador portátil con: Procesador Intel Core I5 de octava generación 1.6Ghz, Memoria de 8 GB de RAM, Tarjeta gráfica de 2GB, y al menos 200 GB de espacio libre en el disco duro.
- 1 computador de mesa con: Procesador Intel Core i7 de 3.4Ghz, Memoria de 32GB, Tarjeta gráfica NVIDIA GeForce RTX 2080.
- 1 computador portátil con: Procesador Intel Core I5 de 2.3Ghz, Memoria de 8GB de RAM, sin tarjeta gráfica.
- Anaconda con Python 3.6
- Librerías para el experimento (Protobuf 3.0.0, Pillow 1.0, Lxml, Jupyter, notebook, Matplotlib, Tensorflow>=1.12.0, Tensorflow-gpu>=1.12.0, Cython, OpenCV, Contextlib2 y Cocoapi).
- API de detección de objetos de Tensorflow.
- Disco duro portátil de 1TB de almacenamiento.
- Cuda toolkit, cuDNN y Driver de NVIDIA.

4.2. ESTRATEGIAS DE COMUNICACIÓN Y DIVULGACIÓN

Las estrategias hasta el momento que se tienen para la comunicación y divulgación del experimento son las siguientes:

- La sustentación de proyecto de grado ante los compañeros de clase, jurados y asesor.
- Mediante una publicación en formato digital en el repositorio de la universidad católica de Colombia.

5. DISCUSIÓN DE RESULTADOS

En las tablas 7, 8, 9, 10, 11 y 12 se ilustran los resultados de las métricas aplicadas a los modelos de detección de armas de mano entrenados en el proyecto.

Primero en la tabla 7 se evidencia los resultados de la métrica mAP (mean Average Precision) en el muestreo de 70% y 30%, en esta medida el mayor porcentaje es de la red Faster RCNN Resnet 101 con el conjunto de datos de 1 segundo, su valor es de 0,6465 y significa que de 100 detecciones que realiza el modelo, 64 son predicciones correctas de armas de mano. Este valor es positivo para un detector de objetos, además esta medida es aún mejor en los valores de IoU=0.5 ya que se considera un margen de error del 50% y se logra obtener Precisión de 0,98 mientras que cuando el IoU=0.75 el margen de error es más pequeño y por consiguiente es más estricto al detectar un arma de mano y se evidencia con un Precisión de 0,77.

Por otro lado, la métrica de Precisión(pequeño) en todas las tablas desde la 6 a la 11 el resultado es -1 ya que el conjunto de datos con el que se trabaja no tiene armas de mano de área pequeña (para que un área se considere pequeña debe tener el área $< 32 \times 32$ p). En cambio, se evidencia que el detector de armas de mano funciona mejor en las áreas grande al tener los promedios de 0,63 y 0,64.

En la tabla 9 se evidencia el AR (Average Recall) del muestreo 70% y 30% y se vuelve a evidenciar que el mejor modelo fue el Faster Resnet 101 en el conjunto de datos de 1 segundo. Este valor de 0,65296 indica que de 10 armas de mano en una imagen el detector logra localizar a 6 de estas armas. También se evidencia que el modelo detecta mejor los objetos grandes que los medianos y que el conjunto de datos no tiene más de 10 armas de mano por imagen al ser tan parecidos los valores entre Recall AR@10 Y AR@100.

Estos resultados son similares a los de las tablas 7 y 8 y sus diferencias son por 1 centésima, por lo que se evidencia que este cambio de muestreo de 70% y 30% a 75% y 25% no marca una diferencia grande. Aun así, en las tablas 10 y 11 aparece el mejor modelo con un muestreo de 80% y 20%.

En las tablas 10 y 11 se obtiene el mejor modelo con Average Precision de 0,6482 y Average Recall de 0,6737. En este modelo de Faster Resnet 101 se evidencia mejores resultados al tener un conjunto de entrenamiento de porcentaje mayor (80%) y al ser entrenado con el conjunto de datos más diverso al contener frames de videos de YouTube y de bases de datos de películas.

También es importante recalcar que este modelo realizó pruebas por aparte con videos que no estaban dentro del conjunto de datos y se encontró muchos falsos positivos con objetos parecidos a un arma de mano.

Siendo más específicos con las tablas 7, 8, 9, 10, 11, 12 se puede identificar cual es el modelo que mejor detecta objetos medianos, objetos grandes y cuales fueron más precisos en las armas de mano por el tamaño de su área. Estos resultados se encuentran en la siguiente tabla.

Medida de rendimiento	Faster RCNN	Conjunto de datos	Muestreo
<i>Mejor identificador de objetos medianos</i>	Faster_inception_v2	1 segundo	75-25
<i>Mejor identificador de objetos grandes</i>	Faster_resnet_101	1 segundo	80-20
<i>Mejor predictor de objetos medianos</i>	Faster_inception_v2	1 segundo	75-25
<i>Mejor predictor de objetos grandes</i>	Faster_resnet_101	1 segundo	80-20

Tabla 11: Mejores resultados según el área del arma de mano.

En Colombia existen multinacionales que brindan soluciones en el campo de tecnología, que ponen al alcance de sus clientes servicios que permiten la detección de objetos basados en aprendizaje profundo, resaltando dos grandes proveedores de estos servicios, IBM con PowerAI Vision⁴⁴ y Microsoft con Cognitive Services – Computer Vision⁴⁵, estas empresas ponen al alcance servicios web que consumen recursos en la nube, pero representan costos elevados.

A diferencia de estas soluciones, el experimento se implemento bajo aplicativos open source que no generan ningún tipo de costo, se puede desplegar en plataformas Windows y Linux, además, por medio de una API puede ser desplegado en dispositivos móviles.

⁴⁴ IBM. IBM PowerAI Vision. En: *IBM* [en línea]. Disponible en: <https://www.ibm.com/cognitive/marketplace/ibm-powerai-vision/details> [consulta: 30/11/2019].

⁴⁵ Microsoft Azure. Computer Vision. En: *Microsoft Azure* [en línea]. Disponible en: <https://azure.microsoft.com/es-es/services/cognitive-services/computer-vision/>

El modelo bajo la arquitectura **Faster RCNN Resnet 101** presenta un moderno método de detección de armas de mano en video, ya que aplicando el enfoque de la red de regiones propuestas permitiendo una reducción en los tiempos y el costo computacional que requiere el algoritmo, por lo tanto, se puede ejecutar en tiempo real. Esto se evidencia con los resultados prometedores en las medidas de desempeño.

Con estos resultados podemos decir que es posible implementar en sistemas de cámaras con propósitos de vigilancia y control, específicamente para la tarea de detección de armas, como: estaciones de transporte públicos (estaciones Transmilenio), instituciones educativas (colegios y universidades), centros de monitoreo (centro de monitoreo de la Policía Metropolitana de Bogotá, centros de vigilancia privada), entre otros. Esto gracias a que el detector de armas de mano a mostrado un alto potencial, incluso en situaciones donde la calidad de video es baja o en videos con poca luminosidad, ha proporcionado resultados satisfactorios.

6. CONCLUSIONES

- De los modelos implementados en este experimento de detección automática de armas de mano, la Faster RCNN Resnet 101 fue la que tuvo mayor Precision y mayor Recall. Esto demuestra que al tener una arquitectura más robusta se mejora la detección.
- De los conjuntos de datos creados el que tuvo mejores valores en las métricas fue el conjunto de frames de 1 segundo con un muestreo de 80% y 20% al tener mayor variedad de frames recopilados de películas y de YouTube.
- La estrategia metodológica proporcionó una mayor claridad de los métodos y fases que componen un algoritmo de aprendizaje profundo, teniendo como arquitectura la red Faster RCNN que apoyo el experimento de detección automática de revólveres y pistolas.
- Con el desarrollo del algoritmo se evidencia que es necesario encontrar una mayor variedad de videos donde las armas de mano tengan más ángulos y se obtengan armas de área pequeña.
- Según las medidas de desempeño Average Recall y Average Precision, el algoritmo presenta una tasa aceptable para la detección de armas de mano, pero en ocasiones identifica muchos falsos positivos (predice que un objeto es un arma cuando realmente no lo es).

7. RECOMENDACIONES

- Para construir un conjunto de datos efectivo se debe contar con una gran cantidad de frames del objeto y estos deben estar en diferentes ángulos para mejorar el rendimiento del algoritmo, además debe estar balanceado en sus clases y tener el objeto en áreas pequeñas, medianas y grandes.
- Al desarrollar la metodología es recomendable hacer una amplia investigación de los métodos de aprendizaje profundo para optar por los adecuados y mejorar el desempeño del algoritmo.
- Para el entrenamiento del modelo es necesario tener un equipo con GPU de 8GB de capacidad para que el modelo realice mayor cantidad de pasos en menos tiempo y sea más eficiente en los tiempos de computo, ya que generalmente son muy largos y esto ayuda a reducirlos, además al probar el modelo favorece en la reproducción fluida del video.
- Se recomienda realizar como trabajo a futuro un método de detección automática en tiempo real e implementarlo en zonas de inseguridad de Bogotá.

8. ANEXOS

Anexo A: Manual de Configuración GPU

Para realizar la respectiva configuración de la GPU de cualquier proyecto que necesite Tensorflow-gpu es necesario realizar el siguiente manual.

- Buscar el Driver actualizado de la GPU que sea conveniente a la tarjeta gráfica que se tenga, en este caso al ser la GeForce RTX 2080 se trabaja con el controlador NVIDIA Studio Driver SD de la versión

Descarga de controladores NVIDIA

Búsqueda avanzada de controladores

Tipo de producto:	Sistema operativo:
GeForce	Windows 10 64-bit
Serie del producto:	Tipo de controlador de Windows:
GeForce RTX 20 Series	estándares ?
Familia del producto:	Idioma:
GeForce RTX 2080	Español (España)
	WHQL/Beta:
	Todos ?

BUSCAR

Nombre	Versión	Fecha de publicación
+ NVIDIA Studio Driver ^{SD}	431.86	4.9.2019

Figura 39: Descarga de controladores NVIDIA.

431.86.

- Instalar CUDA Toolkit 9.0 o versiones mayores, este es requisito para usar esta GPU, además automáticamente se instala Visual Studio y las dependencias. También es importante saber que depende del Driver que se haya descargado anteriormente la descarga del CUDA.

NVIDIA graphics driver R418 or newer	CUDA 10.1
NVIDIA graphics driver R418 or newer	CUDA 10.0
NVIDIA graphics driver R410 or newer	CUDA 9.2
NVIDIA graphics driver R418 or newer	CUDA 9

Tabla 12: Versiones de Controladores y CUDA.

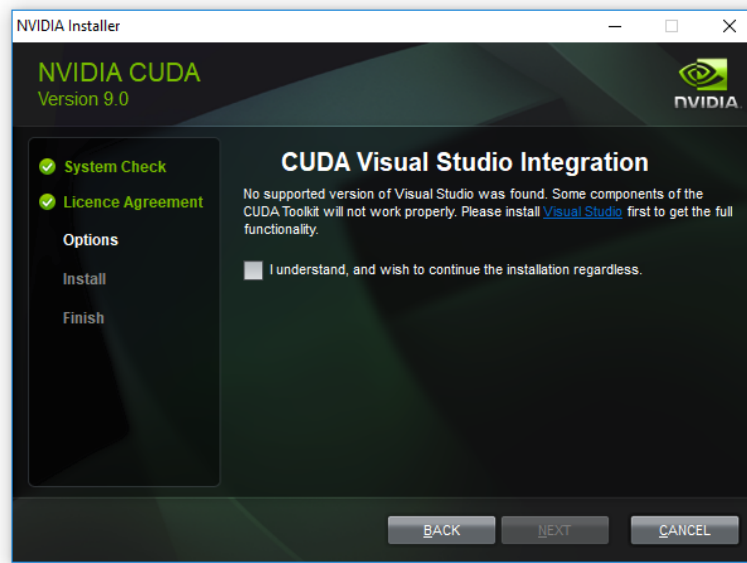


Figura 40: Instalación de CUDA Versión 9.0

- Instalar NVIDIA cuDNN, el cual depende del CUDA que se haya instalado, en este caso debe ser la versión 7.4.2. de cuDNN que se realiza copiando y pegando los archivos *cuda64_7.dll*, *cuda.h* y *cuda.lib* en las carpetas *bin*, *include* y *lib/x64* respectivamente.
- En realidad, antes de instalar Tensorflow se deben realizar los puntos anteriores y luego si elegir que versión de Tensorflow-gpu es la que más beneficia según la siguiente tabla.

Versión tensorflow-gpu	Versión de Python	CuDNN	CUDA
tensorflow-gpu-2.0.0	3.5-3.7	7.4	10
tensorflow-gpu-1.14.0	3.5-3.7	7.4	10
tensorflow-gpu-1.13.0	3.5-3.7	7.4	10
tensorflow-gpu-1.12.0	3.5-3.6	7	9

Tabla 13: Versiones compatibles para Tensorflow-gpu.

- Por último, para verificar que la instalación fue exitosa se utiliza Jupyter notebook, por medio del siguiente comando se verifica que Tensorflow es compatible con CUDA y que no existen errores.

- `tf.test.is_built_with_cuda()`

- Además, para validar que la GPU está disponible se debe ejecutar el siguiente comando:
 - `tf.test.is_gpu_available(cuda_only=False, min_cuda_compute_capability=None)`
- En los dos últimos pasos el resultado debe ser *True*.

Anexo B: Conjunto de Datos

Se realiza la entrega del conjunto de datos con el cual se trabajó el proyecto, que contiene videos y frames con armas de mano (revólveres y pistolas).

- Conjunto de videos
URL: <https://www.kaggle.com/bacriollo19/dataset-of-videos>
- Conjunto de frames y XML
 - 1 Frame x 1 segundo:
URL: <https://github.com/bacriollo19/Frames10>
 - 1 Frame x 0.5 segundos:
URL: <https://github.com/bacriollo19/Frames05>

Anexo C: Repositorio de los Códigos del Proyecto

Se entrega los códigos ejecutados en las etapas de pre procesamiento, muestreo, entramiento y prueba.

URL: <https://github.com/bacriollo19/Codigo>

9. BIBLIOGRAFÍA

Murillo Mojica. Oscar. Seguridad, el tema del fogueo entre Policía y ciudadanía. [en línea]. En: *El tiempo* [Citado el 29 de junio, 2019] Disponible en internet: <<https://www.eltiempo.com/bogota/como-esta-la-seguridad-en-bogota-segun-los-ciudadanos-y-la-policia-348276>>

Secretaría Distrital de Seguridad, Convivencia y Justicia, 2018. Bogotá ya Cuenta con más de 1.600 Cámaras De Vigilancia. [en línea]. En: *Secretaría Distrital de Seguridad, Convivencia y Justicia*. [Citado el 20 de noviembre, 2019]. Disponible en internet: <<https://scj.gov.co/es/noticias/bogot%C3%A1-cuenta-m%C3%A1s-1600-c%C3%A1maras-vigilancia>>

Zhao, Zhong-Qiu & Zheng, Peng & Xu, Shou-Tao & Wu, Xindong, 2019. Object Detection with Deep Learning: A Review. *IEEE. Transactions on Neural Networks and Learning Systems*. [Citado el 16 marzo, 2019]; pp. 1-21.

El Espectador. Preocupa incremento de hurtos violentos en Bogotá. [en línea]. En: *El Espectador*. [Citado el 04 de julio, 2019]. Disponible en internet: <<https://www.elespectador.com/noticias/bogota/preocupa-incremento-de-hurtos-violentos-en-bogota-articulo-846973>>

Velásquez Loaiza, Melissa. Bogotá, ¿cada vez más insegura?: estas son las cifras de los crímenes en la capital colombiana. [en línea]. En: *CNN*. [Citado el 14 de marzo, 2019] Disponible en internet: <<https://cnnespanol.cnn.com/2018/02/06/bogota-cada-vez-mas-insegura-estas-son-las-cifras-de-los-crimenes-en-la-capital-colombiana/>>

El tiempo. En Bogotá sube uso de armas de fuego en hurtos y baja en homicidios. [en línea]. En: *El tiempo*. [Citado el 14 de marzo, 2019]. Disponible en internet: <<https://www.eltiempo.com/bogota/entre-enero-y-febrero-subio-el-uso-de-armas-de-fuego-en-atracos-en-bogota-208274>>

El Espectador. Advierten uso de armas largas para perpetrar delitos en Bogotá. [en línea]. En: *El Espectador*. [Citado el 14 de marzo, 2019]. Disponible en internet: <<https://www.elespectador.com/noticias/bogota/advierten-creciente-uso-de-armas-largas-para-perpetrar-delitos-en-bogota-articulo-869118>>

El tiempo. ¿Qué tanto impactan en la seguridad las cámaras de vigilancia? [en línea]. En: *El tiempo*. [Citado el 14 de marzo, 2019]. Disponible en internet: <<https://www.eltiempo.com/bogota/impacto-de-las-camaras-de-vigilancia-en-la-seguridad-de-bogota-170988>>

SIE. El importante rol del sistema de CCTV en la Seguridad Privada. [en línea]. En: *SIE*. [Citado el 14 de marzo, 2019]. Disponible en internet:

<<https://siesa.com.ar/el-importante-rol-del-sistema-de-cctv-en-la-seguridad-privada/>>

Rodríguez Gómez, José David. Refuerzan videovigilancia en Bogotá con nuevos centros de monitoreo. [en línea]. [Citado el 30 de marzo, 2019] En: *RCN radio*. Disponible en internet: <https://www.rcnradio.com/bogota/refuerzan-videovigilancia-en-bogota-con-nuevos-centros-de-monitoreo>

Kumar Tiwari, Rohit. K. Verma, Gyanendra, 2015. A Computer Vision based Framework for Visual Gun Detection using Harris Interest Point Detector. *ScienceDirect*. Procedia Computer Science. Vol.54, (06, 2015); pp. 703-712.

MJ, Bahmani. AI vs Machine Learning vs Deep Learning. [En línea]. En: *Medium*. [Citado el 23 de abril, 2019]. Disponible en internet: <<https://medium.com/datadriveninvestor/ai-vs-machine-learning-vs-deep-learning-ba3b3c58c32>>

ExpertSystem. What is Machine Learning? A definition. [en línea]. En: ExpertSystem. [Citado el 10 de julio, 2019]. Disponible en internet: <<https://www.expertsystem.com/machine-learning-definition/>>

Colciencias. ¿qué es el aprendizaje profundo? [en línea]. En: Colciencias. [Citado el 21 de marzo, 2019]. Disponible en internet: <<http://www.todoesciencia.gov.co/aprende/innovacion/que-es-el-aprendizaje-profundo>>

BBVA. ¿Qué es la inteligencia artificial? [en línea]. En: *BBVA*. [Citado el 21 de marzo, 2019]. Disponible en internet: <<https://www.bbva.com/es/que-es-la-inteligencia-artificial-2/>>

Shaoqing, Ren. Kaiming, He. Ross, Girshick. Jian, Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. En: Cornell University (arXiv). Volume 3. 2015. Pages 1-14.

Gobierno de España – Ministerio de interior. Clasificación de armas. [En línea]. En: Gobierno de *España – Ministerio de interior*. [Citado el 21 de marzo, 2019]. Disponible en internet: <<http://www.interior.gob.es/web/servicios-al-ciudadano/seguridad/armas-y-explosivos/clasificacion-de-armas>>

SkyMind. Datasets and Machine Learning. [En línea]. En: *SkyMind*. [Citado el 11 de julio, 2019] Disponible en internet: <<https://skymind.ai/wiki/datasets-ml>>

GeeksforGeeks. Data Preprocessing for Machine learning in Python. [En línea]. En: *GeeksforGeeks*. [Citado el 11 de julio, 2019]. Disponible en internet: <<https://www.geeksforgeeks.org/data-preprocessing-machine-learning-python/>>

Virginia Tech. Better Image Scaling. [En línea]. En: VTCourses. [Citado el 11 de julio, 2019] Disponible en internet: <<http://courses.cs.vt.edu/~masc1044/L17-Rotation/ScalingNN.html>>

Katharopoulos, Angelos. Fleuret, François. Not All Samples Are Created Equal: Deep Learning with Importance Sampling. En: Arxiv. Volume 2. 2018. Pages 1-13.

Towards Data Science, 2019. Sampling Techniques. [En línea]. En: Towards Data Science. En: *Towards Data Science*. [Citado el 11 de julio, 2019]. Disponible en internet: <<https://towardsdatascience.com/sampling-techniques-a4e34111d808>>

DeepAI. Sampling Techniques. [En línea]. En: DeepAI. [Citado el 11 de julio, 2019]. Disponible en internet: <<https://deepai.org/machine-learning-glossary-and-terms/feature-extraction>>

Liu, Han Yu Feature Extraction and Image Recognition with Convolutional Neural Networks. En: IOP Conf. Series: Journal of Physics: Conf. Series 1087. 2018. Pages 1-7.

Tryo Labs. Faster R-CNN: Down the rabbit hole of modern object detection. [En línea]. En: *Tryo Labs*. [Citado el 11 de julio, 2019]. Disponible en internet: <<https://tryolabs.com/blog/2018/01/18/faster-r-cnn-down-the-rabbit-hole-of-modern-object-detection/>>

Towards data science. Activation Functions in Neural Networks. [En línea]. En: *Towards data science*. [Citado el 11 de julio, 2019]. Disponible en internet: <<https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>>

Geeksforgeeks. CNN | Introduction to Pooling Layer. [En línea]. En: Geeksforgeeks. [Citado el 23 de septiembre, 2019] Disponible en internet: <<https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/>>

Towards Data Science. Review: Faster R-CNN (Object Detection). [En línea]. En: *Towards Data Science*. [Citado el 11 de julio, 2019]. Disponible en internet: <<https://towardsdatascience.com/review-faster-r-cnn-object-detection-f5685cb30202>>

Mathworks. ¿Qué es el reconocimiento de objetos? [En línea]. En: *Mathworks*. [Citado el 23 de marzo, 2019]. Disponible en internet: <<https://la.mathworks.com/solutions/deep-learning/object-recognition.html>>

R. Caruana and A. Niculescu-Mizil, "Data mining in metric space: an empirical analysis of supervised learning performance criteria", in Proc. of the 10th ACM

SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04), New York, NY, USA, ACM 2004, pp. 69-78.

Sunasra, Mohammed, 2017. Performance Metrics for Classification problems in Machine Learning. [En línea]. En: *Medium*. [Citado el 10 de octubre, 2019]. Disponible en internet: <<https://medium.com/thalus-ai/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b>>

PyImageSearch. Intersection over Union (IoU) for object detection. [En línea]. En: *PyImageSearch*. [Citado el 10 de octubre, 2019]. Disponible en internet: <<https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>>

Greg, Flittonn. Toby, P. Breckon. Najla, Megherbi. A comparison of 3D interest point descriptors with application to airport baggage object detection in complex CT imagery. En: *EI Sevier*. [Citado el 10 de octubre, 2019]. Volume 46. Sept 2013. Pages 2420-2436.

Gesick, Richard. Saritac, Caner. Hung, Chih-Cheng. Automatic Image Analysis Process for the Detection of Concealed Weapons. En: Standford. ACM International Conference Proceeding Series. [Citado el 10 de octubre, 2019].

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro perona, Deva Ramanan, Piotr Dollár, C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. En: SpringerLink. 2014. Pages 740-755. [Citado el 10 de octubre, 2019].

Tiwari, Rohit Kumar. K. Verma, Gyanendra. A Computer Vision based Framework for Visual Gun Detection Using Harris Interest Point Detector. En: ScienceDirect. Volume 54. 2015. Pages 703-712. [Citado el 10 de octubre, 2019].

Fuente: Huang, Jonathan. Vivek, Rathod. Chen, Sun. Menglong, Zhu. Anoop, Korattikara. Fathi, Alireza. Ian, Fischer. Zbigniew, Wojna. Yang, Song. Guadarrama, Sergio. Murphy, Kevin. ArXiv: 1611.10012v3, 2017. Review: Speed/accuracy trade-offs for modern convolutional object detectors. [en línea]. [Citado el 10 de octubre, 2019].

Olmos, Roberto. Tabik, Siham. Herrera, Francisco. [En línea]. ArXiv: 1702.05147v1, 2017. Automatic Handgun Detection Alarm in Videos Using Deep Learning. [Citado el 10 de octubre, 2019]. Disponible en internet: <<https://arxiv.org/abs/1702.05147>>

Ren, Shaoqing. He, Kaiming. Girshick, Ross. Sun, Jian. [En línea]. ArXiv: 1506.01497v3, 2016. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. [Citado el 10 de octubre, 2019] Disponible en internet: <<https://arxiv.org/pdf/1506.01497.pdf>>

GITHUB TENSORFLOW. Tensorflow Object Detection API. [en línea]. En: *GITHUB*. [Citado el 10 de octubre de, 2019]. Disponible en internet: <https://github.com/tensorflow/models/tree/master/research/object_detection>