DESDE 2013
https://repository.uaeh.edu.mx/revistas/index.php/icbi/issue/archive
Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI
ISSN: 2007-6363

Publicación Semestral Pädi Vol. 11 No. Especial 2 (2023) 196–204

Detección de armas tipo pistola mediante el uso de redes convolucionales con una arquitectura tipo YOLO y estereoscopía.
Detecting pistol-type weapons using convolutional networks with YOLO-like architecture and stereoscopy.

A. Schcolnik-Elias [a,*], S. Martínez-Díaz [a], J. E. Luna-Taylor [a], I. Castro-Liera [a]

[a] *Tecnológico Nacional de México, Instituto Tecnológico de La Paz, 23080, La Paz, Baja California Sur, México.*

Resumen

Las cámaras de seguridad y los sistemas de videovigilancia se han convertido en infraestructuras sumamente importantes para garantizar la seguridad de los ciudadanos. Sin embargo, el problema de inseguridad sigue en crecimiento debido en gran parte al fácil acceso a armas de fuego tipo pistola por lo que, la detección temprana de este tipo de armas es de suma importancia para ayudar a prevenir accidentes. El objetivo de este trabajo es implementar un sistema de visión estereoscópica que sea capaz de detectar con alta confianza y en tiempo real este tipo de objetos, además de lograr definir la distancia a la que se encuentra. Para dicha detección se implementó una arquitectura de redes neuronales convolucionales (CNN) con un algoritmo tipo YOLO, empleando aprendizaje transferido, junto con un algoritmo para la estimación estereoscópica de la distancia. El sistema presentado funciona con un valor de Intersección sobre unión (IoU) de 0.6 en el cual el se logró una precisión de 92.2 % además, el error promedio detectado en la estimación de distancia hasta un máximo de 3 metros es de 9.3 centímetros.

Palabras Clave: Redes convolucionales, Red profunda YOLO, Visión estereoscopica, Detección de objetos, Pistola

Abstract

Security cameras and video surveillance systems play a crucial role in ensuring public safety. However, the increasing accessibility of pistol-type firearms contributes to growing concerns about insecurity. Early detection of these weapons is of utmost importance to prevent potential accidents. This study aims to develop a real-time stereoscopic vision system capable of accurately detecting pistol-type objects and determining their distance with high confidence. The approach combines a convolutional neural network (CNN) architecture with a YOLO-type algorithm, utilizing transfer learning, and incorporates an algorithm for stereoscopic distance estimation. The presented system achieves an accuracy of 92.2 % with an Intersection over Union (IoU) value of 0.6. Moreover, the average distance estimation error within a range of 3 meters is only 9.3 centimeters.

Keywords: Convolutional neural networks, YOLO deep neural network, Stereoscopic vision, Object detection, Pistol

1. Introducción

Uno de los problemas de muchas ciudades en el mundo, que se encuentra muy presente en las ciudades de México, es la inseguridad. Para intentar reducir el número de crímenes cometidos se han implementado los sistemas de videovigilancia. Sin embargo, la detección de situaciones de alto riesgo a través de estos sistemas todavía se realiza de forma manual en muchas ciudades (Grega *et al.*, 2016). La falta de mano de obra en el sector de la seguridad y el desempeño limitado de los humanos puede resultar en peligros no detectados o retrasos en la detección de amenazas, lo que representa un riesgo para el público.

Cuando una persona lleva un arma de fuego al aire libre, es un fuerte indicador de una situación potencialmente peligrosa. De acuerdo a (INEGI, 2021), en México en el año 2020, se registraron 36,579 homicidios de los cuales 25,456 fueron llevados a cabo con un arma de fuego.

Una forma de reducir este tipo de violencia es la prevención mediante detección temprana para que los agentes de seguridad o policías puedan actuar. En particular, una solución innovadora a este problema es equipar a las cámaras de vigilancia o control

*Autor para correspondencia: aaronschcolnikelias@gmail.com
Correo electrónico: aaronschcolnikelias@gmail.com (Aarón Schcolnik-Elias), saul.md@lapaz.tecnm.mx (Saúl Martínez-Díaz), jorge.lt@lapaz.tecnm.mx (Jorge Enrique Luna-Taylor), iliana.cl@lapaz.tecnm.mx (Iliana Castro-Liera)

Historial del manuscrito: recibido el 28/03/2023, última versión-revisada recibida el 21/06/2023, aceptado el 30/06/2023, publicado el 11/09/2023. **DOI:** https://doi.org/10.29057/icbi.v11iEspecial2.10727

con un sistema de alerta automático preciso de detección de armas de fuego (Olmos *et al.*, 2018).

Una de las soluciones presentadas para automatizar un sistema de detección consta del uso de las redes neuronales artificiales. Las redes neuronales artificiales son técnicas de aprendizaje automático que simulan los mecanismos de aprendizaje del cerebro humano y su arquitectura consiste en capa de entrada, capas ocultas y capa de salida, en las cuales se encuentran módulos de procesamiento (neuronas) interconectados entre sí, que darán como resultado una predicción (Arballo *et al.*, 2022).

Por otro lado, las redes neuronales convolucionales (CNN, por sus siglas en inglés) son un tipo de red neuronal profunda que han revolucionado el campo del reconocimiento de imágenes. Las CNNs están diseñadas para aprender y extraer características complejas de imágenes de entrada mediante el uso de múltiples capas de filtros convolucionales.

Las CNNs funcionan pasando la imagen de entrada a través de una serie de capas convolucionales que extraen características cada vez más complejas. Estas capas convolucionales usan filtros que escanean la imagen de entrada para identificar patrones o características en los datos. La salida de cada capa convolucional se pasa entonces a través de una función de activación no lineal para introducir la no linealidad en el modelo. La salida final de la red se usa entonces para hacer una predicción sobre la imagen de entrada.

Una manera simplificada de ver a dichas redes es que son un modelo compuesto por una red multicapa con núcleos (kernels) que convolucionan para extraer características relevantes de la imagen de entrada (Arballo *et al.*, 2022).

En los últimos años, el campo del reconocimiento de imágenes ha experimentado una transformación significativa gracias a las CNN, las cuales han obtenido resultados superiores a cualquier otro sistema clásico de aprendizaje automático en métodos de reconocimiento, clasificación, detección y segmentación de imágenes (Flitton *et al.*, 2013) (Gesick *et al.*, 2009). Esta tecnología se ha utilizado ampliamente en varias aplicaciones, incluidos los sistemas de seguridad (Grega *et al.*, 2016). Un área en la que las CNN han demostrado un gran potencial es en la detección y la identificación de objetos en imágenes. La capacidad de las CNN para aprender características y patrones de grandes conjuntos de imágenes las convierte en una opción atractiva para los sistemas de seguridad que requieren una identificación fiable y precisa de los objetos.

Los sistemas de seguridad tradicionales, como se mencionó anteriormente, se basan en la identificación manual, lo que es lento y propenso a errores humanos. Por el contrario, las CNN pueden procesar grandes cantidades de datos e identificar objetos con un alto grado de precisión de manera automática. Además, las CNNs pueden detectar diferencias sutiles en las imágenes, lo que es crucial en aplicaciones de seguridad donde pequeñas variaciones pueden tener una gran diferencia. Por ejemplo, las CNNs se pueden usar para reconocer rostros, detectar armas o identificar actividades sospechosas en áreas concurridas.

En general, el uso de reconocimiento de imágenes mediante redes neuronales convolucionales tiene el potencial de ser un avance significativo en términos de seguridad. Como lo concluido en (Deepthi *et al.*, 2018) donde se presentó el uso de redes convolucionales para el reconocimiento de armas tipo pistola y cuchillos, siendo una de sus grandes limitaciones el uso de un conjunto de datos de prueba demasiado pequeño (319 imágenes de entrenamiento) lo cual limitó su uso en situaciones reales. Sin embargo, gracias a su experimentación llegaron a la conclusión de que el uso de este tipo de redes puede agilizar el trabajo de monitoreo de cámaras en el sector de seguridad.

En (Strbac *et al.*, 2020) se complementó la detección de objetos mediante la implementación de un sistema de cámaras estéreo. La finalidad de dicho artículo era obtener la distancia aproximada a la que se encuentra un coche sin la necesidad de utilizar sensores ultrasónicos o tipo LIDAR. Al finalizar su experimentación el equipo logró obtener una alta precisión hasta una distancia máxima de 20 metros.

En este documento es presentada una metodología para el reconocimiento y detección automático de armas tipo pistola. Ademas de su detección, se lleva a cabo la estimación de la distancia a la que se encuentra dicho objeto. Esto se logra utilizando aprendizaje profundo supervisado con CNNs aplicando el algoritmo You Only Look Once (YOLO), el conjunto de datos utilizado fue obtenido de (Olmos *et al.*, 2018) y modificado con imágenes originales con su respectiva imagen etiquetada (ground truth). Este conjunto de datos puede ser modificado y extendido con múltiples tipos de pistola diferentes para aumentar su precisión y utilidad.

2. Materiales y métodos

2.1. You Only Look Once (YOLO)

La mayoría de los algoritmos utilizados en redes neuronales para la detección y clasificación de objetos consisten de dos redes neuronales, una para la detección y otra para la clasificación, esto causa que dichos algoritmos sean muy lentos para su uso en detección de objetos en tiempo real (Strbac *et al.*, 2020).

YOLO replantea la detección de objetos como un único problema de regresión (figura 1), directamente desde los píxeles de la imagen hasta las coordenadas de la caja de contorno y las probabilidades de clase. Es bastante simple, una sola red convolucional predice simultáneamente múltiples cajas de contorno y probabilidades de clase para esas cajas. El sistema funciona de la siguiente manera:

1. Primero redimensiona la imagen de entrada
2. Después ejecuta una única red convolucional en la imagen
3. Y termina al umbralizar las detecciones resultantes según la confianza del modelo

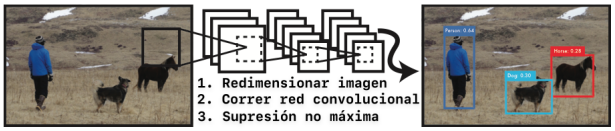

Figura 1: El sistema de detección YOLO (Redmon *et al.*, 2016).

La principal ventaja de esto es la velocidad, los modelos basados en YOLO son extremadamente rápidos ya que no necesitan una arquitectura compleja, YOLO ve toda la imagen durante la capacitación y la prueba, por lo que implícitamente codifica información contextual sobre clases, así como su apariencia.

El único gran inconveniente con respecto a YOLO es que todavía está por detrás de los sistemas de detección de última generación en términos de precisión. Si bien puede identificar rápidamente objetos en las imágenes, tiene dificultades para localizar precisamente algunos objetos, especialmente los pequeños (Redmon *et al.*, 2016).

2.2. *YOLOv8*

YOLOv8 es la octava versión de YOLO creada por el equipo Ultralytics, fue publicada en su versión inicial el 13 de enero del 2023, tomando como base el trabajo inicial realizado en YOLO (Jocher y Ultralytics, 2023). La principal ventaja de esta arquitectura es su alta precisión a comparación de versiones anteriores, especialmente en sus modelos menos complejos conocidos como Yolov8 Nano (Yolov8n) y Yolov8 Small (Yolov8s) y, ademas de tener una alta precisión, sus velocidades son mayores a los demás modelos (figura 2).

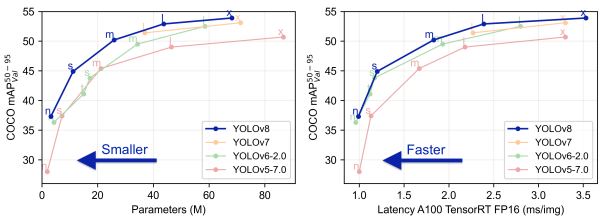

Figura 2: Gráficas comparando la precisión y la velocidad entre los diferentes modelos basados en YOLO (Jocher y Ultralytics, 2023).

Como se ilustró en la figura 2, COCO es un conjunto de datos de subtítulos, segmentación y detección de objetos a gran escala. COCO *mAP* (mean average precision por sus siglas en ingles) toma en cuenta la precisión y el recordatorio de detección de objetivos en dicho conjunto de datos. Cuando hablamos de *map*50 − 95 debemos tomar en cuenta diferentes umbrales de *IoU* (Intersection Over Union, o Intersección Sobre Union en español).

IoU es un valor entre 0 a 1 que cuantifica el grado de superposición entre dos cajas delimitadoras (figura 3). El área de superposición es el área donde se superponen la caja delimitadora de verdad básica (o ground truth en inglés) y la caja delimitadora predecida por el modelo y el área de unión toma en cuenta el área total cubierta entre ambas cajas delimitadoras.

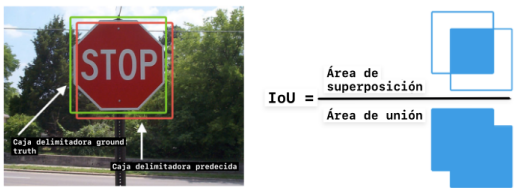

Figura 3: Representación visual de los parámetros necesitados para obtener la *IoU*.

Ademas de la *IoU*, es necesario obtener el valor de **precisión** (fórmula 1) el cual es el porcentaje de predicciones correctas realizadas por el modelo entrenado, en los términos más simples, la precisión es la relación entre los verdaderos positivos y todos los positivos. Cabe mencionar que este valor es dependiente del umbral de *IoU* especificado, lo que significa que este puede mejorar o empeorar de acuerdo a dicho umbral. Por ejemplo, en nuestro campo de estudio esta métrica nos estaría diciendo *que tan correcto es nuestro modelo cuando este mismo detecta que hay un arma en la imagen*. Los **verdaderos positivos** son definidos como una detección correcta de una caja delimitadora ground truth, en nuestro caso es *çuando una pistola se encuentra en la imagen y nuestro modelo la detectó "* mientras que los **falsos positivos** son una detección incorrecta de un objeto inexistente o una detección fuera de un objeto existente, en nuestro caso sería *çuando en la imagen no se encuentra una pistola pero el modelo detectó una "*. Los falsos positivos son considerados como un error tipo *I*, esto debido a que a pesar de que es un error en la detección este no tiene repercusiones muy importantes, por ejemplo, si nuestro modelo detecta una pistola cuando no hay una pistola presente simplemente podemos anular esa detección.

$$Precisión = \frac{Verdaderos\ Positivos}{Verdaderos\ Positivos + Falsos\ Positivos} \quad (1)$$

También necesitamos obtener el valor conocido como **exhaustividad** (recall) (fórmula 2); este valor, que igualmente es un porcentaje, define que tan bueno es nuestro modelo para identificar **verdaderos positivos** correctamente, igual que la precisión este valor es dependiente del umbral de *IoU* especificado. Entonces, en nuestro campo de estudio nos diría *"de todas las imágenes que cuentan con una pistola cuantas de ellas detectamos correctamente como una pistola "*. Los **falsos negativos** son definidos como una caja delimitadora ground truth no detectada, en nuestro caso sería *"la imagen cuenta con una pistola pero nuestro modelo detecto que no hay ninguna pistola en ella "*. Además, los falsos negativos se consideran como un error tipo *II*, esto debido a que es un error en la detección que tiene repercusiones muy importantes, por ejemplo, si nuestro modelo no detecta una pistola en la escena cuando si hay una pistola esto puede facilitar un tiroteo o un asesinato.

$$Recall = \frac{Verdaderos\ Positivos}{Verdaderos\ Positivos + Falsos\ Negativos} \quad (2)$$

El uso de precisión (fórmula 1) y de recall (fórmula 2) puede parecer confuso, pero lo podemos presentar de esta manera: *"la precisión se centra en que tan correctas son las detecciones, mientras que el recall se centra en la completitud de las detecciones "*. Un sistema con alta precisión puede perder algunos objetos, mientras que un sistema con alto recall puede tener muchos falsos positivos. Equilibrar estas dos métricas es crucial para lograr un buen rendimiento general en un sistema de detección de objetos.

*mAP*50, se calcula obteniendo al valor promedio de precisión (fórmula 1) para cada clase y luego obteniendo la media de la precision en todas las clases cuando el umbral *IoU* (figura 3) es igual a 0,50, *mAP*50 − 95 se obtiene de la misma manera con el único cambio siendo que cuenta con multiples umbrales *IoU* desde 0,50 hasta 0,95, con valores tomados cada 0.05, dando un total de 10 valores, donde luego se toma el valor promedio. Cuanto mayor sea el indicador, mejor será la detección del modelo (Liu *et al.*, 2022).

El algoritmo YOLO detecta un objeto y proporciona las coordenadas de los contenedores de fronteras de todos los objetos detectados. Estas coordenadas son presentadas como los parámetros X_1, X_2, Y_1 y Y_2, los cuales son utilizados en la expresión matemática para la estimación de distancia basada en estereoscopía. Además, a pesar de que Yolov8 cuenta con una arquitectura (figura 4) mas compleja que versiones anteriores, esta sigue siendo capaz de funcionar fácilmente en tiempo real en hardware no tan costoso.

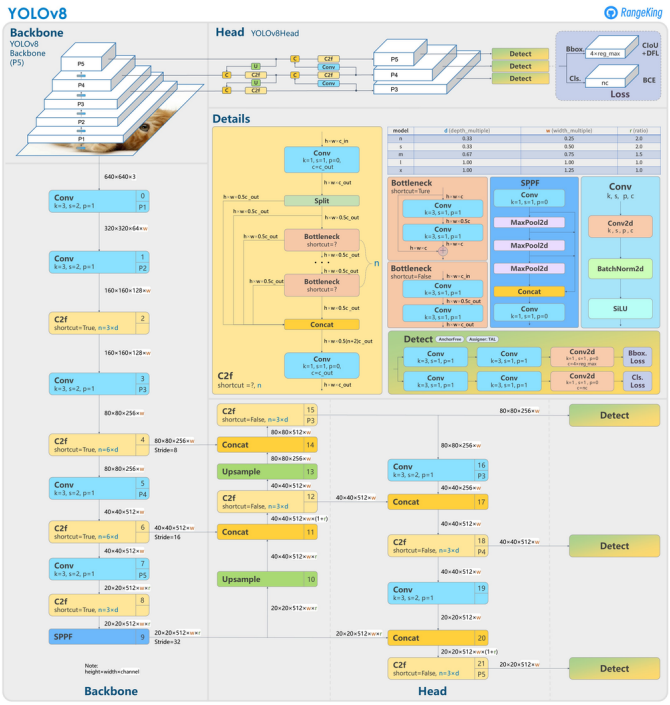

Figura 4: Arquitectura de la red YOLOv8 (RangeKing, 2023).

2.3. *Arquitectura de la red*

La red (figura 4) utiliza una *red de pirámide de características (FPN)* para detectar objetos de diferentes tamaños y escalas dentro de una imagen. Esta *FPN* consta de múltiples capas que detectan objetos a diferentes escalas, lo que permite al modelo detectar objetos grandes y pequeños dentro de una imagen.

Ademas, la arquitectura se puede dividir en dos partes principales: ”la columna vertebral (backbone) y la cabeza (head)”, las cuales constan de diferentes submódulos los cuales son una combinación de capas convolucionales (Conv y C2f), capas de agrupamiento (MaxPool2d), capas de muestreo superior (Upsample), fórmulas de concatenación de tensores (Concat) y fórmulas de perdida (Bbox. Loss y Cls. Loss).

La columna vertebral consiste de una versión modificada de CSPDarknet53, consta de 53 capas convolucionales y emplea una técnica llamada ”*cross-stage partial connections*”(conexiones parciales entre etapas) para mejorar el flujo de información entre las diferentes capas de la red.

La cabeza de YOLOv8 consta de múltiples capas de convolución seguidas de una serie de capas totalmente conectadas. Estas capas son responsables de predecir las cajas delimitadoras, las puntuaciones de objetividad y las probabilidades de clase para los objetos detectados en una imagen, además, una de las principales características de YOLOv8 es el uso de un mecanismo de autoatención en la cabeza de la red, este mecanismo permite al modelo centrarse en diferentes partes de la imagen y ajustar la importancia de diferentes características en función de su relevancia para la tarea.

Para finalizar, esta arquitectura permite utilizar un modelo pre-entrenado en el conjunto de datos COCO (Veit *et al.*, 2016) el cual es un modelo general para detección de objetos, en este artículo se decidió utilizar dicho modelo pre-entrenado.

2.4. *Estereoscopía*

Es la ciencia y tecnología que se ocupa de los dibujos o fotografías en dos dimensiones que cuando se observan con ambos ojos parecen existir en tres dimensiones en el espacio.

La estereoscopía es posible solo debido a la visión binocular, la cual requiere que la vista del ojo izquierdo y la vista del ojo derecho de un objeto se perciba desde diferentes ángulos o, en el caso de este estudio, requiere que se recopilen imágenes con una cámara izquierda a cierto ángulo y una cámara derecha en un ángulo distinto.

2.5. *Calibración de las cámaras*

La calibración y orientación de la cámara es un requisito previo necesario para la extracción de información métrica 3D precisa y fiable de las imágenes. Se considera que una cámara está calibrada si se conocen los parámetros de distancia principal, desplazamiento del punto principal y distorsión de la lente (Remondino *et al.*, 2006).

Para poder llevar a cabo dicha calibración necesitamos colocar ambas cámaras a la misma altura (figura 5), esto con la intención de que los cambios entre las imágenes sean únicamente visibles en el plano horizontal y que el error vertical sea el mínimo posible o inexistente.

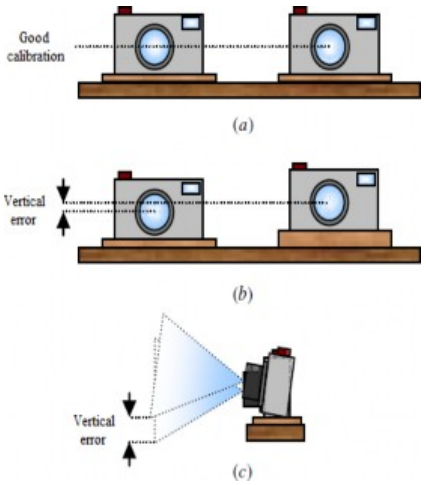

Figura 5: a) Posición de cámaras correcta. b) Posición incorrecta debido a la presencia de un error vertical (Strbac *et al.*, 2020).

El modelo de Zhang (figura 6) es un método de calibración de cámaras que usa técnicas de calibración tradicionales (conocidas como puntos de calibración) y técnicas de autocalibración (correspondencia entre los puntos de calibración cuando están en diferentes posiciones). Para realizar una calibración completa por este método son necesarias al menos tres imágenes diferentes del objetivo a calibrar, ya sea moviendo el objetivo o la misma cámara (figura 7).

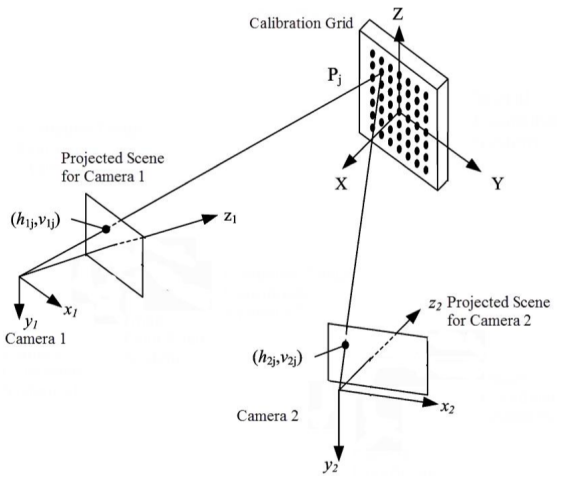

Figura 6: Esquema de calibración para un sistema de visión con 2 cámaras. (Chavelas y Diaz, 2019).

En la figura 6 se presentan los parámetros necesarios para llevar a cabo la calibración de las cámaras y estos son:

- X, Y, Z: Espacio de la malla de calibración.
- x_1, y_1, z_1: Espacio de cámara 1 (o izquierda).
- x_2, y_2, z_2: Espacio de cámara 2 (o derecha).
- h_{1j}, v_{1j}: Punto de la cámara 1 donde se proyecta la imagen de la escena.
- h_{2j}, v_{2j}: Punto de la cámara 2 donde se refleja la imagen de la escena

La principal aportación de este método es el extraer los parámetros intrínsecos K (distancia focal y el punto principal de ambas cámaras) ademas de los parámetros de calibración extrínsecos R y T. Los parámetros extrínsecos están relacionados con la posición 3D de las cámaras, expresan la transformación de un punto del mundo 3D a las coordenadas 2D de la cámara. Donde R denota la rotación y T una traducción, en la calibración estereoscópica, estos valores expresan la posición de una cámara (es decir, la derecha) en relación con la otra (es decir, la izquierda) (Chavelas y Diaz, 2019).

Figura 7: Ejemplo de imágenes utilizadas para la calibración de un arreglo de cámaras estéreo mediante el método Zhang.

2.6. *Caso de estudio*

En este documento presentamos una metodología (figura 8) en la cual se busca detectar armas de fuego tipo pistola mediante el uso de la arquitectura YoloV8 entrenada con un conjunto de datos modificado con imágenes originales.

Además se llevó a cabo la adaptación de dicha arquitectura con un algoritmo que, mediante el uso de imágenes estereoscópicas, es capaz de definir la distancia a la que se encuentra el objeto detectado. El uso de este algoritmo ayuda a eliminar detecciones erróneas como es en el caso de que el objeto detectado sea demasiado grande o demasiado pequeño como para clasificarlo como pistola de fuego.

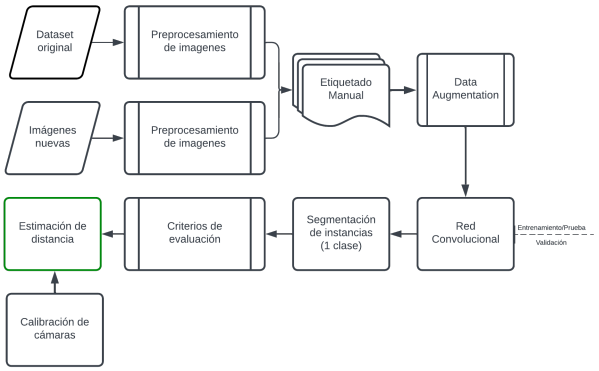

Figura 8: Metodología adoptada en este trabajo.

2.7. *Obtención de datos*

En (Olmos *et al.*, 2018) se presentó un conjunto de datos que consta de 3000 imágenes de armas tipo pistola, este consta de imágenes capturadas por el equipo ademas de imágenes recopiladas en redes sociales y resultados en metabuscadores. Además de estas imágenes se capturaron 500 imágenes originales las cuales constan de personas con un arma tipo pistola en mano así como imágenes del área de control en las que será probado el algoritmo sin armas de fuego presentes, esto con la finalidad de que la CNN logre detectar cuando un arma no está presente, logrando así evitar falsos positivos.

2.8. *Preprocesamiento y elaboración del conjunto de datos*

El preprocesamiento del conjunto de imágenes consistió en:

- Cambio de formato de imagen: Se estableció el formato JPG.
- Estandarización de tamaño a 640x640 pixeles.

Con la herramienta **Roboflow Annotate** (https://roboflow.com) se creó el nuevo dataset que cuenta con imágenes originales así ampliando la capacidad del conjunto de datos original (figura 9) conformando el ground truth de forma manual con la única etiqueta de clase: **Pistol**.

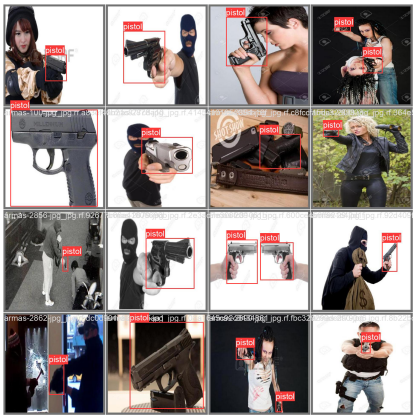

Figura 9: Imágenes presentes en el dataset original que fueron utilizadas en el entrenamiento, las etiquetas son consideradas ground truth.

Las imágenes fueron separadas en diferentes divisiones, las cuales son:

- Imágenes de entrenamiento
- Imágenes de validación
- Imágenes de prueba

Ademas de dicha división las imágenes de entrenamiento fueron aumentadas con los siguientes métodos:

- Rotación de la imagen entre -15 a +15 grados
- Voltear las imágenes horizontal y verticalmente
- Conversión a escala de grises
- Aumento de brillo entre -20 a +20

El aumento de datos se usa cuando se entrenan CNNs para aumentar el tamaño de los conjuntos de datos y ayudar a evitar el sobreajuste (overfitting). Al aumentar el tamaño del conjunto de datos, la red se expone a una mayor variedad de datos, lo que le ayuda a aprender características más robustas y hacer predicciones más precisas. También ayuda a reducir la sensibilidad a pequeñas variaciones en los datos mediante la introducción de transformaciones generadas aleatoriamente, como traducción, rotación y escalado, para aumentar el conjunto de datos existente. De esta manera, el modelo de CNN es capaz de generalizarse mejor ya que se entrena sobre una gama más amplia de puntos de datos. Tras realizar el aumento mencionado el entrenamiento se realizó con 7800 imágenes y la validación con 741 imágenes.

2.9. Algoritmo de estimación de distancia

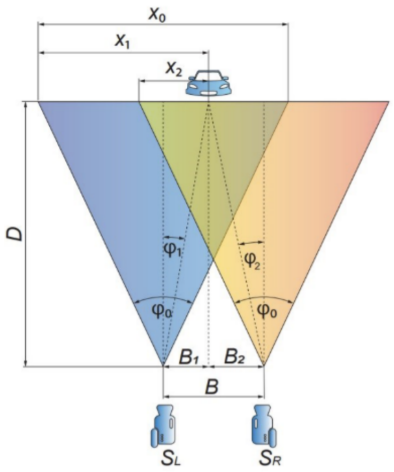

Figura 10: Parámetros importantes para la obtención de distancia mediante estereoscopía (Strbac *et al.*, 2020).

.

En la (figura 10) damos a conocer los parámetros necesarios para la estimación de distancia a la que se encuentra el objeto detectado:

- S_L y S_R representan a la cámara izquierda y derecha respectivamente.
- B es la distancia a la que se encuentran las cámaras entre ellas.
- φ_0 representa el campo de visión (field of view, FoV) de las cámaras (este valor cambia de acuerdo a cada cámara, y se puede obtener en sus especificaciones técnicas).
- D es la distancia existente entre las cámaras y el objeto detectado.

La fórmula para obtener la distancia D se representa en 3:

$$D = \frac{B}{tan(\varphi_1) + tan(\varphi_2} \tag{3}$$

En donde φ_1 y φ_2 son los ángulos entre el eje del lente de la cámara y la dirección al objeto detectado.

Otros tres párametros importantes son X_0, X_1 y X_2.

- X_0 representa el número de pixeles horizontales de las imágenes
- X_1 (imagen izquierda) y X_2 (imagen derecha) son el número de píxeles entre el punto medio del borde horizontal del cuadro delimitador de los objetos y el borde izquierdo de la imagen.

$$D = \frac{B \times X_0}{2\tan(\frac{\varphi_0}{2})(X_1 - X_2)} \tag{4}$$

Con esta fórmula 4 podemos estimar la distancia de cualquier objeto que este presente en ambas imágenes.

El algoritmo que se definió para la obtención de la distancia constará de los siguientes criterios:

1. En ambas detecciones el objeto debe pertenecer a la misma clase (pistola).
2. El objeto en la imagen de la cámara derecha debe estar mas cerca de la orilla izquierda a comparación del mismo objeto en la imagen izquierda (fórmula 5) donde X_i y X_d son el número de pixeles horizontales en la imagen izquierda y derecha respectivamente, W_i y W_d son el ancho del cuadro delimitador izquierdo y derecho respectivamente.

$$X_i + W_i < (X_d + W_d)\ y\ (X_i < X_d). \tag{5}$$

3. El tamaño de los cuadros delimitadores deben ser lo mas parecidos posibles de acuerdo a un umbral estimado. Utilizando la (fórmula 6) donde W_i y W_d son el ancho del cuadro delimitador izquierdo y derecho respectivamente, y, H_i y H_d son la altura del cuadro delimitador izquierdo y derecho respectivamente. La obtención del valor final del umbral se llevó a cabo variando el valor de este mismo en diferentes pruebas. El valor inicial fue de 0,75 y este mismo fue variando en cada prueba hasta 0,95 en incrementos de 0,05 siendo el valor 0,90 el que brindó mejores resultados.

$$(\frac{min(W_i, W_d)}{max(W_i, W_d)} > 0{,}9)\ y\ (\frac{min(H_i, H_d)}{max(H_i, H_d)} > 0{,}9) \tag{6}$$

4. La diferencia entre los centroides en el plano vertical (y) debe ser lo mas pequeña posible, de acuerdo a un umbral definido (fórmula 7). Dicho umbral también fue obtenido mediante la variación de su valor en múltiples pruebas. El valor inicial fue de 1 *pixel* y este mismo fue variando en cada prueba hasta 10 *pixeles* en incrementos de 1 *pixel* siendo el valor de 5 *pixeles* el que brindó mejores resultados.

$$|(Cy_i - Cy_d)| < 5\ pixeles \tag{7}$$

Estos criterios son utilizados para verificar que el objeto que esta siendo detectado en la cámara izquierda es el mismo objeto que está siendo detectado en la cámara derecha, en dado caso de que se cumplan todos los criterios, el algoritmo llevará a cabo su último paso, el calcular la distancia mediante el uso de la fórmula 4.

3. Resultados

3.1. Entrenamiento

El algoritmo optimizador utilizado durante el entrenamiento fue el descenso de gradiente estocástico (SGD por sus siglas en ingles); con un factor de aprendizaje (learning rate) de 0.01; el valor *IoU* predeterminado es de 0.6; 100 epocas de entrenamiento y un tamaño de lote (batch size) de 64.

Los procesos fueron codificados en Python. Para la fase de entrenamiento fue utilizado un equipo con procesador Intel Core i7 9750H, RAM 16GB, así como una Unidad Gráfica de Procesamiento (GPU, por sus siglas en inglés) en tarjeta Nvidia TITAN X. El entrenamiento tomó aproximadamente 4 horas, los siguientes resultados (figura 11) fueron obtenidos durante el entrenamiento con el uso del conjunto de validación, el cual consta de 741 imágenes que nunca han sido, ni serán, presentadas al modelo.

3.2. Métricas

Las métricas presentadas durante el entrenamiento son las mismas que se presentaron en las ecuaciones (1), (2), así como $mAP50$ y $map50-95$. Se guardaron los pesos que brindaron los mejores resultados en el conjunto de evaluación y estos mismos fueron utilizados en los experimentos en tiempo real, los resultados brindados por estos pesos fueron los siguientes (tabla 1):

Tabla 1: Métricas obtenidas tras evaluar el modelo entrenado.

Imagenes	Instancias	Precisión	IoU	Recall	mAP50.
741	783	0.922	0.6	0.815	0.903

3.3. Pruebas de reconocimiento

En la figura 12 se incluyen algunos ejemplos de los resultados obtenidos al evaluar el modelo con imágenes presentes en el conjunto de evaluación, en este caso únicamente se esta presentando la detección de la pistola, no se está intentando obtener la distancia a la que se encuentra el objeto.

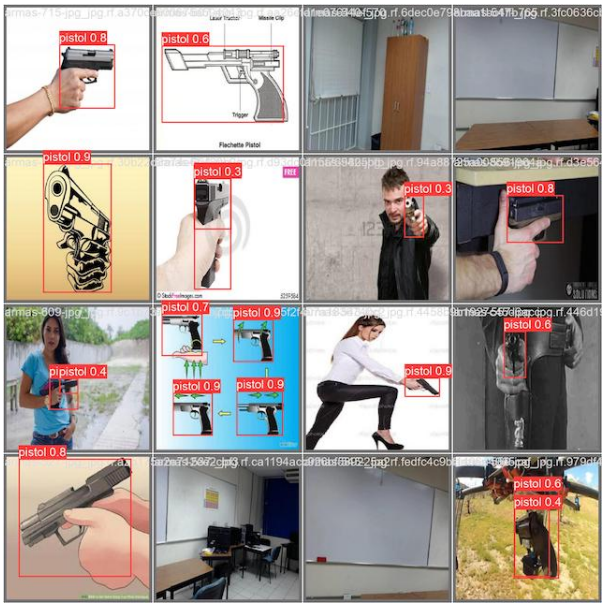

Figura 12: Predicciones realizadas por el modelo, los valores encima de la caja representan la confianza del modelo.

3.4. Reconocimiento en tiempo real

Las pruebas de reconocimiento en tiempo real fueron llevadas en un área controlada en el Instituto Tecnológico de La Paz, esto con la finalidad de experimentar a diferentes distancias sin distracciones por parte del mundo exterior, ademas, la confianza mínima del modelo se colocó en un 0,60 puesto que se consideró que esta confianza es lo suficientemente elevada como para eliminar falsos negativos durante el funcionamiento. Las cámaras utilizadas durante los experimentos son del modelo *"Logitech C920 Pro HD Webcam, 1080p "*.

En este punto se está evaluando que tan viable es utilizar este modelo para la detección de armas tipo pistola en tiempo real además de definir el tamaño del error respecto a la obtención de distancia, los resultados de la estimación de distancia se presentan en la tabla 2 y en la figura 14, además mostramos un ejemplo de la detección y estimación en la figura 13.

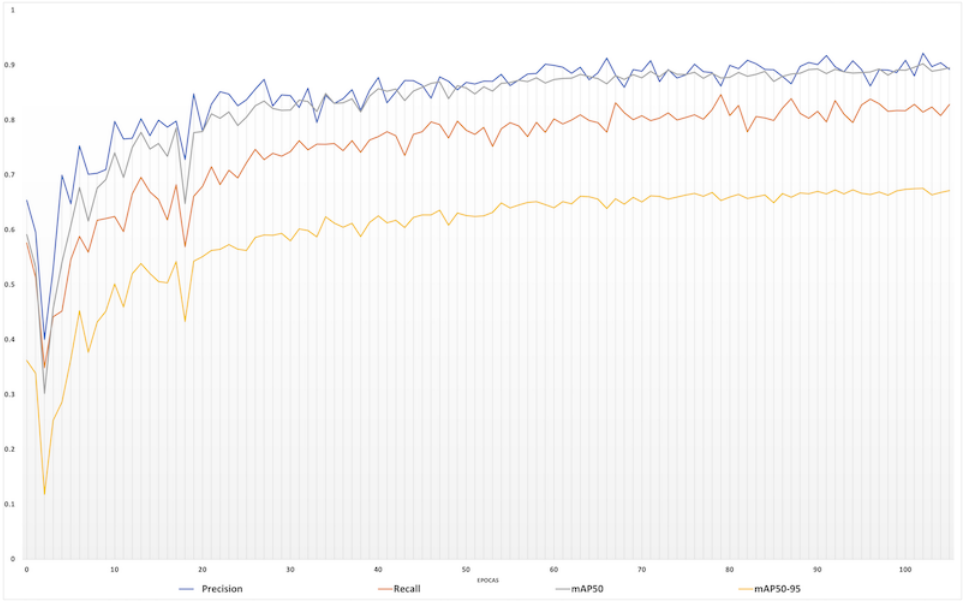

Figura 11: Resultados de la fase de entrenamiento.

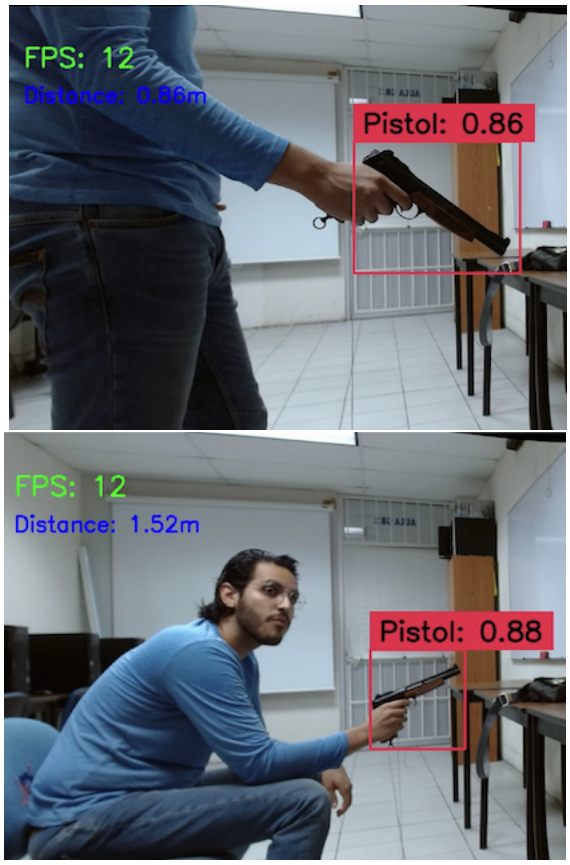

Figura 13: Imágenes de los experimentos llevados a cabo.

Tabla 2: Estimación de distancia.

Distancia Real (m)	Distancia Estimada (m)	Error (m)
0.6	0.591	0.009
0.7	0.732	-0.032
0.8	0.824	-0.024
0.9	0.875	0.025
1.0	0.954	0.046
1.1	1.063	0.037
1.2	1.183	0.017
1.3	1.301	-0.001
1.4	1.352	0.048
1.5	1.456	0.044
1.6	1.507	0.093
1.7	1.573	0.127
1.8	1.718	0.082
1.9	1.846	0.054
2.0	1.925	0.075
2.1	1.978	0.122
2.2	1.978	0.170
2.3	2.138	0.162
2.4	2.272	0.128
2.5	2.378	0.122
2.6	2.367	0.233
2.7	2.421	0.279
2.8	2.726	0.074
2.9	2.675	0.225
3.0	2.822	0.0178
3.1	2.97	0.13
	Error promedio (m)	**.093**

Como se logra ver en dicha la figura 13 el modelo corre a 12 cuadros por segundo de manera constante los cuales son mas que suficientes para un funcionamiento en tiempo real, además podemos apreciar como el modelo detecta el objeto con diferentes confianzas de acuerdo a la cámara que lo esté detectando.

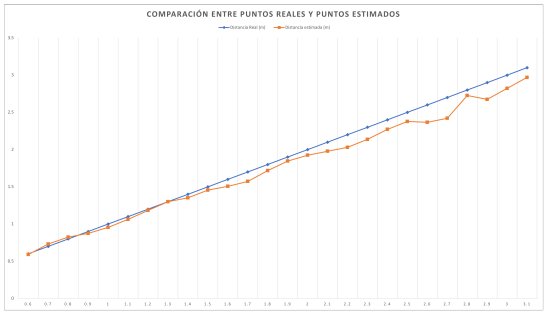

Figura 14: Comparación gráfica entre la distancia real y la distancia estimada

4. Conclusiones

El uso de las redes convolucionales aplicadas a sistemas de videovigilancia constituyen una alternativa viable para la detección de armas con alta precisión, especialmente utilizando un algoritmo tipo YOLO ya que esté facilitó el funcionamiento de este sistema en tiempo real.

Además no solamente se logró el uso del sistema en tiempo real, si no que alcanzamos superar la precisión de múltiples trabajos en el estado del arte. Nuestra precisión final de 92.2 % (tabla 1) ha demostrado ser mas elevada que lo logrado con arquitecturas tipo FASTER R-CNN y VGG16 SSD como se presento en (Olmos *et al.*, 2018) y (SUSMITHA y KUMAR, 2023) donde se alcanzó una precisión máxima de 84.21 % y 84.6 % respectivamente.

El algoritmo de estimación de distancia demostró ser lo suficientemente funcional hasta una distancia de 3.1 metros, manejando un error promedio de 0.093 metros o 9.3 centímetros. Sin embargo, dicho algoritmo se encuentra limitado por la detección precisa del arma, la cual demostró tener problemas en la detección del objeto a distancias mayores a 3 metros ya que este mismo logra detectar el objeto pero con una confianza demasiado baja (menor a 0.5). Como consideraciones futuras se planea ampliar el algoritmo de estimación de distancia mediante el uso de un sistema de reconocimiento de esquinas con la intención de no solo lograr la estimación de la distancia del objeto si no que también estimar el tamaño real del objeto para lograr eliminar detecciones que se consideren demasiado grandes como para realmente ser un arma tipo pistola.

Agradecimientos

Al Consejo Nacional de Ciencia y Tecnología (CONACYT), por el soporte académico brindado mediante la beca de estudios de posgrado.

Al Tecnológico Nacional de México (TecNM) campus La Paz, por la provisión de equipo de computo para las prácticas de este artículo.

Agradezco especialmente a mi director de tesis, el Dr. Saúl Martínez Díaz y a los miembros de mi comité tutorial, el M. en C. Jorge Enrique Luna Taylor y la M. en C. Iliana Castro Liera, por sus aportes y el gran apoyo que recibí de su parte en todo momento.

A mi madre, Martha Lorena Elías Talamantes, mi abuela, Luisa Clementina Talamantes Taylor y a mi padre Osías Schcolnik Corral, por apoyarme en todo mi camino como estudiante y motivarme a alcanzar este objetivo.

Y a mi pareja, Samantha Daniela Vázquez Corona, por ser la persona que me convenció a aceptar este reto y por ser mi mayor apoyo durante esta etapa de mi vida, gracias por todo.

Referencias

Arballo, J. J. P., Diaz, S. M., Liera, M. A. C., y Taylor, J. E. L. (2022). Deteccion de cambio en superficie costera mediante la segmentacion de imagenes aereas utilizando redes neuronales convolucionales. *Padi Boletin Cientifico de Ciencias Basicas e Ingenierias del ICBI*, 10:136–144.

Chavelas, S. M. y Diaz, S. M. (2019). Algoritmo paralelo en gpu para el rastreo de armas de fuego tipo pistola tesis.

Deepthi, T., Gaayathri, R., y Shanthosh, S. (2018). Firearm recognition using convolutional neural network. *academia.edu*.

Flitton, G., Breckon, T. P., y Megherbi, N. (2013). A comparison of 3d interest point descriptors with application to airport baggage object detection in complex ct imagery. *Pattern Recognition*, 46:2420–2436.

Gesick, R., Saritac, C., y Hung, C. C. (2009). Automatic image analysis process for the detection of concealed weapons. *ACM International Conference Proceeding Series*.

Grega, M., Matiolanski, A., Guzik, P., y Leszczuk, M. (2016). Automated detection of firearms and knives in a cctv image.

INEGI (2021). Datos preliminares revelan que en 2020 se registraron 36579 homicidios.

Jocher, G. y Ultralytics (2023). Github yolov8 - ultralytics. Accessed on February 13, 2023.

Liu, F., Zhao, H., y Liu, W. (2022). Improved garment detection algorithm based on yolov5. pp. 1054–1058.

Olmos, R., Tabik, S., y Herrera, F. (2018). Automatic handgun detection alarm in videos using deep learning. *Neurocomputing*, 275:66–72.

RangeKing (2023). Brief summary of yolov8 model structure. Accessed on February 13, 2023.

Redmon, J., Divvala, S., Girshick, R., y Farhadi, A. (2016). You only look once: Unified, real-time object detection. pp. 779–788.

Remondino, F. ., Fraser, C., y Remondino, F. (2006). Digital camera calibration methods. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVI:266–272.

Strbac, B., Gostovic, M., Lukac, Z., y Samardzija, D. (2020). Yolo multi-camera object detection and distance estimation. *2020 Zooming Innovation in Consumer Technologies Conference, ZINC 2020*, pp. 26–30.

SUSMITHA, D. y KUMAR, S. V. (2023). Weapon detection using artificial intelligence and deep learning for security applications. *Journal of Engineering Sciences*, 14(01).

Veit, A., Matera, T., Neumann, L., Matas, J., y Belongie, S. (2016). Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*.