UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE CUENCA

CARRERA DE INGENIERÍA ELECTRÓNICA

Trabajo de Titulación previo a la obtención
del título de Ingeniero Electrónico

PROYECTO TÉCNICO CON ENFOQUE GENERAL:

"DESARROLLO DE UN SISTEMA DE DETECCIÓN DE ARMAS DE FUEGO CORTAS EN EL MONITOREO DE VIDEOS DE CÁMARAS DE SEGURIDAD"

AUTOR:

DAVID ORLANDO ROMERO MOGROVEJO

TUTOR:

ING. CHRISTIAN RAÚL SALAMEA PALACIOS Ph.D

CUENCA – ECUADOR

2018

CESIÓN DE DERECHOS DE AUTOR

Yo, David Orlando Romero Mogrovejo con documento de identificación N° 0301863916 manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de titulación: **"DESARROLLO DE UN SISTEMA DE DETECCIÓN DE ARMAS DE FUEGO CORTAS EN EL MONITOREO DE VIDEOS DE CÁMARAS DE SEGURIDAD"**, mismo que ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

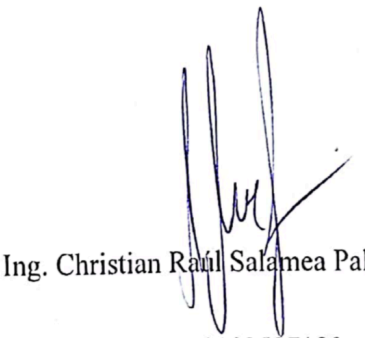
Cuenca, 20 de diciembre del 2018

David Orlando Romero Mogrovejo
CI: 0301863916
AUTOR

CERTIFICACIÓN

Yo declaro que bajo mi **tutoría** fue desarrollado el trabajo de titulación: **"DESARROLLO DE UN SISTEMA DE DETECCIÓN DE ARMAS DE FUEGO CORTAS EN EL MONITOREO DE VIDEOS DE CÁMARAS DE SEGURIDAD"**, realizado por David Orlando Romero Mogrovejo, obteniendo el **Proyecto Técnico con enfoque general** que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana.

Cuenca, 20 de Diciembre del 2018



Ing. Christian Raúl Salamea Palacios Ph.D

CI: 0102537180

TUTOR DEL TRABAJO DE TITULACIÓN

DECLARATORIA DE RESPONSABILIDAD

Yo, David Orlando Romero Mogrovejo con número de cédula CI. 0301863916, autor del trabajo de titulación: **"DESARROLLO DE UN SISTEMA DE DETECCIÓN DE ARMAS DE FUEGO CORTAS EN EL MONITOREO DE VIDEOS DE CÁMARAS DE SEGURIDAD"** certifico que el total contenido del **Proyecto Técnico con enfoque general,** es de mi exclusiva responsabilidad y autoría.

Cuenca, 20 de diciembre del 2018

David Orlando Romero Mogrovejo
CI: 0301863916
AUTOR

AGRADECIMIENTOS

Quiero agradecer a mis padres y familia por brindarme su apoyo en cada momento, por ser un pilar fundamental durante mi carrera universitaria y por todas sus enseñanzas a lo largo de estos años. Al Ing. Christian Salamea, quien con su paciencia, sabiduría y amistad ha sabido orientar y motivar el desarrollo de este proyecto.

David Orlando Romero Mogrovejo

DEDICATORIAS

El presente proyecto va dedicado a mis padres Orlando y Ruth, por todo su amor, apoyo y comprensión durante mi formación académica, a mi hermana Gabriela y mis abuelos Nube y David por ser un pilar fundamental en todo este camino y por la confianza depositada en mi y a todos mis familiares y amigos que estuvieron presentes durante esta etapa de mi vida.

David Orlando Romero Mogrovejo

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

Los sistemas de monitoreo actuales basados en cámaras de seguridad se han convertido en un requisito operacional considerado esencial en el ámbito de la seguridad y protección de entidades comerciales, su funcionamiento está basado en la observación humana mediante el uso de cámaras de seguridad. El operador al realizar una detección en el lugar de monitoreo notifica lo sucedido a las entidades encargadas de la seguridad respectiva para que se proceda a las acciones correspondientes según el caso. Actualmente la estructura de funcionamiento de los sistemas de monitoreo presentan una considerable ineficiencia, debido al tiempo de detección y de reacción, además de las múltiples limitaciones humanas que se presentan en el monitoreo, este supone una enorme carga y dificultad para las personas, además que su costo es demasiado alto para ser factible para muchas entidades comerciales.

Las redes neuronales convolucionales han proporcionado un gran avance en visión artificial en los últimos años, en tareas como la detección de objetos en tiempo real y en clasificación de imágenes, siendo utilizada en muchas aplicaciones como la conducción autónoma. En este proyecto se desarrolló un sistema de detección de armas de fuego cortas en videos de cámaras de seguridad, mediante el uso de redes neuronales convolucionales. Para el desarrollo del sistema se creó una base de datos a partir de la recopilación de imágenes y frames de videos en donde esté presente una arma de fuego corta, así como imágenes en donde no esté presente la misma, se evaluó el modelo implementado mediante diferentes métricas de evaluación, adicionalmente se realizó una integración del modelo en un prototipo de software. Los resultados obtenidos evidencian que el sistema de detección posee valores de recall y precisión altos en sus detecciones, además en las pruebas realizadas presenta una operación óptima en videos en los cuales no fue entrenado el sistema, y en donde existen ambientes muy complejos con múltiples objetos y personas en el entorno.

INTRODUCCIÓN

Los sistemas cerrados de televisión (CCTV) son sistemas compuestos por una o más cámaras de vigilancia conectadas a uno o más monitores de video [1], buscan, entre otras cosas, prevenir situaciones de peligro como intrusiones o robos a mano armada, siendo en la mayoría de los casos utilizada en entidades comerciales. Actualmente estos sistemas de seguridad son sistemas pasivos que no proporcionan la ayuda necesaria en el momento en que ocurre una situación de peligro, como un asalto con arma de fuego, debido a que solo registran lo sucedido para su uso posterior. Este tipo de sistemas se han convertido en algo similar a una persona a la cual tenemos para nuestra protección pero esta solo observa lo que nos sucede pero no realiza ninguna acción en base a ello.

Una forma de reducir y evitar este tipo criminalidad es mediante la detección temprana de las armas de fuego cuando ocurren situaciones de peligro como un robo, lo cual proporcionaría un mejor tiempo de reacción por parte de las entidades encargadas de la seguridad. Un robo dura aproximadamente 3 minutos o menos, lo cual es una de las razones de que la mayoría de los robos que se cometen no se resuelven, debido al tiempo de reacción. El uso del monitoreo en vivo es una opción poco viable para la mayoría de negocios debido a su elevado costo y por las múltiples limitaciones que posee, como tiempos de detección, además de limitaciones humanas en el aspecto del monitoreo.

Una detección instantánea permitiría notificar a las autoridades en el mismo momento en el que la situación está sucediendo, no luego como se realiza actualmente, así como activar sistemas que posean una función disuasiva hacia las personas que cometen estos tipos de hechos. Si bien el presente proyecto se limita a la detección de armas de fuego en videos, el mismo se podría aplicar para la detección en tiempo real de cualquier tipo de arma.

ANTECEDENTES DEL PROBLEMA DE ESTUDIO

La implementación de cámaras de vigilancia en entidades comerciales y en medios de transporte público, buscan registrar lo sucedido cuando se dan situaciones de peligro como robos, sin embargo en el caso de las entidades comerciales en la mayoría de los casos no solicitan la implementación de un monitoreo en vivo a las empresas de seguridad debido en su mayor parte al costo y a la baja eficiencia de esos sistemas, por lo cual en la mayoría de los casos se implementa solamente cámaras. En los medios de transporte público en el caso del Ecuador las cámaras implementadas notifican a las entidades encargadas de la seguridad solamente cuando el conductor presiona un botón de emergencia, para lo cual los conductores deben arriesgar sus vidas para realizar esa acción en el momento de un robo o asalto. Gran parte de estos no se resuelven debido a que las llamadas a las entidades encargadas de la seguridad se las hace luego de que el suceso ha pasado, debido a que se implementan sistemas pasivos que no proporcionan una ayuda en el momento adecuado.

La solución al problema de la detección de situaciones de peligro en las que participan armas de fuego desde una perspectiva de la detección por medio de la visión artificial en las que no participan personas, ha sido muy poco abarcado. Los trabajos que se han enfocado en este problema aplican diferentes métodos, como enfoques de segmentación basado en colores y detectores de puntos de interés “SURF”, características “SIFT” o detector de puntos de interés “Harris”, combinados con algoritmos de agrupación “K-means”, así como “Support vector machines” [2,3,4]. En estos métodos se utilizaron imágenes en donde el arma ocupa todo el espacio existente es decir no fueron realizados con imágenes de situaciones reales las cuales son mucho más complejas. Los autores en [5,6] abarcan la detección de armas de fuego aplicando Machine Learning, así como Redes Neuronales Convolucionales respectivamente, en este último se aplica aprendizaje de transferencia más conocido como “Transfer Learning”, de esta manera se utiliza pequeñas bases de datos para el entrenamiento del modelo, sin embargo se debe utilizar modelos entrenados en tareas similares a la nueva tarea a realizar, para que de esta manera se ocupe correctamente el conocimiento previamente adquirido y aplicarla a la nueva tarea, además todos estos fueron entrenados con bases de datos de situaciones no reales, en donde se presentan ambientes mucho más complejos.

JUSTIFICACIÓN (IMPORTANCIA Y ALCANCES)

En los últimos años en múltiples países se ha dado un incremento de las tasas de criminalidad en los atracos donde se dio uso a un arma de fuego, principalmente en países en donde es legal su posesión, como Estados Unidos. En donde en el año 2017 se produjo mas 200.000 robos con armas de fuego [7], las tres cuartas partes de las lesiones no fatales por arma de fuego son causadas en asaltos en este país, presentándose aproximadamente un número mayor a los 66.000 casos por año [8]. El tipo de crimen en donde se da el uso de un arma de fuego incide en la tasa de homicidios, según las estadísticas de la Oficina de las Naciones Unidas contra las Drogas y el Crimen revelan que los homicidios cometidos durante actos criminales como robos y asaltos son de aproximadamente el 5% en promedio de todos los homicidios en América, Europa y Oceanía cada año [9].

En [10] se presenta que en el Ecuador en el año 2017 se presentó una tendencia creciente en la cantidad de robos a unidades económicas, excepto en el quinto, sexto y doceavo mes, en donde la tendencia de robos descendió. En total se cuentan más 5400 robos en el año y aproximadamente 450 cada mes [10]. La seguridad en las entidades comerciales es un problema que los usuarios tienen que enfrentar actualmente, debido a esto se han tenido que adoptar medidas de seguridad para proteger los bienes de las entidades comerciales y de igual manera de las personas que se encuentran en el lugar, quienes pueden ser perjudicados por hechos delictivos acontecidos en este ámbito. En las entidades comerciales los sistemas de seguridad constan en su mayoría de cámaras, las cuales son elementos que no proporcionan una ayuda en el momento que se requiere. Los sistemas de monitoreo existentes en donde personas observan remotamente son costosos lo cual no es práctico para la mayoría de negocios, además de que su eficiencia es baja debido a que presentan muchas limitaciones humanas. En la mayoría de los casos, múltiples operadores observan cientos de monitores al mismo tiempo, considerando que un robo dura muy poco tiempo, el operador debe ser muy exacto para observar el monitor adecuado en el momento adecuado. Según [4] un estudio publicado en “Security Oz Magazine” un operador a menudo se perderá hasta el 45% de la actividad de la pantalla después de 12 minutos de monitoreo continuo y la tasa de fallo aumenta hasta el 95% después de 22 minutos. Debido a estas consideraciones, este proyecto tiene como objetivo desarrollar un sistema de seguridad de detección de armas de fuego cortas en videos de cámaras de seguridad, si bien el

presente proyecto se limita a la detección en videos, el mismo se podría implementar para detecciones en tiempo real, el cual tendría la capacidad de monitorear cientos de cámaras de seguridad de bancos, tiendas, unidades de transporte entre otros, con lo cual se podría detectar un crimen en progreso y comunicar la detección de armas de fuego en el mismo momento en el que la situación está ocurriendo, no cuando la misma ya ha culminado, esto permitiría la ganancia de tiempo, para que de esta manera las personas encargadas de la seguridad actúen en el mismo momento en el cual la situación de peligro esté sucediendo y no luego, o se podrían activar sistemas que proporcionen una función disuasiva.

El proyecto beneficiará a las unidades económicas en las que constan negocios, entidades financieras, empresas, entre otros, proporcionándoles un sistema de seguridad que pueda detectar armas de fuego cortas cuando se produzca un robo o asalto, con lo cual se podrá dar una ayuda inmediata cuando la situación esté ocurriendo, siendo aplicable no solo para nuestro medio, sino que se puede implementar en cualquier país y lugar en donde se lleve a cabo un monitoreo de cámaras de seguridad.

OBJETIVOS

OBJETIVO GENERAL

- Desarrollar un sistema de detección de armas de fuego cortas en el monitoreo de videos de cámaras de seguridad.

OBJETIVOS ESPECÍFICO

- Crear una base de datos de imágenes de armas de fuego cortas para el entrenamiento de la red neuronal convolucional.
- Diseñar una red neuronal convolucional para el reconocimiento de imágenes de armas de fuego.
- Integrar el modelo entrenado para la detección de armas de fuego en un prototipo de software.
- Evaluar el sistema implementado por medio de las métricas de equal error rate y el costo de detección promedio.

CAPÍTULO 1: ESTADO DEL ARTE

En este capítulo se realiza una descripción del sistema de seguridad electrónico destinado actualmente a enfrentar la detección de situaciones de peligro en las que participa un arma de fuego , el tipo de red neuronal utilizada para el desarrollo del sistema de detección, el sistema de detección y localización de objetos “Yolo”, así como la biblioteca y entorno utilizado para la implementación del sistema de detección.

1.1 DETECCIÓN DE ARMAS DE FUEGO

Los sistemas de seguridad que enfrentan actualmente el trabajo de detección de situaciones de peligro en donde aparecen armas de fuego o cualquier tipo de arma son la seguridad de tipo física que implementa guardias de seguridad y los sistemas de seguridad electrónicos, siendo estos los más comúnmente implementados en lugares como negocios, sistemas de transporte entre otros.

1.1.1 SISTEMAS DE SEGURIDAD ELECTRÓNICOS

Un sistema de seguridad electrónico es la interconexión de recursos, redes y dispositivos cuyo objetivo es proteger la integridad de las personas y su entorno, previniéndolas de posibles peligros y presiones externas. Los sistemas de seguridad electrónicos están orientados a disuadir, prevenir y reducir la ocurrencia de un hecho delictivo, lo que resulta en minimizar las probabilidades de pérdida en el establecimiento comercial [11]. El tipo de seguridad electrónica destinada a detectar hechos en los cuales aparezcan armas de fuego con los sistemas CCTV.

- *Sistemas CCTV*: CCTV (Closed Circuit Television) es un sistema de video vigilancia, que tiene como objetivo la supervisión, el control y el eventual registro de la actividad física dentro de un local, predio o ambiente en general. El sistema puede estar compuesto de una o varias cámaras de vigilancia, conectadas a uno o más monitores o televisores, los cuales reproducen las imágenes capturadas por las cámaras de seguridad [12].

Este tipo sistema de seguridad electrónica el cual actualmente es el más usado para prevenir el tipo situaciones de peligro como un robo en donde aparecen armas de fuego poseen múltiples desventajas entre las más destacables están su alto costo, mantenimiento complejo y la necesidad de operadores que vigilen todo el tiempo la actividad de la cámara [13].

1.2 DETECCIÓN DE OBJETOS MEDIANTE EL APRENDIZAJE PROFUNDO

La tecnología basada en el aprendizaje profundo ha tenido un gran avance en los últimos años obteniendo grandes resultados en tareas como la clasificación de imágenes y la detección de objetos en gran parte debido a la disponibilidad de grandes cantidades de datos y poder computacional. La detección de objetos es el procedimiento para determinar la instancia de la clase a la que pertenece un objeto y estimar la ubicación del mismo mediante la salida de un cuadro delimitador alrededor del objeto, siendo las redes neuronales convolucionales o CNN’s el principal método usado para este tipo de tareas [14].

1.3 REDES NEURONALES CONVOLUCIONALES

Una red neuronal convolucional (CNN) es una clase de red neuronal artificial que está inspirada en la corteza visual del cerebro la cual consiste en capas de células simples y complejas [15]. Este tipo de red neuronal artificial permite realizar tareas de clasificación directamente a partir de imágenes, textos o sonidos, siendo especialmente útiles para localizar patrones en imágenes, con el objetivo de reconocer objetos, caras y escenas, aprendiendo directamente a partir de los datos de las imágenes utilizando patrones para clasificarlas [16].

Las CNN’s aprenden extrayendo de las imágenes una jerarquía de características no lineales que crecen en complejidad a través de sus capas, comenzando en sus primeras capas en donde se detecta bordes, sombras hasta llegar a

capas más profundas en donde se detectan características mucho más complejas como rostros, (Figura 1.1). Estas características extraídas son usadas por sus capas finales para realizar tareas de clasificación o regresión. [17].

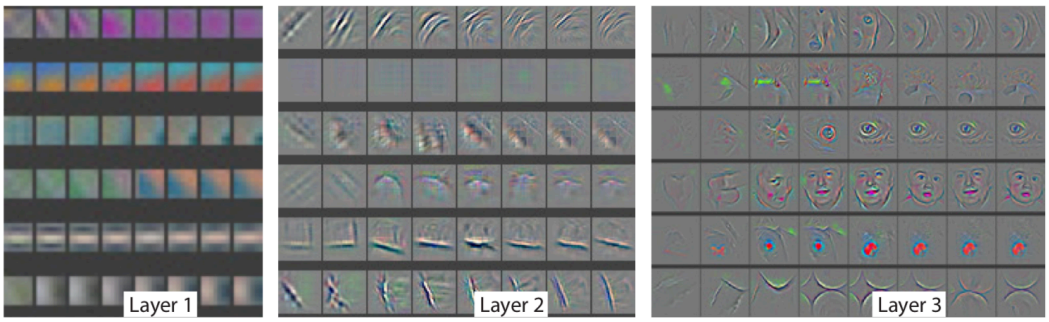


Figura 1.1 *Extracción de características jerárquicas*
Fuente: *[18]*

Las CNN's reciben como ingreso imágenes, las cuales son matrices tridimensionales en donde el tamaño de las dos primeras dimensiones corresponde a la longitud y ancho de las imágenes en pixeles y la tercera dimensión corresponde al número de canales que posee la imagen, si la misma es una imagen RBG esta posee 3 canales y si una imagen es a blanco y negro esta posee 1 canal. Estas matrices ingresan a la red que posee una estructura conformada por capas de convolución, capas de pooling y una red de neuronas totalmente conectadas. Esta estructura se presenta en la Figura 1.2 [19].

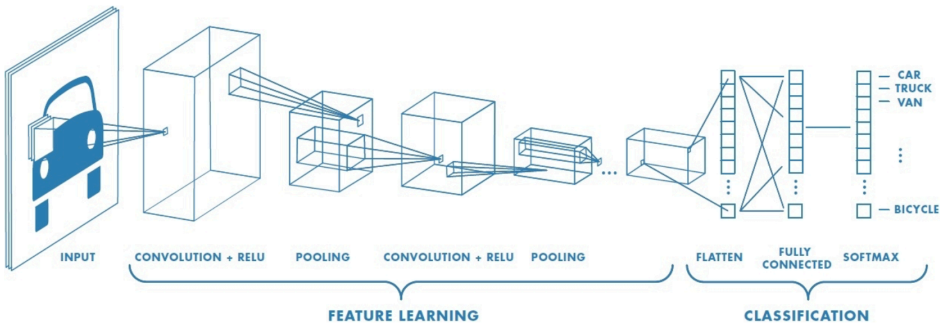


Figura 1.2 *Estructura de una Red Neuronal Convolucional*
Fuente: *[16]*

1.3.1 CAPA CONVOLUCIONAL

Las capas convolucionales sirven como extractores de características, por lo tanto aprenden las características de sus imágenes de entrada, las imágenes ingresan a las capas convolucionales en donde se combinan con "filtros" o "kernels". Al

combinar los filtros con las imágenes de ingreso la CNN realiza una multiplicación de la matriz del filtro por la matriz de la imagen de entrada y luego suma todos los elementos de la matriz resultante para obtener un valor único. Estos valores obtenidos al deslizar el filtro por la imagen realizando la operación de convolución forman el "Mapa de características", esto se observa en la Figura 1.3 [19][20].

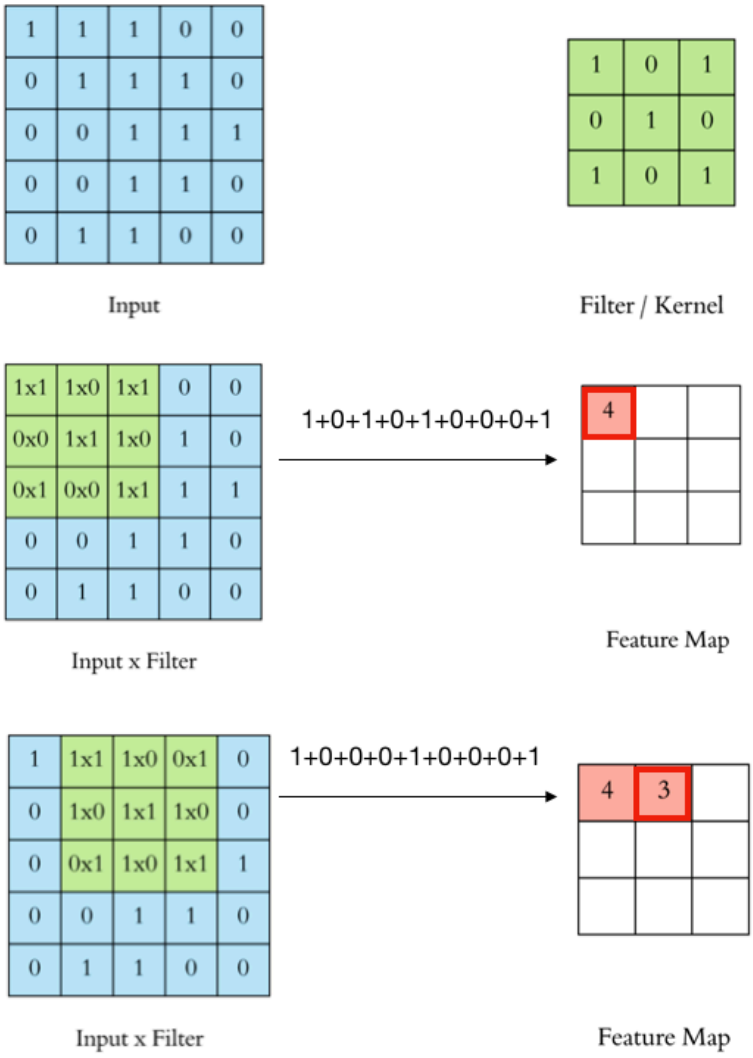


Figura 1.3 *Operación de Convolución*
Fuente: *[19]*

Durante el entrenamiento la red neuronal emplea el algoritmo del descendimiento del gradiente mediante el cual aprende los valores óptimos de las matrices de los filtros que le permiten extraer características significativas como texturas, bordes y formas del mapa de características de ingreso. A medida que

aumenta la cantidad de filtros aplicados a la entrada, aumenta la cantidad de características que la CNN puede extraer [19].

1.3.2 RELU

La función de activación "RELU" (Unidad Lineal Rectificada) es una operación no lineal la cual introduce la no linealidad en el modelo, dando la capacidad al mismo de aprender funciones no lineales y está definida por la siguiente ecuación [21]:

$$f(x) = \max(0, x)$$

Esta función de activación evalúa cero para las entradas negativas y los valores positivos permanecen intactos, proporcionando una velocidad de entrenamiento mucho mayor debido a que los gradientes negativos no se propagan. Esta se puede observar en la Figura 1.4 [21]:

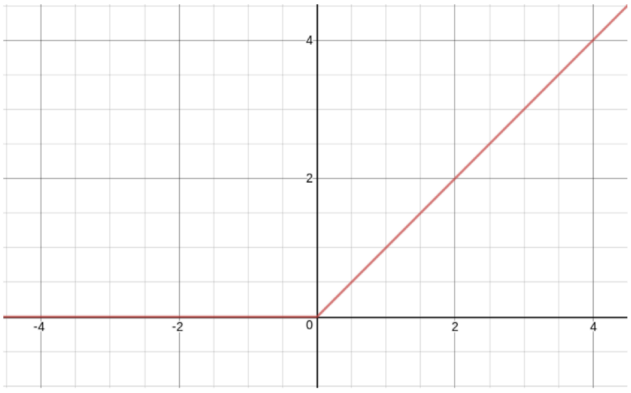


Figura 1.4 *Función de activación RELU*
Fuente: *[21]*

1.3.3 POOLING

Pooling es una operación que reduce las dimensiones espaciales, largo y ancho del mapa de características, no afectando a la profundidad del mismo, esta capa se suele aplicar luego de la capa convolución o luego de la aplicación de una función de activación, esta operación logra reducir el número de parámetros y por lo tanto ayuda a controlar el sobreajuste de la red, además proporciona al modelo una cierta invariancia espacial, lo descrito se observa en la Figura 1.5 [19][20].

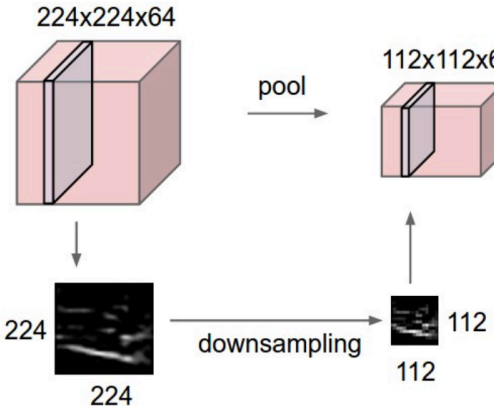


Figura 1.5 *Reducción de dimensiones mediante la operación de Pooling*
Fuente: *[22]*

El algoritmo más usado para la operación de Pooling es "Max Pooling". Max Pooling opera de manera similar a la convolución, esta se desliza sobre el mapa de características utilizando ventanas de un tamaño específico, para cada ventana el valor máximo se envía a un nuevo mapa de características y todos los demás valores se descartan, con lo cual se preservan las características más importantes, esto se puede observar en la Figura 1.6 [19].

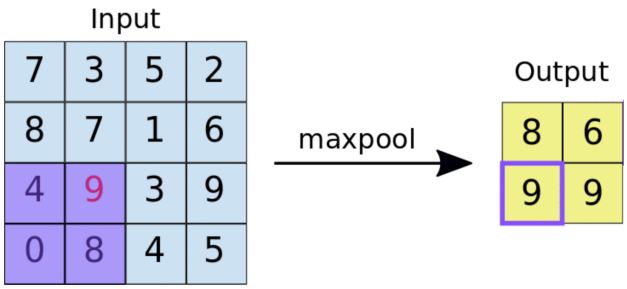


Figura 1.6 *Operación Max-Pooling*
Fuente: *[19]*

1.3.4 CAPAS TOTALMENTE CONECTADAS

Las capas de neuronas totalmente conectadas se implementan al final de la red neuronal convolucional, las cuales realizan la operación de clasificación basados en las características extraídas por parte de la convolución, por lo cual el ingreso a estas capas es una entrada aplanada del mapa de características resultante de la etapa de convolución, esto se puede observar en la Figura 1.7 [19][20].

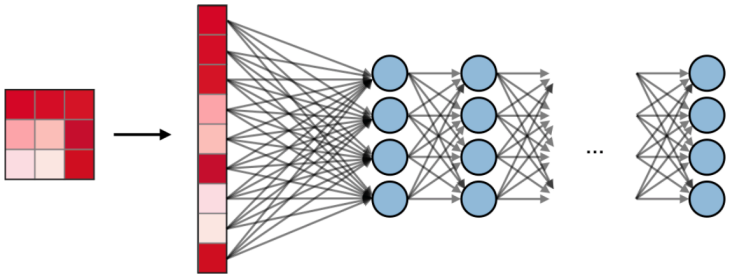


Figura 1.7 *Red neuronal totalmente conectada*
Fuente: *[20]*

1.4 YOLO – REAL TIME OBJECT DETECTION

Yolo (You Only Look Once) es un sistema de detección y localización de objetos en tiempo real [23], el cual está entrenado en la base de datos "COCO" la cual es una base de datos que contiene 300.000 imágenes de 91 tipos de objetos que van desde personas, animales a diferentes tipos de objetos [24]. Yolo es extremadamente rápido y preciso en comparación con otros modelos, esta comparación se puede observar en la Figura 1.8 , en donde se indica la velocidad del modelo vs mAP (mean average precision) [23].

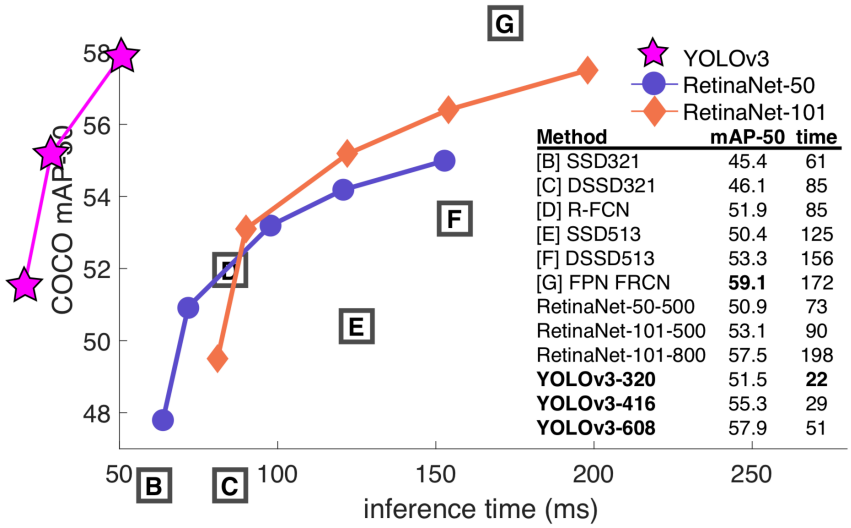


Figura 1.8 *Comparación de modelos*
Fuente: *[25]*

Yolo implementa un enfoque diferente y posee varias ventajas en comparación con otros sistemas de detección basados en clasificadores, estos aplican el modelo a una imagen en múltiples ubicaciones y escalas, mientras que Yolo observa la imagen una sola vez , implementando una sola red neuronal a toda la imagen completa,

haciendo predicciones con una única evaluación de red a diferencia de los sistemas como R-CNN que requieren miles para una sola imagen, esto lo hace extremadamente más rápido, más de 1000 veces más rápido que R-CNN Y 100 veces más rápido que Fast R-CNN [23].

1.4.1 FUNCIONAMIENTO

Yolo lleva a cabo sus detecciones dividiendo la imagen de entrada en una cuadrícula de SxS celdas, cada una de las celdas es responsable de predecir N número de cuadros delimitadores o "bounding boxes" para cada cuadro delimitador la red predice un nivel de probabilidad de que un cuadro delimitador encierre un objeto y la probabilidad de que el objeto encerrado sea de una clase en particular. Los cuadros delimitadores que poseen un nivel de probabilidad bajo de cierto nivel son eliminados, quedando solamente los que tienen una alta probabilidad, a los cuadros delimitadores restantes se aplica una técnica llamada "non-max suppresion" el cual elimina los cuadros delimitadores duplicados que detectaron el mismo objeto, quedando así solamente los cuadros delimitadores que poseen una alta probabilidad que cierran un objeto de una clase en particular, lo descrito se puede observar en la Figura 1.9 [25].

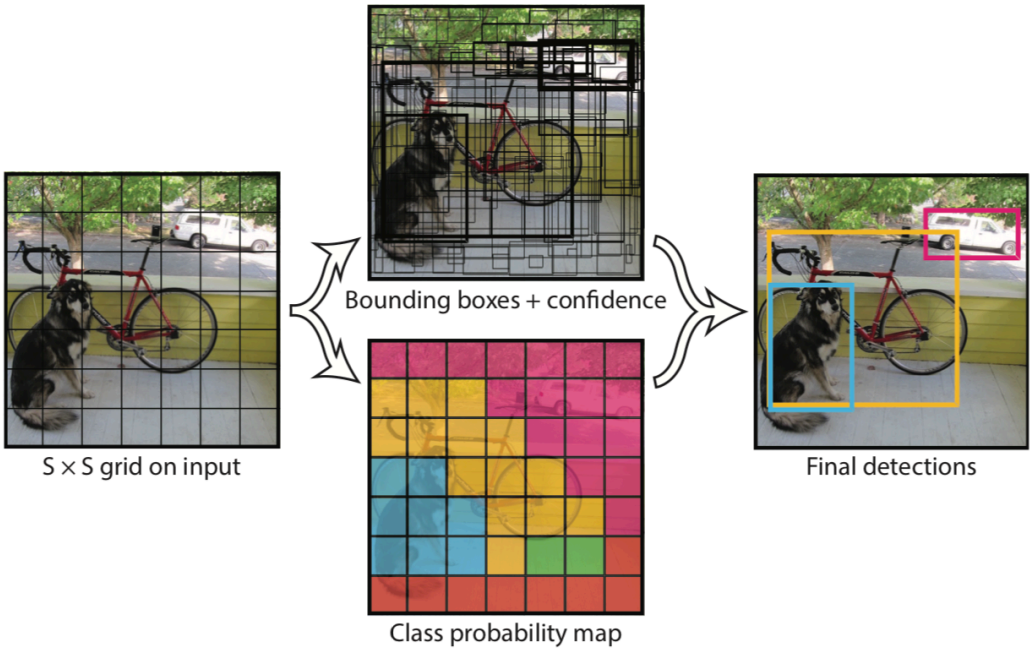


Figura 1.9 *Sistema de detección Yolo*
Fuente: *[26]*

1.5 TENSORFLOW

TensorFlow [27] es una biblioteca de software libre desarrollado por el equipo de Google Brain dentro de la organización de investigación de Google Machine Learning Intelligence, con el propósito de llevar a cabo tareas de aprendizaje automático e investigación de redes neuronales profundas. TensorFlow implementa una forma de programación en grafos de flujo de datos en donde cada nodo en el gráfico representa una instancia de una operación matemática, y cada conexión del grafo es un conjunto de datos multidimensional (tensores) [28]. Un ejemplo de un gráfico computacional que implementa TensorFlow se puede observar en la Figura 1.10.

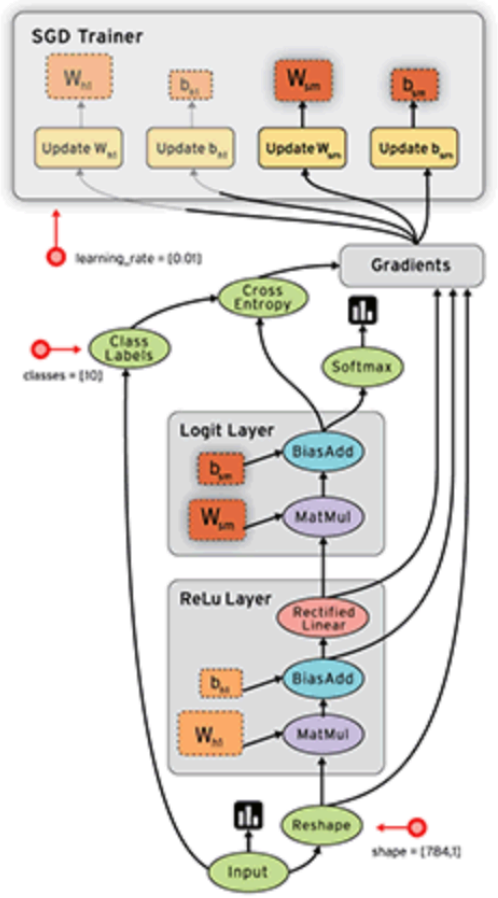


Figura 1.10 *Grafico de flujo de datos - TensorFlow*
Fuente: *[27]*

1.6 QT CREATOR

QT Creator es un entorno de desarrollo integrado multiplataforma para crear aplicaciones C++ y QML para múltiples plataformas de escritorio, integradas y móviles, este entorno posee un editor de código y está integrado con herramientas para

diseñar, codificar, probar, e implementar software. Este entorno posee Python bindings los cuales permiten el desarrollo de interfaces mediante código Python [29].

1.7 GOOGLE CLOUD – GPU

Google Cloud es una plataforma que permite acceder a una capacidad de computación paralela masiva, esta posee GPU's optimizadas para tareas como el aprendizaje profundo, simulación física o modelado molecular, que incrementan la velocidad de procesamiento de tareas complejas, como por ejemplo el entrenamiento de redes neuronales profundas. El uso del servicio en la nube reduce los gastos de capital, optimiza tiempo y costes [30].

CAPÍTULO 2: MARCO METODOLÓGICO

En el presente capítulo se presentan los procedimientos y técnicas aplicadas para la creación de la base de datos, el diseño y desarrollo de la arquitectura de la CNN, además de la descripción de las fases de entrenamiento, evaluación y pruebas.

2.1 BASE DE DATOS

Para el desarrollo del sistema de detección, tal como es para prácticamente todos los sistemas que utilicen el aprendizaje automático como estrategia de análisis, es necesario contar con una base de datos de grandes dimensiones para el entrenamiento del modelo. Debido a la inexistencia en la web de una base de datos de imágenes de armas de fuego o de personas que posean un arma de fuego, se procedió a la creación de esta base de datos, la cual está conformada de dos clases, la primera de personas las cuales están en posesión de un arma de fuego y la segunda de personas sin un arma de fuego.

La base de datos se optó por constituirla con imágenes de personas en posesión de un arma de fuego en vez de imágenes en donde se presente solamente el arma de fuego, debido a dos razones, la primera de proporcionar a la red en su entrenamiento de imágenes similares a las que se va a enfrentar en su funcionamiento, en donde el arma de fuego se presenta en ambientes muy complejos con múltiples objetos a su alrededor y la segunda razón se debe a que en la industria de la seguridad existen personas encargadas de proporcionar la seguridad respectiva en múltiples lugares, como guardias o policías, los cuales poseen armas de fuego que en la mayoría de los casos se encuentran en lugares del cuerpo en donde son visibles y en estos casos no sería adecuado que el sistema realice una detección. Se busca que el sistema detecte al

momento que una persona posee el arma de fuego en sus manos, debido a que en estas situaciones se considera que es una situación de peligro.

La base de datos creada originalmente está conformada por 17684 imágenes y está conformada de dos clases, dividida en un porcentaje aproximadamente de 50% para cada clase, como se observa en la Figura 2.1

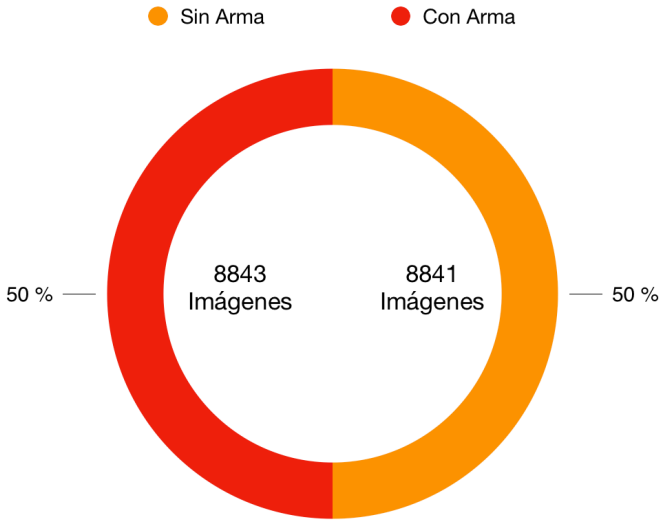


Figura 2.1 *Estructura de la base de datos*
Fuente: Autor

2.1.1 CLASE A – CON ARMA DE FUEGO

La primera clase de la base de datos en donde se encuentra presente un arma de fuego, está conformada de imágenes obtenidas de diferentes sitios de la web así como de imágenes obtenidas de videos de personas que poseen armas de fuego en diferentes situaciones, como en robos reales y en situaciones de personas realizando prácticas de tiro con armas de fuego, las cuales de igual manera se obtuvieron de la web.

En referencia a la información obtenida a partir de videos, los mismos se clasificaron en función de dos situaciones puntuales, la primera, de situaciones de robo reales en donde aparece un arma de fuego. En la web existe un gran número de videos de este tipo, sin embargo, la gran mayoría no posee una calidad adecuada dadas las condiciones propias de las cámaras y del muestreo utilizado para montar el video en la red. Por esta razón, se procedió a escoger los videos que presentasen la calidad más alta posible. El criterio seguido para escoger la calidad adecuada de las imágenes se puede observar en la Figura 2.2, en donde desde la imagen número 3 se considera que

la calidad de la imagen es adecuada para el entrenamiento de la red dadas las condiciones de enfoque, resolución gráfica y luminosidad, provocando que se visualice mejor todas las características de la imagen.

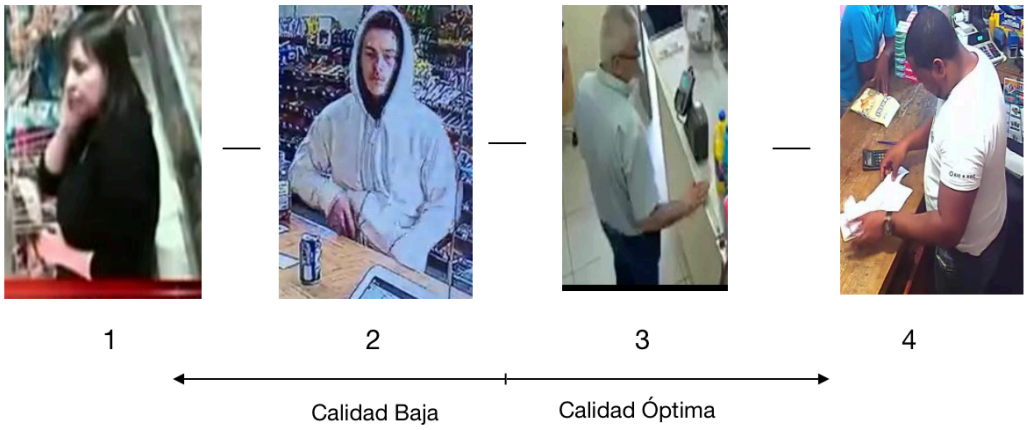


Figura 2.2 *Rango de calidad de imágenes*
Fuente: Autor

El segundo tipo de videos considerados para esta clase presenta a personas realizando prácticas de tiro, en donde se muestran posiciones típicas de personas que portan un arma de fuego que resultan ser muy similar a las que se dan en situaciones de robos reales y que al aparecer en una situación diferente a la de un robo real, le aporta al sistema información de generalización que evitará el sobre aprendizaje de los datos.

2.1.2 CLASE B – SIN ARMA DE FUEGO

La segunda clase de la base de datos está conformada de imágenes en donde no se presentan armas de fuego, el primer tipo de imágenes para esta clase se obtuvieron igualmente de videos en donde se presentaban personas realizando prácticas de tiro, se tomaron las imágenes en donde no se presentaba el arma de fuego. El segundo tipo de imágenes de esta clase fueron obtenidas en la web de personas sin un arma de fuego que se encontraban en diferentes tipos de posiciones y lugares, con estas imágenes se reemplazó a las imágenes de videos de robos reales en las que no se presentaba el arma de fuego debido a que estas presentaban una mejor calidad de imagen, según el criterio considerado en la Figura 2.2.

Con el método propuesto se obtuvieron un total de 2411 videos, y a partir de cada uno se extrajeron los correspondientes “Frames” tal como se observa en la Figura 2.3.

Figura 2.3 *Obtención de Frames de videos*
Fuente: Autor

En los mencionados Frames, el objeto que se desea detectar ocupa una pequeña parte de la imagen en un ambiente muy complejo, el arma de fuego se va a encontrar solamente junto a las personas y no en otro lugar de la imagen, por lo cual solamente los segmentos de la imagen en donde se encuentran las personas son de interés para el sistema. Por esta razón a los frames obtenidos se aplicó el sistema de detección "YOLO", con el fin de detectar y localizar a las personas en la imagen y una vez localizadas segmentar a estas en cada uno de los frames y así obtener solamente ese segmento de la imagen, lo descrito se observa en la Figura 2.4. Debido a que "YOLO" es un sistema implementado en el lenguaje de programación "C" se utilizó una implementación del mismo en TensorFlow obtenido de [31].

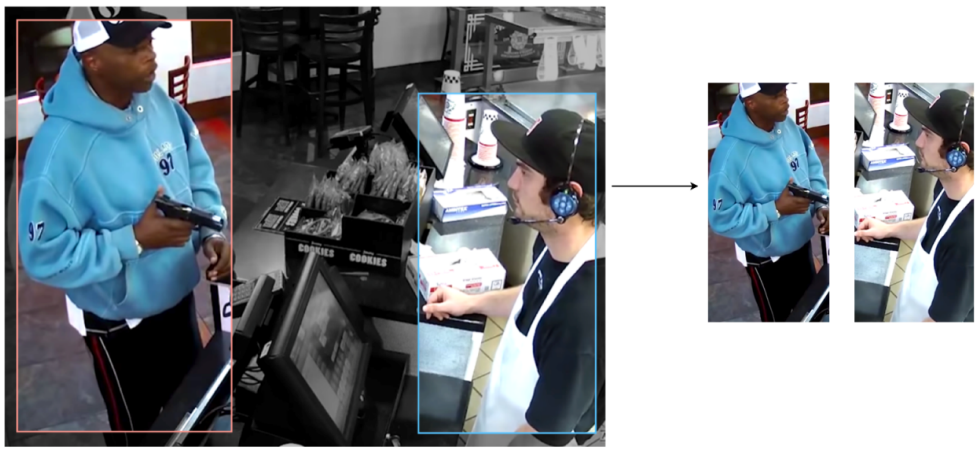


Figura 2.4 *Localización y Segmentación de personas*
Fuente: Autor

A partir de las modificaciones propuestas se pudo reducir imágenes de grandes dimensiones a imágenes pequeñas, en las que se incluye la información más

importante, que para nuestro caso son las personas que se encuentran en posesión de armas de fuego, todo el proceso descrito se puede observar en la Figura 2.5.

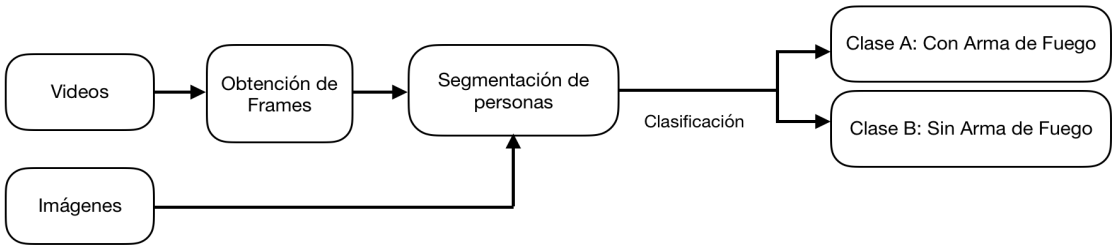


Figura 2.5 *Proceso de Creación de la Base de Datos*
Fuente: Autor

La estructura de cada una de las clases de la base de datos se puede observar en las Figuras 2.6 y 2.7

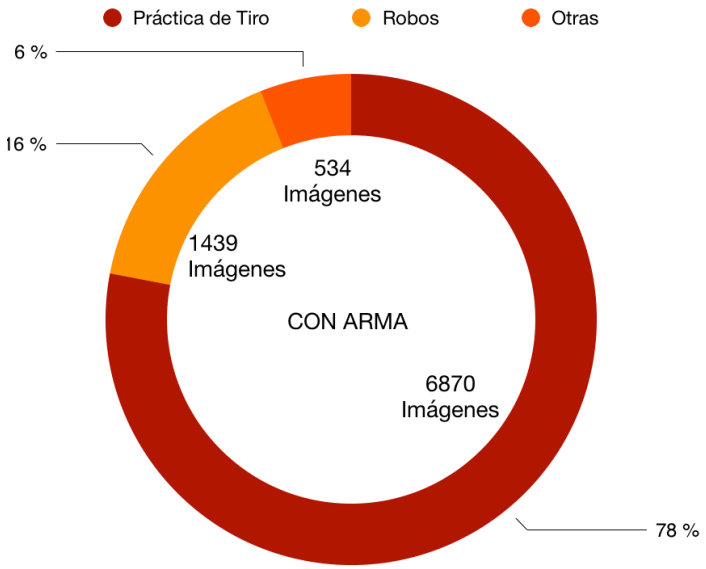


Figura 2.6 *Estructura – Clase A: Imágenes con Arma de Fuego*
Fuente: Autor

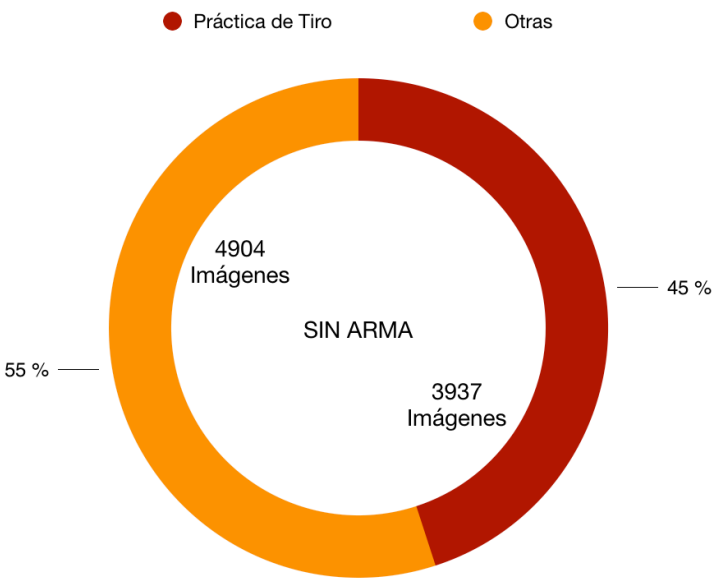


Figura 2.7 *Estructura –Clase B: Imágenes sin Arma de Fuego*
Fuente: Autor

En un robo real, la persona que está cometiendo el hecho puede presentar el arma de fuego de dos maneras diferentes, primero al mostrar el arma de fuego, tal como se muestra en la Figura 2.8 A, en donde el atacante apunta el arma de fuego, esta posición es captada por la cámara al encontrarse esta en una posición lateral al atacante. La segunda área en donde se presenta un arma de fuego se muestra en el caso B de la misma Figura 2.8, en este caso el atacante muestra parcialmente el arma de fuego en esta área sin apuntarla, esta posición es captada por la cámara al encontrarse en una posición frontal al atacante. En la presente base de datos aproximadamente un 95% de las imágenes presentan el arma de fuego en la posición del caso A, debido a esto el sistema de detección proporcionará su funcionamiento óptimo al presentarse esta posición.

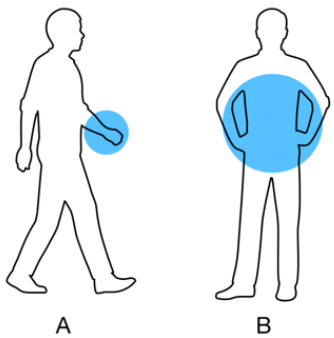


Figura 2.8 *Posiciones en las que se Presenta el Arma de Fuego*
Fuente: Autor

2.1.3 PREPROCESAMIENTO Y AUMENTO DE LA BASE DE DATOS

Previo al uso de la base de datos en la red neuronal convolucional propuesta en este trabajo, es necesario realizar un preprocesamiento de los datos, primeramente en el aspecto de normalización del tamaño de las imágenes, debido a que la entrada a la estructura de la CNN requiere que sea de un tamaño fijo, así como diferentes modificaciones a las imágenes con el objetivo de crear nuevas imágenes para el aumento de la base de datos.

- Normalización del tamaño de las imágenes:

La base de datos obtenida originalmente está conformada por imágenes de varios tamaños y en razón de que la entrada de la CNN tiene que ser de un tamaño fijo, se procedió a establecer el tamaño de las imágenes en 224x224 pixeles, este valor fue establecido luego de haber realizado un conjunto de pruebas, de las cuales se pudo determinar que el arma de fuego no presenta una gran distorsión al realizar un cambio de tamaño a la imagen. Con un tamaño de 224x224 se pudo comprobar que el arma no presenta una distorsión considerable en las imágenes.

- Aumento de la base de datos:

La base de datos originalmente obtenida está conformada por 17684 imágenes, debido a que la dificultad que representa la detección de armas de fuego en ambientes complejos es alta, y que la estructura de la CNN podría tener una gran cantidad de parámetros por la complejidad del problema, se consideró necesario incrementar el tamaño de la base de datos para evitar el sobreentrenamiento de la red. El incremento se ha llevado a cabo mediante la aplicación de diferentes técnicas para la modificación de las imágenes obtenidas inicialmente con el objetivo de crear nuevas imágenes para el aumento de la base de datos.

-*Flip en el eje horizontal:* La primera técnica que se aplicó para el aumento de la base de datos es la realización de un "Flip" en el eje horizontal a la imagen, con lo cual se crean otras imágenes logrando en cierto modo un cambio en la posición del arma de fuego en la imagen.

-*Rotación:* La segunda técnica implementada fue de rotación de las imágenes. Las cámaras de seguridad originalmente tienen una inclinación en los lugares en los cuales son colocadas, por lo cual las imágenes que muestran poseen en cierto grado una inclinación. Por esta razón, se llegó a la conclusión de que una rotación no excesiva

de las imágenes crearía nuevas imágenes similares a las que la red neuronal se enfrentará en el mundo real, se realizó una rotación a las imágenes en los ángulos de: -10, -20, -30, 10, 20 ,30.

Con las técnicas implementadas a la base de datos original de 17.684 imágenes, el tamaño aumentó a 247.576 imágenes en total. Un esquema de las técnicas aplicadas descritas para el aumento de la base de datos se puede observar en la Figura 2.9.

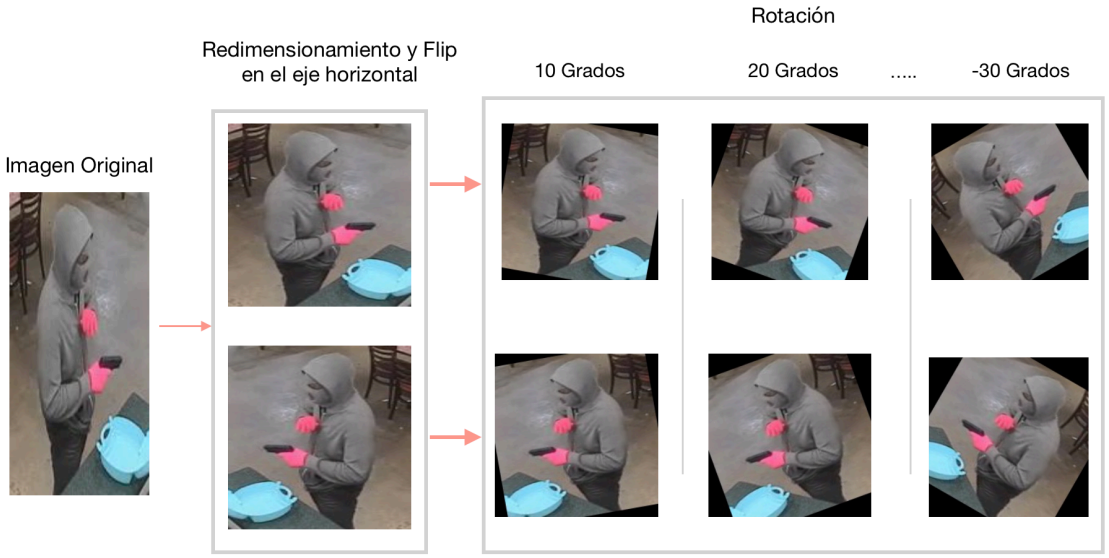


Figura 2.9 *Técnicas Aplicadas para el aumento de la Base de Datos*
Fuente: Autor

2.2 RED NEURONAL CONVOLUCIONAL

Para el desarrollo del sistema de detección propuesto se optó por la utilización de redes neuronales convolucionales (CNN), puesto que actualmente forman parte del estado del arte en el ámbito del reconocimiento de imágenes, además de que éstas presentan características muy importantes, como la extracción automática de las características necesarias de las imágenes para la clasificación.

2.2.1 ARQUITECTURA DE RED

El diseño de la CNN se realizó luego de haber realizado diferentes pruebas, utilizando diferentes configuraciones en la estructura de la red. Estas configuraciones partieron de la idea de dos tipos de arquitecturas, las cuales son arquitecturas de red que han obtenido resultados muy favorables en tareas de detección y clasificación de imágenes, estas son “VGG net” [32] y “ZF net” [33].

2.2.1.1 ARQUITECTURA VGG NET

La primera arquitectura de red propuesta fue basada en la arquitectura de la red "VGG net", la idea en la cual está basada la estructura de esta red neuronal convolucional es mantener a la red simple y profunda. "VGG net" posee una estructura de red formada por filtros pequeños que poseen un tamaño de filtro de 3x3 en todas sus capas, esta estructura de red se caracteriza por su profundidad, implementando una gran cantidad de capas y filtros convolucionales [32]. Las diferentes configuraciones de "VGG net" se puede observar en la Figura 2.10.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figura 2.10 *Configuraciones – VGG net*
Fuente: [32]

La primera arquitectura de red propuesta para el entrenamiento del sistema de detección que está basada en la arquitectura de "VGG net" se puede observar en la Figura 2.11.

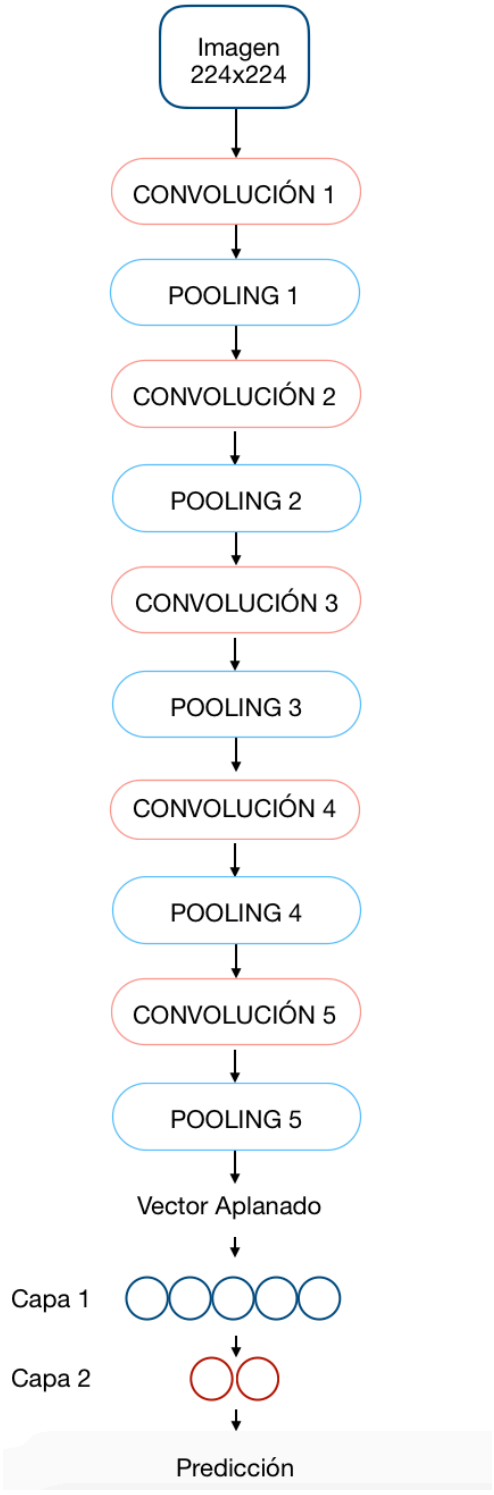


Figura 2.11 *Arquitectura VGG Net: Red Neuronal Convolucional propuesta*
Fuente: Autor

La estructura de cada una de las capas en lo que respecta al tamaño y número de filtros en cada capa en la etapa convolucional así como en la red neuronal totalmente conectada se puede observar en la Tabla 2.1.

Tabla 2.1 Arquitectura VGG Net: Red Neuronal Convolucional Propuesta

Capas	Tamaño de Filtro	Número de Filtros
Convolución 1	3x3	64
Max-Pooling 2x2		
Convolución 2	3x3	128
Max-Pooling 2x2		
Convolución 3	3x3	256
Max-Pooling 2x2		
Convolución 4	3x3	512
Max-Pooling 2x2		
Convolución 5	3x3	512
Max-Pooling		
Capa 1 - Red Neuronal TC – 2048 Neuronas		
Capa 2 - Red Neuronal TC – 2 Neuronas		

2.2.1.2 ARQUITECTURA ZF NET

La segunda estructura de red propuesta está basada en la arquitectura de la red "ZF net", que propone un enfoque diferente con el uso de filtros en la etapa convolucional de un mayor tamaño, siendo estos de 7x7 y 5x5 en las primeras capas y 3x3 en las capas intermedias y finales. Esta estructura posee 5 capas convolucionales y 3 capas de "Pooling". Cabe destacar que esta arquitectura no implementa etapas de "Pooling" luego de cada capa convolucional [33]. La estructura de "ZF net" se puede observar en la Figura 2.12.

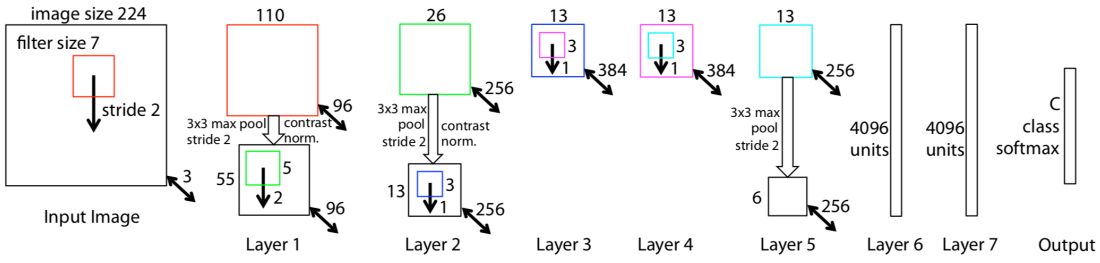


Figura 2.12 *Configuración– ZF net*
Fuente: [33]

La segunda estructura de red propuesta para el entrenamiento del sistema de detección basada en la arquitectura de "ZF net" se puede observar en la Figura 2.13

Figura 2.13 *Arquitectura ZF Net: Red Neuronal Convolucional propuesta*
Fuente: Autor

La estructura de cada una de las capas en lo que respecta al tamaño y número de filtros en cada capa en la etapa convolucional así como en la red neuronal totalmente conectada se puede observar en la Tabla 2.2.

Tabla 2.2 Arquitectura ZF Net: Red Neuronal Convolucional Propuesta

Capas	Tamaño de Filtro	Número de Filtros
Convolución 1	7x7	64
Max-Pooling 3x3		
Convolución 2	5x5	128
Max-Pooling 3x3		
Convolución 3	3x3	192
Convolución 4	3x3	192
Convolución 5	3x3	128
Max-Pooling 3x3		
Capa 1 - Red Neuronal TC – 2048 Neuronas		
Capa 2 - Red Neuronal TC – 2048 Neuronas		
Capa 3 – Red Neuronal TC – 2 Neuronas		

2.3 DESARROLLO DEL MODELO

Las redes neuronales convolucionales se programaron utilizando el lenguaje de programación "Python" [34] ya que facilita la implementación y depuración. Además se empleó la biblioteca de código abierto "TensorFlow" [27] debido a que presenta una gran integración, facilita la implementación de modelos, así como la construcción y entrenamiento de redes neuronales.

2.3.1 PREPARACIÓN DE LA BASE DE DATOS

La biblioteca "TensorFlow" soporta una gran variedad de formatos de archivos en los cuales se puede encontrar la base de datos, sin embargo TensorFlow posee un formato de archivo propio llamado "TF.Record" el cual es formato binario simple. Su uso posee un impacto significativo en la importación de los datos y por lo tanto en el tiempo de entrenamiento del modelo, además que con su uso toda la información se encuentra en un solo archivo haciendo más eficiente el entrenamiento [27]. El proceso seguido para la creación de los archivos TF.Record se puede observar en la Figura 2.14

2.3.2 PROGRAMACIÓN DEL MODELO

La CNN fue programada utilizando "TensorFlow" mediante "Custom Estimators" la cual es una API de "TensorFlow" que simplifica la programación del modelo, ésta encapsula todo lo que conlleva el entrenamiento y evaluación del modelo, además de que se encarga de tareas como la construcción del gráfico computacional,

inicialización de variables, carga de la base de datos, además de la lectura y escritura del modelo en memoria [27].

Previo al entrenamiento del modelo se debe realizar una lectura y decodificación de los archivos TF.Record, permitiendo de esta manera tomar toda la base de datos ya decodificada y de esta tomar conjuntos pequeños de imágenes o "batch" para el ingreso al modelo, el algoritmo para el entrenamiento y evaluación del modelo se puede observar en la Figura 2.15

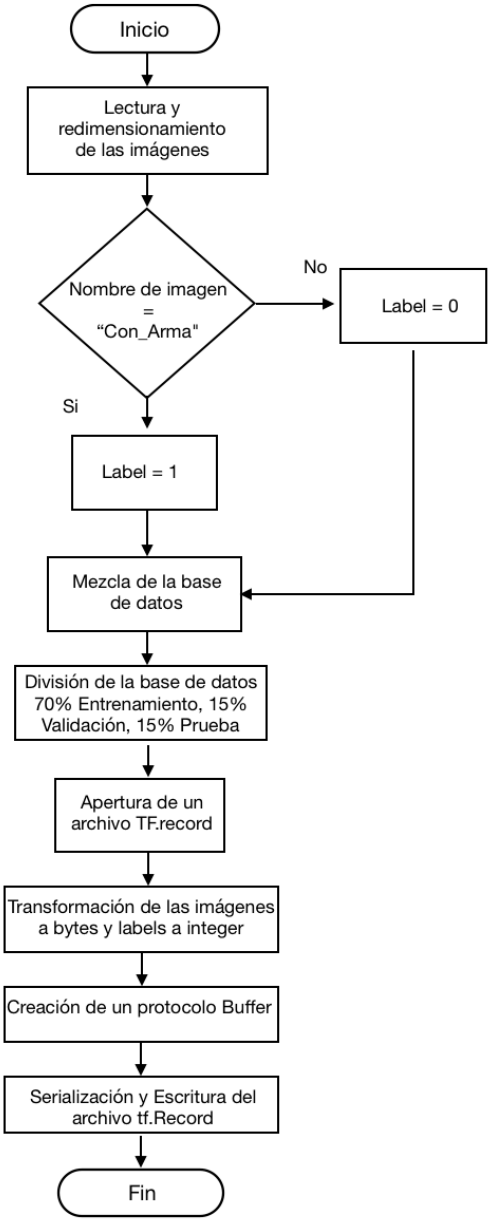


Figura 2.14 *Algoritmo para la creación de los archivos Tf.Record*
Fuente: Autor

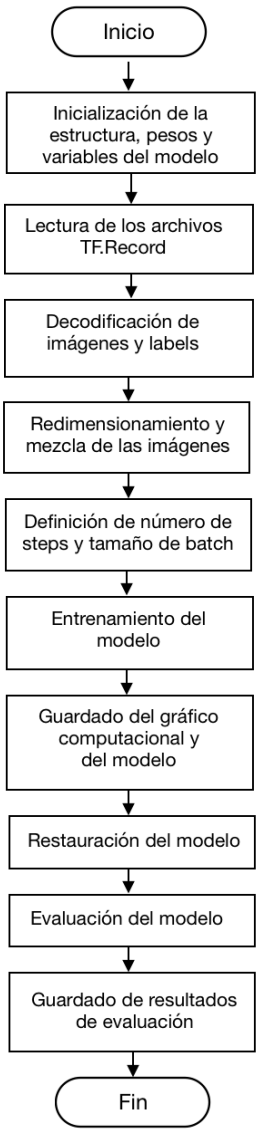


Figura 2.15 *Algoritmo para el entrenamiento del modelo*
Fuente: Autor

2.3 ENTRENAMIENTO DE LA RED

La base de datos original está conformada por 17.684 imágenes y mediante la aplicación de diferentes técnicas se realizó la ampliación de la misma consiguiendo un total de 247.576 imágenes. Para el entrenamiento de la CNN se tomó primeramente la mitad de la base de datos total es decir 123.788 imágenes, esto con el fin de analizar si las nuevas imágenes creadas favorecen la tarea de entrenamiento proporcionando información complementaria a la inicial y no tiendan a sobreentrenar al sistema y analizar si se obtienen valores de entrenamiento favorables.

Se estudiaron ambas arquitecturas, considerando al resultado más favorable en estas pruebas tanto en el entrenamiento como en la evaluación y para su entrenamiento posterior con la base de datos completa. Para el "Entrenamiento" del modelo se utilizó un 70% de la base de datos, para la "Evaluación" y "Pruebas" un 15% respectivamente. El conjunto de "Entrenamiento" es la porción de la base datos usada para preparar al sistema, el conjunto de "Validación" es la porción de la base de datos usada para comprobar el buen funcionamiento del modelo entrenado, este proporciona información para ajustar los respectivos hiperparámetros del modelo, además de proporcionarnos información relacionada con el sobreentrenamiento del sistema y el conjunto de "Prueba" es la parte de la base de datos que se usa para probar el modelo luego de que este haya sido entrenado y ajustado. El entrenamiento del modelo es realizado mediante la utilización de la plataforma "Google Cloud", con el uso de un GPU NVIDIA Tesla K80.

2.4 EVALUACIÓN Y PRUEBA DEL MODELO

La evaluación y pruebas del modelo se realizaron en base de diferentes métricas, con el fin de cuantificar la calidad de las predicciones del sistema de detección.

2.4.1 EXACTITUD

La exactitud se comprende como la cercanía entre un valor de una cantidad obtenida por medición y el valor verdadero de dicha cantidad [35]. En términos de la exactitud de un clasificador es la probabilidad de predecir correctamente la clase de una instancia sin etiquetar [36]

2.4.2 EQUAL ERROR RATE

Equal Error Rate (EER) es el valor en donde la proporción de falsos positivos (FAR) es igual a la proporción de falsos negativos (FRR). Este se encuentra estableciendo la sensibilidad del sistema en un punto óptimo en donde se minimice el número de errores producidos. Mientras más bajo sea el valor de EER mayor será la precisión del sistema [37][38]. FAR Y FRR están definidas mediante las siguientes ecuaciones:

$$FAR: \frac{Número\ de\ falsos\ positivos}{Número\ de\ ejemplos\ de\ clasificación} \quad (1)$$

$$FRR: \frac{Número\ de\ falsos\ negativos}{Número\ de\ ejemplos\ de\ clasificación} \quad (2)$$

Las curvas de FAR y FRR así como el valor de EER se puede observar en la Figura 2.16.

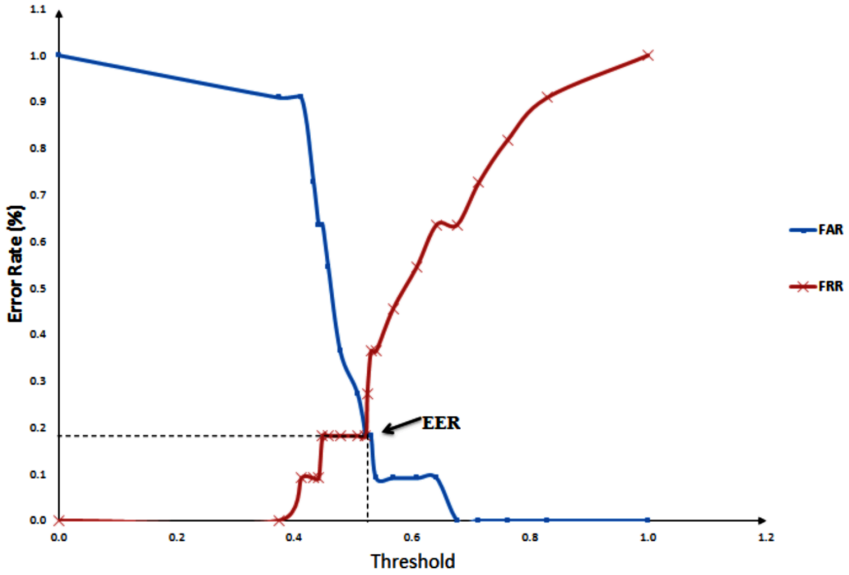


Figura 2.16 *Equal Error Rate*
Fuente: [39]

2.4.3 MATRIZ DE CONFUSIÓN

La matriz de confusión es una herramienta de visualización en la que cada columna de la matriz representa las predicciones de cada clase, mientras que en cada fila se representa las instancias de la clase real. La matriz de confusión permite identificar si el sistema se encuentra confundiendo las clases y proporciona herramientas para seleccionar los modelos posiblemente óptimos y descartar modelos no adecuados, [40], esta se puede observar en la Tabla 3.3.

- TP: Es la cantidad de ejemplos positivos que fueron clasificados como positivos por el modelo.
- FN: Es la cantidad de ejemplos positivos que fueron clasificados como negativos por el modelo.
- FP: Es la cantidad de ejemplos negativos que fueron clasificados como positivos por el modelo.
- TN: Es la cantidad de ejemplos negativos que fueron clasificados como negativos por el modelo.

Tabla 3.3 Matriz de Confusión

		Predicciones	
		Positivos	Negativos
Valor Real	Positivos	Verdaderos Positivos (TP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (TN)

2.4.4 PRECISIÓN

La precisión mide esa fracción de ejemplos clasificados como positivos que son verdaderamente positivos, siendo su valor más alto 1 y su valor más bajo 0, ésta está definida mediante la siguiente ecuación [41]:

$$PRECISION = \frac{tp}{tp + fp} \quad (3)$$

En donde *tp* es el número de verdaderos positivos y *fp* es el número de falsos positivos.

2.4.5 RECALL

Recall mide la fracción de ejemplos positivos que están correctamente etiquetados, siendo su valor más alto 1 y su valor más bajo 0, este está definido mediante la siguiente ecuación [41].

$$RECALL = \frac{tp}{tp + fn} \quad (4)$$

En donde *tp* es el número de verdaderos positivos y *fn* es el número de falsos negativos.

CAPÍTULO 3: IMPLEMENTACIÓN Y ANÁLISIS DE RESULTADOS

En este capítulo se presentan los resultados obtenidos de las diferentes estructuras de red probadas en el entrenamiento del modelo, presentando la arquitectura de red que obtuvo los mejores resultados en el entrenamiento y evaluación así como la obtención de diferentes métricas de evaluación, además se describe la implementación del sistema de detección, describiendo la combinación de modelos realizada, así como la interfaz implementada.

3.1 RESULTADOS DEL ENTRENAMIENTO

Las pruebas realizadas en el entrenamiento de la red utilizan un tamaño de "batch" de 200 imágenes y una tasa de aprendizaje de 0.001, el descendimiento del gradiente es el algoritmo implementado en el entrenamiento para el mejoramiento de los parámetros de la red. La función de activación implementada en la etapa convolucional así como en las capas de la red neuronal totalmente conectada es "Relu", excepto en la capa final en donde se utiliza una función de activación "Softmax".

Para la realización de las pruebas se partió de las arquitecturas "VGG Net" y "ZF Net" propuestas en el capítulo anterior, a partir de éstas también se evaluaron otras configuraciones con el fin de establecer el funcionamiento óptimo del sistema. Se pudo notar que las variaciones son pequeñas entre estas 2 arquitecturas. Los parámetros que se modificaron fueron: el número de neuronas en la red totalmente conectada así como el número de filtros en las capas convolucionales.

3.1.1 ARQUITECTURA VGG NET

Primeramente se realizaron pruebas con la arquitectura VGG Net propuesta en el capítulo anterior y con una segunda configuración que consiste en una variación de esta arquitectura respecto al número de neuronas en la red neuronal totalmente conectada. Estas estructuras de red se observan en la Tabla 3.1.

Tabla 3.1 Arquitectura VGG Net: Configuraciones

Configuraciones	
1	2
Input 224x224	
Conv 3x3 – 64 Filtros	Conv 3x3 – 64 Filtros
Max-Pooling 2x2	
Conv 3x3 – 128 Filtros	Conv 3x3 – 128 Filtros
Max-Pooling 2x2	
Conv 3x3 – 256 Filtros	Conv 3x3 – 256 Filtros
Max-Pooling 2x2	
Conv 3x3 – 512 Filtros	Conv 3x3 – 512 Filtros
Max-Pooling 2x2	
Conv 3x3 – 512 Filtros	Conv 3x3 – 512 Filtros
Max-Pooling 2x2	
RTC – 2048 Neuronas	RTC – 4096 Neuronas
RTC - 2 Neuronas	RTC - 2 Neuronas

Estas configuraciones implementan un paso (“Stride”) de 1 pixel en los filtros convolucionales, este paso es tal que la resolución espacial se conserva luego de la convolución, mientras que el paso implementado en las capas “Max-Pooling” es de 2 pixeles. Las pruebas realizadas con estas configuraciones se presenta en la Tabla 3.2

Tabla 3.2 Arquitectura VGG Net: Pruebas

Pruebas : Arquitectura VGG Net						
Configuración 1						
			Entrenamiento		Evaluación	
Pruebas	Imagen	Steps	Loss	Accuracy	Loss	Accuracy
P1	RGB	10.000	0.32	0.81	0.30	0.84
Configuración 2						
P2	RGB	10.000	0.27	0.83	0.30	0.86
P3	RGB	12.500	0.20	0.84	0.25	0.89
P4	RGB	16.000	0.16	0.87	0.22	0.90
P5	Gris	16.000	0.20	0.85	0.26	0.89

Se puede observar que los mejores resultados en el entrenamiento y evaluación se obtienen con la configuración 2 en las pruebas P4 en imágenes RGB y en la P5 con imágenes en escala de grises , en donde se observa que se llega a los valores más altos de exactitud o “accuracy” sin presentarse valores de pérdidas o “loss” que indiquen un fuerte sobre entrenamiento del sistema, que se presenta cuando los valores de pérdida en la evaluación es mucho mayor que el valor de pérdida obtenido durante el entrenamiento.

3.1.2 ARQUITECTURA ZF NET

Se realizaron pruebas con la arquitectura “ZF Net” propuesta en el capítulo anterior y con dos configuraciones más que son variaciones de esta arquitectura respecto al número de neuronas de la red neuronal totalmente conectada y en el número de filtros convolucionales . Estas estructuras de red se observan en la Tabla 3.3.

Tabla 3.3 Arquitectura ZF Net: Configuraciones

Configuraciones		
1	2	3
Input 224x224		
Conv 7x7 – 64 Filtros	Conv 7x7 – 92 Filtros	Conv 7x7 – 92 Filtros
Max-Pooling 3x3		
Conv 5x5 – 128 Filtros	Conv 5x5 – 192 Filtros	Conv 5x5 – 192 Filtros
Max-Pooling 3x3		
Conv 3x3 – 192 Filtros	Conv 3x3 – 256 Filtros	Conv 3x3 – 256 Filtros
Conv 3x3 – 192 Filtros	Conv 3x3 – 256 Filtros	Conv 3x3 – 256 Filtros
Conv 3x3 – 128 Filtros	Conv 3x3 – 192 Filtros	Conv 3x3 – 192 Filtros
Max-Pooling 3x3		
RTC – 2048 Neuronas	RTC – 4096 Neuronas	RTC – 4096 Neuronas
RTC – 2048 Neuronas	RTC – 2 Neuronas	RTC – 2048 Neuronas
RTC-2		RTC-2 Neuronas

Estas estructuras implementan en las 2 primeras capas convolucionales un paso de los filtros de convolución de 2 pixeles, con lo cual la resolución espacial no se conserva luego de la convolución y en las capas convolucionales siguientes su paso es de 1 pixel en donde la resolución espacial si se conserva luego de la convolución, mientras que el paso implementado en las capas de “Max-Pooling” es de 2 pixeles. Las pruebas realizadas con estas configuraciones se presentan en la Tabla 3.4

Tabla 3.4 Arquitectura ZF Net: Pruebas

Pruebas : Arquitectura ZF Net						
Configuración 1						
			Entrenamiento		Evaluación	
Pruebas	Imagen	Steps	Loss	Accuracy	Loss	Accuracy
P1	RGB	17.000	0.28	0.81	0.28	0.87
P2	Gris	9.000	0.29	0.79	0.33	0.85
P3	Gris	11.000	0.12	0.82	0.35	0.87
Configuración 2						
P2	RGB	19.000	0.21	0.82	0.29	0.86
Configuración 3						
P3	RGB	15.000	0.13	0.85	0.32	0.86

En las pruebas realizadas con este tipo de arquitectura de red se puede observar que no se obtienen valores de exactitud o "accuracy" en las predicciones tan altos como en la arquitectura VGG Net, llegando a un valor máximo de 0.87, además de que los valores de pérdida o "loss" son más altos en comparación con la anterior arquitectura tanto en el entrenamiento como en la evaluación.

Los mejores resultados obtenidos tanto en el entrenamiento como en la evaluación fueron obtenidos con la configuración 2 de la arquitectura "VGG Net", por lo tanto para este problema en particular se obtuvieron mejores resultados utilizando filtros convolucionales pequeños y haciendo a la red más profunda implementando una gran cantidad de filtros convolucionales en cada capa, al contrario de la arquitectura "ZF Net" en donde la red utiliza filtros de mayor tamaño y sin implementar una gran cantidad de filtros en sus capas convolucionales como en la arquitectura de "VGG Net". Además con estos resultados se pudo observar que las nuevas imágenes creadas a partir de la base de datos original si proporcionaron información complementaria a la inicial, debido a que en las pruebas realizadas no se presentó un fuerte sobre entrenamiento en etapas iniciales del entrenamiento del modelo.

Siendo la configuración 2 de la arquitectura VGG Net la red que proporcionó los mejores resultados, se optó por utilizar ésta para el entrenamiento del modelo con la base de datos completa. Primeramente se realizaron pruebas con las imágenes en RGB, la curva de pérdida y de exactitud de la red se puede observar en las Figuras 3.1 y 3.2. La curva de color "anaranjado" es la curva correspondiente a los valores

obtenidos durante el entrenamiento y los puntos azules corresponden a los valores resultantes de la evaluación.

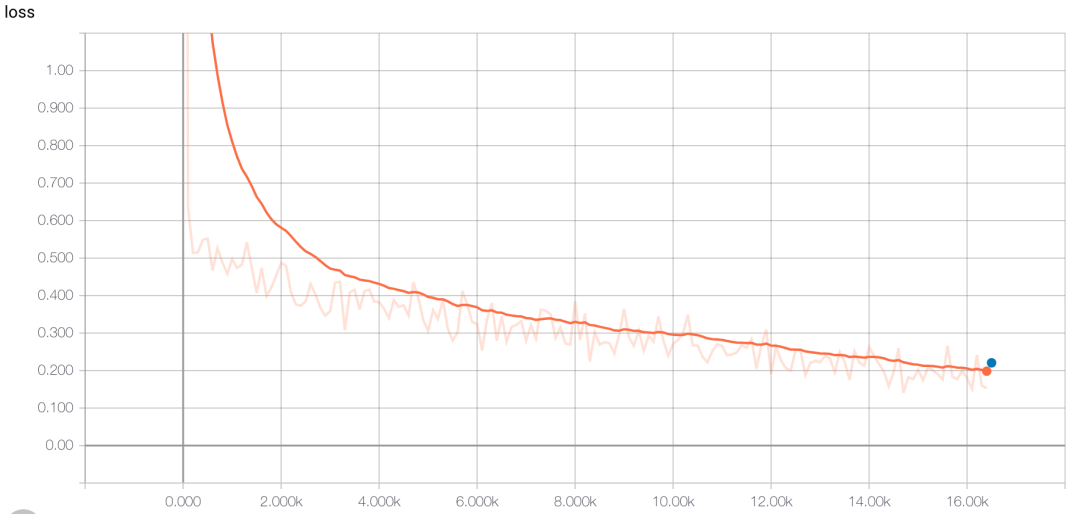


Figura 3.1 *Curva de pérdida del entrenamiento y evaluación con imágenes RGB*
Fuente: Autor

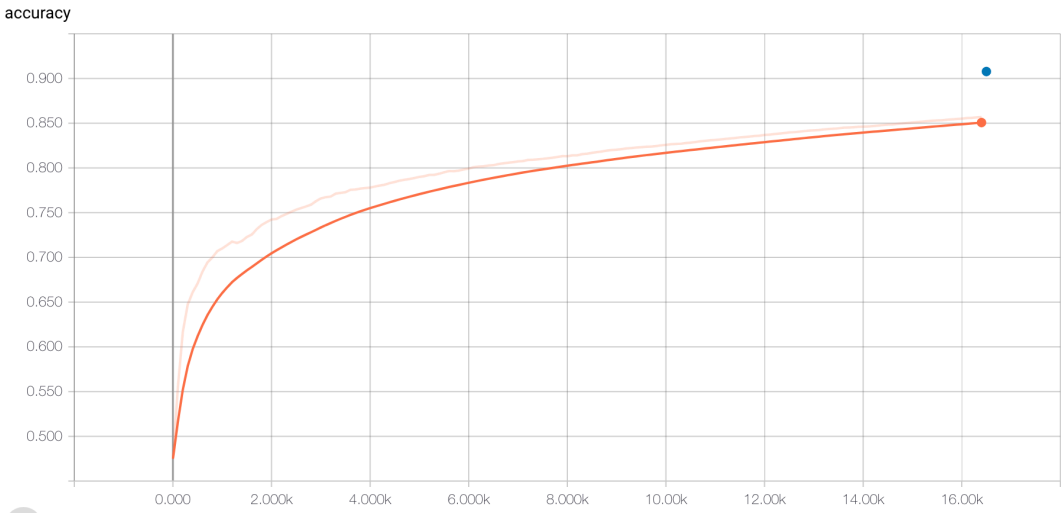


Figura 3.2 *Curva de exactitud del entrenamiento y evaluación con imágenes en RGB*
Fuente: Autor

Las curvas obtenidas durante el entrenamiento y evaluación del modelo con imágenes en escala de grises se observa en las Figuras 3.3 y 3.4

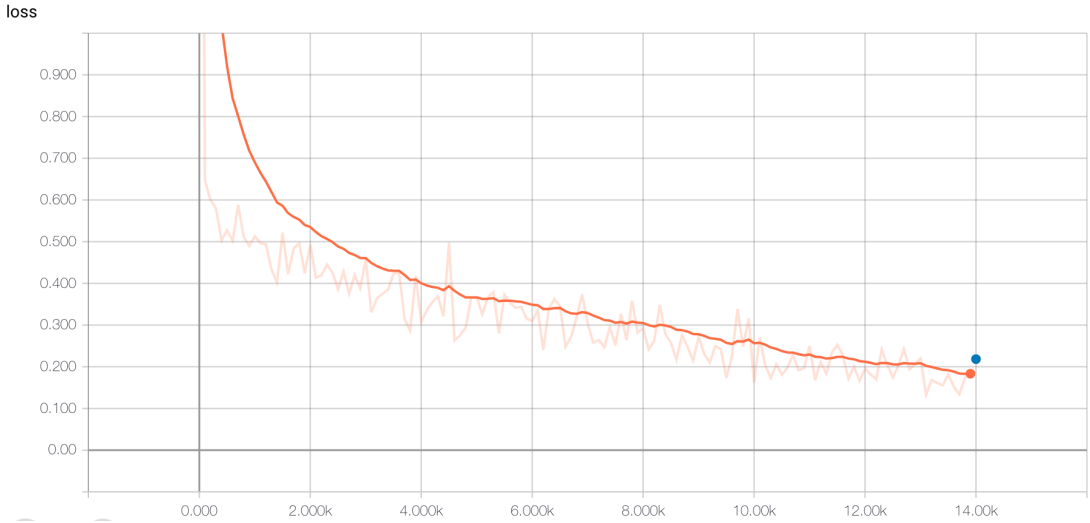


Figura 3.3 *Curva de pérdida del entrenamiento y evaluación con imágenes en escala de grises*
Fuente: Autor

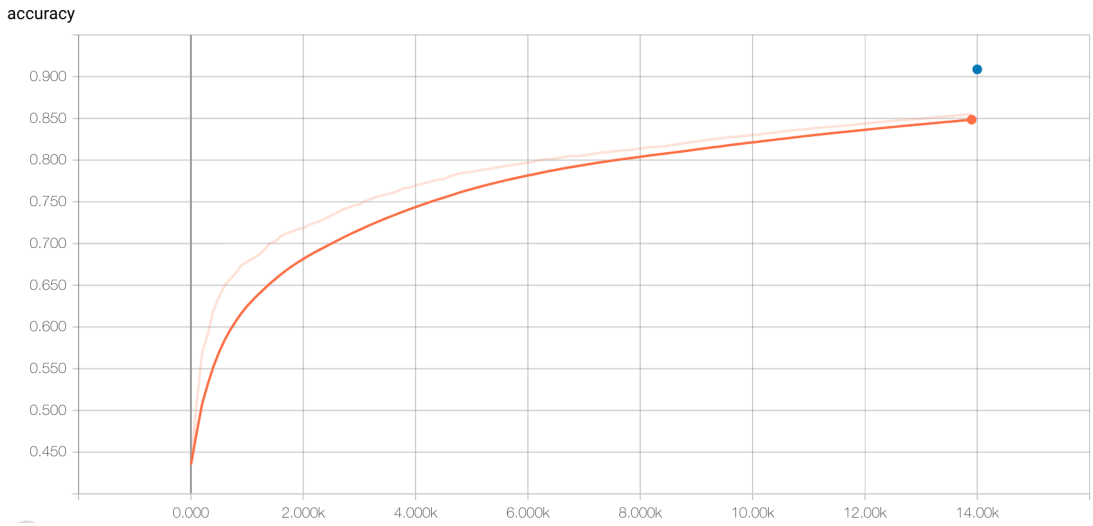


Figura 3.4 *Curva de exactitud del entrenamiento y evaluación con imágenes en escala de grises*
Fuente: Autor

La comparación de los resultados de las pruebas tanto en imágenes en escala de grises como en RGB se puede observar en la Tabla 3.5.

Tabla 3.5 Comparación de Resultados

Pruebas : Arquitectura VGG Net							
Configuración 2							
			Entrenamiento		Evaluación		8
Pruebas	Imagen	Steps	Loss	Accuracy	Loss	Accuracy	Horas
P1	RGB	16.500	0.15	0.8570	0.22	0.9067	11 h 5 min
P2	Gris	14.000	0.17	0.8550	**0.21**	**0.9087**	8 h 6 min

Según los resultados obtenidos se puede observar que los valores de pérdida y exactitud de las predicciones durante el entrenamiento son muy similares, sin embargo durante la evaluación en la prueba P2 se obtienen mejores resultados con las imágenes en escala de grises, debido a que se obtiene un valor más bajo de pérdida y una mayor exactitud en las predicciones. Por lo cual se escoge este modelo entrenado con imágenes en escala de grises para la implementación del modelo. Para ello se procedió a obtener las métricas de evaluación, primeramente se obtuvo el valor de EER, el cual fue obtenido mediante el conjunto de prueba de la base de datos y los valores de FAR y FRR se obtuvieron mediante las ecuaciones 1 y 2. La gráfica para su obtención se puede observar en la Figura 3.5.

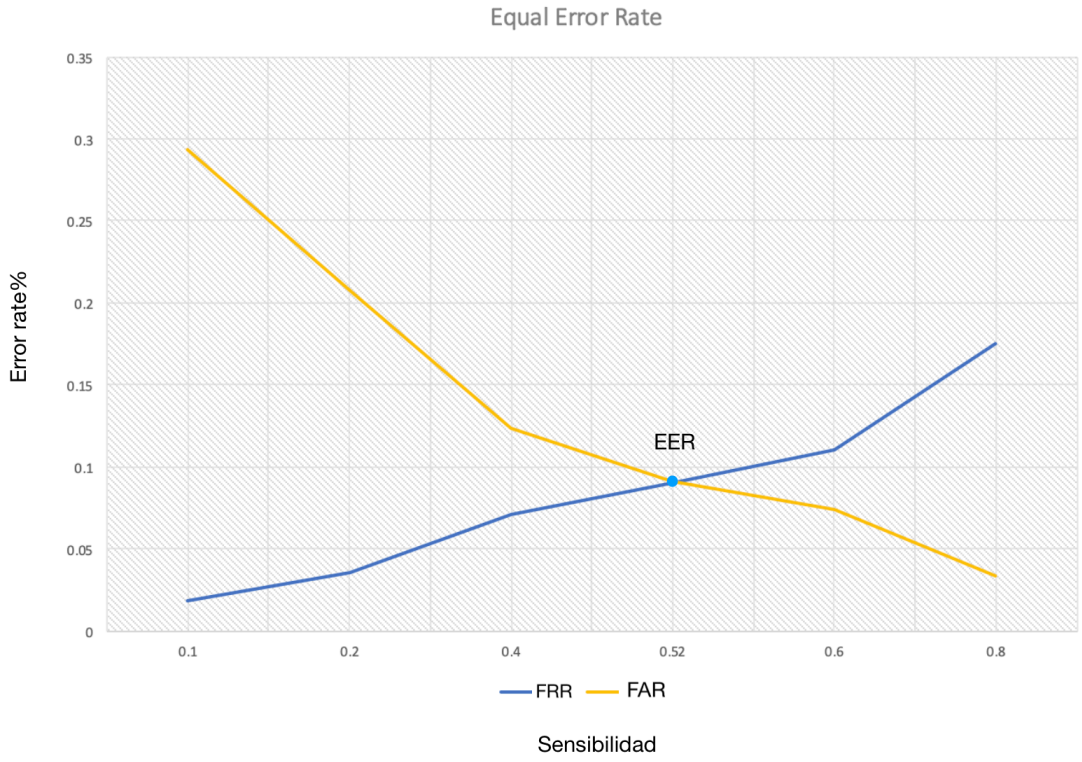


Figura 3.5 *Obtención de EER*
Fuente: Autor

El cruce entre las proporciones de falsos negativos (FRR) y falsos positivos (FAR) proporciona el valor de EER y corresponde a un valor de 0.09 a una sensibilidad de 0.52.

La matriz de confusión obtenida de este modelo para la sensibilidad más óptima para el valor más bajo de EER se puede observar en la Tabla 3.6. Esta matriz de confusión se obtuvo mediante el conjunto de prueba de la base de datos.

Tabla 3.6 Matriz de Confusión:

		Predicciones	
		Con Arma de Fuego	Sin Arma de Fuego
Clases	Con Arma de Fuego	15657	1553
	Sin Arma de Fuego	1561	15649

Se obtienen los valores de recall y precisión del modelo, mediante las ecuaciones 3 y 4, los valores obtenidos se presentan en la Tabla 3.7.

Tabla 3.7 Resultados de Métricas

Métricas	
Recall	0.9097
Precisión	0.9093

Considerando que el valor más alto que se puede obtener para estas métricas es 1, los resultados obtenidos de 0.9097 y 0.9093 tanto en la métrica de recall como en la precisión del sistema de detección respectivamente son muy favorables.

3.2 IMPLEMENTACIÓN

Las cámaras de seguridad usualmente se sitúan en lugares donde transitan muchas personas y existen múltiples objetos en el entorno, por lo cual el video el cual es captado por la cámara posee una alta complejidad en términos de la cantidad de elementos presentes en cada frame del video, en donde realizar una detección de un objeto en estos ambientes complejos puede resultar en la obtención de múltiples falsos positivos en las detecciones.

Un operador que regularmente lleva a cabo tareas de monitoreo pone especial atención al momento de que una persona aparece en el video, principalmente porque sabe que solamente ahí se puede dar una situación en la que se puede presentar un arma de fuego, generalmente no pone la atención necesaria cuando no existen personas en el ambiente que se está observando, porque sabe que en esta situación no va a aparecer un arma de fuego y por lo tanto una situación de peligro. Con el objetivo de simular este enfoque en las personas para realizar detecciones de armas de fuego se

realizó una combinación de dos modelos, el primer modelo es “Yolo” el cual se utiliza con el objetivo de detectar y localizar a las personas en la imagen, con las personas localizadas se obtienen solamente estos segmentos de la imagen las cuales serán las imágenes de ingreso al modelo desarrollado de detección de armas de fuego cortas, logrando de esta manera que el sistema de detección de armas analice solamente los segmentos de la imagen que son de importancia. Lo descrito se puede observar en las Figuras 3.6 y 3.7.

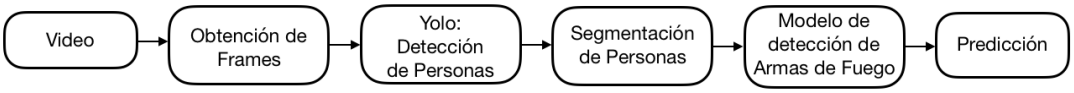


Figura 3.6 *Proceso de detección*
Fuente: Autor

Figura 3.7 *Funcionamiento del sistema de detección*
Fuente: Autor

3.2.2 INTERFAZ

La interfaz creada para la implementación del sistema de detección se puede observar en la Figura 3.8. La misma presenta diferentes elementos como el video de monitoreo, los segmentos de la imagen en donde se dio predicciones de armas de fuego, 2 botones para la selección del modo de funcionamiento y 1 para la selección del video.

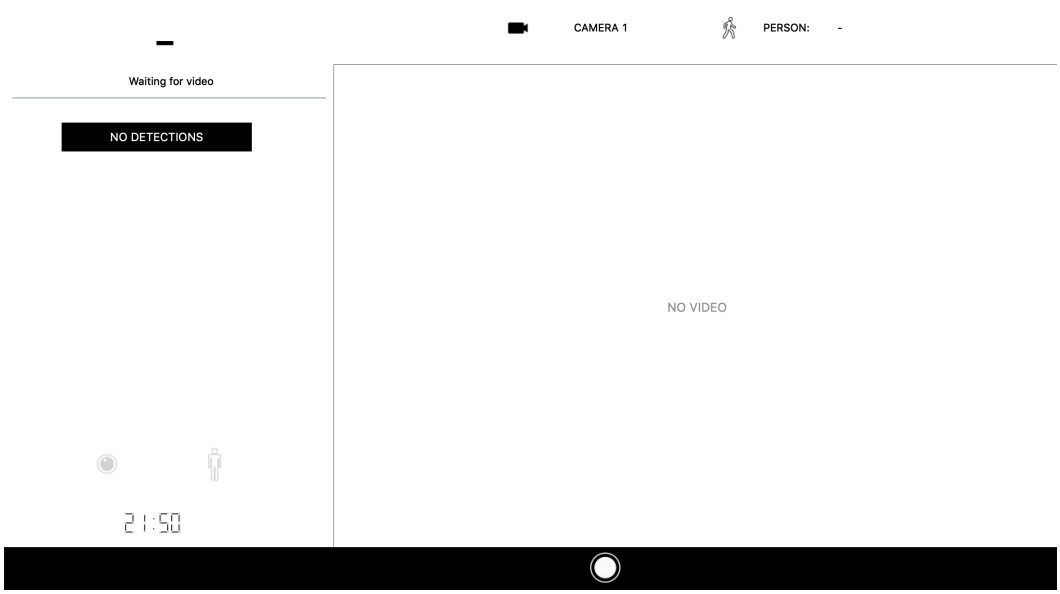


Figura 3.8 *Interfaz – Sin Video*
Fuente: Autor

El sistema en funcionamiento se presenta en las Figura 3.9. En esta se puede observar el video de seguridad en la parte derecha de la interfaz, al momento de que el sistema realiza una detección se procede a localizar el segmento de la imagen en donde se dio lugar la detección del arma de fuego, presentando este segmento en la parte izquierda de la interfaz además de un mensaje de alerta de que un arma de fuego ha sido potencialmente detectada. En esta imagen se puede evidenciar que el sistema puede realizar detecciones de una manera correcta en ambientes sumamente complejos en donde existe gran cantidad de objetos en el entorno.

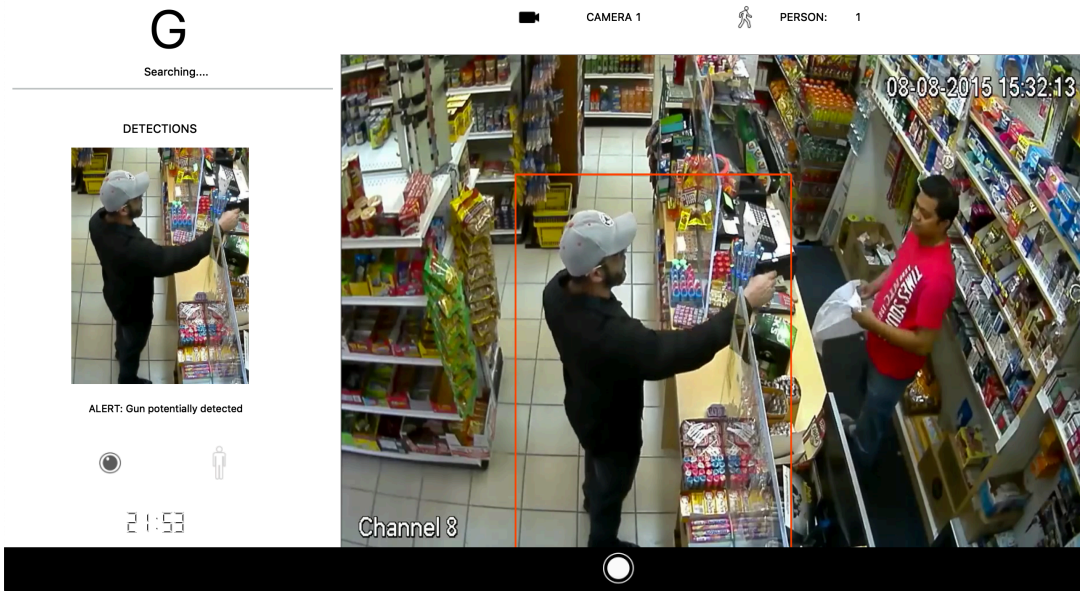


Figura 3.9 *Sistema de Detección de Armas de Fuego*
Fuente: Autor

La interfaz permite además la activación de un segundo tipo de funcionamiento, el cual permite la detección de personas, este podría ser utilizado como un sistema de detección de intrusiones en momentos del día en donde la detección de un arma de fuego no sea necesaria, esto se puede observar en la Figura 3.10.

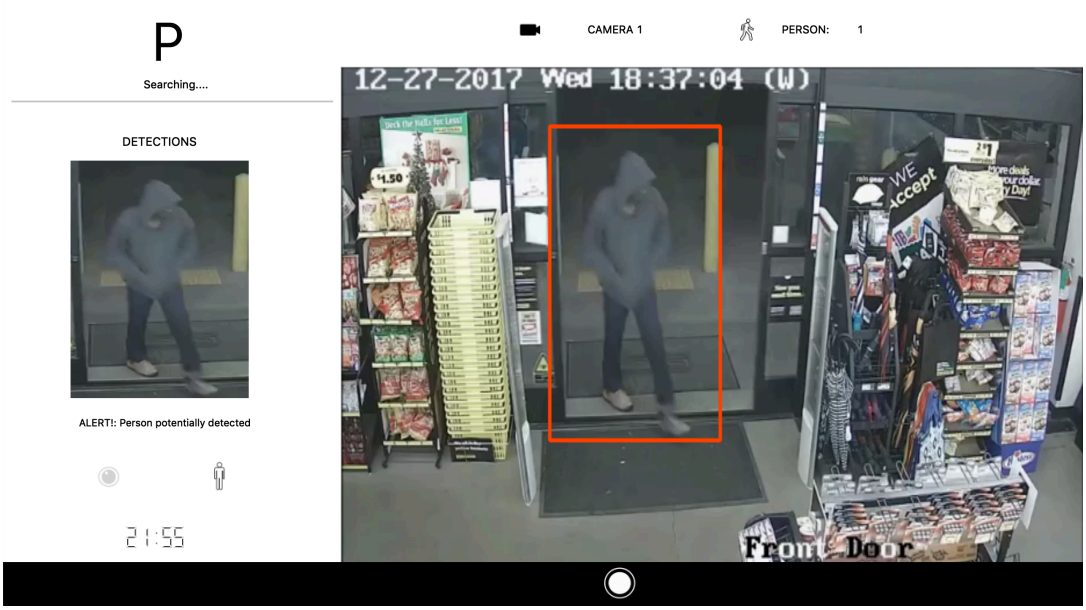


Figura 3.10 *Sistema de detección de Personas*
Fuente: Autor

CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

En el desarrollo del sistema de detección de armas de fuego se trataron dos aspectos importantes, el primero consistió en la creación de una base de datos en donde exista la presencia de un arma de fuego, y segundo, el diseño de varias configuraciones de redes neuronales convolucionales con las que se llevaron a cabo las pruebas en el entrenamiento de la red con el objetivo de encontrar la arquitectura que proporcione los resultados más favorables. En base de los resultados obtenidos en la experimentación se puede concluir que:

- El aumento de la base de datos mediante la implementación de diferentes técnicas para la creación de nuevas imágenes proporcionó información complementaria a la inicial, debido a que en las pruebas realizadas no se presentó un fuerte sobreentrenamiento en etapas iniciales de la red.
- El uso de una red neuronal convolucional que posea una arquitectura que implemente una gran cantidad de filtros pequeños en sus capas convolucionales permiten obtener mejores resultados tanto en el entrenamiento como en la evaluación del modelo.
- El uso de una arquitectura de red que se caracterice por la utilización de un mayor tamaño en sus filtros de convolución y sin implementar una gran cantidad de estos en sus capas convolucionales también proporciona resultados favorables, sin embargo sus valores de exactitud en las predicciones no son mayores a los valores obtenidos con la estructura de red descrita anteriormente.

- El entrenamiento y evaluación de la CNN con imágenes en escala de grises permitió la obtención de mejores resultados tanto en la exactitud de las predicciones como en los valores de pérdida, en comparación con el modelo entrenado en imágenes RGB.
- La utilización del sistema de detección "YOLO" permitió proporcionar al modelo desarrollado los segmentos de la imagen que son de importancia para la detección de armas de fuego, permitiendo al sistema completo enfocarse solamente en los segmentos de la imagen en donde aparezcan personas e ignorar las demás zonas de la imagen en las que no se puede dar la presencia de armas de fuego.

4.2 RECOMENDACIONES

Como recomendaciones se puede expresar:

- El problema de la detección de armas de fuego es un problema que presenta una complejidad alta debido a los diferentes entornos en los que se encuentran las cámaras de seguridad, por lo cual para la obtención de resultados favorables en las detecciones se necesita una base de base de datos de grandes dimensiones para, por un lado, garantizar que el sistema haya sido entrenado con elementos que comúnmente aparecen en un asalto a mano armada, y por otro, evitar el sobre entrenamiento de la red.
- Se recomienda usar imágenes de un tamaño igual a 224x224 pixeles o mayor, debido a que el uso de imágenes de un menor tamaño a estos, el objeto que se desea detectar en este caso el arma de fuego se reduce en su dimensión tendiendo a perder sus características.
- Para el entrenamiento y funcionamiento del sistema es recomendable el uso de una GPU, debido a que el número como el tamaño de las imágenes son de grandes dimensiones, por lo cual se necesita un procesamiento óptimo para reducir los tiempos de entrenamiento del modelo, así como tener un funcionamiento óptimo en la detección.

4.3 TRABAJOS FUTUROS

El sistema de detección de armas de fuego desarrollado puede ser mejorado al seguir entrenando al modelo con una mayor cantidad de imágenes de personas que estén en posesión de un arma de fuego, pudiendo llegar a un punto en el que el sistema se completamente independiente. Además este puede ser implementado para realizar la detección de cualquier tipo de arma, como cuchillos o armas de fuego de mayor tamaño. Finalmente este sistema se puede integrar para proporcionar alarmas disuasivas cuando se realice una detección de un arma de fuego o enviar alertas de detección a operadores encargados para la comunicación inmediata con los entes encargados de la seguridad.

REFERENCIAS BIBLIOGRÁFICAS

[1] W. Deisman, *CCTV: Literature Review and Bibliography*, no. January 2003. 2003.

[2] R. K. Tiwari and G. K. Verma, "A computer vision based framework for visual gun detection using SURF," *Int. Conf. Electr. Electron. Signals, Commun. Optim. EESCO 2015*, no. January, 2015.

[3] N. Ben Halima and O. Hosam, "Bag of words based surveillance system using support vector machines," *Int. J. Secur. its Appl.*, vol. 10, no. 4, pp. 331–346, 2016.

[4] R. K. Tiwari and G. K. Verma, "A Computer Vision based Framework for Visual Gun Detection Using Harris Interest Point Detector," *Procedia Comput. Sci.*, vol. 54, pp. 703–712, 2015.

[5] M. Grega, A. Matiolański, P. Guzik, and M. Leszczuk, "Automated detection of firearms and knives in a CCTV image," *Sensors (Switzerland)*, vol. 16, no. 1, 2016.

[6] R. Olmos, S. Tabik, and F. Herrera, "Automatic Handgun Detection Alarm in Videos Using Deep Learning," University of Granada, 2017.

[7] Stadista, "Number of robberies in the U.S. 2017, by weapon | Statistic," 2017. [Online]. Available: https://www.statista.com/statistics/251914/number-of-robberies-in-the-us-by-weapon/. [Accessed: 13-Dec-2018].

[8] EveryTown For Gun Safety Support Fund, "Gun Violence in America," 2018. [Online]. Available: https://everytownresearch.org/gun-violence-america/. [Accessed: 13-Dec-2018].

[9] United Nations Office on Drugs and Crime Research; Trend Analysis Branch; Division of Policy Analysis and Public Affairs, "Global Study On Homicide

2013," Vienna, 2013.

[10] Instituto Nacional de Estadísticas y Censos - INEC, "Delitos de mayor connotación psicosocial Enero 2018," Ecuador, 2018.

[11] Cámara Nacional de Comercio y Servicios, "Segunda Edición Manual de Recomendaciones: Sistema Seguridad Electrónico," Uruguay, 2017.

[12] M. Muñoz and J. Rubio, "Diseño del Sistema de CCTV para Hospital Centro E.S.E En Planadas Tolima," Universidad Cooperativa de Colombia, 2018.

[13] E. Hidalgo, "SISTEMA CCTV (CIRCUITO CERRADO DE TELEVISIÓN) ENTRE EDIFICIOS, PARA LA SEGURIDAD Y VIGILANCIA EN EL AEROPUERTO INTERNACIONAL COTOPAXI," Universidad Técnica de Ambato, 2012.

[14] A. R. Pathak, M. Pandey, and S. Rautaray, "Application of Deep Learning for Object Detection," *Procedia Comput. Sci.*, vol. 132, pp. 1706–1717, 2018.

[15] W. Rawat and Z. Wang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review," *Mit Press J.*, vol. 29, pp. 2352–2449, 2017.

[16] Matlab, "Redes Neuronales Convolucionales." [Online]. Available: https://la.mathworks.com/solutions/deep-learning/convolutional-neural-network.html. [Accessed: 13-Dec-2018].

[17] T. Dettmers, "Deep Learning in a Nutshell: Core Concepts," 2015. [Online]. Available: https://devblogs.nvidia.com/deep-learning-nutshell-core-concepts/. [Accessed: 13-Dec-2018].

[18] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," New York, 2013.

[19] Google, "ML Practicum: Image Classification," 2018. [Online]. Available: https://developers.google.com/machine-learning/practica/image-classification/convolutional-neural-networks. [Accessed: 13-Dec-2018].

[20] S. Amidi and A. Amidi, "CS 230 - Convolutional Neural Networks Cheatsheet." [Online]. Available: https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks. [Accessed: 13-Dec-2018].

[21] T. Dettmers, "Deep Learning in a Nutshell: History and Training | NVIDIA Developer Blog," 2015. [Online]. Available: https://devblogs.nvidia.com/deep-learning-nutshell-history-training/. [Accessed: 13-Dec-2018].

[22] Stanford, "CS231n Convolutional Neural Networks for Visual Recognition," 2017. [Online]. Available: http://cs231n.github.io/convolutional-

networks/#pool. [Accessed: 13-Dec-2018].

[23] J. Redmon and A. Farhadi, “YOLO: Real-Time Object Detection,” 2018. [Online]. Available: https://pjreddie.com/darknet/yolo/. [Accessed: 13-Dec-2018].

[24] T.-Y. Lin *et al.*, “Microsoft COCO: Common Objects in Context,” May 2014.

[25] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” *arXiv*, 2018.

[26] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” *arXiv*, Jun. 2015.

[27] M. Abadi *et al.*, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,” 2015.

[28] Google, “TensorFlow,” 2017. [Online]. Available: https://opensource.google.com/projects/tensorflow. [Accessed: 13-Dec-2018].

[29] Qt Creator, “Qt APIs & Libraries, Tools and IDE | Qt,” 2018. [Online]. Available: https://www.qt.io/qt-features-libraries-apis-tools-and-ide/. [Accessed: 13-Dec-2018].

[30] Google Cloud, “Unidad de procesamiento de gráficos (GPU) | Google Cloud,” 2018. [Online]. Available: https://cloud.google.com/gpu/. [Accessed: 13-Dec-2018].

[31] P. Kapica, “Implementing YOLO v3 in Tensorflow (TF-Slim) – ITNEXT,” 2018. [Online]. Available: https://itnext.io/implementing-yolo-v3-in-tensorflow-tf-slim-c3c55ff59dbe. [Accessed: 13-Dec-2018].

[32] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *arXiv*, Sep. 2014.

[33] M. D. Zeiler and R. Fergus, “Visualizing and Understanding Convolutional Networks,” *arXiv*, Nov. 2013.

[34] Python Software Foundation, “Python.” [Online]. Available: https://www.python.org/. [Accessed: 13-Dec-2018].

[35] A. Menditto, M. Patriarca, and B. Magnusson, “Understanding the meaning of accuracy, trueness and precision,” *Accredit. Qual. Assur.*, vol. 12, no. 1, pp. 45–47, Jan. 2007.

[36] P. Galdi and R. Tagliaferri, “Data Mining: Accuracy and Error Measures for Classification and Prediction,” in *Encyclopedia of Bioinformatics and Computational Biology*, Elsevier, 2018, pp. 431–436.

[37] A. M. Hamad, D. Salah Elhadary, and A. Omar Elkhateeb, “Multimodal

Biometric Personal Identification System Based On IRIS & Fingerprint," *Int. J. Comput. Sci. Commun. Networks*, vol. 3, pp. 226–230.

[38] K.-A. Toh, J. Kim, and S. Lee, "Biometric scores fusion based on total error rate minimization," *Pattern Recognit.*, vol. 41, no. 3, pp. 1066–1082, Mar. 2008.

[39] T. Owuye, I. Awoyelu, and S.Bamiwuye, "Development of a Multimodal Biometric Model for Population Census," *Am. J. Signal Process.*, vol. 7, pp. 25–37, 2017.

[40] N. Sánchez Anzola, "Máquinas de soporte vectorial y redes neuronales artificiales en la predicción del movimiento USD/COP spot intradiario," *Odeon*, no. 9, p. 113, 2016.

[41] J. Davis and M. Goadrich, "The Relationship Between Precision-Recall and ROC Curves," Pittsburgh, 2006.