



**Implementación de un sistema de alerta temprana para la seguridad física en la
Universidad de las Fuerzas Armadas ESPE con visión artificial para la detección de
armas de fuego**

Collahuazo Collahuazo, José Danilo y Farinango Medina, Paola Cristina

Departamento de Ciencias de la Computación

Carrera de Tecnologías de la Información

Trabajo de integración curricular, previo a la obtención del título de Ingeniero/a en Tecnologías
de la Información

Ing. Tapia León, Freddy Mauricio Ph. D.

01 de marzo del 2024

Hoja de Resultados de la Herramienta



Plagiarism report

MIC_Collahuazo_Farinango_Final_Solo...

Firmado digitalmente por FREDDY MAURICIO TAPIA LEON
Nombre de reconocimiento (DN): c=EC, o=BANCO CENTRAL DEL ECUADOR, ou=ENTIDAD DE CERTIFICACION DE INFORMACION-EC/BCE, l=QUITO, serialNumber=0000310294, cn=FREDDY MAURICIO TAPIA LEON
Fecha: 2024.03.01 20:22:20 -05'00'

Scan details

Scan time: February 29th, 2024 at 3:2 UTC
Total Pages: 53
Total Words: 13210

Plagiarism Detection



Types of plagiarism		Words
Identical	0.6%	83
Minor Changes	0.3%	44
Paraphrased	0%	0
Omitted Words	5.9%	783

AI Content Detection



Text coverage
 AI text
 Human text

🔍 Plagiarism Results: (10)

🌐 (PDF) Detección de armas tipo pistola mediante el uso de redes convoluci... 0.5%

https://www.researchgate.net/publication/373849112_deteccion_de_armas_tipo_pistola_mediante_el_uso_de...

Aarón Schcolnik-Elias

ArticlePDF AvailableDetección de armas tipo pistola mediante el uso de redes convolucionales con una arquitectura tipo YOLO y estereos...

🌐 TesisSchcolnik.pdf 0.3%

<http://posgrado.itlp.edu.mx/uploads/archivos/tesisschcolnik.pdf>

INSTITUTO TECNOLÓGICO DE LA PAZ DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN MAESTRÍA EN SISTEMAS COMPUTACIONALES DETECCIÓN DE ARMA...

🌐 (PDF) Comparison of YOLOv3, YOLOv5s and MobileNet-SSD V2 for Real-Ti... 0.2%

https://www.researchgate.net/publication/353211011_comparison_of_yolov3_yolov5s_and_mobilenet-ssd_v2...

Rakkshab Iyer

Home Anthropology Cultural Anthropology Masks ArticlePDF AvailableComparison of YOLOv3, YOLOv5s and MobileNet-SSD V2 for Real-Time...

🌐 (PDF) Comparison of YOLOv3, YOLOv5s and MobileNet-SSD V2 for Real-Ti... 0.2%

https://www.academia.edu/49690792/comparison_of_yolov3_yolov5s_and_mobilenet-ssd_v2_for_real_time_...

Rakkshab Iyer

Academia.edu no longer supports Internet Explorer. To browse Academia.edu and the wider internet faster and more securely, p...

Certified by

About this report
help.copleaks.com

copleaks.com



**Departamento de Ciencias de la Computación
Carrera de Tecnologías de la Información**

Certificación

Certifico que el trabajo de integración curricular: **“Implementación de un sistema de alerta temprana para la seguridad física en la Universidad de las Fuerzas Armadas ESPE con visión artificial para la detección de armas de fuego”** fue realizado por los señores **Collahuazo Collahuazo José Danilo y Farinango Medina Paola Cristina**, el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizada en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Sangolquí, 20 de marzo del 2024

Firma

FREDDY
MAURICIO
TAPIA LEON

Firmado digitalmente por FREDDY MAURICIO TAPIA LEON
Nombre de reconocimiento (DN):
o=EC, ou=BANCO CENTRAL DEL
EQUADOR, ou=ENTIDAD DE
CERTIFICACION DE INFORMACION-
EGBCE, ou=QUITO,
serialNumber=0000310294,
cn=FREDDY MAURICIO TAPIA LEON
Fecha: 2024.03.20 21:25:48 -0500'

.....
Ing. Tapia León Freddy Mauricio, PhD.

C. C. 1714745690



**Departamento de Ciencias de la Computación
Carrera de Tecnologías de la Información**

Responsabilidad de Autoría

Nosotros, **Collahuazo Collahuazo José Danilo y Farinango Medina Paola Cristina**, con cédulas de ciudadanía n°1725344145 y n°1754884425, declaramos que el contenido, ideas y criterios del trabajo de integración curricular: **Implementación de un sistema de alerta temprana para la seguridad física en la Universidad de las Fuerzas Armadas ESPE con visión artificial para la detección de armas de fuego** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 18 de marzo del 2024

Firma

Collahuazo Collahuazo José Danilo

C.C.: 1725344145

Farinango Medina Paola Cristina

C.C.: 1754884425



**Departamento de Ciencias de la Computación
Carrera de Tecnologías de la Información**

Autorización de Publicación

Nosotros **Collahuazo Collahuazo José Danilo y Farinango Medina Paola Cristina**, con cédulas de ciudadanía n°1725344145 y n°1754884425 autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de integración curricular: **Implementación de un sistema de alerta temprana para la seguridad física en la Universidad de las Fuerzas Armadas ESPE con visión artificial para la detección de armas de fuego** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi/nuestra responsabilidad.

Sangolquí, 18 de marzo del 2024

Firma

Collahuazo Collahuazo José Danilo

C.C.: 1725344145

Farinango Medina Paola Cristina

C.C.: 1754884425

Dedicatoria

Este trabajo está dedicado a Dios, quien nos ha permitido llegar hasta este punto de nuestras vidas. Cada página está impregnada con profunda gratitud y humildad.

A nuestros padres,

Por su amor incondicional y apoyo constante, sin los cuales no habríamos alcanzado esta meta. Este logro es también suyo.

A nuestros hermanos,

Por su paciencia y complicidad, que han hecho este viaje más llevadero y memorable. Su fe en nosotros ha sido crucial para nuestro éxito académico.

A nuestros amigos,

Por ser una luz en nuestro camino y por brindarnos su amistad incondicional. Gracias por estar siempre presentes y por ser una fuente constante de alegría y compañía.

A Segundo Collahuazo, le rendimos homenaje por ser una inspiración constante. Cada palabra de esta tesis lleva consigo su sabiduría y ejemplo.

A Galo Monteverde, expresamos nuestro más sincero agradecimiento por su apoyo inquebrantable y por alentar a Paola a seguir sus sueños.

Este trabajo es el resultado del amor, apoyo y aliento de cada uno de ustedes en nuestra travesía. Gracias por formar parte de este capítulo inolvidable en nuestras vidas.

Agradecimientos

Queremos expresar nuestro agradecimiento a todas las personas que han sido pieza clave de nuestro viaje universitario. En primer lugar, agradecemos a nuestros docentes, quienes desde el primer día han sido personas admirables, que nos han brindado conocimiento, orientación y apoyo incondicional.

Un agradecimiento especial a nuestro tutor Freddy Tapia, cuyo valioso tiempo con sus sabios consejos y constante motivación nos ha impulsado a perseguir y alcanzar nuestra meta. Su apoyo ha sido invaluable para esta etapa académica.

A nuestras familias. Por su apoyo incondicional, palabras de ánimo y ser el pilar constante en nuestras vidas, lo que ha sido esencial para superar los desafíos y alcanzar este logro.

No podemos pasar por alto a nuestros queridos amigos, quienes han sido compañeros leales desde el inicio, les agradecemos por darnos una amistad sólida y crecer juntos. Su amistad ha sido un regalo preciado que llevaremos con nosotros siempre.

Gracias a todos por ser haber sido parte esencial de esta etapa.

Índice de Contenido

Resumen	13
Abstract.....	14
Capítulo I: Introducción	15
Antecedentes	15
Problema	16
Justificación.....	17
Objetivos	18
Alcance	18
Capítulo II: Marco Teórico y Estado del Arte	19
Marco Teórico	19
Visión artificial	19
Aprendizaje profundo	20
Redes Neuronales Convolucionales.....	20
Detección de Objetos	22
Circuito Cerrado de Televisión	22
Roboflow	22
Google Colab	23
Kaggle.....	23
YOLOv5 (You Only Look Once).....	23
Python.....	25
PyTorch.....	25
OpenCV	25
Tkinter	26
Chime.....	26
Estado del Arte.....	26
Revisión de los Estudios Primarios.....	31
Metodología	37
Metodología Scrum	38
Técnica de Recolección de Datos	39

Capítulo III: Desarrollo	40
Construcción del Modelo de Red Convolutiva.....	41
Etapas de la Arquitectura del Modelo	42
Entrenamiento del modelo.....	45
Implementación del Modelo Entrenado.....	59
Vista resultado Interfaz.....	65
Empleo del sistema de videovigilancia	67
Capítulo IV: Resultados	72
Implementación	72
Interfaz	73
Capítulo V: Conclusiones y Recomendaciones.....	76
Conclusiones.....	76
Recomendaciones.....	78
Capítulo VI: Trabajos Futuros	79
Bibliografía.....	80

Índice de Tablas

Tabla 1. Cadena de búsqueda en las diferentes bases de datos	28
Tabla 2. Resultados de la cadena de búsqueda.	30
Tabla 3. Estudios primarios	30
Tabla 4. Detección de Objetos en Tiempo Real - CAM 1	47
Tabla 5. Equipos de Desarrollo e Implementación	71

Índice de Figuras

Figura 1. Diagrama Esquemático de una Red Neuronal Convolutacional.	21
Figura 2. Sistema de Detección Yolo	24
Figura 3. Pasos Manifiesto de Desarrollo Ágil de Software	38
Figura 4 Estructura del Sistema de Detección y Alerta.	40
Figura 5 Arquitectura del Modelo de Red Convolutacional	41
Figura 6. Etiquetado de la Clase	43
Figura 7. Agregación de Rotación en Imágenes de Armas de Fuego	44
Figura 8. Diagrama de Infraestructura de Entrenamiento del Modelo	45
Figura 9. Importación de Librerías Yolov5 y sus requerimientos	46
Figura 10. Importación de Api Key Yolov5 (Dataset)	47
Figura 11. Importación de parámetros y modelo de Entrenamiento Yolov5	47
Figura 12. Métricas de Evaluación de Matriz de Confusión.	49
Figura 13. Generación de Precisión del Modelo.	51
Figura 14. Evaluación de Precisión del Modelo de Detección de Objetos.	52
Figura 15. Resultados Entrenamiento – Ajuste Fino	53
Figura 16. Evaluación de Sensibilidad del Modelo de Detección de Objeto	54
Figura 17. Evaluación de la Sensibilidad de Precisión del Modelo	55
Figura 18. Valor de confianza optimo del Modelo de Detección de Objetos.	56
Figura 19. Matriz de Confusión del Modelo de Detección de Objetos.	57
Figura 20. Resultados del modelo entrenado para identificar armas de fuego en Imágenes.	58
Figura 21. Diagrama de Implementación del Modelo Entrenado	60
Figura 22. Importación de Librerías Python	61
Figura 23. Ruta Estática del Modelo Entrenado	61
Figura 24. Definición del Método para activar las cámaras	62

Figura 25. Integración de Cámara 1 y Modelo Entrenado	63
Figura 26. Activación de Interfaz Principal	64
Figura 27. Pantalla Principal del Sistema de Monitoreo.	65
Figura 28. Detección de Objetos en Tiempo Real – CAM 1	66
Figura 29. Ingreso Sistema de Videovigilancia Hikvision	67
Figura 30. Configuración de Resolución para Video en Tiempo Real	68
Figura 31. Vista en Directo Cámara Hikvision	68
Figura 32. Arquitectura del sistema	69
Figura 33. Funcionamiento del sistema de detección	73
Figura 34. Detección de Arma simultáneamente en dos cámaras.	74
Figura 35. Detección de Arma con Proximidad Cercana	74
Figura 36. Implementación Modelo Deep Learning – Sistema de Videovigilancia Hikvision	75

Resumen

En un contexto mundial donde la amenaza de la violencia armada se intensifica, la expansión de armas de fuego se presenta como un peligro creciente a escala global. Este desafío, sin embargo, no es ajeno a nuestra realidad. En Ecuador, nos enfrentamos a la necesidad de abordar la creciente amenaza de la violencia armada. En respuesta a esta urgencia, se propone un sistema innovador de alerta temprana para la seguridad física en la Universidad de las Fuerzas Armadas ESPE. Este sistema, basado en visión artificial, busca lograr la detección efectiva de armas de fuego, resaltando características distintivas y el uso apropiado de las tecnologías con beneficio a la sociedad.

El desarrollo de este sistema se distribuyó en dos puntos claves, el primero en la creación de una red neuronal convolucional precisa y eficiente; y el segundo, el diseño de una interfaz que, combinada con la red neuronal creada en el punto anterior, le dan un valor agregado en la interacción de la herramienta con el personal a cargo del monitoreo.

El sistema incorpora técnicas y herramientas como Roboflow para el procesamiento de las imágenes de armas de fuego, centrándose en la segmentación de las imágenes y asignación de la correspondiente etiqueta. Para llevar a cabo el proceso se utilizó el modelo de entrenamiento YOLOv5 con el fin de crear un algoritmo de aprendizaje automático para la detección en tiempo real, programado en Python. Este enfoque busca elevar la presión y eficacia del sistema. Finalmente, se comprueba la presión del modelo, con una presión del 95% de efectividad, además de implementar el modelo a un prototipo de monitoreo facilitado la robustez y confiabilidad en su uso.

Palabras Clave: Visión artificial, Detección de armas, Redes neuronales convolucionales, Alerta temprana, Institución Educativa.

Abstract

In a global context where the threat of armed violence intensifies, the expansion of firearms is presented as a growing danger on a global scale. This challenge, however, is not foreign to our reality. In Ecuador, we are faced with the need to address the growing threat of armed violence. In response to this urgency, an innovative early warning system for physical security at the ESPE Armed Forces University is proposed. This system, based on artificial vision, seeks to achieve effective detection of firearms, highlighting distinctive characteristics and the appropriate use of technologies for the benefit of society.

The development of this system was divided into two key points, the first in the creation of a precise and efficient convolutional neural network; and the second, the design of an interface that, combined with the neural network created in the previous point, gives added value in the interaction of the tool with the personnel in charge of monitoring.

The system incorporated techniques and tools such as Roboflow for processing firearm images, focusing on image segmentation and assignment of the corresponding label. To carry out the process, the YOLOv5 training model was used in order to create a machine learning algorithm for real-time detection, programmed in Python. This approach seeks to increase the pressure and efficiency of the system. Finally, the pressure of the model is checked, with a pressure of 95% effectiveness, in addition to implementing the model to a web application to facilitate robustness and reliability in its use.

Keywords: Computer Vision, Weapon Detection, Convolutional Neural Networks, Early Warning, Educational Institution.

Capítulo I: Introducción

Antecedentes

Ecuador enfrenta una grave crisis de seguridad desde hace dos años, con un aumento preocupante de delitos vinculados a la delincuencia organizada, posicionándolo como el tercer país más violento en Sudamérica (Rodríguez, 2023). La falta de supervisión dentro de las prisiones ha permitido que las bandas aprovechen su influencia y tomen el control, generando así un aumento significativo de la violencia, dentro y fuera de estos centros (Reuters, 2024). Como solución a estos actos violentos, en abril del 2023, mediante el decreto ejecutivo número 707 el expresidente Lasso permitió la posesión y porte de armas a toda persona natural bajo ciertas condiciones con forme a la ley. Con el fin de que la ciudadanía se sienta protegida (El Telégrafo, 2023).

Por otro lado, las instituciones de educación superior del país coinciden que permitir el uso de armas promueve la aparición de conflictos y violencia en el entorno, al contrario de su principal objetivo que es proteger la integridad de todos los miembros.

En este contexto, el despliegue de cámaras de vigilancia y sistemas de circuito cerrado no solo redefine nuestra concepción de la seguridad, sino que también introduce una era donde la tecnología desempeña un papel protagónico en la protección de espacios y personas. La detección de armas, la identificación de objetos y la visión artificial son elementos esenciales que han elevado la eficacia de estos sistemas, otorgándoles la capacidad de anticipar y prevenir posibles amenazas.

En referencia a la detección temprana de objetos sospechosos, la clave del proceso radica en la personalización y ajuste de modelos de redes neuronales convolucionales, una técnica avanzada de aprendizaje profundo, para clasificar y posicionar estos objetos en tiempo real. Este método no solo implica un progreso técnico, sino que también marca un logro

significativo en la implementación práctica del sistema para mejorar la seguridad en entornos cruciales (Gali et al., 2022).

No obstante, el buscar mejorar los sistemas de detección de objetos, conlleva la integración de más elementos que aporten facilidad y simplicidad al desarrollo del sistema, empezando por la perfección del modelo mediante el uso de plataformas, como por ejemplo Roboflow que amplifica el nivel del entrenamiento, con el fin de alcanzar una excelente eficiencia (Thiruthanigesan et al., 2021).

Por tal razón, el presente proyecto tiene como finalidad elaborar un modelo de detección en tiempo real, de la mano de un sistema de cámaras que al combinarlos cubran con las necesidades de reconocer y alertar del uso de armas dentro del entorno educativo.

Problema

Según diversas fuentes especializadas, destacan que la persistente problemática de la inseguridad en Ecuador, caracterizada por la violencia y la proliferación del uso de armas, representa un desafío considerable para la serenidad de la sociedad. Los elevados índices de criminalidad han suscitado una creciente inquietud, impactando tanto en la percepción de seguridad de los ciudadanos como poniendo en peligro la integridad física de las comunidades. La presencia constante de armas en diversos incidentes violentos ha contribuido a la creación de un entorno de riesgo, enfatizando la imperiosa necesidad de abordar esta situación de manera efectiva. La complejidad de este problema demanda enfoques innovadores que no solo identifiquen los factores que desencadenan la inseguridad, sino que también propongan soluciones eficaces para fomentar un entorno más seguro y resguardar la integridad de todos los ciudadanos (MELLA, 2023).

En la actualidad, las cámaras de seguridad y los sistemas de videovigilancia desempeñan un papel crucial en la lucha contra la creciente inseguridad, especialmente asociada al fácil

acceso a armas de fuego. La principal función de estas infraestructuras radica en mejorar la detección temprana de armas, contribuyendo significativamente a prevenir incidentes y fortaleciendo la seguridad comunitaria (Schcolnik-Elias et al., 2023).

El propósito de este trabajo es fortalecer la seguridad física del campus universitario de las fuerzas armadas ESPE, se propone la implementación de un sistema de alerta temprana. El sistema aprovecha eficientemente tecnologías como YOLOv5 y Roboflow para ofrecer respuestas inmediatas, evidenciando así la capacidad tecnológica para abordar riesgos de manera proactiva.

Con base en la problemática delineada anteriormente, se formula la siguiente interrogante de investigación:

¿Cómo puede la implementación de un sistema de alerta temprana, basado en tecnologías como YOLOv5 y Roboflow, fortalecer de manera efectiva la seguridad física en el campus universitario de las Fuerzas Armadas ESPE, considerando la persistente problemática de inseguridad en Ecuador caracterizada por la proliferación del uso de armas?

Justificación

La escalada de violencia y el aumento en el uso de armas de fuego han generado una amenaza tangible para la seguridad ciudadana y la integridad de las comunidades, destacando la necesidad urgente de soluciones innovadoras.

La elección de estrategias tecnológicas se basa en la integración de herramientas que no solo permiten la identificación temprana de armas, contribuyendo a prevenir incidentes, sino que también se adapten de manera efectiva a la complejidad y diversidad de situaciones presentes en el entorno universitario. Esta iniciativa no solo fortalecerá la seguridad física del campus, incluso se establecerá un modelo valioso para la aplicación efectiva de tecnologías avanzadas en la gestión integral de la seguridad en entornos educativos y comunitarios.

Por las razones antes mencionadas, se necesita que el sistema este resguardado por tecnologías vanguardista que permitan realizar la detección e identificación en este caso de armas de fuego, además, de alertar de manera temprana, la presencia del objeto a mención dentro de la Universidad, al personal encargado. Después de todo, el sistema debe de brindar respuestas en tiempo real, siendo esta una medida esencial para contrarrestar la problemática persistente.

Objetivos

General

Desarrollar un sistema de alerta temprana basado en visión artificial para la detección de armas de fuego con el fin de fortalecer la seguridad y protección de la comunidad universitaria, permitiendo una respuesta ágil ante situaciones de emergencia y detectando posibles amenazas.

Específicos

- Realizar una revisión completa de la literatura científica y técnica relacionada con la detección de amenazas y la alerta temprana utilizando visión artificial en entornos similares, como campus universitarios u otros espacios públicos.
- Desarrollar e implementar el sistema de alerta temprana con la integración de un modelo de detección de amenazas que permita la generación anticipada de alertas.
- Evaluar la eficiencia del sistema para proporcionar resultados detallados que servirán como base en la elaboración de una guía de superación.

Alcance

Este proyecto se enfocará específicamente en la implementación de un sistema de alerta temprana para fortalecer la seguridad en las residencias ubicadas dentro de la Universidad de las Fuerzas Armadas ESPE. La intervención se centrará en los bloques residenciales A y B,

caracterizados por edificios de 5 pisos, cada uno cuenta con pasillos equipados con un par de cámaras de seguridad.

La ejecución del sistema abarcará el análisis y adaptación de las tecnologías seleccionadas con el fin de detectar de armas en tiempo real dentro de los pasillos residenciales. La implementación se llevará a cabo considerando la distribución de las cámaras existentes y su integración efectiva con el sistema de alerta temprana.

Es fundamental destacar que el ámbito de este proyecto se limita a las áreas residenciales designadas, priorizando la seguridad en este contexto específico dentro del campus universitario. La implementación y validación del sistema se realizarán con un enfoque preciso en los bloques A y B, contribuyendo así a la seguridad integral de la comunidad estudiantil en dichas residencias.

Capítulo II: Marco Teórico y Estado del Arte

En este capítulo se exponen conceptos elementales que aportan al desarrollo de este proyecto en conjunto con la revisión del estado del arte. De manera que se realiza la descripción de los enfoques acorde a las diferentes técnicas, seguida por las herramientas que se aplican. Finalmente, se detalla el análisis del proceso que se llevó a cabo para la revisión del estado del arte.

Marco Teórico

Visión artificial

La visión artificial o también conocida como visión por computadora se ha vuelto cada vez más crucial en la sociedad actual debido a la abrumadora cantidad de datos visuales generados diariamente a nivel global. Este campo tiene diversas aplicaciones prácticas, desde la detección de objetos y personas en videos de vigilancia hasta el desarrollo de nuevas tecnologías. Los recientes avances en el aprendizaje profundo han tenido un impacto significativo al mejorar la

precisión y eficacia de la visión por computadora en el procesamiento de grandes volúmenes de datos visuales (Szeliski, 2022).

Aprendizaje Profundo

La técnica del aprendizaje profundo, o Deep Learning, habilita a los modelos computacionales para adquirir representaciones de datos con diversos niveles de abstracción, a través de la implementación de múltiples capas de procesamiento (Goodfellow et al., 2017). Esta estrategia ha transformado significativamente diversos campos, marcando avances notables en áreas como el reconocimiento de voz, la identificación visual de objetos, la detección de objetos. En el proceso de aprendizaje profundo, se emplean algoritmos de retro propagación con el fin de guiar los ajustes de los parámetros internos de la máquina en cada capa, tomando como referencia la representación de la capa precedente. Las redes convolucionales profundas se destacan en el procesamiento de imágenes, videos, voz y audio, mientras que las redes recurrentes resultan eficaces para manejar datos secuenciales como texto y voz (Lecun et al., 2015).

Redes Neuronales Convolucionales

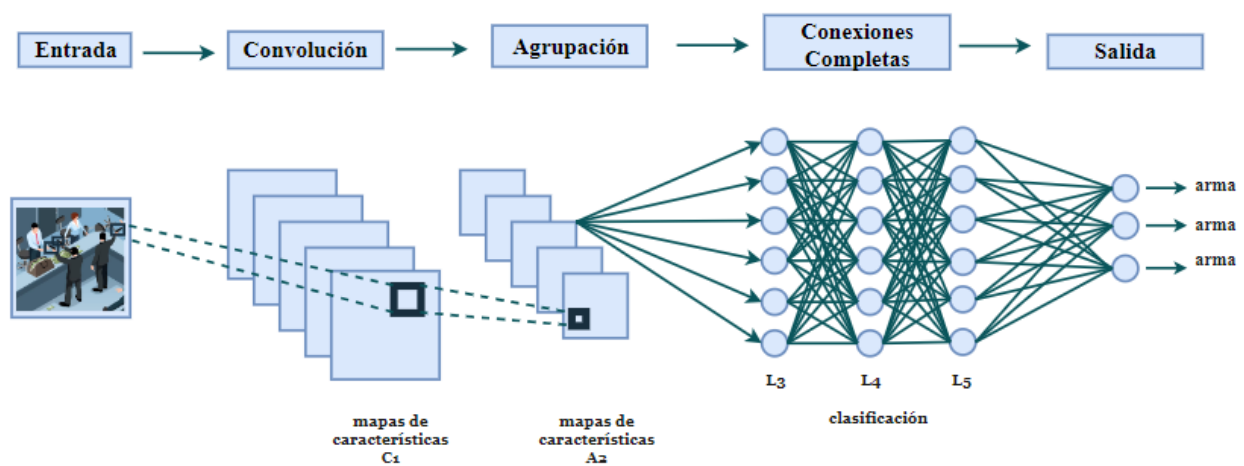
Una red neuronal convolucional o CCN, es un modelo de red neuronal que se forma de características relevantes de los datos automáticamente, en particular de imágenes mediante la mejora de los filtros, es decir, píxeles que poseen las imágenes. La eficiencia que maneja nace de la convolución dado que esto reduce la necesidad de usar conexiones finalizadas, por otro lado, se emplean tanto para el reconocimiento de imágenes como procesamiento de lenguaje natural, entre otros, que tengan relación con la visión artificial. Por ende, este tipo de red requiere menos procesamiento que otros algoritmos y se resalta su forma de aprendizaje autónomo sin la interacción humana para la extracción de características (Phung & Rhee, 2019).

El proceso de las redes neuronales convolucionales comienza con el ingreso de las imágenes, las cuales son tratadas de acuerdo con el tipo de matriz que está constituido, es decir,

si son escalas de blanco y negro posee un único canal, mientras que, las de color cuentan con tres canales (Rojo, Verde, Azul). Una vez detectada la matriz que corresponde, estas ingresan a la red, constituida por las capas de convolución, agrupación y la red neuronal con conexiones completas(Google, 2022).

Figura 1.

Diagrama Esquemático de una Red Neuronal Convolutiva.



Nota. Esta figura muestra la arquitectura de una red neuronal convolutiva, en la que consta la presencia de 5 capas, la capa de entrada en que se reciben los datos, seguida por la capa de convolución en la cual se procede a la detección de características relevantes, luego, la capa de agrupación en la que se realiza el proceso de reducir la dimensionalidad del objeto, la capa completa de conexiones en la cual se realizan las decisiones y finalmente se tiene la capa de salida en la cual se exponen los resultados obtenidos. La comunicación de estas capas permite que la red neuronal aprenda por medio de una adaptación de patrones complejos, con el fin de ser una red robusta y eficiente. Adaptado de *High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets*, por V.H Phung y E.J Rhee, 2019, Applied Sciences, Volumen 9, pagina 4500, 9(21).

Detección de Objetos

La detección se centra en prever tanto la ubicación como la clase de los objetos en una imagen. Este proceso implica la identificación de la posición mediante el uso de cajas rectangulares (bounding boxes), que requieren la predicción de variables como el nombre de la clase, las coordenadas del vértice superior izquierdo de la caja delimitadora, así como el ancho y alto de la misma (x, y). En este enfoque, el modelo debe abordar el desafío de manejar la presencia variable de objetos en la imagen, generando resultados de manera dinámica. Además, el modelo no solo debe localizar los objetos sino también clasificarlos según su tipo. Se distinguen dos tipos de detectores de objetos: los de un solo paso (one-stage), y de dos pasos (two-stage), siendo YOLO un ejemplo de un detector de un solo paso que realiza la localización y clasificación simultáneamente (Roboflow, 2022).

Circuito Cerrado de Televisión

Las cámaras de Circuito Cerrado de Televisión (CCTV) cumplen un papel esencial en abordar desafíos de seguridad, siendo un requisito crucial en este ámbito. Su despliegue en espacios públicos tiene como objetivo principal brindar seguridad, facilitar investigaciones y respaldar estrategias de detección y prevención. Las imágenes capturadas por estas cámaras son pruebas fundamentales en procedimientos legales (Bhatti et al., 2021).

Roboflow

Es una plataforma versátil que permite a los usuarios configurar, etiquetar y mejorar conjuntos de datos para visión por computadora. Proporciona herramientas para procesamiento, anotación y almacenamiento, con una interfaz de programación de aplicaciones para integración en flujos de trabajo existentes. Además, ofrece funcionalidades de visualización y análisis de datos para facilitar el proceso de mejora de modelos. Esta plataforma capacita a los desarrolladores desde la carga y organización de datos hasta el entrenamiento y despliegue

eficientes, utilizando algoritmos avanzados y técnicas de aprendizaje automático (Roboflow, 2022).

Google Colab

Es un servicio basado en Jupyter Notebook que ofrece acceso gratuito a recursos computacionales, incluyendo GPU y TPU. Este entorno en la nube es especialmente útil para disciplinas como el aprendizaje automático, la ciencia de datos y la educación, proporcionando un medio accesible para investigadores individuales interesados en tecnologías avanzadas como el aprendizaje profundo y las redes neuronales. Su funcionalidad se extiende a la ejecución y almacenamiento de código en Python, destacando la posibilidad de utilizar GPU de manera gratuita para tareas como el entrenamiento de modelos y procesos de procesamiento en la nube, sin incurrir en costos adicionales (Google Colab, 2023).

Kaggle

Kaggle emerge con una herramienta invaluable que se centra en la ciencia de datos y el aprendizaje automático. Proporcionando un entorno basado en la nube que permite la carga y ejecución de códigos sin necesidad de realizar operaciones localmente. Su interfaz intuitiva, la disponibilidad de bases de datos y bibliotecas compartidas por la comunidad, junto con la posibilidad de utilizar unidades de procesamiento gráfico (GPU), son aspectos que consolidan su destacada posición, especialmente en el ámbito del eficiente entrenamiento de redes neuronales complejas (Kaggle, 2023).

YOLOv5 (You Only Look Once)

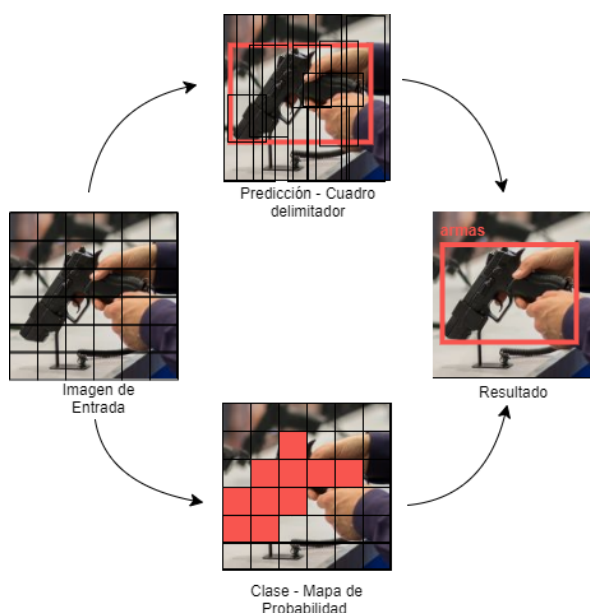
Es una perspectiva innovadora en la detección de objetos destinados al procesamiento en tiempo real (Hui, 2018), al abordarla como un problema de regresión en lugar de recurrir a enfoques convencionales de adaptación de clasificadores. Este método predice directamente las coordenadas de las cajas delimitadoras y las probabilidades asociadas a las clases mediante un solo modelo de red neuronal, permitiendo una evaluación única y una optimización integral del

rendimiento de detección. La arquitectura unificada de YOLO se destaca por su notable velocidad en el proceso de imágenes a 45 cuadros por segundo en tiempo real, mientras que su versión más pequeña, Fast YOLO, alcanza una velocidad de 155 cuadros por segundo, manteniendo el doble de mAP que otros detectores en tiempo real. A pesar de posibles errores de localización, la capacidad de YOLO para minimizar falsos positivos en el fondo y su habilidad para aprender representaciones generales de objetos lo consolidan como un líder en detección, incluso mostrando superioridad en la generalización a diversas áreas (Redmon et al., 2016).

YOLOv5 como parte de la serie YOLO para visión por computadora, representa un avance significativo en la detección de objetos desde su lanzamiento en mayo de 2020. Desarrollado por Glenn Jocher, este modelo se destaca por su cambio a Python. Con versiones desde la pequeña hasta la extragrande. Además, ofrece una detección eficiente de objetos en una sola etapa, consolidándose como una herramienta avanzada y versátil en visión por computadora y detección de objeto (Solawetz, 2020a).

Figura 2.

Sistema de Detección Yolo



Nota. Esta figura representa como Yolo lleva a cabo sus detecciones dividiendo la imagen de entrada en cuadrículas de celdas, cada una de ellas responsables de predecir N números de cuadros delimitadores o conocidos como “Bounding Boxes” para cada cuadro delimitador la red convolucional predice un nivel de probabilidad como resultado encierre un objeto y que la misma probabilidad de que el objeto encerrado sea de la clase definida en particular, en nuestro caso “armas”.

Python

Python (*Welcome to Python.Org, 2023*) es un lenguaje de programación de alto nivel y orientado a objetos. Es versátil y puede ejecutar diversos tipos de programas. Se caracteriza por ser interpretado, lo que implica que puede ejecutar el código fuente sin necesidad de ser compilado, lo que agiliza el desarrollo a costa de una potencial menor velocidad de ejecución. Python es ampliamente utilizado en el desarrollo web y de aplicaciones informáticas, destacando por su semántica dinámica integrada.

PyTorch

PyTorch (PyTorch, 2023) es una biblioteca de código abierto que utiliza Python como base, y está específicamente creada para llevar a cabo cálculos numéricos mediante el uso de tensores. Destaca por su eficiencia en el desarrollo de aplicaciones de aprendizaje profundo. Su capacidad para ejecutar cálculos en unidades de procesamiento gráfico (GPU) acelera tanto el proceso de entrenamiento como la ejecución, convirtiéndola en una opción ampliamente preferida en el ámbito del desarrollo de inteligencia artificial.

OpenCV

Es la biblioteca de visión por computadora más extensa a nivel mundial, se destaca por su naturaleza de código abierto y su extenso repertorio de más de 2500 algoritmos. Bajo la administración de la Fundación de Open Source, se utiliza principalmente para la detección de objetos y rostros, destacando en diversas áreas (OpenCV - Open Computer Vision Library, 2023).

Tkinter

Es una biblioteca del lenguaje de programación Python diseñada para facilitar la creación y desarrollo de aplicaciones de escritorio con interfaz gráfica. Funciona como la interfaz por defecto para el kit de herramientas de GUI TK y está disponible en una amplia variedad de plataformas, abarcando desde sistemas Unix hasta Windows (Tkinter, 2023).

Chime

Es una biblioteca del sonido de Python diseñada para emitir notificaciones de sonidos al terminar un proceso. No presenta dependencias adicionales. Los sonidos que se reproducen son asincrónicas y no bloqueantes. La función tarda alrededor de 2 milisegundos en ejecutarse, pese a la duración del sonido. Estos sonidos se los puede utilizar de manera libre de acuerdo a las necesidades que presente el usuario(Halford, 2021).

Estado del Arte

Este apartado tiene como propósito realizar una revisión exhaustiva de la literatura con el fin de analizar los enfoques existentes en aprendizaje profundo para la detección de objetos en tiempo real. Además, se examinarán las soluciones implementadas hasta la fecha para abordar este desafío específico. El objetivo principal de esta revisión es identificar los desafíos inherentes a la detección de los objetos y en tiempo real, con base en este análisis, proponer una nueva perspectiva para la aplicación de esta tecnología. Esto se traducirá en el desarrollo de un modelo de aprendizaje profundo con la capacidad de detectar eficazmente los diversos tipos de armas de fuego en tiempo real.

Para este estudio, los artículos se adquirieron de las bases de datos IEEE, Scopus y Google Scholar, a fin de, realizar una revisión previa a la literatura, de modo que, atribuya a la validación de la propuesta.

Este proceso está compuesto de varias etapas, así como, identificar el problema, criterios de inclusión y exclusión para obtener los aspectos que deben de tener los artículos de referencia. Tomando en cuenta lo antes mencionado, se valida la cadena de búsqueda, se realiza la selección de los artículos primarios base. Por último, se mencionan los resultados obtenidos en el proceso.

Criterios de inclusión y exclusión

Para la investigación, se han seleccionado y examinado artículos publicados a partir del año 2020, asegurando así la relevancia y actualidad de la información. Se da preferencia a aquellos que abordan la detección de objetos en tiempo real guiado en redes neuronales convolucionales, especialmente bajo la implementación de la arquitectura de YOLO. Considerando estos elementos, se implementaron los siguientes criterios de inclusión:

- Artículos que estén enfocados en la detección de objetos en tiempo real mediante herramientas, técnicas de aprendizaje e integración de un sistema de CCTV.
- Artículos que comparen metodologías y técnicas para la detección de objetos en tiempo real.
- Artículos cuyo contenido sea enfocado en la integración de estas tecnologías.

Cadena de Búsqueda

Para guiar el proceso se hizo uso de los términos más relevantes que se enlazan a conectores lógicos, así como, "OR" y "AND". Las temáticas de interés buscadas en las bases de datos se presentan en detalle a continuación:

- Técnicas para la extracción de características, etiquetado, procesamiento de las imágenes.
- Soluciones utilizando procesos de detección de objetos.

- Soluciones con algoritmos de aprendizaje profundo en combinación con sistemas de videovigilancia.

A continuación, se detalla las cadenas de búsqueda que se emplearon en las bases de datos mencionadas anteriormente.

Tabla 1.

Cadena de búsqueda en las diferentes bases de datos

Base de Datos	Cadena de Búsqueda	Número de Artículos
IEEE	("All Metadata":"vision" OR "All Metadata":"computer vision" OR "All Metadata":"image processing") AND ("All Metadata":"gun" OR "All Metadata":"firearm" OR "All Metadata":"weapons") AND ("All Metadata":"detection" OR "All Metadata":"recognition" OR "All Metadata":"classification")	346
SCOPUS	"vision" OR "computer vision" OR "image processing" OR "firearm detection" OR "algorithms you only look once" OR "object detection algorithms" AND "detection" OR "recognition" OR "classification" AND "gun" OR "firearm" OR "weapons" AND "machine learning" OR "incident prevention" AND "artificial intelligence" OR "surveillance systems" OR "image analysis" OR "video analysis" OR "integration off security"	275
GOOGLE SCHOLAR	"Roboflow" AND "vision" OR "computer vision" OR "image processing" OR "firearm detection" OR "algorithms you only look once" OR "YOLOv5" OR "object detection algorithms" AND "artificial intelligence" OR "surveillance systems" OR	395

"image analysis" OR "video analysis" OR "integration off
security cameras" AND un" OR "firearm" OR "weapons"

Nota. En esta tabla se exponen los resultados de las búsquedas obtenidas en las bases de datos, sin embargo, el número de artículos obtenidos son resultados de las temáticas en cuestión. Por ende, se debe realizar una validación de los resultados, a fin de, obtener mejores resultados que favorezca al desarrollo del proyecto.

Validación de estudios

Tras llevar a cabo la búsqueda en las bases de datos mediante la cadena de búsqueda general, se procedió a realizar la depuración de los resultados con respecto a los artículos obtenidos en la búsqueda. Este proceso se llevó a cabo mediante la aplicación de los siguientes filtros:

- Idioma: inglés – español.
- Título del artículo - 1° Filtro.
- Resumen del artículo - 2° Filtro.
- Lectura de todo el artículo - 3° Filtro.

En base a los puntos mencionados anteriormente, se excluyeron aquellos artículos que no cumplían con los criterios de inclusión y exclusión previamente establecidos. El resultado de este proceso se encuentra detallado en la Tabla.

Tabla 2.*Resultados de la cadena de búsqueda.*

Base de datos	Idioma	1° Filtro	2° Filtro	3° Filtro
IEEE	Inglés	24	8	2
SCOPUS	Inglés	6	4	2
GOOGLE	Español	10	3	2
SCHOLAR	Inglés	20	10	2

Nota. Esta tabla detalla los filtros aplicados para obtener los artículos relevantes en las distintas bases de datos.

Tras completar el proceso de selección, se obtuvo un total de 8 artículos. Con el fin de aclarar los objetivos de esta investigación y analizar la pertinencia de cada artículo seleccionado, se identificaron características distintivas de cada estudio.

Tabla 3.*Estudios primarios*

Código	Título	Cita
AP1	Weapon Detection in Real-Time CCTV Videos Using Deep Learning.	(Bhatti et al., 2021)
AP2	Weapon Identification using YOLO V5 Algorithm.	(Reddy A et al., 2023)
AP3	Detección de armas tipo pistola mediante el uso de redes convolucionales con una arquitectura tipo YOLO y estereoscopía.	(Schcolnik-Elias et al., 2023)

AP4	Deep Learning approach for event monitoring system.	(Vikas et al., 2022)
AP5	Comparison of YOLOv3, YOLOv5s and MobileNet-SSD V2 for Real-Time Mask Detection.	(Iyer et al., 2021)
AP6	Face Mask Recognition System with YOLOV5 Based on Image Recognition.	(Yang et al., 2020)
AP7	Sistema de detección de aves mediante análisis de imágenes.	(Marcela Silva-Labanda & Carlos Cobos-Torres, 2021)
AP8	State of-the-Art Analysis of Multiple Object. Detection Techniques using Deep Learning.	(Sharma et al., 2023)

Nota. Esta tabla refleja los artículos seleccionados después de un completo análisis, denominándolos Estudios primarios. Por otro lado, cada artículo cuenta con un código, código que figura a los estudios primarios (AP).

Revisión de los Estudios Primarios

AP1: Weapon Detection in Real-Time CCTV Videos Using Deep Learning.

En este estudio realizado por Bhatti et al., (2021) despliega diversas metodologías y recursos para concebir un sistema automatizado de detección de armas en tiempo real. Se enfoca en la aplicación de modelos de aprendizaje profundo, especialmente en las redes neuronales convolucionales (CNN). Mediante una estrategia de aprovechamiento de conocimientos, se integran modelos pre entrenados para optimizar los procesos. Dicho artículo

propone incluir clases de confusión y filtros OpenCV con el fin de mejorar la calidad de las imágenes en la clasificación. Los resultados obtenidos con el modelo seleccionado YOLOv4, fueron de una puntuación del 91% de efectividad. Por lo que este estudio nos ofrece técnicas que refuerzan al modelo entrenado, a fin de elevar el rendimiento y la precisión a un 100% de efectividad.

AP2: Weapon Identification using YOLO V5 Algorithm.

El estudio realizado por Reddy A et al., (2023) se enfoca en el progreso de las tecnologías de visión por computadora para la detección de armas en tiempo real, con énfasis en el uso del algoritmo YOLOv5. En lugar de depender de métodos tradicionales que requieren intervención manual o sistemas basados en reglas limitadas, la metodología propuesta utiliza el poder del aprendizaje profundo para automatizar el sistema. Se detallan técnicas que optimizan los hiperparámetros y el aumento de datos, a fin de, garantizar la robustez del modelo. Los resultados subrayan la eficacia del enfoque propuesto. Este estudio ofrece técnicas para equilibrar la velocidad y precisión del algoritmo en diversas condiciones y orientaciones.

AP3: Detección de armas tipo pistola mediante el uso de redes convolucionales con una arquitectura tipo YOLO y estereoscopía.

(Schcolnik-Elias et al., 2023) aborda el problema de inseguridad y el acceso a armas de fuego mediante un sistema de visión estereoscópica. El propósito es detectar y clasificar los objetos, como pistolas, con cierto grado de confianza, además de identificar su distancia. Empleando una arquitectura de redes neuronales convolucionales (CNN) con el algoritmo YOLO para la detección, junto con un algoritmo para la estimación estereoscópica de la distancia. La metodología se basa en aprendizaje profundo supervisado, utilizando un conjunto de datos diverso modificado para aumentar su precisión y utilidad. La investigación demuestra que el sistema, basado en YOLOv5, supera métodos tradicionales y presenta una precisión del 92.2%

en la detección, ofreciendo una solución eficiente. Dicho estudio ofrece métodos para ampliar el algoritmo de estimación de distancia para mejorar su rendimiento a distancias mayores.

AP4: Deep Learning Approach for Event Monitoring System.

Vikas et al. (2022) enfocan su investigación en reforzar la seguridad mediante un sistema de vigilancia CCVT combinado con el modelo YOLOv5 para detección de armas en tiempo real. El proceso requirió la mejora del conjunto de datos el emplear una variación de imágenes en escala de grises, versiones borrosas y aplicar el desenfoque gaussiano y bloques residuales para su procesamiento. Es así como el sistema arroja resultados prometedores, con una precisión de 93%. De ahí que el estudio sea una fuente confiable de varias técnicas para la construcción y el fino ajuste del modelo.

AP5: Comparison of YOLOv3, YOLOv5s and MobileNet-SSD V2 for Real-Time Mask Detection.

En la investigación de Iyer et al., (2021) da a conocer que se ocupa de la detección del uso de mascarillas mediante algoritmos de aprendizaje profundo, enfocándose en la comparación de YOLOv3, YOLOv5 y MobileNet-SSD V2. Enfatizando la relevancia de usar tecnologías avanzadas de visión por computadora. La metodología implica la recopilación manual de datos, el entrenamiento de modelos en una instancia de Google Cloud, y la comparación de su rendimiento en diversos dispositivos. Los resultados indican que YOLOv5 se destaca como el modelo más eficiente para el despliegue en tiempo real, mientras que YOLOv3 y MobileNet-SSD V2 también exhiben viabilidad en distintos escenarios. En conclusión, este estudio aporta una referencia técnica para la selección del modelo y la evaluación de rendimiento orientado a las diferentes condiciones específicas.

AP6: Face Mask Recognition System with YOLOV5 Based on Image Recognition.

(Yang et al., 2020) aborda la aplicación de algoritmos de aprendizaje profundo, especialmente YOLOv5, para la identificación de mascarillas en entornos públicos. La propuesta tiene como objetivo sustituir la inspección manual mediante la implementación de un método basado en aprendizaje profundo para la vigilancia del uso de mascarillas en espacios públicos. El sistema que se divide en fases diversas, abarcando desde la mejora de las imágenes de mascarillas, la segmentación de información relativa a las mismas, el reconocimiento facial y la interacción con la interfaz. Los resultados revelan una tasa de éxito del 97.9%, destacando la eficacia y viabilidad de la propuesta. El estudio brinda diversas estrategias para garantizar la solidez del modelo, así como la aplicabilidad en entornos públicos.

AP7: Sistema de detección de aves mediante análisis de imágenes.

En la investigación de Marcela Silva-Labanda & Carlos Cobos-Torres (2021), desarrollan un sistema de detección de aves mediante el uso de YOLOv5, aplicándolo en la Reserva Ecológica de Arenillas. Se emplean 2712 imágenes etiquetadas con Roboflow, y el modelo alcanza una precisión del 80% en un entorno natural. La metodología incluye la adquisición de imágenes, procesamiento, entrenamiento y pruebas. Se destaca un 88% de precisión en el entrenamiento y un promedio del 85% haciendo uso de una cámara Logitech C92 en la detección de aves en la reserva. Este estudio ofrece referencia en la técnica e implementación para el análisis de las imágenes, además de un marco sólido en pruebas en espacios abiertos.

AP8: State of-the-Art Analysis of Multiple Object Detection Techniques using Deep Learning.

Este estudio realizado por Sharma et al., (2023) se adentra en el mundo de la detección de objetos, un tema vital en el análisis de videos e interpretación de imágenes. Destacando el avance que Deep Learning (DL) ha aportado, superando las limitaciones de los enfoques

tradicionales. La investigación se centra en examinar los diseños comunes y las modificaciones que impulsan el rendimiento en la detección de objetos, destacando el modelo YOLOv5. La metodología aplica un riguroso proceso de preprocesamiento de datos mediante la plataforma Roboflow y estrategias de aumento de datos. Se profundiza en los elementos clave de YOLOv5, abordando el modelo Backbone, Neck y Head. Además, este estudio apoya en tácticas de mejora de los datos, resaltando la eficiencia y ligereza del algoritmo YOLOv5 en comparación con otros modelos contemporáneos, por medio de herramientas Open Source que facilitan la eficiente implementación.

Respuesta a las Preguntas de Investigación

RQ1: ¿Cómo se evalúa la eficiencia y precisión de los modelos para la detección en tiempo real, y qué métricas específicas se emplean para medir su rendimiento en comparación con otros enfoques?

La evaluación de la eficiencia y precisión propuesta en los estudios recopilados en los modelos de detección en tiempo real recalcan una etapa decisiva en el proceso de desarrollo de proyectos con visión artificial.

Las métricas permiten la comprensión profunda y comparativa de su rendimiento, además, de una orientación en la elección y mejora de la presión al llevarlo a prueba.

Este tipo de evaluación involucra la inferencia en el grupo de datos de prueba, comparando las predicciones con las etiquetas de referencia para calcular métricas clave como la precisión y recall. Al mismo tiempo, se anota el tiempo de inferencia. Estos resultados permiten el ajuste y optimización del modelo, asegurando su eficacia y permitiendo comparaciones con otros enfoques.

RQ2: ¿Cómo han demostrado ser relevantes las tecnologías Roboflow y YOLOv5 en las investigaciones recopiladas?

La relevancia de estas tecnologías en los estudios compilados se demuestra en su versatilidad y éxito en los diferentes escenarios a los que fueron sometidos. Donde la adaptabilidad y personalización son clave para la efectividad de estas tecnologías, como, por ejemplo, los entornos universitarios, siempre y cuando que se realicen las adecuadas configuraciones y se aborden las debidas especificaciones, asociadas con la privacidad, integración y ética de acuerdo con las necesidades que se desea implementar de acuerdo con cada proyecto.

Además, la compatibilidad de ambas tecnologías, han demostrado ser esenciales al afrontar desafíos críticos en las investigaciones y desarrollo de soluciones innovadoras. Su capacidad de integración no solo eleva el nivel de eficiencia en los procesos, sino que, incrementa las posibilidades de consolidar la optimización de los resultados.

RQ3: ¿Cómo influye el manejo y gestión de los datos en el proceso de aprendizaje y desarrollo de la propuesta?

Las investigaciones recopiladas se enfocan en el desempeño que cumplen el proceso de aprendizaje, destacando la organización, accesibilidad y calidad de los datos quienes impactan en el desarrollo del modelo. Por ende, la adecuada gestión permite ampliar el proceso de entrenamiento y la capacidad de validar condiciones del mundo real.

También cabe mencionar, que la buena gestión permite identificar y ajustar, a fin de, contrarrestar los sesgos. A su vez, contribuye en reducir el tiempo y recursos en la fase de entrenamiento.

Por lo tanto, se puede decir que el manejo y la gestión del grupo de datos es elemental para obtener el éxito en el proceso de aprendizaje profundo, dado que, influye no solo en la calidad del rendimiento, sino que, en la capacidad de adaptación del modelo propuesto.

Conclusión del Estado del Arte

Los estudios examinados abarcan técnicas de interés y relevancia, que aportan con los objetivos expuestos para el desarrollo del sistema planteado. En particular, se centran en la segmentación de las imágenes, de igual forma, en los aspectos que colaboran con el proceso de etiquetado, tal como, el tiempo, la precisión, la posición del objeto, el procesamiento y entrenamiento de los datos. De manera que estos enfoques logren perfeccionar el modelo para alcanzar el resultado deseado.

En esa misma línea, el empleo de herramientas que son pieza clave para el desarrollo del modelo, entre las cuales se destacan Roboflow y SelectiveSearch, ambas demostrando eficiencia en su aplicación, en virtud de los resultados obtenidos en la preparación y presión del modelo, convirtiéndolo en confiable.

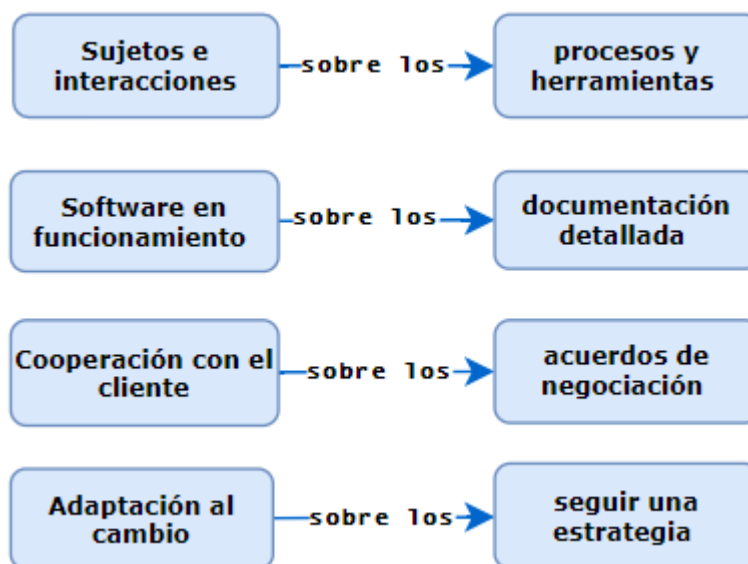
Por último, la combinación de estas tecnologías en los diferentes escenarios, brindan buscar soluciones a las problemáticas sociales, con especial énfasis en los temas de seguridad. Con ello los sistemas de videovigilancia buscan ir de la mano con temas relacionados al campo de la visión artificial y del aprendizaje profundo. Elevando la precisión de detectar objetos sospechosos.

Metodología

Para llevar a cabo el desarrollo metodológico de este trabajo, hemos optado por integrar la metodología ágil y Scrum como marcos fundamentales. Siguiendo las directrices delineadas por Beck y otros autores(Beck et al., 2001) en el "Manifiesto por el Desarrollo Ágil de Software", es crucial subrayar que esta metodología se centra especialmente en los siguientes elementos:

Figura 3.

Pasos Manifiesto de Desarrollo Ágil de Software



Nota. Es fundamental destacar que cada elemento se desglosa en dos valores, los cuales reflejan la adaptabilidad al cambio. Asimismo, es relevante señalar que el valor situado a la izquierda evidencia un enfoque más destacado hacia nuestro proyecto.

Metodología Scrum

La elección de Scrum como nuestro marco de trabajo no fue una decisión casual, sino una cuidadosa selección. Este método es reconocido por su enfoque colaborativo y su habilidad para adaptarse a medida que avanza el desarrollo, estructurando las tareas en sprints: ciclos breves donde se planifican, implementan y entregan incrementos de funcionalidad de manera eficiente. La singularidad de Scrum trasciende su organización temporal; su habilidad para mejorar la comunicación y agilizar la toma de decisiones es evidente. Al mismo tiempo, inyecta en el equipo un dinamismo que fomenta la colaboración, impulsando la innovación y el rendimiento global. La efectividad comprobada de Scrum se manifiesta en varias implementaciones exitosas, resaltando su capacidad para fortalecer la flexibilidad y adaptarse ágilmente a cambios en los requisitos del proyecto (Torrás, 2015).

Técnica de Recolección de Datos

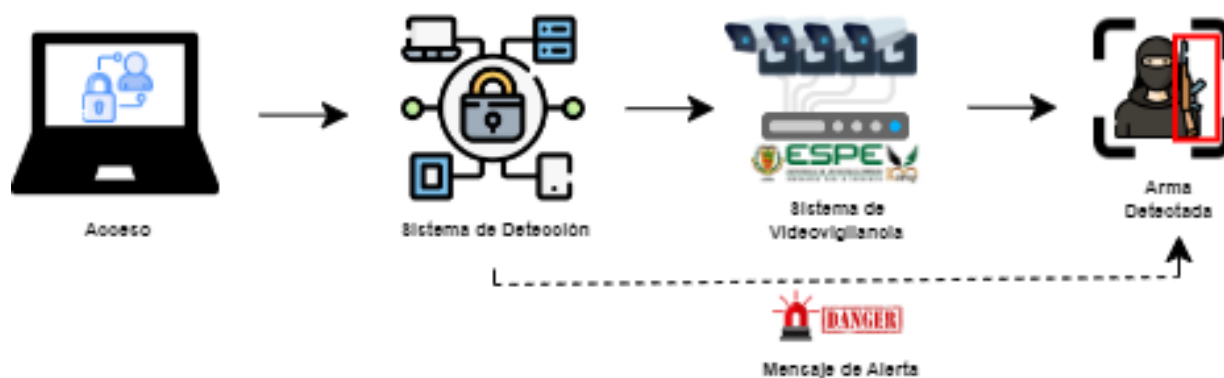
Para iniciar, nos embarcamos en la obtención de información necesaria para la construcción del modelo de detección. Este proceso implica la recopilación de imágenes provenientes de repositorios públicos. Utilizaremos herramientas en línea, como Google y Kaggle, para facilitar la creación de un dataset robusto y ampliamente representativo. La elección y calidad del dataset desempeñan un papel crucial en la capacidad predictiva y la generalización del modelo resultante, este modelo será implementado en las cámaras de seguridad del campus matriz de la Universidad de las Fuerzas Armadas ESPE, con el propósito de llevar a cabo la detección de objetos de manera efectiva.

Capítulo III: Desarrollo

En esta sección se presenta el proceso de desarrollo del sistema de alerta temprana para la detección de armas de fuego. Describiendo la serie de pasos que se aplicaron y los resultados obtenidos durante el progreso del sistema, ofreciendo una percepción total de su evolución y funcionalidad. Como se muestra en la figura adjunta.

Figura 4

Estructura del Sistema de Detección y Alerta.



Nota. En esta figura se observa la estructura del proceso de desarrollo del sistema.

Para un desarrollo adecuado, se ha dividido el proyecto en dos etapas: i) Construcción del modelo; ii) Integración en un aplicativo de escritorio. En la primera etapa se enfoca en la construcción del modelo de aprendizaje profundo para identificar las armas con una precisión superior al 90%. En la segunda, se dedica a la elaboración de una interfaz web y la integración del modelo previamente entrenado, adicional se incluye una alerta garantizando que el sistema sea efectivo en su operación. La combinación de estas dos fases logra el despliegue completo y eficiente del sistema de detección de armas, asegurando su óptimo funcionamiento.

Construcción del Modelo de Red Convolutiva

En esta primera fase se presenta el procedimiento que se llevó a cabo para la construcción del modelo de Deep Learning, modelo que es pieza clave en el desarrollo del sistema de detección de armas dentro del campus de la Universidad de las Fuerzas Armadas ESPE. La metodología, herramientas y ajustes usados permiten proporcionar la comprensión del proceso de creación y relevancia del modelo.

Arquitectura del Modelo de Red Convolutiva

La arquitectura adopta una estructura organizada en fases claramente definidas, como se puede apreciar en la figura abajo descrita. Este enfoque permite una progresión sistemática y ordenada, facilitando una comprensión exhaustiva de cada etapa del proceso.

Figura 5

Arquitectura del Modelo de Red Convolutiva



Nota. Esta figura muestra la arquitectura de alto nivel y las etapas de solución.

Etapas de la Arquitectura del Modelo

Origen de los Datos

En esta fase, se lleva a cabo la recopilación y organización de las imágenes de armas de fuego utilizando repositorios y recursos públicos, con el fin de construir un dataset diverso y representativo para el conjunto de datos a utilizar. Este proceso se detalla a continuación:

- Se procede a la búsqueda y descarga de 1084 imágenes desde la web.
- Se recopilan 1500 imágenes del repositorio de kaggle (Overgaard, 2022).
- Se obtienen 1500 imágenes del repositorio de kaggle (Reda, 2019)

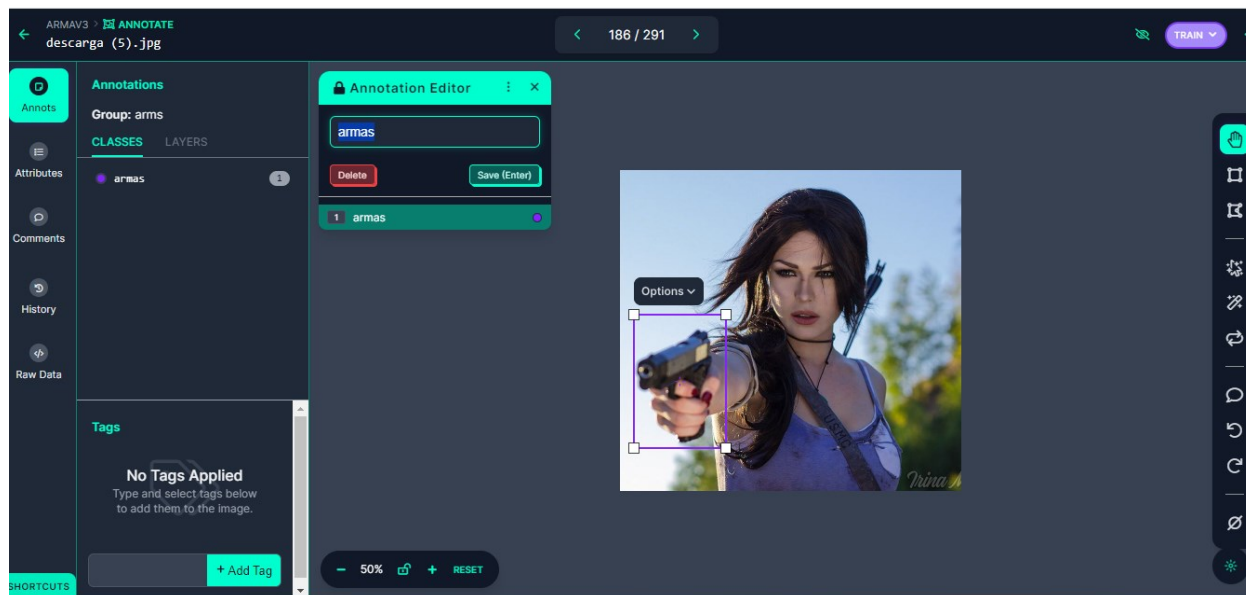
Se recopilaron 4084 imágenes en total, todas esta con una excelente calidad.

Etiquetado

Durante esta etapa se lleva a cabo el etiquetado de las imágenes del dataset utilizando la herramienta de Roboflow. Donde se otorga una única clase denominada “armas” y la etiquetación se realiza por medio de un cuadro delimitador de color morado, según se muestra en la siguiente Figura. Este proceso es significativo para el desarrollo del modelo, puesto que, se asignan y definen los identificadores al objeto relevante en una imagen, facilitando información clave para el entrenamiento del modelo y el reconocimiento del objeto en cuestión.

Figura 6.

Etiquetado de la Clase



Nota. En esta figura se muestra como es la forma de etiquetar la imagen, a su vez se observa cual es el nombre de la etiqueta asignada, en este caso “arma”.

Aumento de datos

Cabe mencionar que, para realizar esta técnica, hacemos uso de una de las características que posee Roboflow, que es el aumento de imágenes, con el fin de mejorar el rendimiento del modelo.

Para ello se realizó el uso del aumento de rotaciones¹ a 90 grados, sentido horario y antihorario, además del volteo horizontal. Con el propósito de, enriquecer el dataset, de tal forma que se genere una gran variedad de imágenes a partir de las imágenes ya existentes.

Así como, se muestran en las siguientes figuras.

Figura 7.**Agregación de Rotación en Imágenes de Armas de Fuego**

Nota. En esta figura se observa la imagen etiquetada con la argumentación de rotación de 90° y voltear. En la rotación de 90ª se visualiza la rotación horaria y antihoraria en la parte inferior.

Tras completar el proceso de la generación del dataset, se debe mencionar que al principio se hizo la carga de 4084 imágenes. Por otra parte, con la aplicación de las técnicas de aumento, como la rotación de 90° y el volteo, se logró ampliar el dataset considerablemente, alcanzando un total de 9212 imágenes. A fin de, contribuir en la variedad y complejidad del conjunto de datos.

División del Dataset

Para comprobar la validez del modelo de aprendizaje, se debe dividir el conjunto de datos en entrenamiento y pruebas.

El conjunto de datos para el entrenamiento, es el parte que almacena los datos para entrenar el modelo; mientras que, validación es usada para comprobar el buen funcionamiento del modelo entrenado, además de ajustar los parámetros, y el conjunto de pruebas se usa con el fin de evaluar el rendimiento del modelo de aprendizaje (Solawetz, 2020b).

Por lo tanto, se procede a la división de los datos, designando el 80% al entrenamiento, para Validación y Pruebas un 10% respectivamente. Además de mencionar que, los datos se mezclan en el proceso y estos se distribuyen a los conjuntos de datos mencionados anteriormente, generando de esta forma la confiabilidad en el rendimiento (Solawetz, 2020b).

Entrenamiento del modelo

Para el proceso de entrenamiento, se accede al servicio de Google Colab, donde el usuario debe autenticarse a través de una cuenta de Google. En el cuaderno de trabajo, se incorpora el conjunto de datos YOLOv5 exportado de la herramienta Roboflow, junto con otros parámetros de entrada necesarios para el entrenamiento del modelo. La siguiente figura ilustra la infraestructura de software que proporciona el servicio de entrenamiento.

Figura 8.

Diagrama de Infraestructura de Entrenamiento del Modelo



Nota. En la figura podemos observar cómo se lleva el proceso de entrenamiento del modelo. Aplicando la herramienta de Google Colab y como se combinan con el algoritmo de YOLO por medio del lenguaje de Python.

Para empezar el proceso de entrenamiento, se procede a configurar el cuaderno de trabajo en Colab. En la primera línea, se procede a importar el algoritmo de YOLOv5, abriendo paso al procedimiento de detección de los objetos y la preparación del modelo para el entrenamiento.

Figura 9.

Importación de Librerías YOLOv5 y sus requerimientos



```
#clone YOLOv5 and
!git clone https://github.com/ultralytics/yolov5 #clone repo
%cd yolov5
%pip install -qr requirements.txt #install dependencies
%pip install -q roboflow

import torch
import os
from IPython.display import Image, clear_output # to display images

print(f"Setup complete. Using torch {torch.__version__}({torch.cuda.get_device_properties(0).name if torch.cuda.is_available() else 'CPU'})")

Cloning into 'yolov5'...
```

Nota. El código detalla el uso del algoritmo de YOLOv5.

A continuación, una de las características que cuenta la herramienta de Roboflow, consiste en facilitar un código o clave de acceso que permite acceder a nuestro espacio de trabajo nombrado "ArmasV3", este punto es clave para el siguiente paso, que es el entrenamiento del modelo.

--data	Define la ubicación del archivo de configuración YAML que {dataset.location}/data.yaml : comprende de información sobre el conjunto de datos y rutas.
--weight:	Hace uso del archivo de pesos preentrenados.
--cache:	Dispone del almacenamiento en caché para agilizar el entrenamiento.
--name:	Se agrega un nombre para el archivo, donde se almacenan los resultados del entrenamiento.

Nota. En esta tabla se detallan los parametros empleados, el entranamiento se configuro para que reaalice 100 épocas, y con un tamaño de lote de 16, lo que significa que sea efectiva la deteccion de los objetos.

Resultados del Entrenamiento

Una vez finalizado con todo el proceso del entrenamiento del modelo de Deep Learning descrito en el apartado anterior, seguido procedemos, a evaluar el desempeño del modelo.

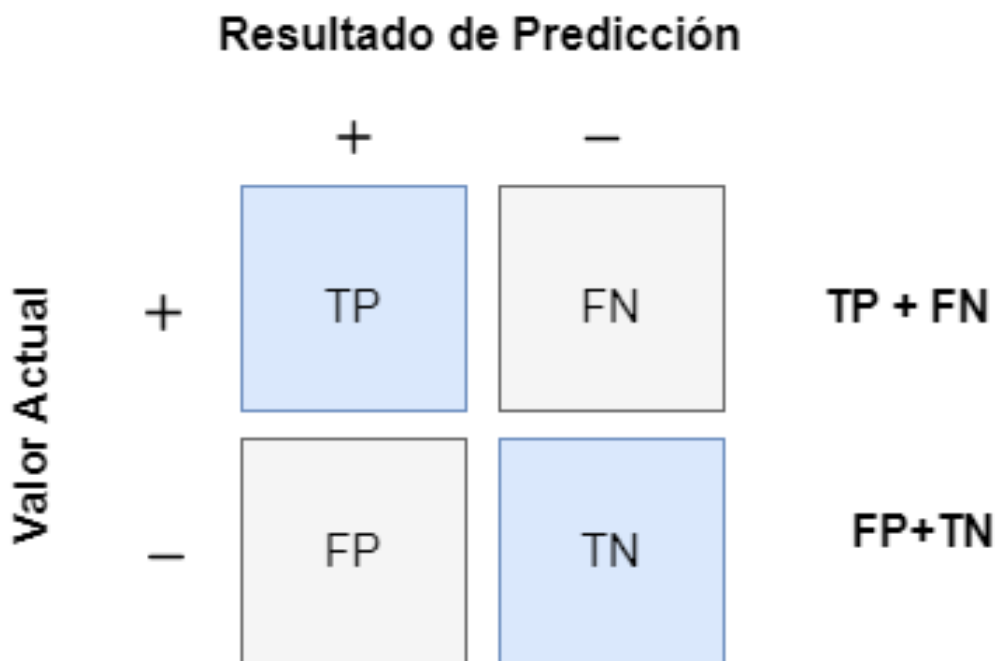
La utilización de las métricas para la evaluación del rendimiento de un modelo de aprendizaje automático es sustancial, puesto que, facilita la mejora constante (Avramidis, 2013)

Métricas de evaluación

El evaluar un modelo de redes neuronales convolucionales es necesario aplicar parámetros que permitan realizar el análisis, tanto de la eficacia y el rendimiento del modelo desarrollado (Pykes, 2021).

Figura 12.

Métricas de Evaluación de Matriz de Confusión.



Nota. En la figura se muestra cómo se evalúa el modelo en base a la matriz de confusión con el valor actual y resultado obtenido de la precisión, cada uno de ellos posee un valor positivo y negativo, su combinación nos da los resultados de la eficacia. Tomado de *The Most Common Evaluation Metrics In NLP por Kurtis Pykes de Towards Data Science*.

A continuación, se describe las métricas aplicadas en la evaluación:

- **Métrica de Sensibilidad o Recall:** Esta métrica brinda información sobre la capacidad del modelo en identificar correctamente los casos positivos. Se calcula en base al número real de positivos que ha clasificado el modelo en función a la suma de los valores positivos y falsos negativos. Un valor alto evidencia que el modelo es óptimo en la identificación de positivos, minimizando los caos perdidos. Mientras que, si el valor es bajo el modelo presenta problemas, dado que, deja pasar los casos positivos (Pykes, 2021)

$$R = \frac{TP}{TP + FN}$$

- **Métrica de precisión:** Evalúa la exactitud de las predicciones positivas efectuadas por el modelo. Siendo este el número de verdaderos positivos en función del número de los valores predichos. Cabe mencionar que, si la precisión es alta el modelo realiza escasa predicciones positivas incorrectas (Pykes, 2021).

$$P = \frac{TP}{TP + FP}$$

- **Métrica puntuación F1:** Esta métrica es una medida para la eficacia del sistema de predicción. Se combinan las métricas de precisión y sensibilidad, a fin de lograr una significativa predicción y capturar un amplio número de casos positivos. La puntuación es de 0 y 1. Donde 0 es bajo y 1 alto (Pykes, 2021).

$$F1 = 2 * \frac{P * R}{P + R}$$

- **Curva PR:** Permite distinguir la habilidad de las clases al medir las predicciones positivas sobre las negativas a través de umbrales de confianza. Al detallar el rendimiento del modelo mientras la clase positiva es menos común. (Pykes, 2021)

Para cumplir con la evaluación del modelo de redes neuronales convolucionales, se dividió la colección de datos en dos grupos: entrenamiento y pruebas. Posteriormente, se entrena el modelo y se analizan los datos prueba. Con la aplicación de las métricas se busca valorar la suficiencia del modelo y determinar los contrapesos en la inestabilidad de la clasificación de la clase.

Análisis de resultados

Precisión: La precisión del modelo entrenado se ajusta a un fragmento apartado exclusivo para evaluar los ejemplos con exactitud al reconocer los objetos expuestos.

Figura 13.

Generación de Precisión del Modelo.

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
0/149	1.52G	0.06931	0.02106	0	42	416: 100% 500/500 [01:24<00:00, 5.93it/s]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% 26/26 [00:08<00:00, 3.04it/s]
	all	810	886	0.801	0.738	0.807 0.518
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
1/149	1.67G	0.04542	0.01762	0	33	416: 100% 500/500 [01:18<00:00, 6.41it/s]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% 26/26 [00:06<00:00, 4.08it/s]
	all	810	886	0.842	0.805	0.872 0.604
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
2/149	1.67G	0.04354	0.01717	0	38	416: 100% 500/500 [01:16<00:00, 6.52it/s]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% 26/26 [00:06<00:00, 3.79it/s]
	all	810	886	0.84	0.559	0.675 0.428
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
3/149	1.67G	0.04218	0.01719	0	32	416: 100% 500/500 [01:15<00:00, 6.58it/s]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% 26/26 [00:06<00:00, 4.14it/s]
	all	810	886	0.859	0.704	0.81 0.568
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
4/149	1.67G	0.04016	0.01657	0	40	416: 100% 500/500 [01:15<00:00, 6.62it/s]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% 26/26 [00:06<00:00, 3.81it/s]
	all	810	886	0.811	0.767	0.835 0.577
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
5/149	1.67G	0.0382	0.016	0	32	416: 100% 500/500 [01:13<00:00, 6.80it/s]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% 26/26 [00:06<00:00, 3.85it/s]
	all	810	886	0.864	0.778	0.862 0.605

Nota. En esta figura se observa como el modelo alcanza un nivel de precisión, demostrando que el entrenamiento del modelo ha sido capaz de realizar las predicciones o clasificaciones con un alto grado de exactitud y confiabilidad, respaldando la eficacia en su tarea específica.

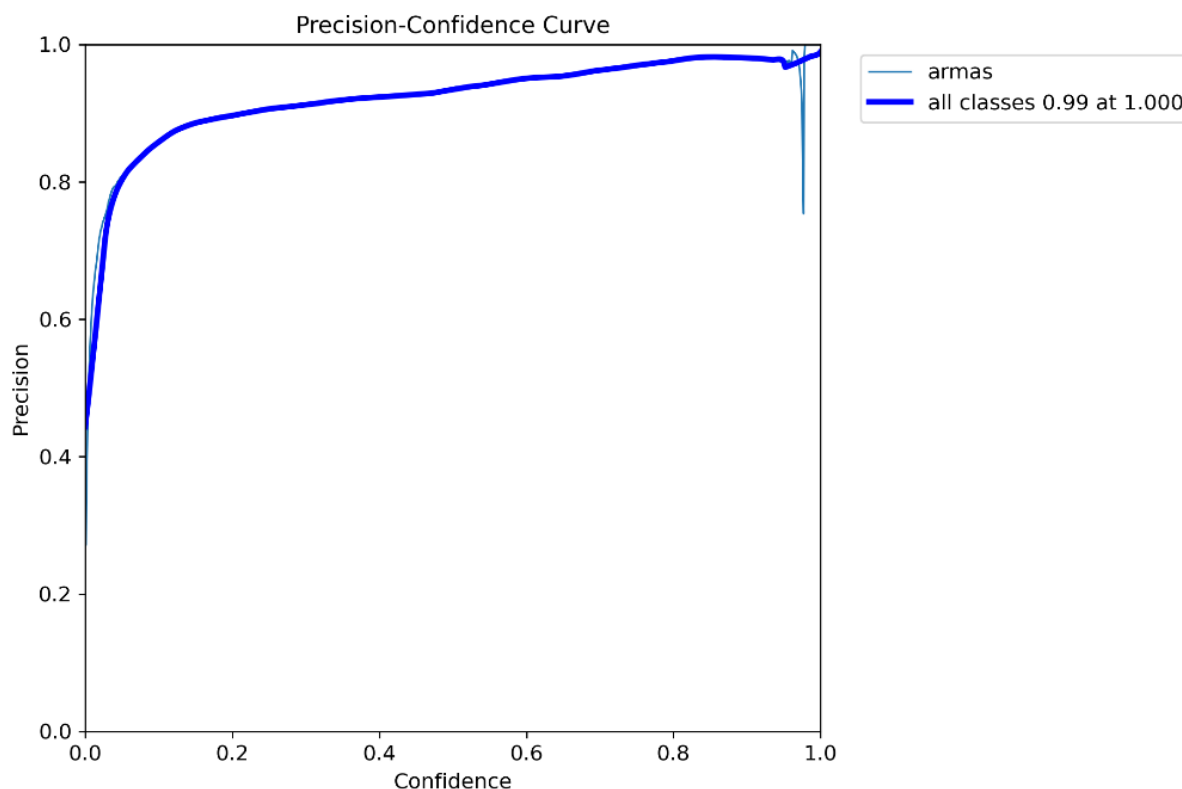
Cada epoch o época simboliza un ciclo completo de entrenamiento, en donde se ajusta sus pesos y los patrones aprenden de los datos de entrada.

Como resultado, se observa la gráfica donde se señala que a partir de la 10va época el algoritmo logra un entrenamiento significativo, cabe mencionar que, el número de épocas empleadas es de 150, alcanzando el 99% de efectividad. Siendo este un punto clave, dado que,

a partir de este resultado, se evidencia la capacidad del modelo para aprender, y de tal forma seguir perfeccionando el modelo hasta alcanzar un 100% de efectividad.

Figura 14.

Evaluación de Precisión del Modelo de Detección de Objetos.



Nota. Esta figura indica que durante el entrenamiento como la curva muestra la precisión del modelo, cambia a medida que se ajusta al umbral de confianza. El incremento significativo de la curva a medida que va aumentando la confianza, de tal forma que la precisión de sus predicciones es notable.

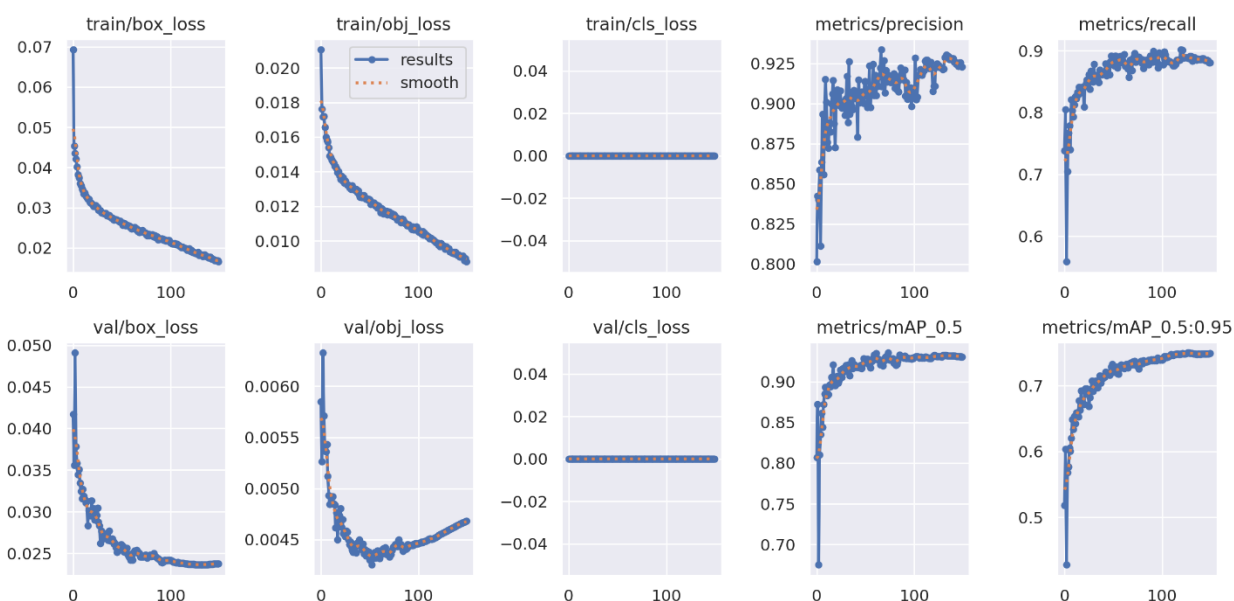
Dicho de otro modo, más claro, el modelo elaborado ha sido lo suficiente capaz de alcanzar un 99% de precisión en la captura las muestras de armas de fuego que fueron generadas del grupo de entrenamiento. Demostrando que la predicción es sumamente claro y efectivo.

Por otro lado, tenemos las pérdidas en la predicción del desajuste² de las anotaciones reales. Es decir, son los errores en la capacidad de predecir las clases específicas del modelo en los datos reales previos.

Entre tanto, se observa sucesivamente los resultados obtenidos, en el cual se resumen las pérdidas y métricas durante la etapa de ajuste. Para ello se detalla, a continuación.

Figura 15.

Resultados Entrenamiento – Ajuste Fino

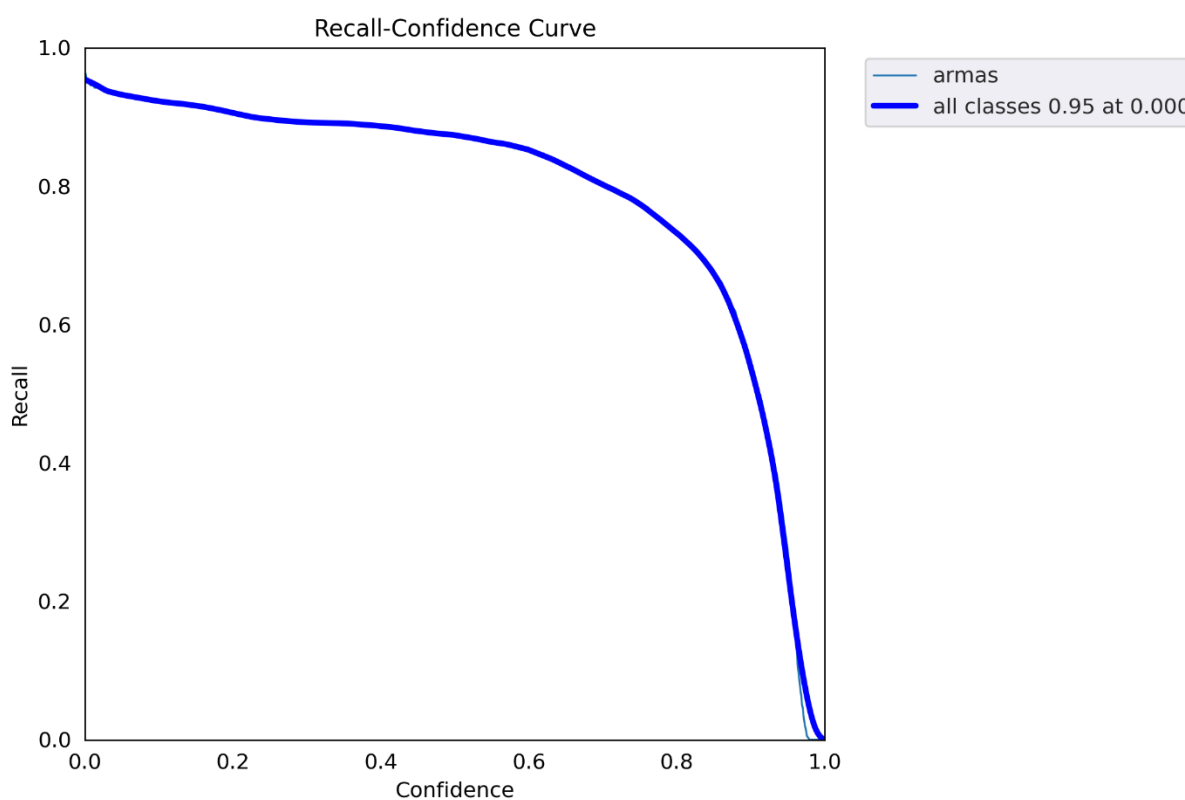


Nota. En esta figura se indica el ajuste fino³ obtenido, las pérdidas generadas indican que el modelo ajusta los cuadros de predicción, a fin de predecir y clasificar los objetos que han sido entrenados. Además, se evalúan las pérdidas para garantizar los umbrales de confianza, siendo esto crucial para entender y mejorar el rendimiento del modelo.

Recall: Tiene la capacidad de identificar los elementos positivos de la clase del modelo a evaluar. El valor de esta métrica permite tener una visión completa del rendimiento del modelo en cuanto a la capacidad que tiene de capturar los elementos destacados de la categoría determinada.

Figura 16.

Evaluación de Sensibilidad del Modelo de Detección de Objeto

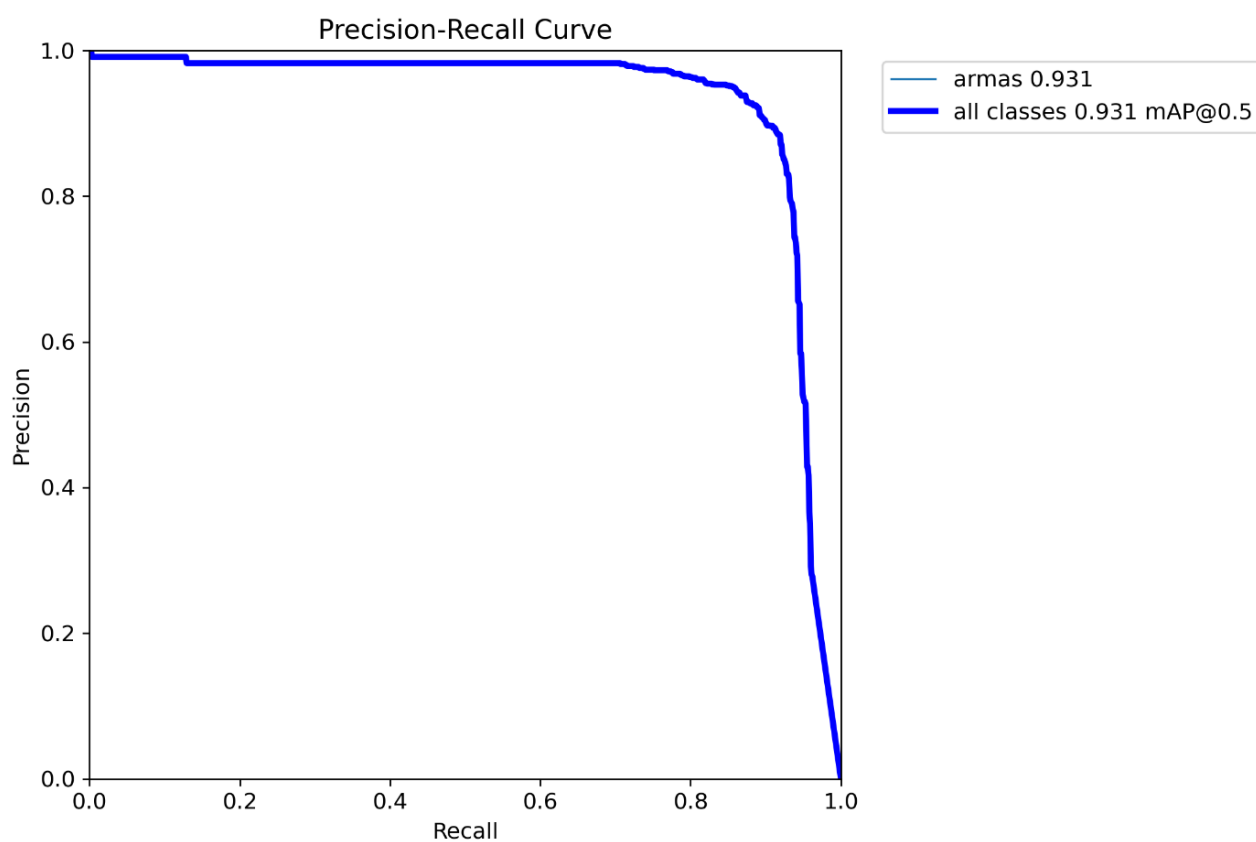


Nota. En esta figura se observa el resultado de la métrica de recall o sensibilidad nos dio una captura de 0.95 de las instancias positivas, es decir, que nuestro modelo tiene un excelente desempeño para identificar los resultados.

Curva PR: Se observa como varia la precisión en función con la sensibilidad de los diferentes puntos de decisión del modelo con respecto al rendimiento. Es decir, que esta curva detalla el comportamiento del modelo con los diferentes umbrales de decisión, resultando útil en las situaciones de inestabilidad en la clasificación.

Figura 17.

Evaluación de la Sensibilidad de Precisión del Modelo

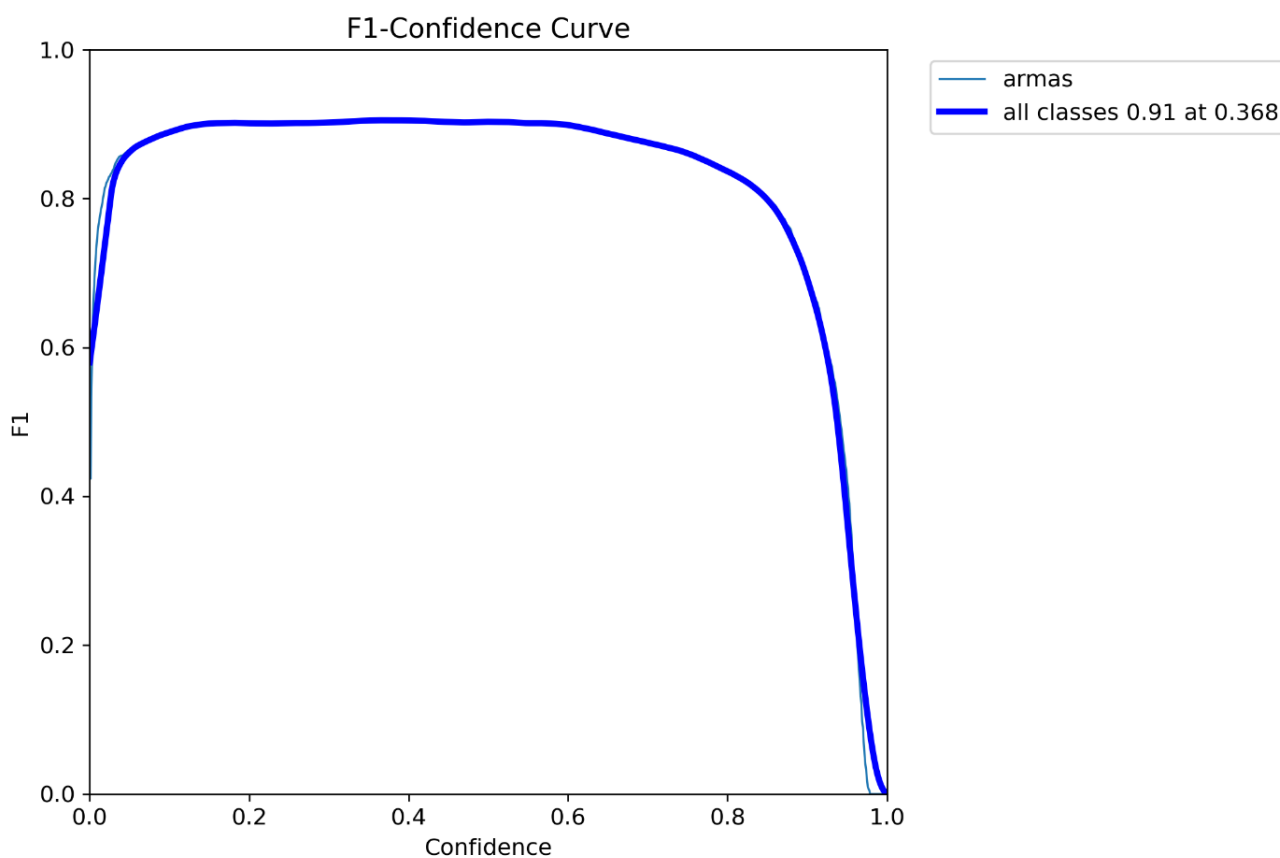


Nota. En la figura se muestra la calidad del rendimiento en la detección de objetos mAP es de 0.931 con un umbral de confianza de 0.5, el cual, determina el seguro de efectividad del resultado del modelo.

Puntuación F1: Se convierte en una solución que nos facilita el rendimiento de las categorías descritas, dado que esta métrica es útil para buscar el equilibrio entre la precisión y sensibilidad donde se halla una inestabilidad entre las clases.

Figura 18.

Valor de confianza óptimo del Modelo de Detección de Objetos.

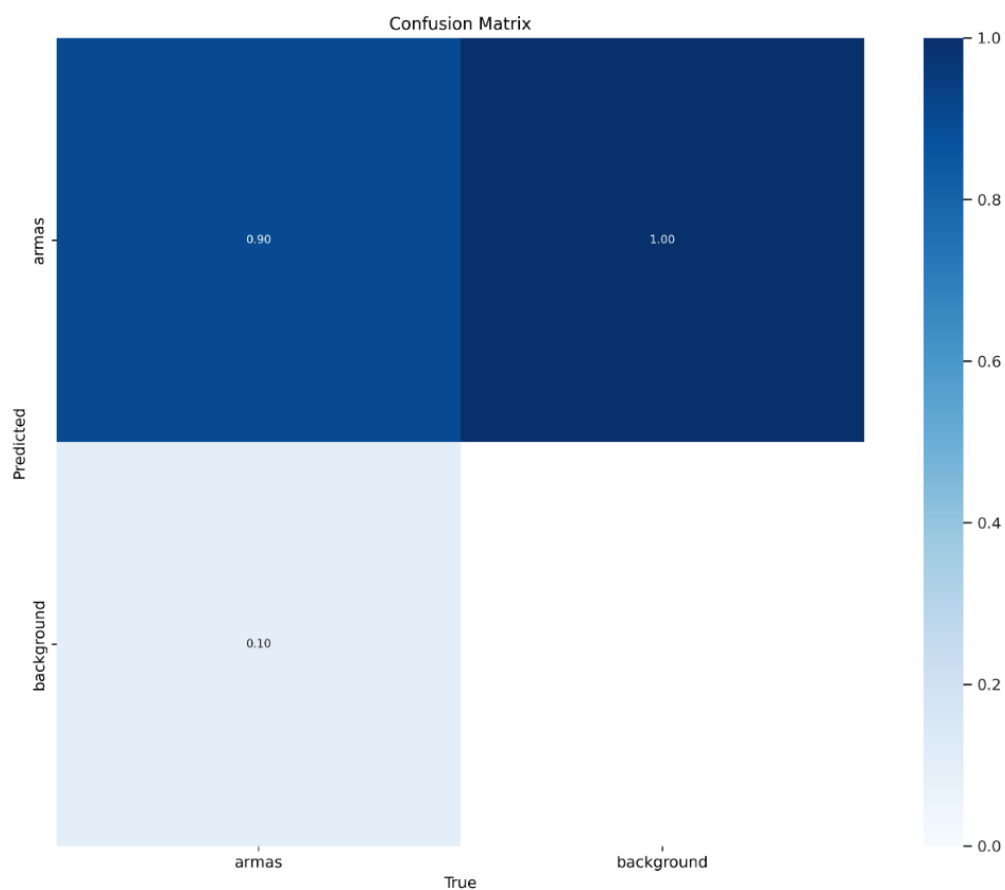


Nota. La puntuación F1 de nuestro modelo es del 0.91, es decir, ha logrado un buen equilibrio entre la precisión y la sensibilidad en sus predicciones. Lo que significa que nuestro modelo ha minimizado tanto los falsos positivos como los falsos negativos de manera efectiva.

Matriz de confusión: Presenta una detallada descripción de los aciertos y error del modelo, de acuerdo con la clase. Reflejando la cantidad de los elementos clasificados en la combinación entre la clase real y las predichas. Esta herramienta es muy practica al momento de detallar el rendimiento del modelo en cada clase, de tal modo, que se facilite a profundidad su capacidad de predicción.

Figura 19.

Matriz de Confusión del Modelo de Detección de Objetos.



Nota. En esta figura se detallan los valores generados en la matriz de confusión. Especificando los resultados de aciertos y error en cada clase de modelo.

La representación de esta matriz es clave en el análisis del rendimiento del modelo, como se observa, la precisión de nuestro modelo contiene la siguiente predicción:

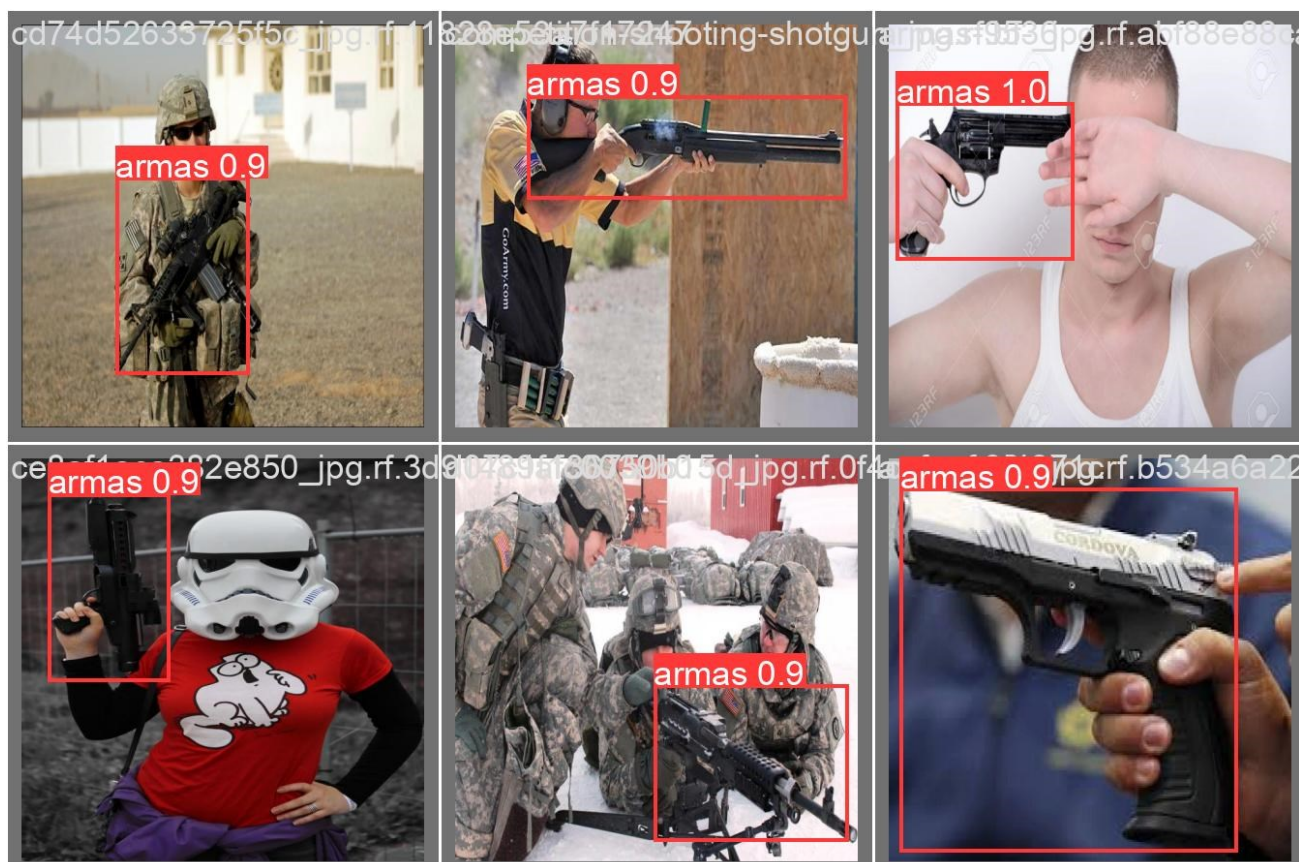
- **TP:** Verdadero Positivo con 0.90 de casos.
- **FP:** Falso Positivo con 0.10 de casos.
- **FN:** Falso negativo con 1.00 de casos.
- **TN:** Verdadero Negativo con 0 de casos.

De acuerdo con los resultados obtenidos podemos decir que nuestro modelo tiene un alto nivel de porcentaje de los verdaderos positivos (TP), lo cual es realmente positivo. Sin embargo, al obtener un 10% de falsos positivos y falsos negativos, el modelo puede llegar a obtener una precisión más perfecta.

Posteriormente, se observa un conjunto de imágenes de muestra con su respectiva etiqueta y su cuadro delimitador. Se visualiza como fue el etiquetado de las armas de fuego y como se ha determinado las regiones de valor, lo que nos lleva a deducir como se entrenó el modelo, reflejando en el modelo un efectivo rendimiento y la capacidad de obtener predicciones precisas.

Figura 20.

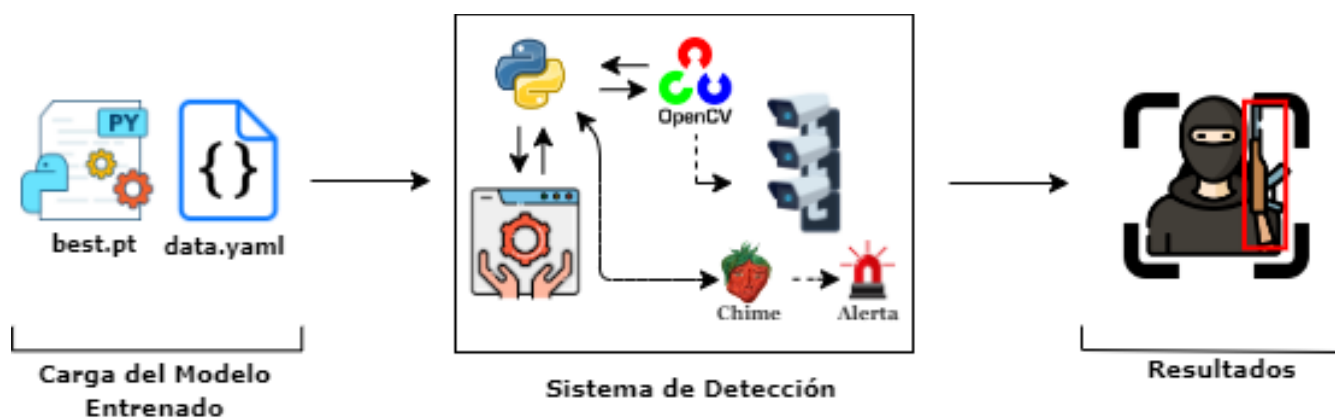
Resultados del modelo entrenado para identificar armas de fuego en Imágenes.



Nota. En esta figura se presenta los resultados obtenidos mediante el modelo entrenado, donde se destaca un nivel de predicción óptimo.

Implementación del Modelo Entrenado

El diseño del software de nuestro sistema se realizó bajo el criterio de separar funciones, de tal forma que se pueda construir el sistema sin ningún contratiempo. Esta estructura permite el desarrollo, mantenimiento y la escalabilidad del sistema, de tal forma que la gestión de cada componente sea eficiente. Por lo que, a continuación, se muestra el diagrama del modelo de entrenamiento a realizar.

Figura 21.*Diagrama de Implementación del Modelo Entrenado*

Nota. Esta Figura detalla el diagrama de implementación del modelo entrenado diseñado en tres componentes.

El primer componente implica la carga del modelo entrenado previamente, el cual se realizó en la fase inicial del desarrollo. Por otro lado, se procede al desarrollo del sistema de detección, el cual se combina con el modelo elaborado y el sistema de videovigilancia. La finalización de este proceso resultara con la obtención de resultados y validación del sistema.

Esto facilita un desarrollo flexible y adaptable a cambios, sino que también admite la retroalimentación durante el proceso, además permite ajustar de forma precisa las demandas requeridas en el proyecto.

Configuración de Librerías y Rutas:

Para empezar, se importan 7 librerías esenciales para la construcción de la aplicación. Entre estas librerías se incluyen `tkinter`, `threading`, `torch`, `cv2`, `numpy`, `chime` y `pathlib`. Cada una de estas librerías cumplen una función en particular, como, el manejo múltiple de los hilos de ejecución, la carga del modelo YOLOv5, el procesamiento de las imágenes y la reproducción de

sonidos. La compatibilidad y facilidad de integración con el entorno de desarrollo asegura el flujo de trabajo eficiente, de mismo modo, al evitar conflictos entre los componentes.

Figura 22.

Importación de Librerías Python



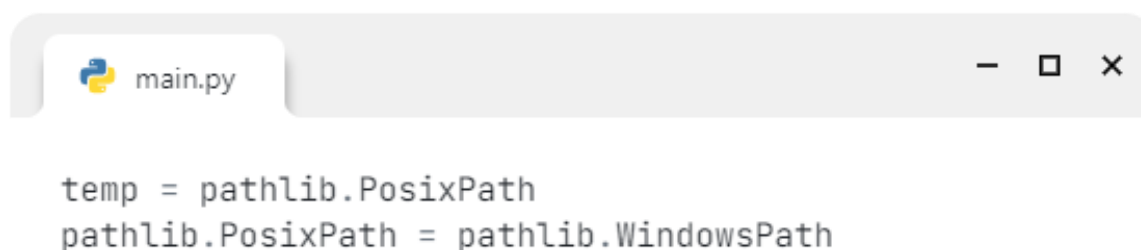
```
import tkinter as tk
import threading
import torch
import cv2
import numpy as np
from PIL import Image, ImageTk
import chime
import pathlib
```

Nota. El código muestra las librerías utilizadas para la realización de la integración del modelo entrenado en un prototipo de interfaz de monitoreo.

A su vez, se realiza una modificación en la clase “**pathlib**”. **PosixPath** para garantizar la compatibilidad con sistema Operativo. Esta adaptación es primordial para una exitosa ejecución y el funcionamiento coherente del código en los diferentes entornos.

Figura 23.

Ruta Estática del Modelo Entrenado



```
temp = pathlib.PosixPath
pathlib.PosixPath = pathlib.WindowsPath
```

Nota. El código detalla la ruta que se utiliza para responder la compatibilidad del Sistema Operativo con el modelo entrenado en Yolov5.

Inicialización de la Interfaz Gráfica:

Para desplegar la interfaz utilizamos la clase **ObjectDetectorApp** para definir el constructor **__init__**, el cual, accede a las dimensiones y fondo de la interfaz. De igual modo, se crean las etiquetas, botones y variables de manejo de las cámaras y de la alerta. Por otro lado, se carga el modelo YOLOv5.

Activación de Cámaras:

Para este punto, hacemos uso del método `active_cam1` y `active_cam2`, a fin, de activar las cámaras respectivamente. A su vez, se crean métodos con `thread` para activar los hilos separados para la detección de objetos con cada una de las cámaras, esto permite que la interfaz no se bloquee durante la ejecución.

Figura 24.

Definición del Método para activar las cámaras



```

> def active_cam1(self):

def active_cam2(self):
    self.cap_cam2 = cv2.VideoCapture(2)
    if not self.cap_cam2.isOpened():
        print("No se puede abrir la cámara CAM 2")
        return

    self.is_running_cam2 = True
    self.thread_cam2 = threading.Thread(target=self.detect_objects_cam2)
    self.thread_cam2.start()

```

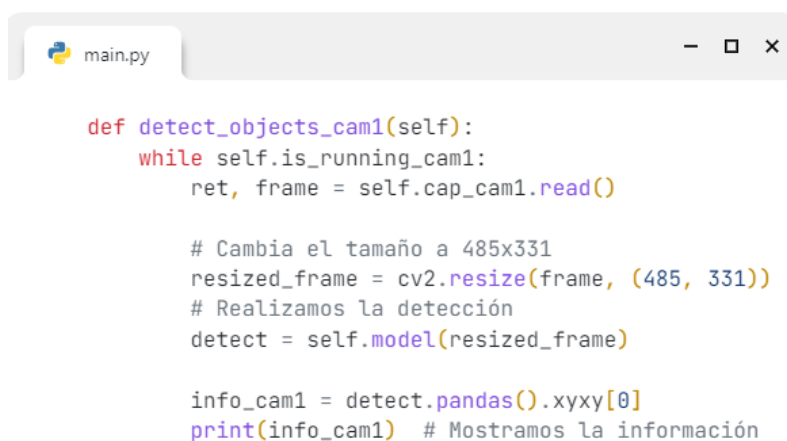
Nota. El código de la presente figura detalla la estructura de activación de las cámaras en hilos (threads) separados en el instante que la cámara 1 se active, también permitirá activar la cámara 2, como se observa en la Figura 28.

Detección de objetos en tiempo real:

Para la captura en tiempo real de las cámaras, se emplea el método **detect_objects**, que permite redimensionar y se envía al modelo YOLOv5 a realizar la detección de objetos.

Figura 25.

Integración de Cámara 1 y Modelo Entrenado



```

def detect_objects_cam1(self):
    while self.is_running_cam1:
        ret, frame = self.cap_cam1.read()

        # Cambia el tamaño a 485x331
        resized_frame = cv2.resize(frame, (485, 331))
        # Realizamos la detección
        detect = self.model(resized_frame)

        info_cam1 = detect.pandas().xyxy[0]
        print(info_cam1) # Mostramos la información

```

Nota. El código detalla la captura del video de la cámara 1, se redimensiona, y ejecuta a través del modelo entrenado, y finalmente muestra la información de la primera detección (si existe) en la consola.

Interfaz gráfica principal:

Para mantener activa la aplicación e interacción con el usuario. Se crea una instancia con la clase **ObjectDetectorApp**, la cual inicia el bucle principal de la interfaz gráfica mediante **root.mainloop()**. Asegurando la respuesta fluida y continuidad con los resultados obtenidos en la detección de objetos.

Figura 26.

Activación de Interfaz Principal



```
if __name__ == "__main__":
    root = tk.Tk()
    app = ObjectDetectorApp(root)
    root.mainloop()
```

Nota. En código configura y ejecuta una aplicación de interfaz gráfica utilizando Tkinter, con una ventana principal y un bucle que permite la interfaz gráfica responda al comportamiento específico determinado por la clase **ObjectDetectorApp** la misma que realiza la detección de la clase “armas”.

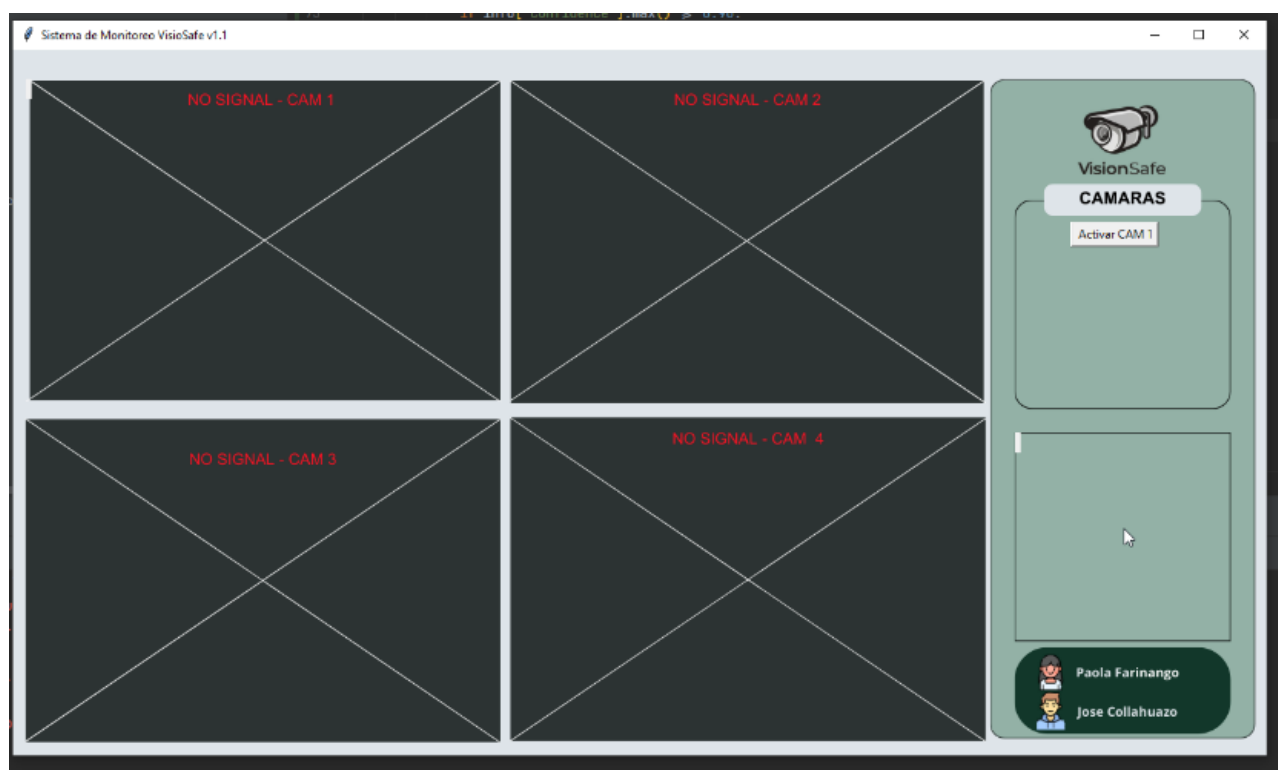
En conjunto, este proceso permite crear un equilibrio en el monitoreo de seguridad en tiempo real a partir de sus alertas visuales y auditivas, mejorando de tal forma la capacidad de respuesta y vigilancia.

Vista resultado Interfaz

Una vez finalizada la aplicación, se observa la interfaz gráfica, que, combinada con cada uno de los elementos mencionados en su desarrollo, al permitir un monitoreo efectivo e intuitivo. Esta visualización brinda una experiencia total para la interpretación y supervisión de la detección del objeto, en este caso “armas de fuego” detectados por parte del sistema.

Figura 27.

Pantalla Principal del Sistema de Monitoreo.

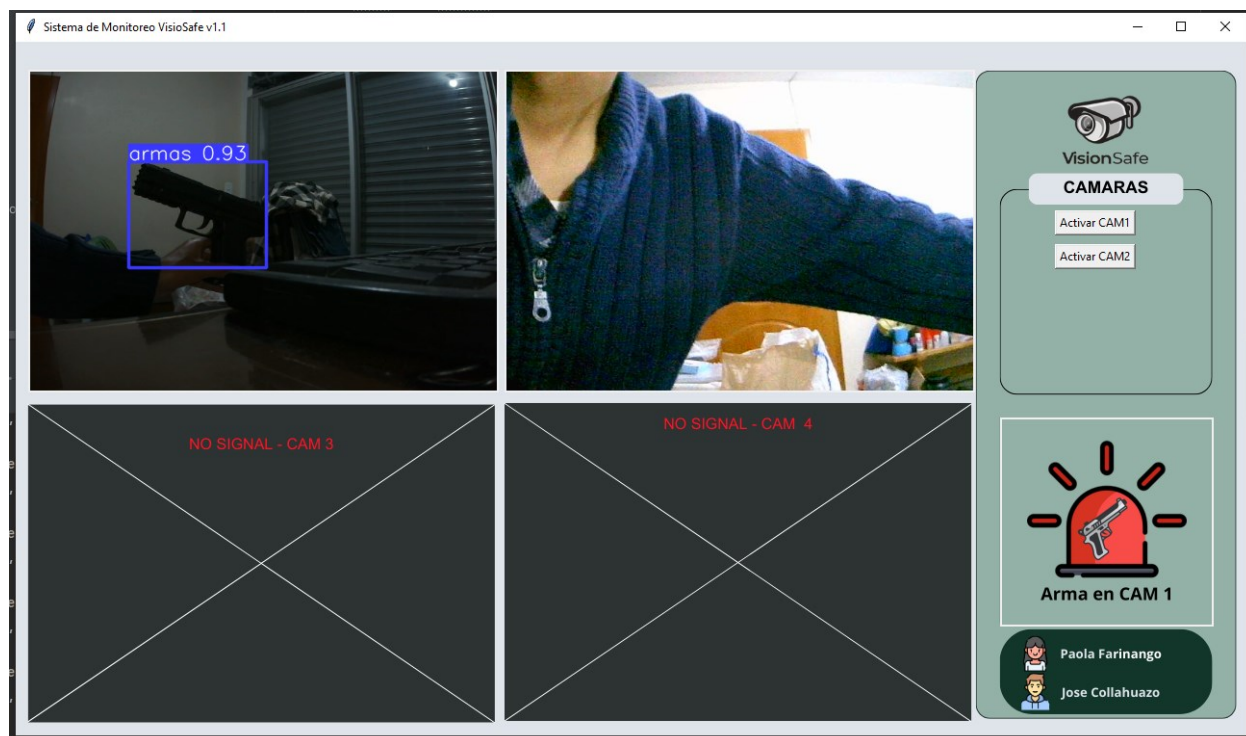


Nota. En esta figura se observa la interfaz gráfica del sistema intuitivo con las respectivas alertas visuales y auditivas, además de una pantalla de monitoreo, botones de activación para las cámaras.

Además, se detecta un objeto con una confianza superior al 93% en imágenes de video, emitiendo un sonido de alerta y la visualización de una alerta visual en la parte inferior de la interfaz.

Figura 28.

Detección de Objetos en Tiempo Real – CAM 1



Nota. En esta figura se observa el funcionamiento de sistema, donde se observa la activación de dos cámaras, en la cual, la cámara 1 detecta el objeto.

Como resultado, se demostró la capacidad del sistema en reconocer el objeto para el cual fue entrenado. Además, se destaca la precisión de reconocimiento del 93%. Este desempeño se debe a la resolución de la cámara utilizada para detectar el arma de fuego, dispositivo que cumple un papel fundamental para la detección precisa del objeto.

Las alertas emitidas en el sistema, tanto con la capacidad visual como auditiva proporcionan una comprensión más profunda de su funcionamiento. Al combinar estas

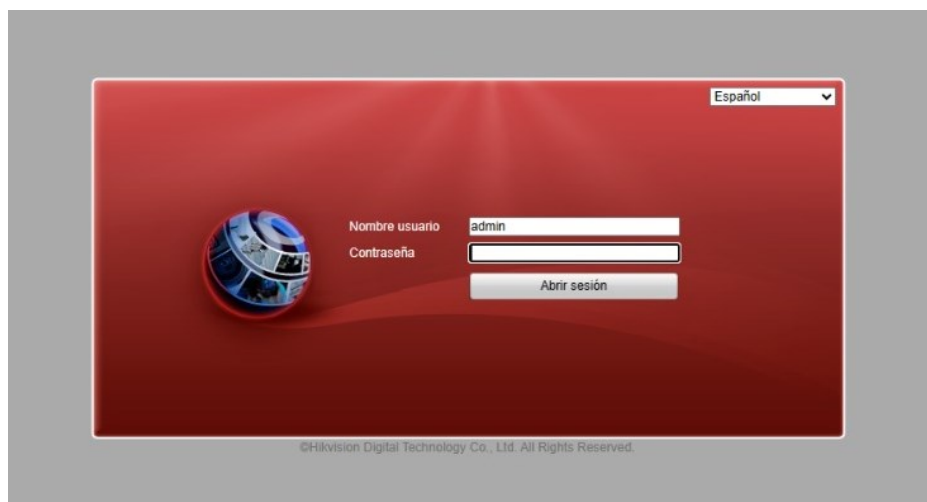
capacidades permite reforzar la confiabilidad del sistema de forma detalla y completa del desempeño obtenido.

Empleo del sistema de videovigilancia

El sistema de videovigilancia empleado en este trabajo es HIKVISION, el cual está diseñado para facilitar la detección de objetos en movimiento de forma precisa, garantizando así la protección del sitio en el que se encuentra instalado. Para ingresar al sistema, accedemos a la interfaz e introducimos las credenciales correspondientes.

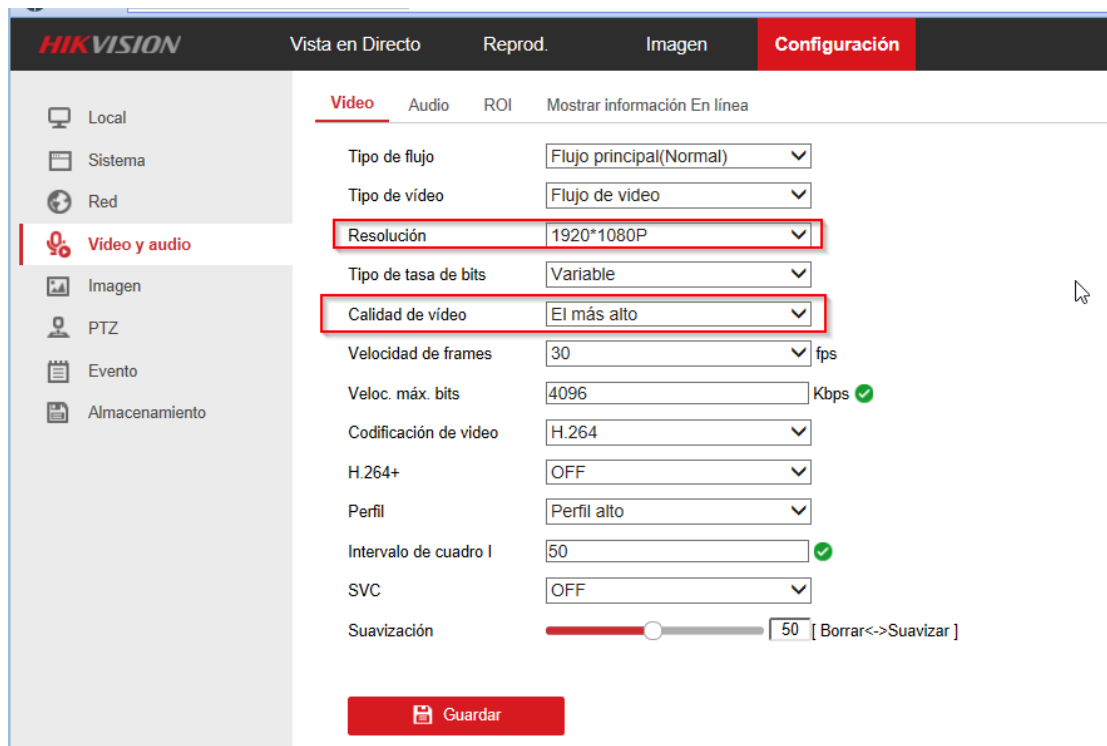
Figura 29.

Ingreso Sistema de Videovigilancia Hikvision



Nota. En esta figura se observa el Login de ingreso al sistema de HIKVISION.

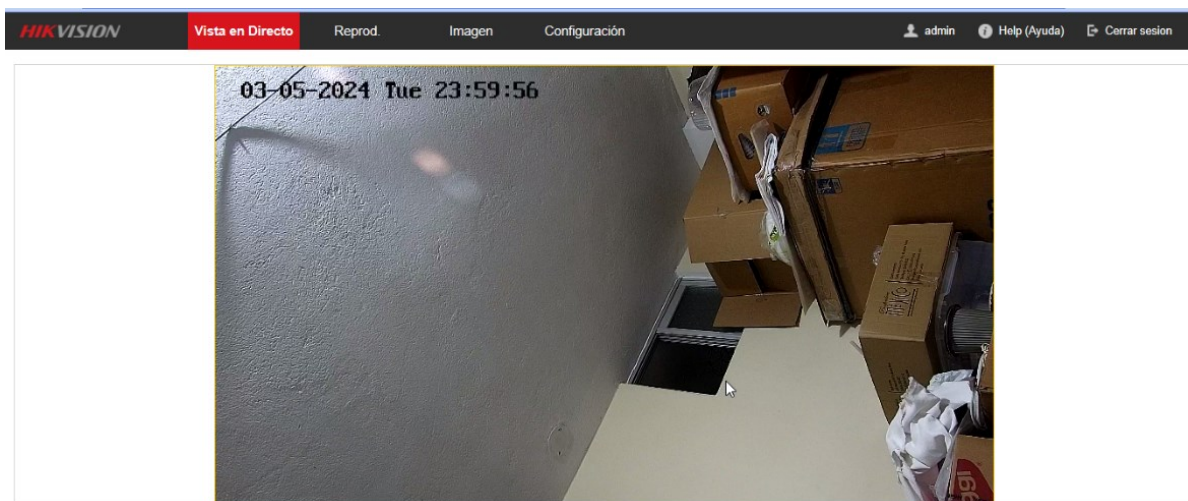
Una vez dentro del sistema se procede a realizar las diferentes configuraciones con respecto a la resolución y calidad de video, siendo este un punto clave para comprobar el rendimiento del modelo.

Figura 30.*Configuración de Resolución para Video en Tiempo Real*

Nota. En esta figura se detalla las respectivas configuraciones realizadas en el sistema de HIKVISION.

Finalmente, una vez realizado las configuraciones, se observa como se despliega la pantalla de monitorio del sistema de videovigilancia.

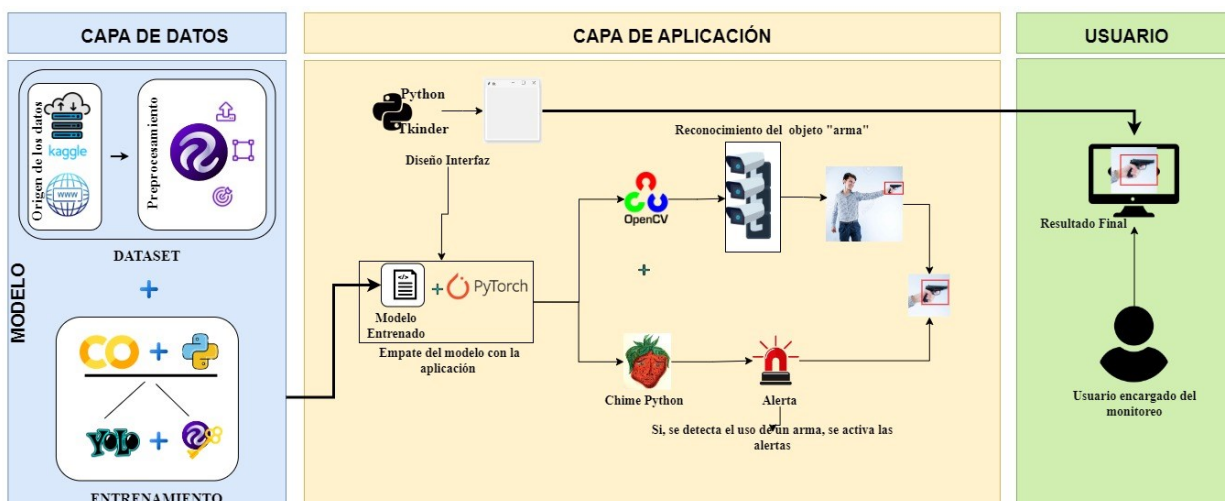
Figura 31.*Vista en Directo Cámara Hikvision*



Nota. En esta figura se observa la pantalla de monitoreo del sistema de videovigilancia.

Figura 32.

Arquitectura del sistema



Nota. En esta figura se detalla la arquitectura en capas implementada durante el desarrollo del sistema propuesto.

Al concluir con el proceso de desarrollo, nuestro sistema se consolida en una arquitectura en capas, siendo este la pieza clave del funcionamiento. Conformada de tres componentes, que cumplen con el objetivo asignado.

Capa de Datos: Se realiza la creación del modelo, mediante la técnica de recolección de los datos, seguido de la carga y ajuste del conjunto de los datos, a fin de, entrenar el modelo aprovechando los recursos computacionales de Google Colab para realizar esa tarea, de tal forma, que el modelo sea preciso para su implementación.

Capa de Aplicación: Se crea la interfaz de usuario y combina el modelo entrenado. A fin de incorporar los aspectos claves que cumplen con la misión de detectar el objeto, el cual al ser reconocido se emitirá las alertas al brindar una respuesta inmediata, de tal forma se establezcan los caminos para la detección de las armas.

Capa de usuario: En esta capa se hace uso de la aplicación desarrollada, en aspectos de evaluar el rendimiento del sistema al llevarlo al escenario de despliegue.

La ejecución sincronizada de cada capa permite, llevar a cabo, la construcción de un sistema eficiente e integral, sino que también, garantiza el correcto funcionamiento y su capacidad de adaptabilidad a los requisitos cambiantes.

Equipos y Materiales

La implementación del sistema requirió de la utilización de ciertos dispositivos. Aquellos que desempeñaron roles específicos con el fin de garantizar el desempeño del sistema. El uso de estas tecnologías contribuye a la precisión y eficacia del sistema en el reconocimiento y alertas a posibles amenazas.

Tabla 5.*Equipos de Desarrollo e Implementación*

EQUIPOS	DESCRIPCIÓN
	<p>Laptop Asus VIVOBOOK, equipada con un procesador Intel Core i7 de 10ma de 2.30 GHz Generación y 16 GB de RAM, y almacenamiento de 512 GB.</p>
	<p>ELP Cámara USB de Seguridad Industrial con gran angular de luz baja HD con lente de ojo de pez de 170 grados. Alta velocidad de fotogramas: entrega 30fps en resolución de 1280* 720 Hight Definition.</p>
	<p>Cámara Web MPC01 que ofrece una resolución de 5 MP, sensor CMOS de 1/4, velocidad de transmisión de 1280 x 720, 1920x 1080, capacidad para transmitir y grabar video HD 1080p con detalles realistas y vibrantes, con un rango de imagen de 70 – 90 cm.</p>
	<p>Cámara Domo de Red PTZ DS-2DE7230IW-AE para exteriores de 2MP con visión nocturna de la marca Hikvision, panorámica de 360°, un sensor CMOS de barrido progresivo, que permite DWDR (Digital Wide Dynamic Range), es compatible con los formatos de compresión H.264 Y MJPEG con resolución de 1920 x 1080 Full HD, zoom óptico de 30x.</p>

Nota. En esta tabla se detallan los equipos utilizados con sus respectivas características.

Capítulo IV: Resultados

En este capítulo, se muestran los resultados provenientes del proceso de desarrollo del sistema de detección de armas de fuego, al igual que, la validación efectuada en las instalaciones de la matriz de la Universidad de las Fuerzas Armadas ESPE, mismo que, nos permitió verificar el rendimiento del sistema completo. Cumpliendo, de tal forma con los objetivos específicos propuestos.

Implementación

Generalmente las cámaras de seguridad se sitúan en áreas donde hay una gran frecuencia de personas y de objetos en el entorno. Por lo que las cámaras al captar en tiempo real tienen a ser altamente complejas, debido a la cantidad de elementos que posee cada cuadro de video capturado en el tiempo.

La persona designada para el monitoreo se debe prestar atención a las personas u objetos que aparecen en los cuadros de video que capturan las cámaras de seguridad. Pero siendo esta actividad de gran responsabilidad en algunos casos las personas del monitoreo se distraen y no prestan la atención necesaria al momento de la vigilancia del entorno. Por tal razón, el objetivo de este trabajo fue la combinación de tecnologías y de la situación cotidiana de resguardar la seguridad de los miembros que con parte del campus matriz de la Universidad de las Fuerzas Armadas ESPE. En el cual se integra un modelo de Deep Learning con una interfaz intuitiva, el cual cumple con la finalidad de detectar el uso de armas de fuego en el entorno al que esta implementado.

Para realizar las pruebas de desempeño, hemos contado con el uso de dos cámaras de video. De los cuales destacamos como punto relevante, la resolución de las cámaras a prueba, una es full HD y la otra HD. Debido a esto la precisión varia dentro de un rango de efectividad del 93% en adelante.

Figura 33.*Funcionamiento del sistema de detección*

Nota. En esta figura se observa el rendimiento del sistema.

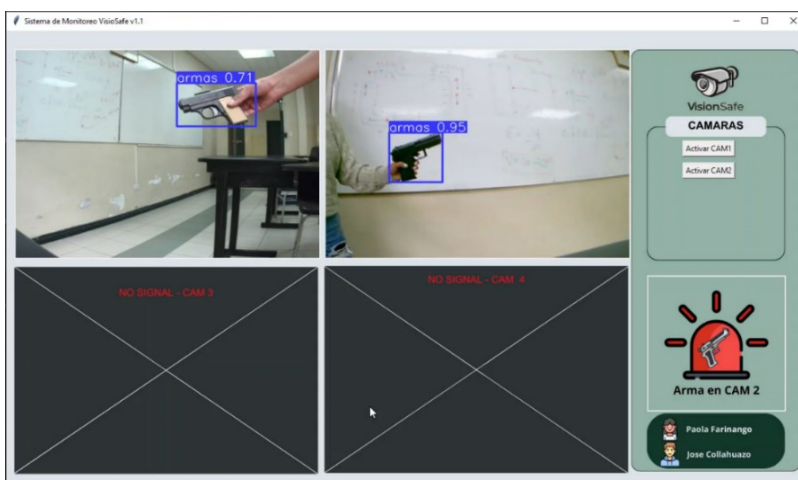
Interfaz

El sistema en marcha se observa en la siguiente figura. Donde se puede visualizar el momento en que el sistema realiza una detección, este procede a ser capturado en un cuadro delimitador en donde se detecta eficazmente el arma de fuego, además se generan las alertas tanto visual como auditiva en la parte inferior derecha, esto con el fin de facilitar en su actividad al personal encargado del monitoreo.

Como se evidencia, la precisión al reconocer el arma de fuego, de acuerdo a la cámara, varía el desempeño la cámara 1 detecta un 71% debido a su resolución, mientras que, la cámara 2 detecta un 95% en su precisión.

Figura 34.

Detección de Arma simultáneamente en dos cámaras.

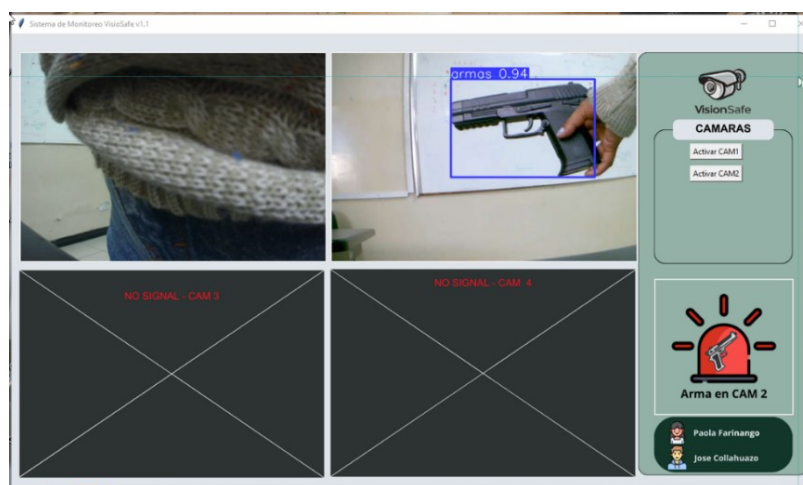


Nota. En la figura se observan la detección del arma de fuego en los escenarios establecidos.

En otro escenario, se observa, de igual forma como el sistema detectar el arma de fuego con una precisión del 94% de acuerdo con los factores del entorno.

Figura 35.

Detección de Arma con Proximidad Cercana

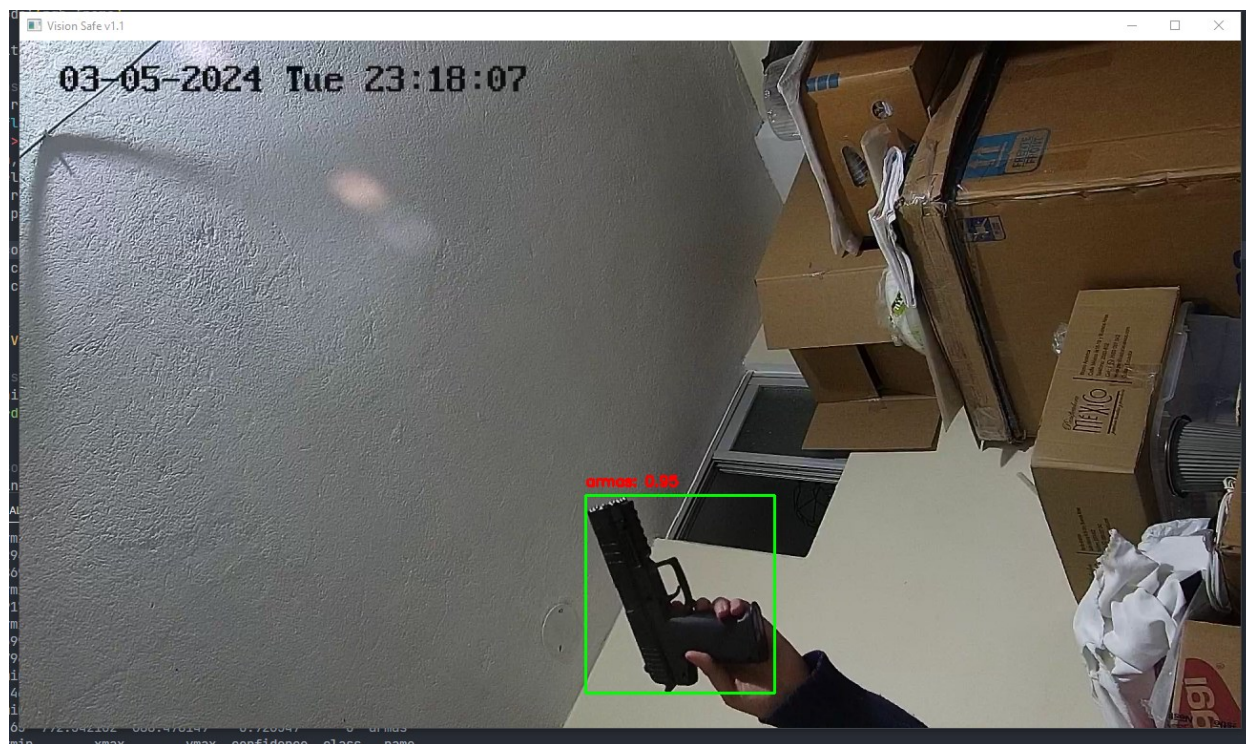


Nota. En esta figura se observa otro ejemplo de detección del arma de fuego, en el cual el objetivo se lo identifica por encontrarse a una distancia cercana de la cámara, permitiendo lograr una precisión del 94% previsto en el modelo.

En una situación distinta, ya con nuestro modelo implementado en el sistema de videovigilancia de HIKVISION, se procede a realizar una prueba de su funcionamiento, donde se puede evidenciar un 95% de precisión. Además, se destaca la importancia de la resolución y calidad de transmisión de video, ya que su facilidad de detectar es más eficaz, a su vez, la capacidad del sistema en adaptarse a entornos donde varía la iluminación resulta crucial, debido a que demuestra el desempeño que tiene nuestro modelo.

Figura 36.

Implementación Modelo Deep Learning – Sistema de Videovigilancia Hikvision



Nota. En esta figura se observa la implementación de un sistema de videovigilancia en combinación con nuestro modelo de Deep Learning, el cual logra obtener una precisión del 95% de efectividad.

Capítulo V: Conclusiones y Recomendaciones

Conclusiones

Cumpliendo con los objetivos propuestos, se observa que la validación del algoritmo nos proporciona una precisión del 99%, sensibilidad del 95%, F1-score con 91% y exactitud del 93% en la detección de armas de fuego. Esto debido a que, las pruebas realizadas dentro de la Universidad de las Fuerzas Armadas ESPE, con una instalación básica de vigilancia, se logra obtener un 95% en la detección de las armas de fuego en tiempo real tomando en cuenta la posición del arma, del mismo modo la distancia es un factor base para la detección, además, vale recalcar que las condiciones del lugar como de las herramientas tecnológicas tiene un impacto significativo al momento de arrojar los resultados.

El tiempo de reacción de la alerta al capturar el arma de fuego es de 2 milisegundos, además, la alerta no solo emite un sonido, sino que se visualiza a través de un mensaje en la interfaz, el mensaje especifica que cámara capta el uso del arma del fuego. En conjunto, esta combinación marca una diferencia significativa en la eficacia y satisfacción del sistema en general.

A pesar de la eficiencia obtenida del modelo en cuestión, la no obtención del 99% de efectividad prevista se atribuye, en parte, a las limitaciones que tienen los dispositivos utilizados para efectuar las pruebas debido a la resolución que cuentan. Este hallazgo resalta lo importante de considerar los factores externos que pueden actuar en el rendimiento del modelo.

La calidad de imágenes y videos es indispensable para la implantación del modelo de red neuronal en el sistema de detección de armas de fuego, a través del uso de cámaras de seguridad. Por lo que, una opción factible es integrar tecnologías como sensores de alta definición (HD) y alto rango dinámico (HDR).

El optar por cámaras de mayor calidad no solo mejora la claridad visual, sino que, también impulsa la precisión del algoritmo de detección en condiciones desafiantes. De manera que, la elección minuciosa de los dispositivos de video como cámaras, es primordial para perfeccionar el rendimiento del sistema al identificar de forma precisa el arma de fuego.

La capacidad del equipo en uso contribuyó significativamente en el desempeño del modelo de red neuronal convolucional (CNN) para la detección de armas de fuego. A lo largo de la ejecución del modelo y la activación de las cámaras integradas, se observó un consumo de memoria RAM, alcanzando un 85%. A su vez, se destaca la eficiencia del modelo YOLOv5, por su diseño compacto y efectivo en entornos con recursos computacionales limitados. Por lo que, el control riguroso de los recursos se muestra cómo, factores críticos para asegurar el rendimiento efectivo del sistema.

Es crucial tomar en cuenta que la detección de armas en tiempo real continúa siendo un desafío sustancial, debido a las variaciones de ángulos de visión, las condiciones del entorno, es decir, la claridad del espacio, así también la forma en que el portador lleva el arma. Estos factores son un obstáculo para que el modelo de detección sea eficaz, sino que también destaca la importancia de continuar con investigaciones, con el fin de superar estos desafíos y llegar a obtener un modelo preciso y robusto.

Para finalizar, este trabajo de titulación no solo fue la combinación de conocimientos obtenidos en el transcurso de la carrera, sino también, de investigaciones exploradas en áreas de tecnología en tendencia, incluso se incorporan aspectos sociales en este caso seguridad y educación. Siendo estos los dos puntos clave para desarrollar el sistema, dado que es importante promover entornos seguros que busque el bienestar colectivo.

Recomendaciones

El problema en este tipo de implementación se evidencia en la complejidad en los entornos donde se alojan las cámaras, además de la resolución de estas en transmitir, siendo este el factor crítico, para garantizar resultados favorables.

Se recomienda usar imágenes con una resolución de 72 pixeles/pulgada, dado que, si son menores a esta, las dimensiones del objeto en cuestión, pierde características, lo que hace que el sistema no detecte con precisión del arma de fuego.

Para la fase del entrenamiento del modelo es importante el uso del GPU que viene integrado en la herramienta de Google Collab Pro, debido a que, por el número de las imágenes tratadas, se requiere de un óptimo procesamiento minimizando los tiempos en la detección del arma.

Capítulo VI: Trabajos Futuros

Este sistema en particular es la base para la construcción para nuevos proyectos que pueden fomentar la seguridad del entorno de la Universidad de las Fuerzas Armadas ESPE, que en este punto es muy importante resguardar la integridad de todos los individuos que integran el plantel.

Para el desarrollo de aplicaciones de detección en tiempo real, se destaca el uso del modelo Yolov5 implementado para el presente sistema de alerta temprana basado en visión artificial para la detección de armas de fuego. La elección de este modelo se basó en su eficiencia en la detección en tiempo real, siendo crucial la presión y velocidad del sistema. Además, cabe mencionar la implementación del modelo YOLO v8 el más reciente, con el fin de mantenerse actualizado y consumir las últimas innovaciones asegurando la eficiencia del modelo a largo plazo.

El modelo entrenado puede mejorar su capacidad hasta alcanzar un nivel autónomo en su predicción, adquiriendo más imágenes, con el fin de llegar a detectar cualquier otro tipo de armas, no solo armas de fuego sino también las armas blancas como cuchillos o armas de fuego de mayor tamaño. Por último, a este sistema se pueden añadir más aspectos con poses en posición de arma de fuego, que crearán el sistema más robusto, práctico y eficiente.

Bibliografía

- Avramidis, E. (2013). RankEval: Open Tool for Evaluation of Machine-Learned Ranking. *The Prague Bulletin of Mathematical Linguistics NUMBER, 100*, 63–72.
<https://doi.org/10.2478/pralin-2013-0012>
- Beck, K., Beedle, M., Bennekum, A. van, Cockburn, A., Cunningham, W., Fowler, M., Grennig, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., & Marick, B. (2001). *Manifesto for Agile Software Development*. <https://agilemanifesto.org/iso/en/manifesto.html>
- Bhatti, M. T., Khan, M. G., Aslam, M., & Fiaz, M. J. (2021). Weapon Detection in Real-Time CCTV Videos Using Deep Learning. *IEEE Access, 9*, 34366–34382.
<https://doi.org/10.1109/ACCESS.2021.3059170>
- El Telégrafo. (2023). *El Telégrafo - Presidente Guillermo Lasso permite el uso y porte de armas a civiles en defensa propia y otras medidas*. Presidente Guillermo Lasso Permite El Uso y Porte de Armas a Civiles En Defensa Propia y Otras Medidas.
<https://www.letelegrafo.com.ec/noticias/actualidad/44/presidente-guillermo-lasso-permite-el-uso-y-porte-de-armas-a-civiles-en-defensa-propia>
- Gali, M., Dhavale, S., & Kumar, S. (2022). Real-Time Image Based Weapon Detection Using YOLO Algorithms. *Communications in Computer and Information Science, 1614 CCIS*, 173–185. https://doi.org/10.1007/978-3-031-12641-3_15/COVER
- Goodfellow, I., Bengio, Y., Courville, A., & Heaton, J. (2017). Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning. *Genetic Programming and Evolvable Machines 2017 19:1, 19(1)*, 305–307. <https://doi.org/10.1007/S10710-017-9314-Z>
- Google. (2022, September 26). *Práctica de AA: Clasificación de imágenes | Machine Learning | Google for Developers*. <https://developers.google.com/machine-learning/practica/image-classification/convolutional-neural-networks?hl=es-419>
- Google Colab. (2023). *colab.google*. <https://colab.google/>
- Halford, M. (2021). *Chime*. <https://github.com/MaxHalford/chime?tab=readme-ov-file>

- Hui, J. (2018, March 17). *Real-time Object Detection with YOLO, YOLOv2 and now YOLOv3 | by Jonathan Hui | Medium*. <https://jonathan-hui.medium.com/real-time-object-detection-with-yolo-yolov2-28b1b93e2088>
- Iyer, R., Shashikant Ringe, P., Varadharajan Iyer, R., & Prabhulal Bhensdadiya, K. (2021). Comparison of YOLOv3, YOLOv5s and MobileNet-SSD V2 for Real-Time Mask Detection. *Article in International Journal of Research in Engineering and Technology*, 25. <https://www.researchgate.net/publication/353211011>
- Kaggle. (2023). *Kaggle: Your Machine Learning and Data Science Community*. <https://www.kaggle.com/>
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature* 2015 521:7553, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Marcela Silva-Labanda, L. I., & Carlos Cobos-Torres, J. I. (2021). Sistema de detección de aves mediante análisis de imágenes. *Dominio de Las Ciencias*, 7(6), 1464–1484. <https://doi.org/10.23857/DC.V7I6.2404>
- MELLA, C. (2023, June 10). *La inseguridad en Ecuador escala a niveles históricos y se impone como prioridad del próximo Gobierno | Internacional | EL PAÍS*. <https://elpais.com/internacional/2023-07-10/la-inseguridad-en-ecuador-escala-a-niveles-historicos-y-se-impone-como-prioridad-del-proximo-gobierno.html>
- OpenCV - Open Computer Vision Library. (2023). <https://opencv.org/>
- Overgaard, J. (2022). *validated_weapons_v5*. <https://www.kaggle.com/datasets/jakobovergaard/validated-weapons-v5/discussion>
- Phung, V. H., & Rhee, E. J. (2019). A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets. *Applied Sciences* 2019, Vol. 9, Page 4500, 9(21), 4500. <https://doi.org/10.3390/APP9214500>
- Pykes, K. (2021, April 9). *The Most Common Evaluation Metrics In NLP | by Kurtis Pykes | Towards Data Science*. <https://towardsdatascience.com/the-most-common-evaluation->

metrics-in-nlp-ced6a763ac8b

PyTorch. (2023). <https://pytorch.org/>

Reda, S. (2019). *Weapons*. <https://www.kaggle.com/datasets/sheroook/weapons>

Reddy A, D., Poojitha, M., Puspallalitha, G., Vishnu Vardhan Reddy, M., Ashok Kumar, K., & Nithya Sree, G. (2023). Weapon Identification using YOLO V5 Algorithm. *International Journal of Science and Research (IJSR)*, 12(11), 550–564.

<https://doi.org/10.21275/SR23827102656>

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You Only Look Once: Unified, Real-Time Object Detection* (pp. 779–788). <http://pjreddie.com/yolo/>

Reuters. (2024). *¿Cómo llegó Ecuador a la actual ola de violencia? ¿Cómo Llegó Ecuador a La Actual Ola de Violencia?* <https://www.vozdeamerica.com/a/como-llego-ecuador-a-la-actual-ola-de-violencia/7435738.html>

Roboflow. (2022, October 2). *Detección de objetos - Roboflow*.

https://help.roboflow.com/es_ES/object-detection

Rodriguez, R. (2023). *Informe Mundial 2024: Ecuador | Human Rights Watch*.

<https://www.hrw.org/es/world-report/2024/country-chapters/ecuador>

Scholnik-Elias, A., Martínez-Díaz, S., Luna-Taylor, J. E., & Castro-Liera, I. (2023). Detección de armas tipo pistola mediante el uso de redes convolucionales con una arquitectura tipo YOLO y estereoscopia. *Pädi Boletín Científico de Ciencias Básicas e Ingenierías Del ICBI*, 11(Especial2), 196–204. <https://doi.org/10.29057/ICBI.V11IESPECIAL2.10727>

Sharma, K., Rawat, S., Parashar, D., Singh Rawat, S., Sharma, S., Roy, S., & Sahoo, S. (2023). State-of-the-Art Analysis of Multiple Object Detection Techniques using Deep Learning. *Article in International Journal of Advanced Computer Science and Applications*, 14(6), 2023. <https://doi.org/10.14569/IJACSA.2023.0140657>

Solawetz, J. (2020a, June 18). *What is YOLOv5? A Guide for Beginners*.

<https://blog.roboflow.com/yolov5-improvements-and-evaluation/>

Solawetz, J. (2020b, August 4). *Train, Validation, Test Split and Why You Need It*.

<https://blog.roboflow.com/train-test-split/>

Szeliski, R. (2022). *Computer Vision*. <https://doi.org/10.1007/978-3-030-34372-9>

Thiruthanigesan, K., Nawarathna, R. D., & Ragel, R. G. (2021). Suspicious Object Detection in Environments with Tear Gas. *2021 IEEE 16th International Conference on Industrial and Information Systems, ICIIIS 2021 - Proceedings*, 185–190.

<https://doi.org/10.1109/ICIIIS53135.2021.9660682>

Tkinter. (2023). *tkinter — Python interface to Tcl/Tk — Python 3.12.0 documentation*.

<https://docs.python.org/3/library/tkinter.html>

Torras, I. (2015). *Introducción — Scrum 1 documentation*.

<https://metodologiascrum.readthedocs.io/en/latest/Scrum.html#que-es-scrum>

Vikas, K., Rajabrahmam, T., Venu, P., & Hariharan, S. (2022). Deep Learning Approach for Event Monitoring System. *International Journal of Managing Information Technology*, 14(3), 23–30. <https://doi.org/10.5121/IJMIT.2022.14302>

Welcome to Python.org. (2023). <https://www.python.org/>

Yang, G., Feng, W., Jin, J., Lei, Q., Li, X., Gui, G., & Wang, W. (2020). Face Mask Recognition System with YOLOV5 Based on Image Recognition. *2020 IEEE 6th International Conference on Computer and Communications, ICC3 2020*, 1398–1404.

<https://doi.org/10.1109/ICC351575.2020.9345042>