# Detector de Personas con Armas de Fuego a Partir de un Sistema de Visión Artificial Basado en el Análisis de Posturas Corporales

Alfonso Durán Caicedo

Trabajo de grado para optar el título de Maestría en Ingeniería Electrónica

Director

Juan Manuel Calderón Chávez

PhD. In Electrical Engineering

Universidad Santo Tomás

Facultad de Ingeniería Electrónica

Maestría en Ingeniería Electrónica

2022

## AGRADECIMIENTOS

En primer lugar, a Dios quien es mi ayuda incondicional, soporte invaluable de mi fortaleza física y mental, en quien siempre encuentro los recursos necesarios para no desfallecer, para poder hacer frente a toda vicisitud presentada.

Quiero hacer un reconocimiento especial a mi esposa por el apoyo y comprensión brindados, quien durante el tiempo de estudio de la Maestría y aún más, el dedicado al desarrollo y ejecución de este Proyecto de Investigación fue gran baluarte para superar todos y cada uno de los obstáculos presentados.

A mi madre, que me apoyó en todo aspecto y de quien recibí palabras de aliento en momentos difíciles que me ayudaron a esforzarme y no desfallecer en mi empeño.

A mis hijos, quienes me brindaron su apoyo moral y mostraron siempre su deseo ferviente para que logrará alcanzar este objetivo.

A mis familiares y amigos que directa e indirectamente colaboraron en el cumplimiento de esta meta.

Al Ingeniero Carlos Enrique Montenegro, decano de la facultad por su acompañamiento y colaboración dados en el tiempo dedicado a este estudio posgradual.

A mi director, el Ingeniero Juan Manuel Calderón quien, a pesar de sus innumerables ocupaciones, apartó de su tiempo para brindarme el acompañamiento y dirección necesarios durante el transcurrir de la ejecución y desarrollo de este trabajo.

Finalmente, a mi alma mater, a mis compañeros y docentes de la Maestría, ya que con ellos pude compartir conocimiento, reafirmando conceptos, los que a la postre fortalecieron y redundaron en el resultado de esta investigación.

Contenido

Lista de tablas

Lista de figuras

**Resumen**

En el entorno social cada día se aprecia como la inseguridad es un mal agobiante, tanto para las personas como para la comunidad en general. Los objetos más utilizados para perpetrar este tipo de actos criminales son las armas de fuego. Con este proyecto se propone implementar un sistema que basado en visión artificial que pueda mediante algoritmos Deep Learning y herramientas de posicionamiento del cuerpo como OpenPose reconocer una persona con un arma de fuego. La clasificación de los objetos o armas de fuego se implementan con redes neuronales convolucionales (CNN). Para hacer más rápido y efectivo este procesamiento de imágenes, se utilizarán varias técnicas de desarrollo sobre Google Colab, Jupyter Lab, Oracle Máquina Virtual y Git Bash, aprovechando también la utilización de una tarjeta GPU modelo RTX A5000 NVIDIA para mayor rapidez en la ejecución de cada uno de los pasos del desarrollo propuesto.

Las fases del proyecto propuesto son cinco principalmente: elaboración base de datos, entrenamiento del modelo, validación, implementación OpenPose, resultados del modelo. También tiene varias subfases que permiten la implementación eficiente del proyecto. Finalmente, el sistema permitió confirmar la actividad peligrosa con arma de fuego mediante detecciones obtenidas a través de videos en tiempo real.

**Abstract**

In the social environment, every day it is appreciated how insecurity is an overwhelming evil, both for people and for the community in general. The most used objects to perpetrate this type of criminal acts are firearms. With this project, it is proposed to implement a system based on artificial vision that can, through Deep Learning algorithms and body positioning tools such as OpenPose, recognize a person with a firearm. The classification of objects or firearms is implemented with convolutional neural networks (CNN). To make this image processing faster and more effective, several development techniques will be used on Google Colab, Jupyter Lab, Oracle Virtual Machine and Git Bash, also taking advantage of the use of an NVIDIA RTX A5000 model GPU card for faster execution of each of the steps of the proposed development.

The proposed project phases are five: database development, model training, validation, OpenPose implementation, model results. It also has several sub-phases that allow the efficient implementation of the project. Finally, the system made it possible to confirm the dangerous activity with a firearm through detections obtained through videos in real time.

## Introducción

Los niveles de inseguridad que aquejan a este país hace ya un largo tiempo, pero que han venido acrecentándose en los últimos años, son reflejo de una sociedad permeada por la criminalidad y el caos social, situación que cada día afecta más. “Los resultados de un reciente estudio del Centro de Estudios Futuros Urbanos en el cual, basado en cifras de la Secretaría de Seguridad, se reveló que el hurto de celulares con arma de fuego creció un 65% entre los años 2019 y 2020 siendo los buses del SITP los más afectados por este flagelo. Además, se evidenció que en los últimos tres años las zonas con más denuncias de hurto a personas son Chicó-Lago y las localidades con mayor hurto a personas entre 2018 y 2020 fueron Ciudad Bolívar y san Cristóbal.” [1].

En el entorno social cada día se aprecia cómo la inseguridad y las conductas delictivas son una constante. Para los años 2018 y 2019, según el Instituto Nacional de Medicina Legal y Ciencias Forenses de nuestro país el número de homicidios fue de 11.297 y 11.630 respectivamente¨ [2].

Con relación al resultado de esta implementación tecnológica se requiere que este sistema este acompañado de la captación de imágenes en tiempo real, las que serán evaluadas constantemente por una variedad de instrucciones computacionales permitiendo detectar e identificar positivamente los objetos propuestos, es decir las armas de fuego y las personas con armas dado su posicionamiento corporal. Los sistemas utilizados para detección de armas de fuego en general se basan únicamente en el reconocimiento del objeto. No se tienen en cuenta parámetros adicionales, ni se utilizan otras herramientas que involucren la postura corporal y la métrica de inclinación de las extremidades superiores para validar el reconocimiento.

Es así como se propone un sistema de reconocimiento de armas de fuego con visión artificial basado en la postura corporal. Este sistema validará efectivamente la detección del arma teniendo en cuenta la postura corporal del individuo que la porta y los ángulos de inclinación que los brazos que portan el arma de fuego y que se miden en la acción [3].

Este sistema va más allá de los modelos convencionales aportando el reconocimiento del arma de fuego y su validación con base en la posición corporal de la persona que la porta y observando adicionalmente las métricas relacionadas. En conjunto el sistema permite confirmar una actividad peligrosa, posiblemente delictiva en tiempo real [4].

# 1. Detector de Personas con Armas de Fuego a Partir de un Sistema de Visión Artificial Basado en el Análisis de Posturas Corporales

## 1.1 Planteamiento del problema

En muchos lugares del mundo se sufre continuamente con uno de los más significativos problemas actuales, la inseguridad [5], [6], [7], [8]. Para atacar este flagelo y minimizar la cantidad de acciones delincuenciales se deben tener sistemas de captación de video que autónomamente sean capaces de detectar cualquier posible actividad peligrosa con armas de fuego. En este tipo de sistema lo primordial es lograr en un tiempo real la identificación concreta de las armas y las personas que las portan y que se pueda efectuar a través de una imagen visualizada por una cámara de vigilancia.

Lo importante aquí es que se pueda apreciar como objeto principal el arma de fuego, si esto sucede, posiblemente se está llevando a cabo algún acto delictivo [9]. Cuando se ha detectado un arma de fuego dentro de un video de una cámara de vigilancia, el siguiente paso será que se haga la debida identificación y adicionalmente, con el apoyo de otras herramientas lógicas evalúe la postura del cuerpo de una persona que apunta su arma, pudiendo comparar el posicionamiento de su cuerpo, de su cabeza, brazos con relación al arma y a su torso, con las métricas previamente asignadas para posturas corporales. Mientras se hace este rastreo el sistema podrá generar una señal audible que alertará a la comunidad y a las autoridades sobre la situación suscitada, esperando que puedan desplazarse al sitio prontamente.

Lo que se desea con este trabajo es detectar en tiempo real mediante video a una persona con un arma de fuego, observando adicionalmente, la postura del cuerpo con respecto al arma mientras apunta con ella, reforzando la certeza de una situación peligrosa, potencialmente delictiva con el uso de armas de fuego.

A partir de la problemática previamente planteada, se establece la siguiente pregunta de investigación:

¿Qué características debe tener un Sistema Tecnológico basado en Visión Artificial con capacidad de detectar personas con armas de fuego basado en posturas corporales que permita identificar y minimizar posibles acciones delictivas o lesivas hacia las personas?

Asimismo, se formulan las siguientes hipótesis:

- Es factible detectar personas con armas de fuego con un sistema basado en visión artificial.
- Es factible detectar y minimizar posibles acciones delictivas a partir de la de la identificación de personas con armas de fuego y su postura corporal con un sistema basado en visión artificial.
- Es factible integrar sistemas de visión artificial para detectar y minimizar posibles conductas delictivas o lesivas en los seres humanos.

**1.2 Justificación**

Los niveles de inseguridad que aquejan a este país hace ya un largo tiempo, pero que han venido acrecentándose en los últimos años, son reflejo de una sociedad permeada por la criminalidad y el caos social, situación que cada día nos afecta más. "...basado en cifras de la Secretaría de Seguridad, se reveló que el hurto de celulares con arma de fuego creció un 65% entre los años 2019 y 2020 siendo los buses del SITP los más afectados por este flagelo..." [1].

Así mismo, lo que más inquieta es el grado de violencia que se ha intensificado en estos hechos de criminalidad que cada vez son más frecuentes en esta sociedad y para los cuales no se ve una pronta solución, pues así se incremente el pie de fuerza policial, se sabe que la delincuencia aprovecha sus descuidos o simplemente, espera que la autoridad se desplace del lugar para atacar y cometer sus actos criminales [10].

Es por esto que, para poder minimizar este flagelo lacerante en el país, se hace necesario aplicar soluciones eficaces que por medio de la tecnología y los últimos logros en el ámbito de Inteligencia Artificial y más puntualmente de Visión Artificial, permitan detectar e identificar armas de fuego y personas que las utilicen para amenazar o simplemente apunten con ellas pudiendo ocasionar un mal o daño grave a algo o alguien. No obstante, haberse implementado sistemas de reconocimiento de armas basados en modelos de aprendizaje profundo [11] aún persisten los falsos positivos que afectan su eficiencia.

Con relación al resultado de esta implementación tecnológica se requiere que este sistema este acompañado de la captación de imágenes en tiempo real. Estas serán evaluadas constantemente por un conjunto de algoritmos computacionales [4],[12], que permitirán realizar detección e identificación positivamente de los objetos propuestos, pudiendo alertar en debida forma en primer lugar a la comunidad y por supuesto, a las autoridades competentes que deberán hacer presencia rápidamente en el lugar de los hechos para someter a él o los delincuentes y llevarlos ante la justicia para que se les aplique y haga efectivo todo el peso de la Ley. Este sistema propone una solución efectiva a esta problemática en tiempo real, apoyándose en la visión artificial y herramientas de análisis de postura corporal humana.

La ciencia y la tecnología brindan una serie de elementos que, si son bien aplicados y encaminados en la manera justa, permitirán mitigar y porque no, algún día poder erradicar por completo la inseguridad y la delincuencia del territorio nacional y porque no del mundo entero.

## 1.3 Objetivos

### *1.3.1 Objetivo general*

Desarrollar un Sistema Tecnológico basado en Visión Artificial, que permita detectar personas con armas de fuego que impliquen acciones delictivas o lesivas hacia las personas.

### *1.3.2 Objetivos específicos*

- Crear un repositorio compuesto de imágenes de armas de fuego y posturas corporales de personas a partir de dataset públicos o creados de forma propia.
- Entrenar un sistema de aprendizaje de máquina que permita reconocer armas de fuego y posturas corporales de personas en las imágenes del repositorio creado para este proyecto.
- Entrenar un modelo de aprendizaje de máquina, que permita realizar la detección de situaciones peligrosas, a partir de las características extraídas.
- Validar el desempeño de los modelos entrenados, a partir de métricas apropiadas según los métodos seleccionados y muestras previamente etiquetadas.

# 2. Marco referencial

## 2.1 Marco teórico

Se hace indispensable tener claridad en los conceptos sobre las técnicas y las herramientas que se utilizan en la ejecución y desarrollo de este Proyecto investigativo, ya que la documentación y la información relacionadas son parte importante en la fundamentación y el alcance de los objetivos propuestos.

### *2.1.1 Visión por computador*

Dentro de la cobertura que tiene la Inteligencia Artificial, está rama se direcciona a que los computadores puedan adquirir datos e información de cualquier imagen, para que a partir de esto se originen mejores resultados que se puedan aplicar en el diario vivir.

Los computadores pueden "ver" con el uso de aparatos de video, específicamente cámaras con las que estén conectados. Todo esto ocurre a raíz de la captura de imágenes por fotografía y video con el uso de equipos especializados para este propósito. Con estas imágenes se procede a realizar una conversión en otras imágenes con información mejorada, conociéndose como "procesamiento de imágenes". Acto seguido, la información resultante es utilizada según los requerimientos planteados, en la busca de soluciones efectivas.

Estas cámaras pueden tomar fotografías o videos. Las imágenes se tratan y procesan para convertirlas en nuevas imágenes con mejor información. Esto se logra utilizando herramientas que permiten hacer mejoras en sus características. Comúnmente se involucra el uso de filtros para implementar el método de procesamiento de imágenes [13].

Los algoritmos de entrenamiento en la visión por computador, utilizan tecnologías que tienen su eje central en el aprendizaje profundo, el cual se desprende del aprendizaje automático. Muchas de las técnicas y aplicaciones para visión por computador tienen su soporte principal en

las redes neuronales convolucionales. Estas CNN son utilizadas para hacer que un computador aprenda acerca de las características de objetos a partir de imágenes varias. Con una buena cantidad de datos el computador distinguirá una imagen de otra. El Deep Learning o aprendizaje profundo permite entrenar algoritmos direccionados a esta área de la computación. Es así como en las redes neuronales convolucionales (CNN) encuentran su asidero una gran cantidad de expertos en el campo de la visión computarizada.

Las redes neuronales convolucionales facilitan que un computador a partir de imágenes aprenda, ya que si se le dan los datos necesarios podrá aprender a diferenciar una imagen de otra. Es así que, mediante la aplicación de un modelo de estas redes, las imágenes ya etiquetadas pasan por un proceso de pixelado, esto se conoce como anotación de imágenes conjunto del aprendizaje automático [4].

Por ende, el etiquetado es requerido en estas convoluciones y posteriores predicciones para también confirmar su fiabilidad y precisión en el cumplimiento de lo esperado.

**2.1.1.1 Cerebro humano base de la visión por computador.** La visión por computador trabaja reconociendo imágenes, "observando" imágenes parecidas, como lo haría un ser humano a partir del aprendizaje alcanzado fruto de las peculiaridades y patrones extractados, así como el grado de confianza alcanzado. Al fin de cuentas, lo que las redes neuronales convolucionales hacen es imitar al hombre en su toma de decisiones [14], [15].

**2.1.1.2 La visión artificial y el hombre.** El aprendizaje profundo involucra un gran costo computacional, teniendo mucho que ver los recursos computacionales y una buena cantidad de datos para el entrenamiento de sus modelos.

Las máquinas aprenden autónomamente mediante algoritmos de aprendizaje profundo, sin que intervenga algún desarrollador ni de una programación para reconocimiento de imágenes haciendo uso de particularidades específicas [16].

Algunas funcionalidades de los sistemas de visión por computador se relacionan [17] así:

- Adquisición de imágenes: Se utiliza regularmente un equipo captador de imágenes para entregar la información recaudada de la imagen o el video. Para esto se puede utilizar cualquier clase de cámara.
- Preprocesamiento: Toda imagen a ser utilizada requiere un trato previo en espera de mejor calidad y resultado en el proceso de la visión por computador. En este paso se incluyen mejoras en diferentes parámetros.
- Algoritmo de visión por computador: Se encarga de detectar el objeto, segmentar la imagen y también su clasificarla, todo dependiendo de la fuente de imagen o video. El algoritmo de procesamiento de imágenes, realiza la detección de objetos, la segmentación de imágenes y la clasificación en cada imagen o fotograma de video.
- Lógica de automatización: El resultado obtenido como salida se caracteriza teniendo en cuenta los datos recopilados luego del ejercicio computacional.

El propósito esencial de la visión por computador no es otro que apoyar los computadores en la comprensión e interpretación de imágenes.

Se basa principalmente en los datos recopilados de las imágenes involucradas. Dicha información puede ser dada por acomodación y posicionamiento de la cámara, reconociendo, agrupando y buscando.

### *2.1.2 Las redes neuronales convolucionales (CNN)*

En estas redes neuronales las “neuronas” en función de campos receptivos se semejan las neuronas de la corteza visual humana. Esta red se desprende del llamado “perceptrón” pero en este caso de varias capas, no obstante, que utilizan en su ejecución matricialmente dos dimensiones, muestran gran eficacia en aplicaciones de visión artificial, así como en las de clasificación [18].

Los núcleos de las CNN se asemejan a receptores que pueden responder a diversas características. Las funciones de activación simulan una función que solo las señales eléctricas neuronales pueden superar. Una información base se transmite a la siguiente neurona. Funciones de pérdida y optimizadores son usadas para enseñar a todo el sistema CNN a aprender lo que se espera aprenda.

En comparación con las redes neuronales artificiales [19], CNN posee muchas ventajas:

1) Conexiones locales. cada neurona ya no está conectada a todas las neuronas de la capa anterior sólo a un pequeño número de neuronas, que es eficaz en reducir parámetros y acelerar la convergencia.

2) Intercambio de pesos: Un grupo de conexiones puede compartir los mismos pesos, lo que reduce aún más los parámetros.

3) Muestreo descendente por reducción de dimensionalidad. Una capa de agrupación aprovecha el principio de correlación local de imagen para reducir la muestra de una imagen y la cantidad de datos mientras conserva información útil. También puede reducir el número de parámetros para eliminar características superficiales. Estas características hacen de las CNN una de las herramientas más representativas en el campo del aprendizaje profundo.

La convolución es un paso fundamental para la extracción de características. Las salidas de la convolución se pueden llamar mapas de características. Al establecer un núcleo de convolución con un cierto tamaño, se pierde información en el borde [20]. Por lo tanto, el relleno es introducido para ampliar la entrada con valor cero, que puede ajustar el tamaño indirectamente.

Los mapas de características contienen una gran cantidad de características que tienden a causar problema de sobreajuste. Como resultado, la agrupación (también se conoce como reducción de muestreo) se propone para obviar la redundancia, incluida la agrupación máxima y la agrupación promedio.

Las redes neuronales convolucionales pueden aprovechar diferentes funciones de activación para expresar características complejas. Similar a la función del modelo neuronal del cerebro humano, la función activación es una unidad que determina qué información debe transmitirse a la siguiente neurona. Cada neurona de la red neuronal acepta el valor de salida de las neuronas de la capa anterior como entrada, y pasa el valor procesado a la siguiente capa. En una red neuronal multicapa, hay una función entre dos capas. Esta función se llama función de activación.

Si no se utiliza una función de activación o se utiliza una función lineal, la entrada de cada capa será una función lineal de la salida de la capa anterior. En este caso, no importa cuántas capas tenga la red neuronal, la salida es siempre una combinación lineal de la entrada, lo que significa que las capas ocultas no tienen ningún efecto. Esta situación es el perceptrón primitivo que tiene una capacidad de aprendizaje limitada. Por esta razón, las funciones no lineales se introducen como funciones de activación [21].

Teóricamente, las redes neuronales profundas con funciones no lineales de activación pueden aproximarse a cualquier función, que mejora en gran medida la capacidad de las redes neuronales para ajustar los datos.

La función sigmoidea es una de las funciones de activación no lineales más típicas. De igual forma, la Unidad Lineal Rectificada (ReLU) es otra función de activación efectiva. En comparación con la función sigmoidea una ventaja significativa de usar la función ReLU es que puede acelerar el aprendizaje.

### *2.1.3 Programación paralela*

El objetivo principal de la programación paralela es aumentar el rendimiento de una aplicación, ejecutándola en múltiples procesadores. No obstante, esta programación se ha asociado tradicionalmente con la llamada "comunidad informática de alto rendimiento", se está volviendo más frecuente para la computación convencional, debido al desarrollo reciente de la conocida "arquitectura multinúcleo de productos básicos".

Existen dos enfoques primordiales para la paralelización de aplicaciones, la automática y la programación paralela. Estas difieren entre sí en cuanto al rendimiento que puede llegar a alcanzar la aplicación y la facilidad para realizarlo.

La programación paralela da como resultado una mejor ganancia en el rendimiento que la paralelización automática. Es muy importante tener en cuenta también los aspectos cualitativos de los modelos.

Es sin lugar a dudas un avance informático que facilita poder desarrollar modelos matemáticos o implementaciones a la par. Así las cosas, ejercicios extensos también es posible dividirlos en partes más pequeñas para ser solucionadas al mismo tiempo.

Se tiene claro que hay casos en que esta herramienta informática es diáfana, pero así mismo se encuentran situaciones en que los que se fundamentan en la concurrencia causan dificultades en su ejecución ya que se pueden presentar errores de código [22].

**2.1.3.1 Speedup.** Este proceso se puede evidenciar cuando se corre un software de computador usando varios procesadores (CPU) de forma coincidente a cambio de solo usar un procesador. El Speedup o aceleración relativa, se ocupa del paralelismo inherente del algoritmo paralelo, mostrando la posible eficiencia en la implementación.

También este dado por la relación de tiempo transcurrido del algoritmo paralelo en un procesador al tiempo transcurrido del algoritmo paralelo en N procesadores. La razón para usar la aceleración relativa es que el rendimiento de los algoritmos paralelos varía con el número de procesadores disponibles, comparando el propio algoritmo con diferente número de procesadores da información sobre las variaciones y degradaciones del paralelismo [23].

### *2.1.4 Unidad de procesamiento gráfico (GPU)*

Es el conjunto de componentes electrónicos especializados que se encargan de implementar un procesador con varios núcleos que garantizan la funcionalidad de una memoria para incrementar y hacer más veloz la generación de una imagen o un resultado computacionalmente hablando.

En sus comienzos, la GPU era un procesador de función fija, construido alrededor de una tubería de gráficos, que se destacó en gráficos 3D. Desde entonces, la GPU se ha convertido en un potente procesador programable, con API y hardware centrándose cada vez más en los aspectos programables. El resultado es un procesador con una enorme capacidad aritmética [24].

La GPU tiene una arquitectura distintiva centrada en torno a un gran número de procesadores paralelos de grano fino. En los últimos años, la GPU ha evolucionado de una función

fija, procesador de propósito especial en un procesador programable paralelo completo con funcionalidad adicional de función fija y propósito especial. Más que nunca, los aspectos programables del procesador han ocupado un lugar central.

El desafío para los proveedores e investigadores de GPU ha sido atacar el equilibrio adecuado entre el acceso de bajo nivel al hardware para habilitar el rendimiento y los lenguajes de programación de alto nivel y herramientas que permiten al programador flexibilidad y productividad, todo frente al rápido avance del hardware.

### *2.1.5 CUDA*

Es un tipo de plataforma o capa lógica que apoyada en el paralelismo computacional ofrece la facilidad de acceso a las unidades de procesamiento gráfico (GPU) de la marca NVIDIA, consta de un conjunto de extensiones de lenguaje C y una biblioteca de tiempo de ejecución que proporciona API para controlar la GPU [25].

De esta manera el investigador podrá aprovechar al máximo todas las ventajas que estas brindan, haciendo mucho más eficientes y rápidos los desarrollos e implementaciones propuestos, teniendo como soporte el “kernel” o núcleo de cálculo que se ejecuta en la GPU [26].

### *2.1.6 YOLO*

Al ser un algoritmo de código abierto permite utilizarlo para la detección de objetos en tiempo real. Se puede hacer uso de diferentes tipos de red CNN ya que ofrece desde una red Nano (YOLOv5n) que es la red más sencilla. Además, como se aprecia en la Figura 1 se tienen como opciones otros cuatro tipos de red siendo la más grande XLarge (YOLOv5x). Esta última a pesar de ser más robusta da un rendimiento en velocidad de hasta 20 ms por cada imagen [27].

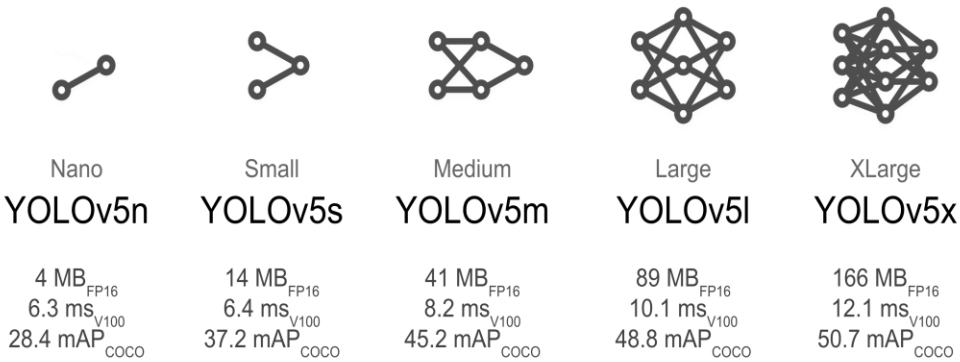


Figura 1. Representación de Modelos Redes YOLOv5. Texto original en idioma inglés tomado de la fuente.
Fuente: https://hashdork.com/es/yolo/

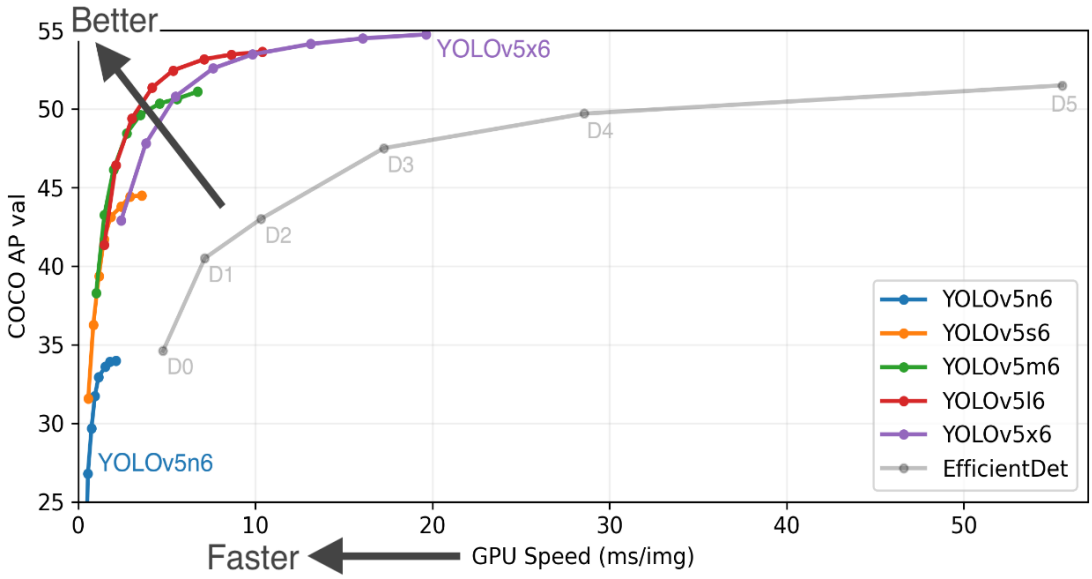


Figura 2. Rendimiento Modelos Redes YOLOv5. Fuente: https://github.com/ultralytics/yolov5

En la Figura 2 se observa el rendimiento de los 5 modelos de las redes (YOLOv5) cuando se implementan con un conjunto de datos llamados COCO. Es un conjunto de imágenes creado con el propósito de obtener mejores resultados con los modelos de reconocimiento de objetos como las redes neuronales convolucionales CNN.

### *2.1.7 Matriz de Confusión*

Para entender adecuadamente los resultados obtenidos se debe en primer lugar tener claro el concepto de lo que es la matriz de confusión y sus parámetros asociados, también denominados

métricas. Así que a continuación en la Figura 3, se visualiza el ejemplo de una matriz de confusión representada con un mapa de calor.

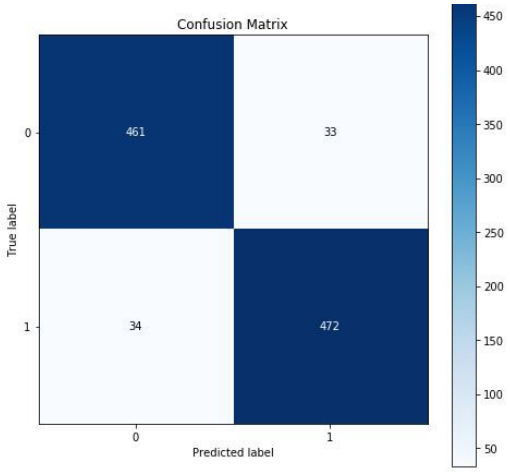


Figura 3. Ejemplo de Matriz de Confusión. Fuente: https://towardsdatascience.com/metrics-and-python-ii-2e49597964ff

La matriz de confusión es una herramienta que permite realizar el análisis de resultados conseguidos en las pruebas de identificación y clasificación.

Adicionalmente es muy importante aclarar los conceptos y estimación de cada una de las métricas de valoración que permiten evaluar un modelo implementado en relación con los resultados obtenidos.

En este ejemplo se aprecia una matriz para 2 tipos de clasificación. Así mismo, se observa que también se relacionan etiqueta verdadera y etiqueta predicha. Cabe indicar que los valores 461 y 472 que se relacionan en la figura mencionada corresponden en una matriz de confusión a las posiciones de a y d respectivamente. Ahora bien, si se habla en términos más aplicados se puede decir que 461 corresponde a TN (True Negative) y 472 corresponde a TP (True Positive).

De igual forma, tomando los valores restantes, se relacionan 33 y 34 que corresponden a las posiciones b y c respectivamente. Ahora bien, si se habla en términos más aplicados se puede decir que 33 corresponde a FP (False Positive) y 34 corresponde a FN (False Negative).

Se debe indicar que según estén ubicadas las etiquetas True y Predicted en la matriz los valores anteriormente descritos variarán en su correspondencia,

A continuación, se determinan las métricas más relevantes y se hallan los valores resultantes luego de efectuar las operaciones correspondientes, reemplazando en las fórmulas los términos por los valores descritos anteriormente.

**Exactitud (accuracy):** Es el porcentaje de predicciones acertadas con referencia al total.

Exactitud(accuracy)= $\frac{(TN+FP)}{(TN+TP+FP+FN)}$ **Ec.1** =

$$\frac{(461+33)}{(461+472+33+34)}$$

$$\frac{494}{1000} = 0{,}494$$

**Precisión:** Se refiere al porcentaje de casos positivos detectados

Precisión = $\frac{TP}{(TP+FP)}$ **Ec.2**

$$\frac{472}{472+33}$$

$$\frac{273}{275} = 0{,}934$$

**Sensibilidad**: (recall): Es la proporción de casos positivos que fueron correctamente identificados por el algoritmo

Sensibilidad = $\frac{TP}{(TP+FN)}$ **Ec.3**

$$\frac{472}{472+34} = 0.932$$

**Especificidad**: Son los casos negativos que el algoritmo clasificó correctamente

Especificidad = $\frac{TN}{(TN+FP)}$ **Ec.4**

$$\frac{461}{461+33} = 0{,}933$$

**Exactitud:** Se refiere a lo cerca que está el resultado del valor verdadero, es decir la cantidad de predicciones positivas que fueron correctas.

F1: Es de gran utilidad cuando la distribución de las clases es desigual

$$F1 = \frac{2 * Precision * Sensibilidad}{Precision + Sensibilidad} = \frac{2 * 0.934 * 0.932}{0.934 + 0.932} = \frac{1.740}{1.866} = 0.932 \quad \textbf{Ec.5}$$

**2.2 Estado del arte:**

Se han venido utilizando diferentes técnicas y herramientas para detectar armas de fuego. En primer lugar, se utilizó sistema de rayos X para este tipo de detección que se implementaron en aeropuertos y estaciones de metros, en varias ciudades del mundo. Sistemas basados en este tipo de detección han sido nombrados en artículos publicados [28]. Se utilizó segmentación y vectores para detección de armas escaneando equipajes con rayos X [29].

Con el uso del algoritmo AdaBoost también se implementaron detecciones de armas de fuego. A pesar de grandes intentos para la eficiencia de estas detecciones la técnica no permitió su rendimiento. Por esta razón aparecieron nuevos desarrollos basados en sistemas de videoseguridad. La detección automática de pistolas con visión artificial se hizo por primera vez en el año 2015. Este desarrollo se basó en segmentación de colores y agrupamiento de k-means para limpiar la imagen de objetos adicionales [30].

Últimamente se ha hecho uso del aprendizaje profundo en detección de armas de fuego mediante cámaras CCTV [31]. En primer lugar, un desarrollo mostraba una red CNN mientras que otro modelo utilizaba una red RCNN [32] consiguiendo mejores resultados. Otros autores han implementado otras técnicas similares [33], [34].

Para bajar los porcentajes de falsos positivos otros desarrolladores e investigadores han planteado diferentes sistemas como el de una cámara simétrica dual con el fin de aumentar el

rendimiento del modelo. Algunos modelaron los falsos positivos como errores filtrándolos para mejorar el rendimiento [35].

A pesar de que estos sistemas se preguntaron por las falsas detecciones, están basados en el objeto y su apariencia. Este proyecto utiliza un modelamiento adicional como es la postura corporal del individuo para mejorar la salida de este.

En los últimos años se han hecho estudios e investigaciones con el uso de las técnicas de aprendizaje profundo. Turau y Hlavac implementaron un método de reconocimiento de actuaciones humanas en imágenes usando posicionamiento corporal [36]. Implementan con un componente de descripción HOG, reconociendo actividades. También se hicieron enfoques de combinación de clasificador orientado con modelos mediante ángulos articulares y orientación del cuerpo.

En otras técnicas se encontraron redes recurrentes en el reconocimiento de acciones humanas [37]. Ese desarrollo usaba celdas de memoria a corto plazo para la clasificación del movimiento realizado, pero se utilizaba en este modelo la postura del cuerpo sino datos obtenidos de un móvil puesto en la cintura del individuo. Después ampliaron este sistema con el uso de Pose2D para un modelo LSTM. Esta pose bidimensional es empleada para reconocer la actividad humana. Más investigaciones referidas a este tema en función de la pose corporal y la identificación de imágenes han sido desarrolladas encontrando resultados de reconocimiento de acciones y actividades de las personas.

A pesar de que la postura corporal ha venido siendo implementada en sistemas de visión por ordenador de reconocimiento de gestos y actividades [38], no se le ha dado gran uso para la detección de armas de fuego. No obstante, algunos modelos han pretendido incorporar la información postural para detección de armas de fuego, pero aún no se ha demostrado su eficacia.

Varios enfoques de personas y armas de fuego utilizando detectores se comportan como filtros eliminando pistolas detectadas, pero persisten los falsos negativos [39].

Este trabajo aplica información de la postura corporal de un individuo que porta un arma de fuego y que a través del reconocimiento y la medición de ángulos de sus extremidades superiores se deduce en el modelo en conjunto si la actividad es delictiva o no.

# 3. Método

En esta sección se indican las partes de la técnica propuesta para la detección de armas de fuego y personas con armas de fuego desde el preprocesamiento de los datasets obtenidos hasta la detección sobre videos en tiempo real. Adicionalmente se observan los resultados obtenidos sobre varios entrenamientos de las redes utilizadas y se evalúan los pasos efectuados y las diferentes herramientas usadas en todo el procedimiento implementado. Finalmente se evalúan los experimentos al igual que los resultados conseguidos.

## 3.1 Método de detección

El procedimiento de clasificación tiene varias etapas, entre las cuales se destaca: la adquisición de imágenes para el dataset, etiquetado (labeling), preprocesamiento de imágenes, uso de transfer learning apoyado en el modelo de detección de objetos YOLOv5, entrenamiento de varios modelos que YOLOv5 ofrece como son YOLOv5s, YOLOv5l, YOLOv5x entre otros. Adicionalmente, el uso de herramientas colaborativas en la nube e interfaces de usuario, software de virtualización multiplataforma de código abierto. Se utilizan aplicaciones para entornos y medio de detección de posturas implementando Open Pose. Se requiere de diferentes pasos correlacionados para ajustar la información.

Mediante diagrama en bloques en la Figura 4 se presenta el funcionamiento general del método. Allí se aprecian 5 fases principales. Estas fases se subdividen en subgrupos para facilitar la comprensión de la metodología implementada.

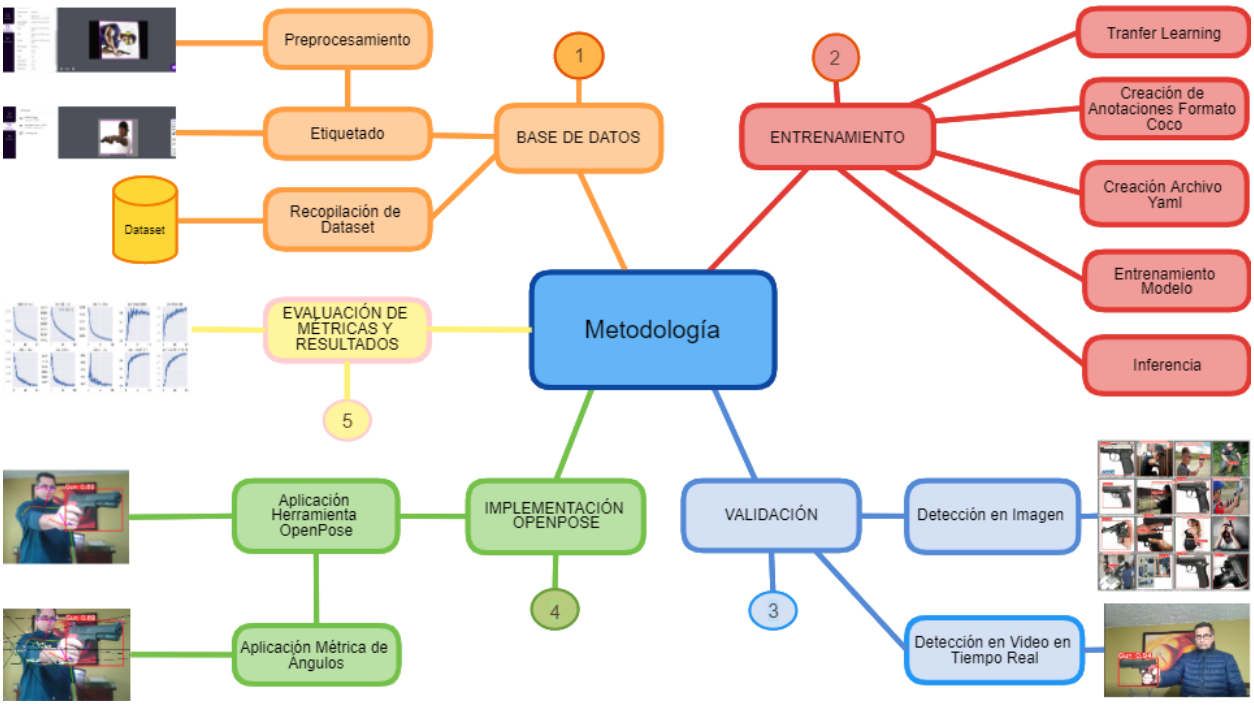


Figura 4. Metodología del proyecto. Fuente: Propia

### *3.1.1 Repositorio*

Conformado por un ***c***onjunto de imágenes de los objetos a clasificar que se recopilan y almacenan de manera organizada para formar el dataset necesario para el desarrollo de las fases posteriores de este proyecto. En esta base de datos se tiene un número significativo de imágenes de armas de fuego y de las personas portando armas y apuntando con ellas que han sido preprocesadas y etiquetadas para proseguir con la fase de entrenamiento [40],[41],[42].

**3.1.1.1 Recopilación de dataset.** Este conjunto de datos se implementa con la recopilación de imágenes extraídas de repositorios públicos obtenidos con motores de búsqueda de internet. Algunas imágenes muestran diferentes tipos de armas de fuego, en su mayoría pistolas y revólveres, otras presentan personas portándolas, apuntando con ellas y también en postura de amenaza a otras personas. En total se recopilan inicialmente alrededor de 3.000 imágenes. Luego de realizar una nueva revisión esta vez más exhaustiva se eliminan alrededor de 500 imágenes que no presentan buena resolución, por lo cual se efectúa una nueva búsqueda de imágenes logrando

captar 3.000 imágenes más para un total de 5.500 imágenes adquiridas. En la Figura 5 se relacionan los sitios web donde se obtuvieron los repositorios y sus respectivas direcciones electrónicas.

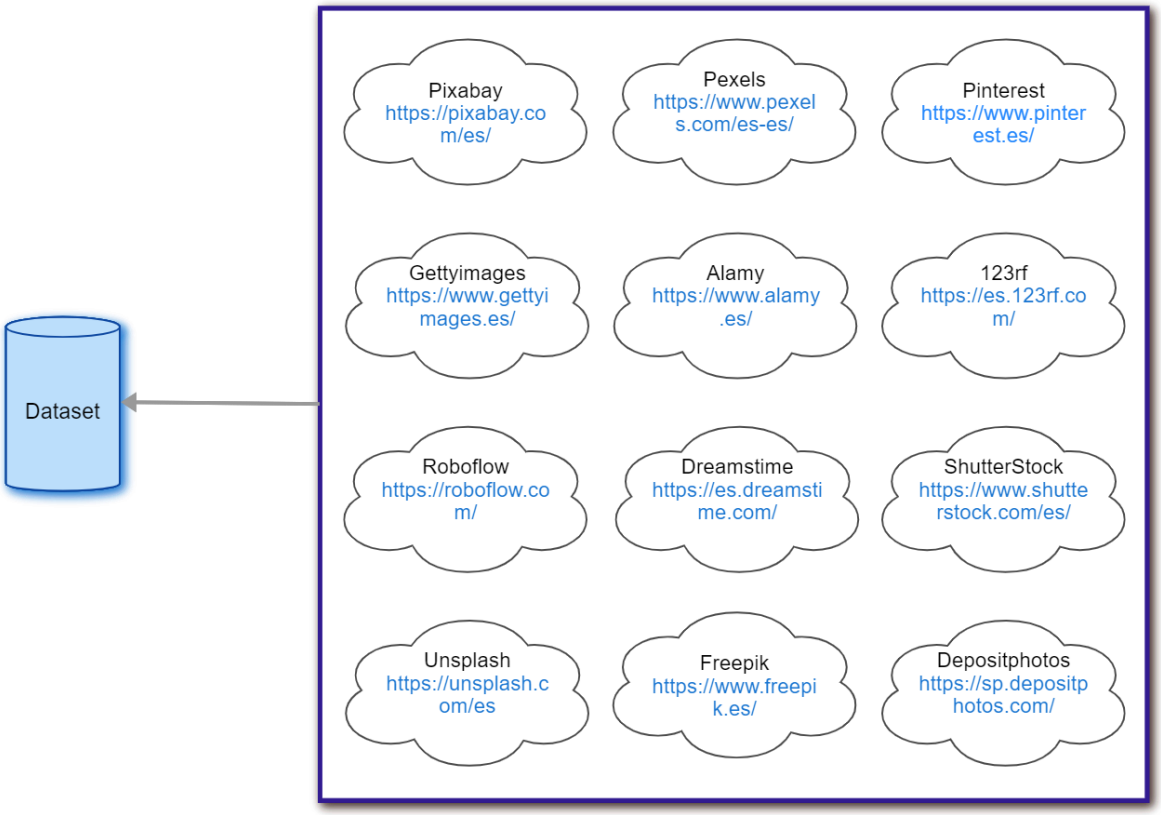


Figura 5. Relación repositorios. Fuente: Propia

**3.1.1.2 Etiquetado**. En este paso se realiza el etiquetado de las imágenes del dataset utilizando la herramienta de desarrollo Roboflow, con la cual se etiquetan dos clases de objetos Gun y Person With Gun. En la Figura 6 se puede observar el etiquetado de una imagen de un arma de fuego. Esta etiqueta se realiza a través de un cuadro delimitador (color verde). Por otra parte, se nombra la clase como “Gun” en el editor de anotación. Las herramientas que se usan en la fase de etiquetado en esta herramienta son 14 elementos numerados por el autor del 1-14 en color blanco o negro sobre la imagen original para un mejor entendimiento.

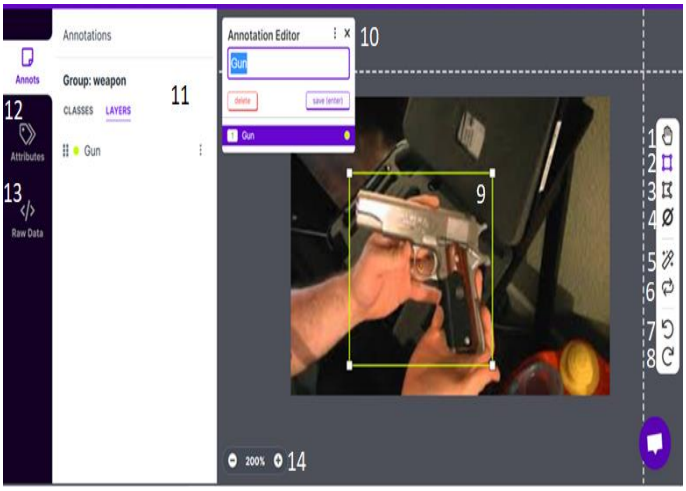


Figura 6. Etiquetado clase “Gun”. Fuente: https://app.roboflow.com/duran/gun-rj2ux/annotate/job/jtl2v5z6bP5zR6mXiWys

Si se observa la Tabla 1, aquí se relacionan cada uno de los 14 elementos que pueden ser usados en el etiquetado y que permiten mayor facilidad en esta fase.

| ***Elementos*** | ***Detalle*** |
|---|---|
| 1 | Herramienta de arrastre del cuadro delimitador |
| 2 | Permite crear el cuadro delimitador de forma manual |
| 3 | Permite crear el cuadro delimitador de forma manual, pero a la forma del objeto punto a punto |
| 4 | Permite anular el cuadro delimitador para volver a marcarlo sobre el objeto |
| 5 | Asistente de etiquetado para asignar nombre al dataset y tipo de modelo |
| 6 | Permite borrar el cuadro delimitador marcado anteriormente |
| 7 | Permite retroceder los pasos a pasos anteriormente ejecutados |
| 8 | Permite avanzar a pasos realizados posteriormente |
| 9 | Imagen del cuadro delimitador de color verde el objeto seleccionado |
| 10 | Editor de anotación para asignar el nombre o clase del objeto etiquetado |
| 11 | Cuadro de anotaciones donde se aprecian las clases etiquetadas |
| 12 | Describe las características de la imagen, como tipo de imagen, tamaño, fecha y hora de etiquetado |
| 13 | Permite ver el archivo de anotaciones del etiquetado |
| 14 | Porcentaje de aumento y disminución de zoom, que va de 50% al 1000% |

Tabla 1. Elementos para etiquetado. Fuente: Propia

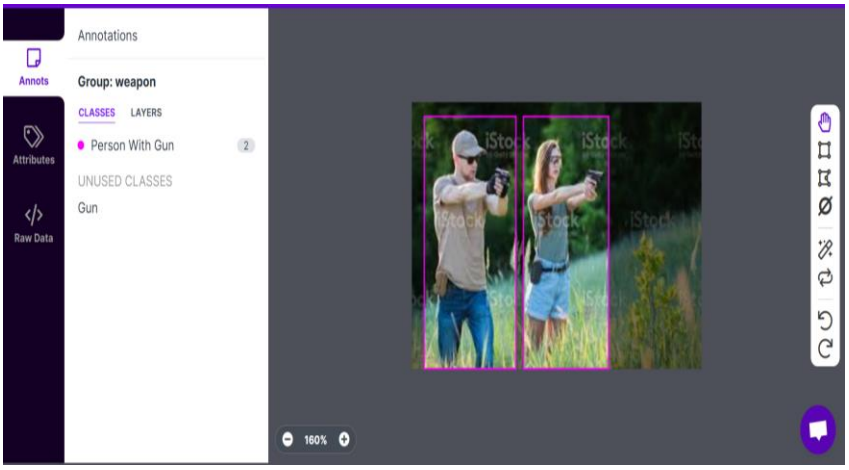


Figura 7. Clase “Person with Gun”. Fuente: Propia

En la Figura 7 se puede apreciar el resultado final de un etiquetado de 2 personas apuntando con armas de fuego. Estas etiquetas se visualizan por medio de los 2 cuadros delimitadores de color violeta. Las 2 etiquetas se nombran como clase “Person With Gun”.

De esta manera se etiqueta un total de 5.500 imágenes. Se aclara que este proceso se realiza de forma manual con cada una de las imágenes del dataset.

**3.1.1.3 Técnicas de preprocesamiento**. Siendo el preprocesamiento una herramienta indispensable previa al entrenamiento, se aplican diferentes opciones dadas por la plataforma de Roboflow como: crop, saturation, brightness, exposure, noise y otras, lográndose un incremento del dataset hasta completar un total de 6.337 imágenes etiquetadas.

En este proceso se aplican técnicas como las relacionadas en la Tabla 2, que son fundamentales en la tarea descrita en este aparte.

| ***Elementos*** | ***Función*** |
|---|---|
| *Crop* | Su función es recortar a través de un rectángulo delimitador que tiene manejadores en las |
| *Saturation* | Proporciona intensidad y limpieza en el matiz de la imagen |
| *Brightness* | Permite más intensidad lumínica. Es decir, mayor brillo |
| *Exposure* | Permite regular la cantidad de luz, capturando detalles de luces y sombra |
| *Noise* | Funciona con un filtro de tipo Gaussiano, por superposición agrega una capa de gris. También |
| *Hue* | Es el color mismo y se define por la longitud de onda, desde 380 hasta 780 nanómetros |

| | |
|---|---|
| *Grayscale* | Permite cambiar a diferentes tonalidades de grises |
| *Shear* | Permite recortar o esquilar contornos de la imagen |
| *Flip* | Permite hacer giros en la imagen seleccionando los grados y la dirección |

Tabla 2. Relación de elementos de preprocesamiento. Fuente: Propia

En la Figura 8 se aprecia el resultado obtenido luego de la aplicación de algunas técnicas de preprocesamiento en una de las imágenes etiquetadas.

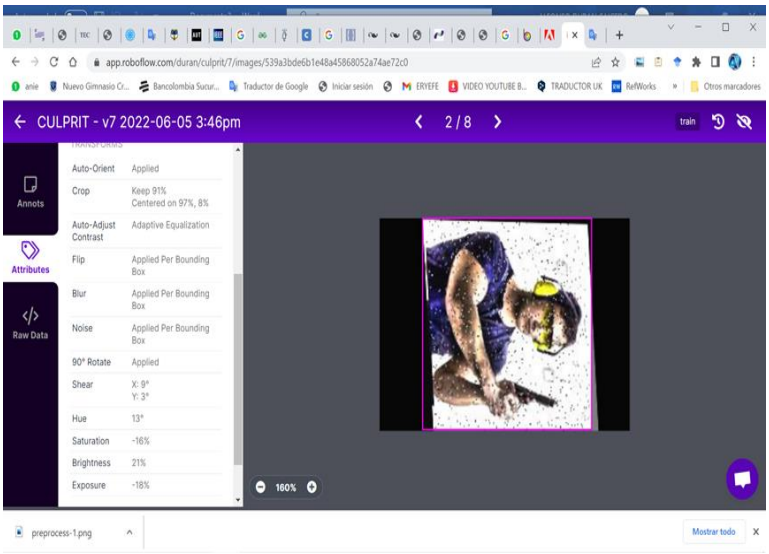


Figura 8. Ejemplo implementación preprocesamiento. Fuente: Propia

En la Figura 9 se aprecia el resultado de la aplicación de las herramientas de preprocesamiento descritas. Entre otras graysclale, crop y shear, esta última incide en los ejes x,y de la imagen para dar apariencia de recorte de sus lados.

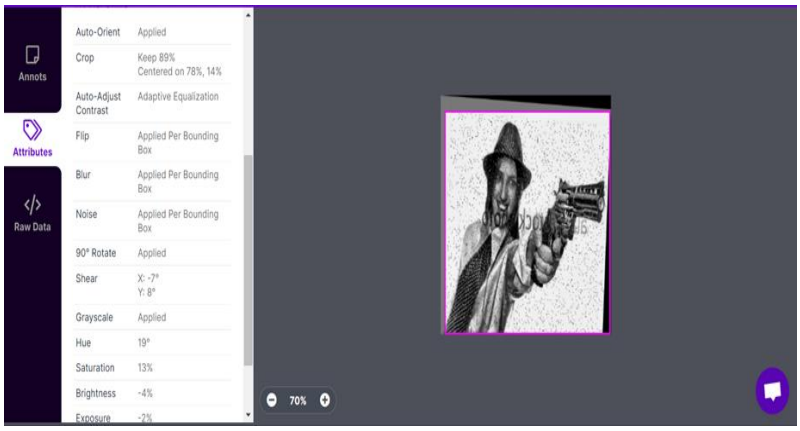


Figura 9. Ejemplo de Preprocesamiento Imagen. Fuente: propia.

Algunas técnicas se pueden implementar previo al etiquetado. Sin embargo, al aplicarse después permite obtener un aumento sobre el conjunto de imágenes delimitadas y nombradas para la fase de entrenamiento.

### *3.1.2 Entrenamiento*

Esta fase se divide en cinco subfases. Cada subfase tiene una condición especial que permite implementar el entrenamiento en una manera de eficaz.

**3.1.2.1 Transfer learning.** Utilizando el modelo de detección de objetos de YOLOv5 y aprovechando el resultado del entrenamiento preexistente, se toman los pesos del modelo ya entrenado continuando con el entrenamiento de la nueva red neuronal con las clases predefinidas, utilizando varios tipos de red CNN YOLOv5.

En la Figura 10 se aprecia el mapa de confusión obtenido luego del entrenamiento de la red tipo YOLOv5s.

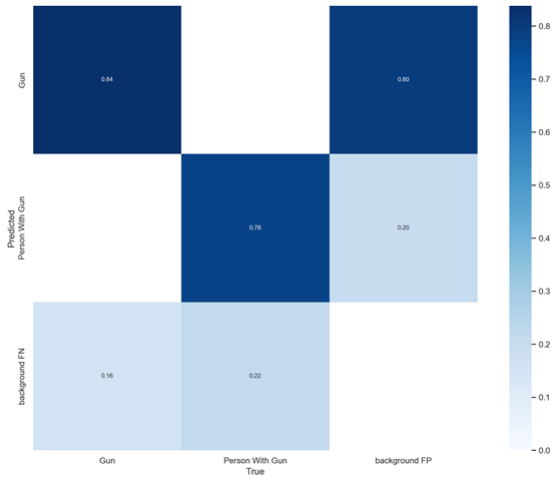


Figura 10. Mapa de confusión experimento 129. Fuente: Propia

Se pueden extractar en una escala posible de 0.0 a 1.0 como máximo, los siguientes resultados:

TP de la clase Gun= 0.84

TP de la clase Person With Gun=0.78

FP de la clase Gun= 0.80

FP de la clase Person With Gun= 0.20

Con los anteriores valores se calcula el recall que es igual a $\frac{TP}{TP+FN}$, reemplazando se obtiene recall= $\frac{0.84}{0.84+0.16} = \frac{0.84}{1} = 0.84$ es decir 84%, habilidad del modelo para detectar los casos relevantes.

Por otra parte, la precisión es igual a $\frac{TP}{TP+FP}$, reemplazando se obtiene, precisión= $\frac{0.84}{0.84+0.0.80}$

de donde $\frac{0.84}{1.64} = 0.512$ es decir 51.2%, el modelo puede ser preciso más no exacto.

$$F1 = \frac{2*Precision*Sensibilidad}{Precision+ Sensibilidad} = \frac{2* 0.512*0.84}{0.512+0.84} = \frac{0.86}{1.352} = 0.636 \quad \textbf{Ec. 1}$$

En este entrenamiento se implementan 300 épocas con un batch de 80 lotes. El optimizador usado para este entrenamiento es del tipo SGD. Los resultados obtenidos se evidencian en las figuras que relacionan las métricas de este entrenamiento.

En la Figura 11 se aprecian las curvas obtenidas, que muestran las relaciones métricas de Confidence/F1en el modelo entrenado entre las clases Gun y Person With Gun. En esta observamos que el nivel de confianza llega hasta un máximo de 0.8 para Person With Gun. Para la clase Gun se obtiene un máximo de 0.7 F1. Estos valores decrecen a medida que la confianza aumenta y se acerca a 1.

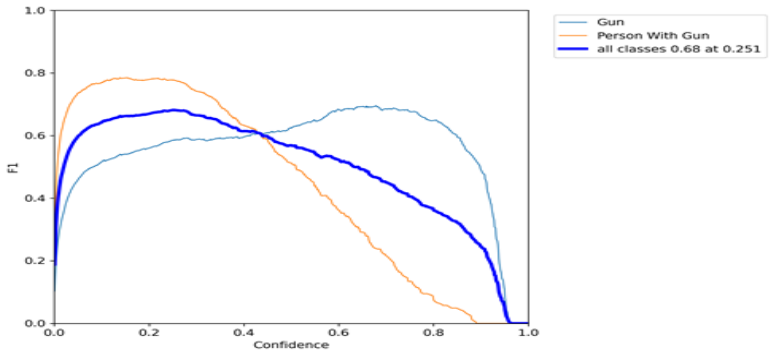


Figura 11. Curva de relación Confidence/F1. Fuente: Propia

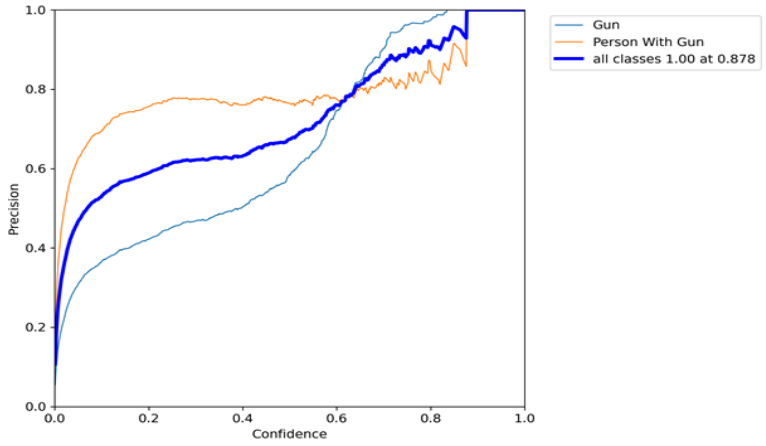


Figura 12. Relación de métricas Precision/Confidence. Fuente: Propia

En la Figura 12 se evidencia que el nivel de confianza del modelo en la detección de las clases aumenta con el tiempo de entrenamiento elevando el nivel de precisión del sistema, alcanzado valores de confianza de 0.878 acercándose a una precisión de 0.9.

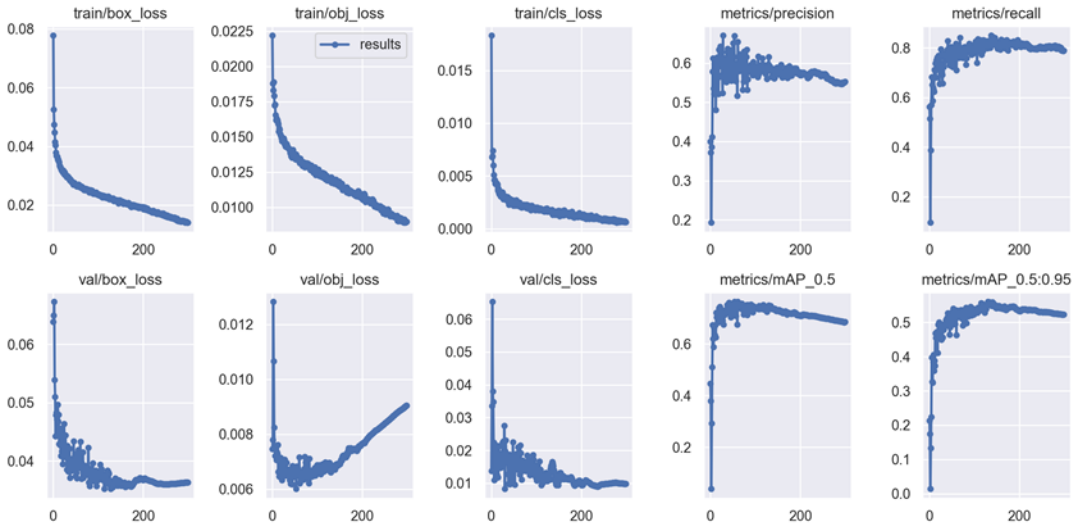


Figura 13. Gráfica de métricas experimento 129. Fuente: Propia

En la Figura 13 se aprecia el gráfico de pérdidas del entrenamiento de los objetos. En las 300 épocas inicialmente la pérdida es de 0.0225 y luego de 25 entrenamientos comienza a decrecer hasta un porcentaje de 1.5 para llegar a las 200 épocas a una pérdida aproximada de 1% para este experimento.

El entrenamiento del sistema de reconocimiento de armas de fuego y de personas con armas se hace a través de redes neuronales convolusionales principalmente con dos tipos de redes, YOLOv5s y YOLOv5x. También se utilizan diferentes lotes (batch), como máximo 140 para una red pequeña. Para la red grande se usó un batch de 48.

La Figura 14 es un ejemplo de un lote de entrenamiento del sistema y permite apreciar las cajas delimitadoras con sus clases asignadas de los objetos identificados.

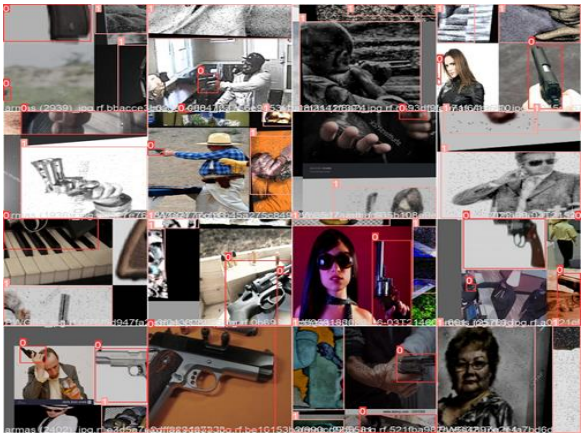

Figura 14. Batch de entrenamiento. Fuente: Propio

### *3.1.3 Validación*

En esta fase el sistema reconoce sobre imágenes diferentes a las asignadas en el entrenamiento los objetos clasificados como “Gun” y “Person With Gun”, esto sirve para verificar la eficiencia del modelo implementado y del previo entrenamiento realizado.

**3.1.3.1 Detección en imagen**. Luego de efectuar el entrenamiento se procede a verificar si se realiza una detección efectiva del arma de fuego (Gun) y de la persona con arma (Person With Gun), esta detección es positiva siempre y cuando el modelo permita en primer lugar identificar la clasificación asignada, reconociendo los objetos dentro de una caja de limitadora con un valor numérico que sea mayor de 0 (cero) y preferiblemente cercano a uno (1).

En la Figura 15 se percibe un grupo de figuras que son reconocidas por sus clases y etiquetas, y con valores cercanos a 0.9. Siendo una validación que da muestra de la efectividad del modelo.

Figura 15. Batch de validación. Fuente: Propia

**3.1.3.2 Detección en video en tiempo real**. A continuación, se relacionan las imágenes o frames tomados de los videos resultantes de las detecciones obtenidas en tiempo real de las armas de fuego (Gun) y Persona con Arma (Person With Gun).


Figura 16. Detección verdadera positiva tiempo real. Fuente: Propia

La detección en tiempo real se aprecia un ejemplo en la Figura 16. Se reconocen las dos clases etiquetadas previamente como “Gun” y “Person With Gun”. Se observan reconocidas en cuadros delimitadores.


Figura 17. Detección de arma de fuego con postura brazo extendido frontal. Fuente: Propia

En la Figura 17 se evidencia el reconocimiento del sistema en tiempo real del arma de fuego. Este reconocimiento arroja un valor de 0.7.


Figura 18. Otra detección de arma de fuego con postura brazo extendido frontal. Fuente: Propia

En la Figura 18 se verifica el buen funcionamiento del modelo de reconocimiento en tiempo real del arma de fuego. Este reconocimiento confirma su eficiencia, dando como resultado un valor de 0.94, sabiendo que el valor máximo es 1.0.

### *3.1.4 Implementación OpenPose*

En esta fase utilizando esta herramienta lógica de estimación de pose humana, se visualiza y evalúa con base en videos en tiempo real el posicionamiento del cuerpo de una persona con arma

de fuego. También se comprueba apoyado en herramientas de medición de ángulos como Geogebra, Kinovea y Tracker, los grados de inclinación de los brazos con respecto al arma. Estas métricas se obtienen en diferentes posiciones de las extremidades superiores de la persona que la porta. Como se observa en la Figura 19, se implementa un modelo para detectar delictiva o peligrosa.

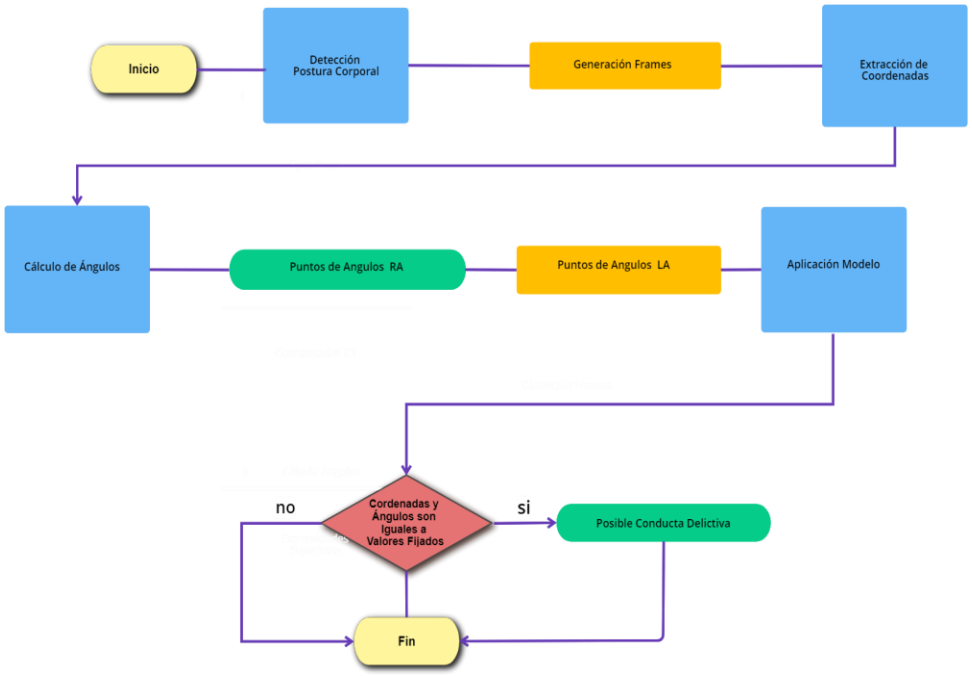


Figura 19. Modelo de detección actividad delictiva o peligrosa. Fuente: Propia

**3.1.4.1 Aplicación herramienta OpenPose.** Esta subfase permite con el uso de esta herramienta (OpenPose) hacer localización de los puntos de las articulaciones de las extremidades superiores en un cuerpo humano. Es así como se utilizan los puntos correspondientes a las articulaciones del hombro, brazo y mano de la extremidad superior derecha. De la extremidad superior izquierda se toman dos puntos de articulación correspondientes al hombro y brazo.


Figura 20. Estimación postural con arma de fuego/brazo semiflexionado. Fuente: Propia

En la Figura 20 se ve la aplicación de la herramienta OpenPose sobre un frame de video tomado en tiempo real de una persona apuntando con un arma de fuego. La posición del brazo que sostiene el arma es semiflexionada.


Figura 21. Estimación postural con arma de fuego/brazo extendido. Fuente: Propia

Se observa en la Figura 21 la imagen de una persona apuntando un arma de fuego en posición de brazo extendido. Con el uso de la herramienta OpenPose se identifican los puntos de las articulaciones de las extremidades superiores.

Para obtener los ángulos de inclinación y articulación, adicionalmente se utilizan herramientas de medición de ángulos como Geogebra, Kinovea y Tracker, con las que se puede calcular y medir.


Figura 22. Medición de ángulos brazo semiflexionado. Fuente: Propia

El resultado que se obtiene al implementar las herramientas de medición de ángulos y coordenadas se puede apreciar en la Figura 22. Estos ángulos son necesarios para la implementación del modelo OpenPose.


Figura 23. Medición de ángulos brazo extendido. Fuente: Propia

En la Figura 23 observamos los ángulos de referencia de la herramienta OpenPose. Estos ángulos son recurrentes en el modelo a implementar.

| ***Puntos de identificación*** | |
|---|---|
| *Brazo derecho: Right Arm (RA)* | *Brazo izquierdo: Left Arm (LA)* |
| 1RA | 1LA |
| 2RA | 2LA |
| 3RA | ---- |

Tabla 3. Puntos de identificación. Fuente: Propia

La Tabla 3 relaciona cada uno de los puntos nombrados e identificados sobre la imagen dada por la herramienta OpenPose. Estos puntos son primordiales en la validación de los ángulos y de las coordenadas x, y necesarios para la implementación del modelo.

| ***Etiqueta de los ángulos de las extremidades*** | |
|---|---|
| *Anotación* | *Significado* |
| 0 | Punto central y de unión con 1RA y 1RL |
| 1RA | Punto de articulación hombro derecho |
| 2RA | Punto de articulación codo derecho |
| 3RA | Punto de articulación mano derecha |
| 1LA | Punto de articulación hombro izquierdo |
| 2LA | Punto de articulación codo izquierdo |

Tabla 4. Etiqueta de los ángulos de las extremidades. Fuente: Propia

En la Tabla 4 se especifican las articulaciones usadas en el modelo y su correspondiente anotación en la imagen implementada en OpenPose.


Figura 24. Relación de puntos de articulación y ángulos brazo semiflexionado al frente. Fuente: Propia

La relación de los puntos de las articulaciones identificados como se expresa en la Tabla 4 y los ángulos formados entre sí, se visualizan en la Figura 24.


Figura 25. Relación de puntos de articulación y ángulos brazo extendido al frente. Fuente: Propia

En la Figura 25 se observa la relación del valor obtenido de los ángulos y los puntos de articulación con la herramienta OpenPose. Estos valores se dan para un brazo extendido al frente.

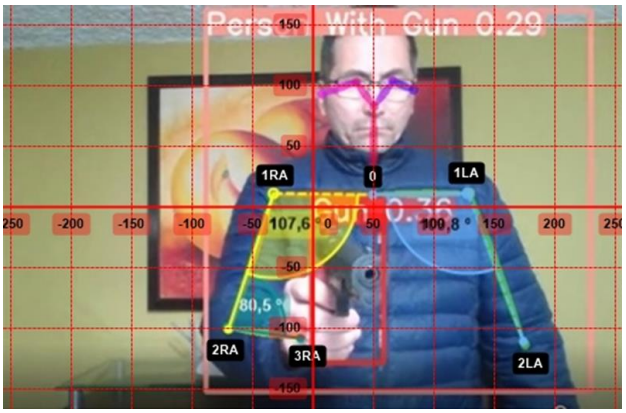


Figura 26. Relación de puntos de articulación, ángulos y coordenadas brazo semiflexionado al frente. Fuente: Propia

Como se aprecia en la Figura 26 a la imagen de la persona apuntando con arma de fuego se le aplica un modelo matemático para extraer coordenadas x,y. La persona expone una postura de brazo semiflexionado al frente.


Figura 27. Relación de puntos de articulación, ángulos y coordenadas brazo extendido al frente. Fuente: Propia

La Figura 27 muestra la imagen de una persona con arma de fuego en brazo extendido al frente. A esta imagen se le extraen coordenadas x,y.

# 4. Evaluación de métricas y resultados

## 4.1 En entrenamiento y validación

En el entrenamiento del sistema se realizan varias correlaciones entre las métricas existentes. Las redes que se implementan influyen levemente en los resultados obtenidos. Es importante mencionar que cada entrenamiento que se realiza tiene como mínimo 100 épocas.De igual manera el número de batch mínimo en una red pequeña es de 80 lotes.

La Tabla 5 permite ver la tasa de iteraciones por segundo (1.28it/s) para un número total de imágenes de prueba de 697. Estas imágenes son de las 2 clases nombradas y previamente etiquetadas. Las capas implementadas en la red convolucional profunda son 213. En este experimento se trabaja con la función de activación ReLu para las capas profundas.

| ***Clasificación*** | ***Gun*** | ***Person With Gun*** |
|---|---|---|
| **TIPO DE RED DE ENTRENAMIENTO** | YOLOv5s | |
| **TIEMPO DE ENTRENAMIENTO** | 0.7770 | |
| **ÉPOCAS** | 100 | |
| **BATCH** | 80 | |
| **LAYERS** | 213 | |
| **PARAMETERS** | 7015519 | |
| **IMAGES** | 697 | |
| **LABELS** | 340 | 438 |
| **AVERAGE INTERACTIONS (it/s)** | 1.28 | |
| **TP (d)** | 0.81 | 0.70 |
| **TN (a)** | 1.19 | 1.81 |
| **FP (b)** | 0.81 | 0.19 |
| **FN (c)** | 0.19 | 0.30 |
| **RECALL** | 0.835 | 0.801 |
| **ACCURACY** | 0.67 | 0.84 |
| **PRECISION** | 0.412 | 0.736 |
| **SENSIBILIDAD** | 0.81 | 0.70 |
| **ESPECIFICIDAD** | 0.595 | 0.905 |
| **F1** | 0.552 | 0.767 |

Tabla 5 Métricas experimento 149

En la Tabla 6 observamos la tasa de iteraciones por segundo (1.93it/s) para un número total de imágenes de prueba de 697. Estas imágenes son de las 2 clases nombradas y previamente etiquetadas. Las capas implementadas en la red convolucional profunda son 444. En este experimento se trabaja con la función de activación ReLu. El tiempo de entrenamiento es de 2.733 horas. Es mayor al anterior dado que la CNN es más grande, tiene casi el doble de capas que la anterior red.

Efectuando un paralelo de igualación de los dos modelos se aprecia que el valor de eficacia está en un rango de 0.682 para el modelo de red grande y de 0.67 para la red pequeña. Esto con la clase “Gun”. Para la segunda clase los valores oscilan entre 0.817 red grande y 0.84 red pequeña. Con estos valores se aprecia que la rata de eficiencia entre modelos varía entre clases siendo mayor la pérdida del modelo de red convolucional grande en la segunda clase en un rango de 0.023. Esto muestra la estabilidad del sistema a pesar de entrenar con otra red neuronal.

Sin embargo, se indica la mejor respuesta de la red pequeña en la identificación de verdaderos positivos en las dos clases propuestas. Los resultados conseguidos de 0.835 y 0.801 evidencian la eficacia del Recall en el modelo.

Los resultados mostrados en estos experimentos en conjunto con las validaciones efectuadas, muestran la eficacia del sistema con respecto a la identificación de armas de fuego “Gun” y de las personas que las portan “Person With Gun”. No se debe dejar de lado que la validación no solo se efectúa a imágenes estáticas. Se valida a través de detección realizada a videos en tiempo real con iteraciones en el rango de 0.33ms por frame.

| ***Clasificación*** | ***Gun*** | ***Person with gun*** |
|---|---|---|
| **TIPO DE RED DE ENTRENAMIENTO** | YOLOv5x | |
| **TIEMPO DE ENTRENAMIENTO** | 2.733 | |

| | | |
|---|---|---|
| **ÉPOCAS** | 100 | |
| **BATCH** | 30 | |
| **LAYERS** | 444 | |
| **PARAMETERS** | 86180143 | |
| **IMAGES** | 697 | |
| **LABELS** | 340 | 438 |
| **AVERAGE INTERACTIONS (it/s)** | 1.93 | |
| **TP (d)** | 0.84 | 0.66 |
| **TN (a)** | 1.2 | 1.79 |
| **FP (b)** | 0.79 | 0.21 |
| **FN (c)** | 0.16 | 0.34 |
| **RECALL** | 0.846 | 0.712 |
| **ACCURACY** | 0.682 | 0.817 |
| **PRECISION** | 0.418 | 0.763 |
| **SENSIBILIDAD** | 0.84 | 0.66 |
| **ESPECIFICIDAD** | 0.603 | 0.895 |
| **F1** | 0.559 | 0.737 |

Tabla 6. Métricas experimento 150

### 4.2 En implementación OpenPose

En la Tabla 7 se observan los resultados conseguidos luego de realizar 4 experimentos de implementación OpenPose. Los valores graduales en cada una de las mediciones obtenidas refieren una congruencia entre sí. Las articulaciones de hombro, codo y mano son los puntos de referencia más importantes. Por esto se usan como base primordial en esta implementación. El sistema Openpose se soporta en estos puntos a nivel superior del cuerpo.

La posición corporal difiere de la actividad realizada. En este caso para el modelo propuesto interesan los puntos ubicados en las extremidades superiores, especialmente la extremidad que sostiene el arma en la acción postural. Se aclara que puede también ser usada la mano contraria, pero de igual manera los valores obtenidos se mantienen iguales y son correspondientes en el caso de uso de la extremidad contraria para la tenencia del arma de fuego.

Se sabe que la articulación del hombro presenta un movimiento natural en condiciones normales que oscila entre -60 y 180 grados. Así mismo la articulación del codo permite un movimiento natural que oscila entre 20 y 180 grados.

Teniendo en cuenta estos valores y relacionándolos con los valores de las Tablas 7 y 8 se obtienen rangos para los casos relacionados en los 4 experimentos. Para la primera condición de brazo semiflexionado al frente se pondera una rata entre 70 y 110 grados involucrando las articulaciones de hombro y codo.

| ***Ángulos calculados brazo derecho semiflexionado al frente*** | | | |
|---|---|---|---|
| Frame | RA | | LA |
| | (0,1RA, 2RA) | (1RA, 2RA, 3RA) | (0, 1LA, 2LA) |
| 1 | 107,60 | 80,50 | 109,80 |
| 2 | 108,60 | 80,10 | 110,30 |
| 3 | 108,10 | 80,30 | 109,70 |
| 4 | 108,70 | 80,60 | 110,40 |

Tabla 7. Ángulos calculados brazo derecho semiflexionado al frente. Fuente: Propia

En la Tabla 7 se aprecian los diferentes valores de ángulos calculados en cuatro experimentos sobre imágenes tomadas de video en tiempo real. Estos valores son para un brazo semiflexionado al frente.

| ***Ángulos calculados brazo derecho extendido al frente*** | | | |
|---|---|---|---|
| Frame | RA | | LA |
| | (0,1RA, 2RA) | (1RA, 2RA, 3RA) | (0, 1LA, 2LA) |
| 1 | 71,40 | 130,90 | 91,60 |
| 2 | 71,41 | 130,91 | 90,98 |
| 3 | 71,60 | 131,20 | 91,80 |
| 4 | 71,50 | 131,10 | 91,70 |

Tabla 8. Ángulos calculados brazo derecho extendido al frente. Fuente: Propia

La Tabla 8 expone el resultado de los valores de los ángulos obtenidos, luego de cuatro experimentos evaluados de videos en tiempo real para un brazo extendido al frente.

Por otra parte, para la segunda condición de brazo extendido al frente se pondera una rata entre 80 y 140 grados que involucra las articulaciones de hombro, codo y mano.

Finalmente, se realiza una asociación entre valores graduales para los dos casos, es decir brazo semiflexionado al frente y brazo extendido al frente para obtener un valor gradual que unifique las dos condiciones. Este valor resultante esta dado entre 70 y 140 grados. El modelo luego de verificar esta rata porcentual y advertir el cumplimiento de la condición, identifica y alerta la posible conducta delictiva mediante señal audible con el mensaje “peligro”.

## 5. Conclusiones

El modelamiento de un sistema de detección de imágenes aplicado al reconocimiento de clases, en especial en armas de fuego se debe implementar con redes neuronales convolucionales CNN, ya que estas redes neuronales permiten que en este tipo de análisis de imágenes cada componente se entrene de forma independiente y se ejecute en corto tiempo. Se indica la necesidad de aplicar mejoras previas a la fase del entrenamiento, es decir realizando ajustes con el uso de herramientas lógicas en la fase de etiquetado y preprocesamiento para obtener dataset más fiables y con mejores resultados.

Por otra parte, La Tabla 5 permite ver la tasa de iteraciones por segundo (1.28it/s) para un número total de imágenes de prueba de 697 y de 100 épocas. Así mismo en la Tabla 6 la tasa de iteraciones por segundo (1.93it/s), para un mismo número de imágenes de prueba y de épocas que en el experimento 149. Lo anterior, aunado a otros varios experimentos realizados previamente evidencia que el número de épocas y de iteraciones que se realizan en el entrenamiento influyen en el rendimiento del sistema, ya que se puede presentar overfitting o sobrentrenamiento.

En el desarrollo de este trabajo se validan imágenes estáticas que respecto a las métricas evaluadas dieron como resultado valores que se relacionan en las Tablas 5 y 6 en el acápite evaluación de métricas y resultados, obteniendo valores de desempeño cercanos a los esperados para el modelo propuesto. Se resaltan las validaciones efectuadas por el detector de imágenes de video en tiempo real, siendo esta condición de métricas fluctuantes, ya que intervienen en el resultado otras circunstancias como son la luminosidad, el espacio, la resolución de la cámara, las condiciones climáticas, la distancia focal y demás aspectos que influyen en el resultado final.

Como se puede apreciar en los resultados obtenidos en la Tabla 5 y Tabla 6 para las métricas de los experimentos 149 y 150 respectivamente, se destacan los valores de eficiencia obtenidos durante el entrenamiento del modelo de detección de “Gun” y “Person With Gun” logrando porcentajes de 0.817 y 0.840, dando una tasa de error del sistema de 0.023, equivalente a un 96% de eficiencia resultante.

Con respecto a la implementación de la herramienta OpenPose para la identificación de la actividad peligrosa con arma de fuego, cabe señalar que como se muestra en Tabla 4, la relación entre los puntos de articulación hombro, codo y mano son referencia importante para el desarrollo del modelo propuesto. La medida de los ángulos articulares se realiza teniendo en cuenta la posición que el individuo adopta al apuntar con el arma de fuego. Es decir, con el brazo semiflexionado orientado hacia adelante o con el brazo extendido con la misma orientación. El modelo finaliza su detección cuando verifica si el rango articular está en una tasa entre 70 y 140 grados. En caso afirmativo identifica y alerta la posible conducta delictiva con un mensaje audible “peligro”, de lo contrario continuará con la aplicación del modelamiento propuesto.

Para mejoras futuras a este desarrollo se hace indispensable recolectar un dataset mucho más amplio que involucre armas de todo tipo, no solo de fuego sino también cortopunzantes, contundentes y demás que puedan causar daño en algún momento a las personas y bienes, tanto públicos como privados, pudiéndose paralelamente implementar nuevas y novedosas aplicaciones con las cuales se abarque un espectro mucho más robusto en los objetivos a alcanzar.

Dados los experimentos efectuados y los resultados obtenidos en los experimentos que se implementaron y desarrollaron con los modelos propuestos como se puede observar en las Figuras 16 a 27 se pudo evidenciar que entre más robusto sea el dataset utilizado, es decir entre más imágenes se puedan recopilar de cada uno de los objetos a clasificar, se asegurará un entrenamiento

efectivo, claro está, teniendo mucha precaución de no caer en sobre entrenamiento de la red neuronal implementada.

Para perfeccionar este desarrollo se pueden implementar diferentes variaciones al mismo en el método de preprocesamiento, el número de cuadros extraídos, la cantidad de ángulos para los vectores, igualmente se puede añadir un clasificador Bi-LSTM para tener exactitud en el resultado final.

Con base en los resultados que se aprecian en las Tablas 5 a 8 y las imágenes que se relacionan en las Figuras 16 a 27, cabe destacar que las pruebas sobre este aplicativo arrojaron los resultados esperados conforme a los objetivos planteados para la detección, referenciando una tasa de error del sistema de 0.023, equivalente a una eficiencia del 96% del modelo. Se hace énfasis en que todas las tareas del sistema se realizaron dentro de un entorno y tiempo real.

Por último, cabe indicar que, como evidencia tecnológica del modelo desarrollado, al realizar la detección propuesta se obtiene como resultado final un reconocimiento positivo del arma de fuego que en consonancia con la persona que la porta, da como resultado una alarma audible que indica la situación peligrosa.

## Referencias

[1] Consejo de Bogotá. https://concejodebogota.gov.co/la-inseguridad-en-bogota-no-da-tregua/cbogota/2021-08-18/130620.php 2021.

[2] Medicina Legal. https://www.medicinalegal.gov.co/documents/20143/386932/Forensis+2018.pdf/be4816a4-3da3-1ff0-2779-e7b5e3962d60

[3] C. Wang, Y. Wang, & Yuille, A. L. An approach to pose-based action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 915-922). 2013.

[4] B. Cyganek, &, J. P. Siebert (2011). An introduction to 3D computer vision techniques and algorithms. John Wiley & Sons

[5] ONU-HABITAT https://onuhabitat.org.mx/index.php/violencia-en-inseguridad-en-las-ciudades

[6] Universidad Nacional de Cuyo. http://www.politicaspublicas.uncu.edu.ar/articulos/index/inseguridad-un-analisis-desde-la-estructura-social

[7] T. Gil Márquez. "Una preocupación por la seguridad: Jaume Curbet, in memoriam." 2012.

[8] I. Hereu, J. Curbet. "Inseguridad global, seguridad mundicéntrica." Revista Catalana de Seguretat Pública. 2011.

[9] E. Spinelli, and ESTRELLA MARIELA. "Criminalística: Lugar del hecho." Especialización en Medicina Legal Dra. RattoNielsen. Instituto Universitario de Ciencias de la Salud Fundación HA Barcelona.

[10] L. Barros. Planificacion de la Actividad Delictual en casos de Robo con Violencia o Intimidación. 2016.

[11] C. Gutiérrez Lancho, "Detección de armas en vídeos mediante técnicas de Deep Learning." 2019.

[12] J. G. Jiménez. Visión por computador. Paraninfo. 2000.

[13] D. A. Montoya. Implementación y evaluación del rendimiento de redes neuronales densas en FPGA para la inferencia rápida, aplicadas a problemas en física y visión artificial. 2020.

[14] J. M. Pérez. Inteligencia computacional inspirada en la vida (Vol. 36). Servicio Publicaciones UMA. 2010.

[15] P. Loncomilla. "Deep learning: Redes convolucionales." Recuperado de https://ccc. inaoep. mx/~ pgomez/deep/presentations . 2016.

[16] K. H. Tan, &, B. P. Lim. The artificial intelligence renaissance: Deep learning and the road to human-level machine intelligence. APSIPA Transactions on Signal and Information Processing, 7. 2018.

[17] A. Verma, P. Singh and J. S Rani., "Modified Convolutional Neural Network Architecture Analysis for Facial Emotion Recognition," in 2019 International Conference on Systems, Signals and Image Processing (IWSSIP), pp. 169-173, 2019.

[18] E. G. Sánchez, J. A. Nalvaiz, & Lozano, L. O. Introducción a las redes neuronales de convolución. Aplicación a la visión por ordenador. 2019.

[19] F. M. Quiroga. Medidas de invarianza y equivarianza a transformaciones en redes neuronales convolucionales (Doctoral dissertation, Universidad Nacional de La Plata). 2020.

[20] D. Erroz Arroyo. Visualizando neuronas en redes neuronales convolucionales. 2019.

[21] F. L., Saca, A. F. Ramírez, C. A. Cruz, J. V. Cortez, , A. Z López., &, E. R. Martínez. Preprocesamiento de bases de datos de imágenes para mejorar el rendimiento de redes neuronales convolucionales. Research in Computing Science, 147(7), 35-45. 2018.

[22] F. Almeida, , D. Giménez, , J. M. Mantas, &, A. M. Vidal. Introducción a la programación paralela. Thompson Paraninfo. 2008.

[23] R., Vuduc, A Chandramowlishwaran, J. Choi, , M Guney, &, A. Shringarpure. On the limits of GPU acceleration. In Proceedings of the 2nd USENIX conference on Hot topics in parallelism (Vol. 13, No. 0). 2010.

[24] M. Bernaschi, A. Di Lallo, , R Fulcoli, , E. Gallo, &, L. Timmoneri Combined use of graphics processing unit (GPU) and central processing unit (CPU) for passive radar signal & data elaboration. In 2011 12th International Radar Symposium (IRS) (pp. 315-320). IEEE. 2011.

[25] F. Bozkurt, , M. Yaganoglu, &, F. B. Günay. Effective Gaussian blurring process on graphics processing unit with CUDA. International Journal of Machine Learning and Computing, 5(1), 57. 2015.

[26] J. Cheng, , M. Grossman, &, T. McKercher. Professional CUDA c programming. John Wiley & Sons. 2014.

[27] P. Jiang, , D. Ergu, , F. Liu, , Y. Cai, &, B. Ma. A Review of YOLO algorithm. 2022.

[28] Gunfire on School Grounds in the United States. Accessed:Jul. 20, 2021. [Online]. Available: https://everytownresearch.org/gunfirein-school/#ns. 2021

[29] R. A. Tessler, S. J. Mooney, C. E. Witt, K. O'Connell, J. Jenness,M. S. Vavilala, and F. P. Rivara, ''Use of firearms in terrorist attacks:Differences between the United States, Canada, Europe, Australia, and New Zealand,'' JAMA Internal Med., vol. 177, no. 12, pp. 1865–1868,

[30] N. Vállez, G. Bueno, and O. Déniz, ‘‘False positive reduction in detector implantation,’’ in Artificial Intelligence in Medicine (Lecture Notes inComputer Science), vol. 7885, N. Peek, R. M. Morales, and M. Peleg, Eds.Berlin, Germany: Springer, , doi: 10.1007/978-3-642-38326-7_28. 2013.

[31] A. Castillo, S. Tabik, F. Pérez, R. Olmos, and F. Herrera, ‘‘Brightness guided preprocessing for automatic cold steel weapon detection in surveillance videos with deep learning,’’ eurocomputing, vol. 330, pp. 151–161, Feb. 2019.

[32] S. Nercessian, K. Panetta, and S. Agaian, ‘‘Automatic detection of potential threat objects in X-ray luggage scan images,’’ in Proc. IEEE Conf. Technol.Homeland Secur., , pp. 504–509. May 2008.

[33] Z. Xiao, X. Lu, J. Yan, L. Wu, and L. Ren, ‘‘Automatic detection of concealed pistols using passive millimeter wave imaging,’’ in Proc. IEEE Int. Conf. Imag. Syst. Techn. (IST), Sep., pp. 1–4. 2015.

[34] G. Flitton, T. P. Breckon, and N. Megherbi, ‘‘A comparison of 3D interest point descriptors with application to airport baggage object detection in complex CT imagery,’’ Pattern Recognit., vol. 46, no. 9, pp. 2420–2436, Sep. 2013.

[35] G. Flitton, A. Mouton, and T. P. Breckon, ‘‘Object classification in 3Dbaggage security computed tomography imagery using visual codebooks,’’Pattern Recognit., vol. 48, no. 8, pp. 2489–2499, Aug. 2015.

[36] 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, Jan. 2015.

[37] N. B. Halima and O. Hosam, ‘‘Bag of words based surveillance system using support vector machines,’’ Int. J. Secur. Appl., vol. 10, no. 4, pp. 331–346, Apr. 2016.

[38] R. Girshick, ‘‘Fast R-CNN,’’ in Proc. IEEE Int. Conf. Comput. Vis. (ICCV),Dec., pp. 1440–1448. 2015.

[39] G. K. Verma and A. Dhillon, ‘‘A handheld gun detection using faster R-CNN deep learning,’’ in Proc. 7th Int. Conf. Comput. Commun. Technol. (ICCCT), , pp. 84–88. 2017.

[40] J. Ponce. "Dataset issues in object recognition." Toward category-level object recognition. Springer, Berlin, Heidelberg, 2006.

[41] G. Olague "Evolutionary computer vision." Proceedings of the 9th annual conference companion on Genetic and evolutionary computation. 2007.

[42] E. Alegre &, R. Fernandez &, E. Mora & Es, Morae@unican &, A. Alonso. Bases de Datos de Imágenes: Arquitectura de los Sistemas de Recuperación de Imágenes Basados en Contenido. 2005.