



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL**

CARRERA DE TELECOMUNICACIONES

Tema:

**SISTEMA CCTV CON ALERTA TEMPRANA Y RECONOCIMIENTO DE
ARMAS DE FUEGO UTILIZANDO MACHINE LEARNING**

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la
obtención del título de Ingeniero en Telecomunicaciones

ÁREA: Programación y redes

LÍNEA DE INVESTIGACIÓN: Tecnologías de la información

AUTOR: Jorge Miguel Unapucha Unapucha

TUTOR: Ing. Elizabeth Paulina Ayala Baño, Mg.

Ambato - Ecuador

febrero - 2025

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: SISTEMA CCTV CON ALERTA TEMPRANA Y RECONOCIMIENTO DE ARMAS DE FUEGO UTILIZANDO MACHINE LEARNING, desarrollado bajo la modalidad Proyecto de Investigación por el señor Jorge Miguel Unapucha Unapucha estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, febrero 2025.

ELIZABETH
PAULINA
AYALA
BANO

Firmado digitalmente por ELIZABETH
PAULINA AYALA BANO
DN: cn=ELIZABETH PAULINA
AYALA BANO, o=EC, ou=SECURITY
DATA S.A., 2.5=ENTIDAD DE
CERTIFICACIÓN DE
INFORMACIÓN
Motivo: Soy el autor de este
documento
Ubicación: n:
Fecha: 2025-02-24 11:31:05:00

Ing. Elizabeth Paulina Ayala Baño, Mg.
TUTORA

AUTORÍA

El presente trabajo de titulación con el tema: SISTEMA CCTV CON ALERTA TEMPRANA Y RECONOCIMIENTO DE ARMAS DE FUEGO UTILIZANDO MACHINE LEARNING es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, febrero 2025.



Jorge Miguel Unapucha Unapucha

C.C. 1805355938

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, febrero 2025.



Jorge Miguel Unapucha Unapucha

C.C. 1805355938

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Jorge Miguel Unapucha Unapucha, estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA CCTV CON ALERTA TEMPRANA Y RECONOCIMIENTO DE ARMAS DE FUEGO UTILIZANDO MACHINE LEARNING nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con el señor Presidente del Tribunal.

Ambato, febrero 2025.



Firmado electrónicamente por:
**FRANKLIN OSWALDO
MAYORGA MAYORGA**

Ing. Franklin Oswaldo Mayorga Mayorga, Mg.

PRESIDENTE DEL TRIBUNAL



Firmado electrónicamente por:
**GEOVANNI DANILO
BRITO MONCAYO**

Ing. Geovanni Brito Moncayo, Mg.

PROFESOR CALIFICADOR



Firmado electrónicamente por:
**JUAN PABLO PALLO
NORONA**

Ing. Juan Pablo Pallo Noroña, Mg.

PROFESOR CALIFICADOR

DEDICATORIA

El presente proyecto está dedicado a mi familia, en reconocimiento a su apoyo incondicional, su paciencia y, sobre todo su confianza a lo largo de este proceso.

A mis padres, Hernán y María, por ser mi fuente de inspiración constante y por su inquebrantable respaldo en cada etapa de este camino.

A mis hermanas y hermano, por su cariño, su ayuda y por estar siempre alentándome a seguir adelante.

A mis amigos, por su compañía y por la motivación que me brindaron para alcanzar esta meta.

AGRADECIMIENTO

En primer lugar, quiero agradecer a Dios por darme la sabiduría y el valor para sobrellevar este largo proceso.

A mi tutora, Ing. Paulina Ayala, expreso mi más sincero agradecimiento por compartir sus conocimientos, experiencia y, sobre todo, por su invaluable guía a lo largo de este proceso de titulación. Sus consejos y apoyo académico han sido esenciales para desarrollar de mejor manera este trabajo.

A la Ing. Pamela Castro, por su paciencia y orientación a lo largo del desarrollo de este proyecto.

Agradezco profundamente a mis padres por su inquebrantable confianza en mí y por ser la inspiración que me motiva a superarme cada día.

A mis hermanas, por su infinita confianza, su apoyo incondicional y por haber creído en mí en todo momento.

A mis amigos Yuli, Eve, Karito, Santy y Ariel, por estar a mi lado en los momentos más difíciles y brindarme su apoyo emocional, que fue invaluable y significativo a lo largo de todo este proceso.

A Fer y Juan por ser mis amigos desde inicios de la carrera, por el tiempo que hemos compartido y sobre todo por su inigualable amistad y apoyo.

A Nikolle, por todos estos años de amistad, paciencia y sobre todo por estar en los momentos más difíciles, infinitas gracia

ÍNDICE GENERAL DE CONTENIDOS

PORTADA.....	i
APROBACIÓN DEL TUTOR.....	ii
AUTORÍA.....	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DEL TRIBUNAL DE GRADO.....	v
DEDICATORIA.....	vi
AGRADECIMIENTO	vii
ÍNDICE GENERAL DE CONTENIDOS	viii
ÍNDICE DE TABLAS.....	xii
ÍNDICE DE FIGURAS.....	xiv
ÍNDICE DE ANEXOS	xvii
RESUMEN EJECUTIVO	xviii
ABSTRACT	xix
CAPÍTULO I. MARCO TEÓRICO.....	1
1.1 Tema de investigación.....	1
1.1.1 Planteamiento del problema	1
1.2 Antecedentes investigativos	2
1.3 Fundamentación teórica	5

1.3.1 Sistemas de seguridad	5
1.3.2 Sistemas CCTV	7
1.3.3 Tipos de sistemas CCTV	9
1.3.4 Inteligencia Artificial	13
1.3.5 Métodos de aprendizaje.....	15
1.3.6 Modelos de detección.....	16
1.3.7 Softwares de etiquetado	18
1.3.8 Microcomputador	20
1.3.9 Plataformas de entrenamiento	23
1.3.10 Software de programación	24
1.3.11 Situación actual de seguridad.....	26
1.3.12 Armas de fuego	27
1.4 Objetivos	30
1.4.1 Objetivo general	30
1.4.2 Objetivos específicos	30
CAPÍTULO II. METODOLOGÍA	32
2.1 Materiales	32
2.2 Métodos.....	32

2.2.1 Modalidad de la investigación	32
2.2.2 Recolección de información.....	33
2.2.3 Procesamiento y análisis de datos	33
CAPÍTULO III. RESULTADOS Y DISCUSIÓN.....	35
3.1 Desarrollo de la propuesta.....	35
3.1.1 Definición de los elementos para la propuesta.....	35
3.1.2 Elementos por detectar en el sistema	36
3.1.3 Esquema general del proyecto	37
3.1.4 Selección de tecnologías	40
3.1.5 Descripción del sistema de detección	46
3.1.6 Implementación sistema CCTV	47
3.1.7 Desarrollo del dataset.....	52
3.1.8 Entrenamiento del sistema	56
3.1.9 Procesamiento y detección de armas en imágenes de video	58
3.1.10 Visualización y alertas	62
3.2 Resultados	64
3.2.1 Resultados del entrenamiento	64
3.2.2 Pruebas de funcionamiento	75
3.2.3 Pruebas de generación de alertas.....	86
3.3 Costos.....	89

3.3.1 Costos de insumos del proyecto	89
3.3.2 Costos de desarrollo técnico	89
3.3.3 Costo total	90
CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES	91
4.1 Conclusiones	91
4.2 Recomendaciones.....	92
REFERENCIAS BIBLIOGRÁFICAS	94
ANEXOS	102

ÍNDICE DE TABLAS

Tabla 1. Comparativa de los tipos de DVR.	10
Tabla 2. Modelos de cámaras de seguridad	10
Tabla 3. Materiales usados en un sistema CCTV.	11
Tabla 4. Comparativa de modelos de detección de objetos	17
Tabla 5. Versiones de Yolo	18
Tabla 6. Comparativa de microcomputadores	22
Tabla 7. Plataformas de entrenamiento basadas en la nube	23
Tabla 8. Comparativa entre softwares de programación.....	25
Tabla 9. Tipos de armas de fuego	27
Tabla 10. Cantidad de armas de fuego decomisadas.....	28
Tabla 11. Ciudades con índice de robos a hogares más alto.....	29
Tabla 12. Ciudades con índice más alto de robos a personas	29
Tabla 13. Porcentaje de confianza de las personas para denunciar delitos.....	30
Tabla 14. Recolección de información.....	33
Tabla 15. Tipos y clasificación de armas	36
Tabla 16. Diferentes formas de cubrir el rostro	37
Tabla 17. Comparativa entre UPS.....	43
Tabla 18. Comparativa Yolov5 y YoloV8	45
Tabla 19. Comparativa de datasets.....	52
Tabla 20. Datos del dataset finalizado	55

Tabla 21. Parámetros por considerar en el entrenamiento.	56
Tabla 22. Librerías utilizadas en el sistema	59
Tabla 23. Importación de librerías	59
Tabla 24. Aplicación del modelo entrenado en el código	60
Tabla 25. Estructura básica de comunicación mediante RTSP	61
Tabla 26. Líneas del código para la parte de detección	61
Tabla 27. Codificación básica de la ventana principal.....	62
Tabla 28. Código para ejecución de la llamada	63
Tabla 29. Codificación de alarma	64
Tabla 30. Comparativa de resultados de entrenamiento obtenidos.....	65
Tabla 31. Valores de predicción obtenidos	66
Tabla 32. Matriz de confusión no normalizada.....	74
Tabla 33. Valores de métricas obtenidos	75
Tabla 34. Costos de insumos del proyecto.....	89
Tabla 35. Total de horas laborales utilizadas en el proyecto	89
Tabla 36. Salario por hora de un ingeniero en Telecomunicaciones	90

ÍNDICE DE FIGURAS

Figura 1. Sistema de seguridad.	6
Figura 2. Detección de arma de fuego.	6
Figura 3. Sirena comunitaria	7
Figura 4. Sistema CCTV	8
Figura 5. Estructura – Sistema CCTV con reconocimiento	8
Figura 6. Diagrama del procesamiento de señales	13
Figura 7. Capas de un modelo de detección.....	16
Figura 8. NVIDIA Jetson Nano.	20
Figura 9. Raspberry Pi 5.	21
Figura 10. Coral Dev Board	22
Figura 11. Cifras de robos a personas con armas de fuego 2022 – 2024.....	26
Figura 12. Cifras de robos a hogares 2022 - 2024	27
Figura 13. Arquitectura del sistema	38
Figura 14. Etapa de entrenamiento detallada	39
Figura 15. Esquema del sistema por bloques	47
Figura 16. Zonas muertas.....	48
Figura 17. Vista de cámara 1 – sala	49
Figura 18. Vista de cámara 2 - cocina.....	49
Figura 19. Configuración de parámetros de red del DVR	50
Figura 20. Asignación y apertura de puertos disponibles.	51

Figura 21. Configuración de las imágenes de video de salida	51
Figura 22. Ventana para cargar imágenes en Roboflow	53
Figura 23. Proceso de etiquetado en Roboflow	54
Figura 24. Dataset terminado	54
Figura 25. Dataset Descargado en Raw URL	55
Figura 26. Descarga de Yolov5 en Google Colab	57
Figura 27. Instalación de requerimientos de Yolov5	57
Figura 28. Comando para descargar dataset desde Roboflow	57
Figura 29. Línea de comando para comenzar con el entrenamiento – 45 épocas.....	58
Figura 30. Línea de comando para comenzar con el entrenamiento – 65 épocas.....	58
Figura 31. Diagrama de flujo general del sistema.....	58
Figura 32. Resultados de entrenamiento con 65 épocas.	65
Figura 33. Resultados de entrenamiento	65
Figura 34. Matriz de confusión	66
Figura 35. Curva de confianza - F1.....	67
Figura 36. Métrica de precisión	68
Figura 37. Curva de precisión y confianza.....	68
Figura 38. Métrica de recall	69
Figura 39. Curva de sensibilidad.....	70
Figura 40. Métrica mAP50.....	71
Figura 41. Métrica mAP50:95	72

Figura 42. Resultados generales.....	73
Figura 43. Ingreso de credenciales.....	76
Figura 44. Credenciales incorrectas	76
Figura 45. Interfaz de visualización de cámaras en tiempo real	77
Figura 46. Activación cámara 1 – sin detección	78
Figura 47. Activación cámara 2 – sin detección	78
Figura 48. Cámara 1 y 2 activadas – sin detección.....	79
Figura 49. Detección de arma corta – cámara 1.....	80
Figura 50. Detección de arma corta – cámara 2.....	80
Figura 51. Detección de arma larga – cámara 1.....	81
Figura 52. Detección de arma larga – cámara 2.....	82
Figura 53. Detección de intruso – cámara 1.....	82
Figura 54. Detección de intruso – cámara 2.....	83
Figura 55. Detección de arma larga e intruso – cámara 1.....	84
Figura 56. Detección de un arma larga y corta – cámara 1.....	84
Figura 57. Imágenes de detección guardadas.....	85
Figura 58. Membretado de imágenes almacenadas.....	86
Figura 59. Alerta activada.....	87
Figura 60. Detección de clases.....	87
Figura 61. Generación de llamada telefónica.....	88
Figura 62. Llamada de emergencia generada a partir de la detección	88

ÍNDICE DE ANEXOS

Anexo A. Especificaciones técnicas de la Jetson Nano en video	102
Anexo B. Especificaciones técnicas DVR HiLook.....	103
Anexo C. Elección de Python	104
Anexo D. Arquitectura de Yolov5	105
Anexo E. Montaje del sistema CCTV.....	106

RESUMEN EJECUTIVO

El aumento continuo de la inseguridad en Ecuador se refleja cada año a través de diversos actos delictivos, siendo los asaltos y secuestros con armas de fuego en hogares el foco principal de este análisis. En este marco, se llevó a cabo una investigación orientada a diseñar un sistema basado en técnicas de machine learning para la identificación de armas de fuego utilizadas en asaltos a hogares. Generar alertas tempranas es el objetivo primordial del sistema, con el fin de acelerar la intervención de las autoridades correspondientes.

El sistema está compuesto por las etapas de adquisición, procesamiento, y visualización. En la etapa de adquisición de datos, se emplea un Circuito Cerrado de Televisión, conformado por 2 cámaras de 2MP, y un Grabador de Video Digital marca HiLook para la obtención de frames en tiempo real. Los datos captados son enviados al microordenador NVIDIA JETSON NANO para su posterior procesamiento. El sistema se desarrolló en Python y YOLOv5 fue el modelo seleccionado como algoritmo de visión artificial, este es el encargado de procesar los frames adquiridos, obteniendo como resultado la identificación de armas de fuego en asaltos o secuestros dentro de hogares. La visualización de los elementos identificados y las alertas generadas son emitidas a través de una llamada telefónica proporcionando una descripción detallada de la situación y objeto identificado, además de una alerta sonora incorporada.

Es importante resaltar que los datos son analizados y transmitidos de forma inmediata, garantizando una respuesta rápida frente a posibles actos delictivos. Como resultado de las pruebas realizadas, se alcanzó una efectividad del 85%.

Palabras clave: Robos, secuestros, machine learning, YOLOv5, armas

ABSTRACT

The continuous increase in insecurity in Ecuador is reflected annually through various criminal activities, with armed assaults and kidnappings in homes being the main focus of this analysis. In this context, a study was conducted aimed at designing a system based on machine learning techniques for the identification of firearms used in home assaults. The primary goal of the system is to generate early alerts to expedite the intervention of the relevant authorities.

The system comprises three stages: acquisition, processing, and visualization. In the data acquisition stage, a Closed-Circuit Televisión consisting of two 2MP cameras and a Hilook brand Digital Video Recorder is used to capture frames in real time. The captured data is sent to the NVIDIA JETSON NANO microcomputer for further processing. The system was developed in Python, and YOLOv5 was selected as the artificial vision model responsible for processing the acquired frames, resulting in the identification of firearms in home assaults or kidnappings. The visualization of the identified elements and the generated alerts is carried out through phone calls and spam messages, providing a detailed description of the situation and the identified object, along with an integrated audible alert.

It's important to highlight that the data is analyzed and transmitted immediately, ensuring a fast response to potential criminal acts. As a result of the tests conducted, the system achieved an effectiveness of 85%.

Keywords: Robberies, kidnappings, machine learning, YOLOv5, firearms.

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

SISTEMA CCTV CON ALERTA TEMPRANA Y RECONOCIMIENTO DE ARMAS DE FUEGO UTILIZANDO MACHINE LEARNING.

1.1.1 Planteamiento del problema

Mediante diferentes fuentes de información, se ha podido constatar la amplitud del problema relacionado con los robos y asaltos a hogares y locales comerciales, el Instituto Nacional de Estadísticas y Censos, detalla los tipos de delitos más comunes en Ecuador, indicando que los robos a domicilios han aumentado un 20% desde el 2023 hasta lo que va del 2024, también indica el considerable aumento de personas detenidas por posesión de armas de fuego sin permiso. Estas recopilaciones de datos estadísticos evidencian el problema de seguridad por el cual está atravesando el país lo cual requiere una solución pronta y oportuna. [1]

Los datos presentados por la Universidad Técnica de Ambato, en el informe denominado “Delitos contra la propiedad”, revelan información relevante acerca de hurtos, robos y horarios más comunes en la provincia de Tungurahua, estos datos están respaldados por la fiscalía general de Estado (FGE) del Ecuador. El informe detalla una variación mensual del 3.8%, y un 23.8% anual, relacionado al robo a propiedades. También muestra los horarios más comunes en los cuales se lleva a cabo este delito, indicando 23 delitos en las madrugadas como mínimo y 60 en las noches, como máximo en periodos mensuales. Estos datos demuestran la necesidad de mejorar las medidas existentes para combatir este delito. [2]

Los robos a viviendas se ubican en el tercer lugar como el delito más común en la ciudad de Ambato, según el Observatorio Económico y Social de Tungurahua, teniendo como primer y segundo lugar el robo a personas (sin darse cuenta) y los robos de partes de vehículos, con datos obtenidos en el periodo 2020 – 2022. Los puntos donde se comete con mayor frecuencia el tercer delito más común de la ciudad son los sectores Letamendi, Techo Propio y Huachi Chico. Este informe también detalla la

confianza de las víctimas de robos hacia la policía nacional, 5 mostrando un promedio de índice de decrecimiento que bordea los 55 puntos sobre 100, lo cual no es muy convincente y requiere una solución para mejorar la seguridad de los ciudadanos. [3]

La implementación de circuitos cerrados de televisión, cámaras de seguridad o sistemas de video vigilancia, a controlado una parte de la ola de delincuencia por la cual atraviesa el país actualmente, pero la falta de actuación de las fuerzas del orden ha causado que las personas busquen más alternativas con respecto a su seguridad y la de sus negocios, locales comerciales u hogares.

Esta brecha en la capacidad de respuesta de las fuerzas del orden ha llevado a un aumento en la demanda de alternativas de seguridad más efectivas por parte de los ciudadanos y propietarios de negocios. La sensación de vulnerabilidad ante la amenaza constante de la delincuencia ha impulsado la búsqueda de soluciones más avanzadas y proactivas que no solo proporcionen vigilancia pasiva, sino que también permitan una detección temprana y una respuesta inmediata ante cualquier situación de riesgo.

En este contexto, surge la necesidad de explorar e implementar soluciones de seguridad más avanzadas y proactivas que integren tecnologías innovadoras, como el reconocimiento de armas y alerta temprana mediante machine learning, en los sistemas de video vigilancia existentes. Estas soluciones tienen el potencial de mejorar significativamente la capacidad de respuesta rápida ante la delincuencia, proporcionando una mayor tranquilidad y protección tanto para los individuos como para las comunidades en su conjunto.

1.2 Antecedentes investigativos

Como resultado de la búsqueda de referencias en artículos científicos y trabajos de titulación, en los repositorios de diferentes instituciones dentro y fuera del país, de la misma manera en bases de datos en línea de artículos científicos, se presentan los siguientes antecedentes investigativos que aportan al tema de investigación:

En 2019, David Orlando Romero Mogrovejo llevó a cabo una investigación titulada "DESARROLLO DE UN SISTEMA DE DETECCIÓN DE ARMAS DE FUEGO CORTAS EN EL MONITOREO DE VIDEOS DE CÁMARAS DE SEGURIDAD"

en la Universidad Politécnica Salesiana de Cuenca, dado que los sistemas de monitoreo actuales basados en cámaras de seguridad se han convertido en un requisito operacional considerado esencial en el ámbito de la seguridad y protección de entidades comerciales, pero que su funcionamiento está basado en la observación humana mediante el uso de cámaras de seguridad, para evitar eso, el autor empleó técnicas de Aprendizaje Profundo (Deep Learning), y creando una base de datos a partir de la recopilación de imágenes y frames de videos en donde esté presente una arma de fuego corta. Se evaluó el modelo implementado mediante diferentes métricas de evaluación sobresaliendo la arquitectura ZF NET con 3 configuraciones diferentes, adicionalmente realizó una integración del modelo en un prototipo de software. Los resultados obtenidos evidencian que el sistema de detección posee valores de recall y precisión altos en sus detecciones dando un 87% de precisión en detección de armas cortas y un 2% de pérdidas, al igual que la relación o cruce entre las proporciones de falsos positivos y falsos negativos, dando como resultado 0.09 a una sensibilidad de 0.52. [4]

La investigación realizada por Fernando Patricio Oñate Miranda, en su investigación “DISEÑO Y CONSTRUCCIÓN DE NODOS INTELIGENTES PARA DETECCIÓN DE ARMAS DENTRO DE UNA RED DE VIDEOVIGILANCIA UTILIZANDO VISIÓN ARTIFICIAL”, en la Escuela Superior Politécnica de Chimborazo, en el año 2020, se enfocó en la implementación de un sistema de detección de armas de un ambiente video-vigilado, con la integración de nodos inteligentes, cada uno siendo capaz de captar información, procesarla y posteriormente enviarla hacia una base de datos. Con la recopilación de imágenes para el entrenamiento de las cámaras, creación de vectores y clasificación de los mismos, el autor continuó con la implementación del sistema. Los resultados obtenidos denotan detecciones exitosas de armas de fuego a plena luz del día, con un porcentaje de 98.33% de efectividad, siendo el valor más alto, y un 93.33% de efectividad, en días nublados. Mientras, que un valor de 85% de efectividad en un 9 día lluvioso, pero considerando la detección de armas dentro de un automóvil como nodo más vulnerable. [5]

Otra mención relevante es la propuesta por Alfonso Durán Caicedo, en el año 2022 con el tema “DETECTOR DE PERSONAS CON ARMAS DE FUEGO A PARTIR DE UN SISTEMA DE VISIÓN ARTIFICIAL BASADO EN EL ANÁLISIS DE

POSTURAS CORPORALES”, hace referencia a la inseguridad en un entorno o conjunto en general, considerando las armas de fuego como objetos principales en ese tipo de delitos, mediante el proyecto de investigación realizado por el autor, el cual realizó un sistema que basado en visión artificial que pueda mediante algoritmos Deep Learning y herramientas de posicionamiento del cuerpo como OpenPose reconocer una persona con un arma de fuego y la clasificación de los objetos o armas de fuego se implementaron con redes neuronales convolucionales (CNN). Las métricas manejadas para cada prueba fueron “Gun” y “Person with gun”, logrando porcentajes de 0.817 y 0.840, esto indica una tasa de error del sistema de 0.023, dando un equivalente de 96% de eficiencia resultante. De esta forma el sistema permitió confirmar la actividad peligrosa con arma de fuego mediante detecciones obtenidas a través de videos en tiempo real. [6]

El trabajo de titulación realizado por Aaron Schcolnik Elias, del Instituto Tecnológico de la Paz en agosto del 2023, se enfoca en la “DETECCIÓN DE ARMAS TIPO PISTOLA MEDIANTE EL USO DE REDES CONVOLUCIONALES CON UNA ARQUITECTURA TIPO YOLO Y ESTEREOSCOPIA”, se refiere a la manipulación constante del personal de seguridad en los sistemas de videovigilancia para la detección de sucesos o actividades delictivas, haciendo énfasis en la necesidad de automatización de dichos sistemas conformados por cámaras. Desarrolló un algoritmo basado en redes convolucionales (CNN), el cual detecta pistolas, este algoritmo se basa en el aumento de datos para el aumento del rendimiento, al igual que mejorando características de precisión y velocidad del mismo. El uso de OpenCV para en entrenamiento y procesamiento con técnicas de redimensionamiento, normalización, recolección corrección y reducción del color, dan como resultado una precisión del 0.922 y un recall 0.815, tomados como promedio de 741 imágenes procesadas y 783 instancias, siendo así indicadores 10 suficientes para obtener el 95% de efectividad del sistema, de esta forma siendo fiable al momento de la detección de armas y movimientos corporales en situaciones delictivas. [7]

En 2024, en la Universidad Técnica de Ambato, Sandoval Robayo Erick Fabián desarrolló un estudio titulado “SISTEMA DE DETECCIÓN DE SITUACIONES DELICTIVAS EN ESTABLECIMIENTOS COMERCIALES USANDO

INTELIGENCIA ARTIFICIAL”, en donde debido al incremento constante de la inseguridad en Ecuador que se manifiesta anualmente mediante una variedad de delitos, siendo los robos con armas en establecimientos comerciales, llevó a cabo un estudio con el propósito de desarrollar un sistema basado en inteligencia artificial para la detección de situaciones delictivas en locales comerciales. El fin principal de este sistema fue generar alertas ante posibles asaltos, con la finalidad de agilizar la respuesta por parte de las autoridades competentes. Este sistema fue hecho con las etapas de adquisición de datos, procesamiento, y visualización, dando como resultado el reconocimiento de los indicadores asociados al robo con armas. En función a las armas de fuego, indica una precisión de 0.94, sensibilidad de 0.89 y exactitud de 0.85, siendo un promedio de 84% de efectividad del sistema, por ende, ese mismo porcentaje para la eficacia en detección de armas de fuego. [8]

1.3 Fundamentación teórica

1.3.1 Sistemas de seguridad

Es un sistema integral de medidas técnicas y procedimientos diseñados para proteger la confidencialidad, integridad y disponibilidad de la información almacenada o transmitida en sistemas informáticos y redes. Combina diversas tecnologías y estrategias de seguridad, incluyendo la protección física y la gestión de datos, con el propósito de proporcionar una defensa sólida, coordinada y eficaz frente a riesgos y amenazas potenciales. [9]

a. Sistemas de monitoreo por video

La videovigilancia mediante circuitos cerrados de televisión basados en videos IP se refiere a una solución de seguridad la cual utiliza cámaras conectadas a una red de IP (Protocolo de Internet), esto para la captura de imágenes y videos en áreas específicas. Estas grabaciones se transmiten a través de la red, y las cámaras empleadas pueden ser tanto analógicas como digitales, con opciones de resolución estándar o alta definición. [10]

La Figura 1 detalla los elementos básicos que conforman un sistema de videovigilancia.

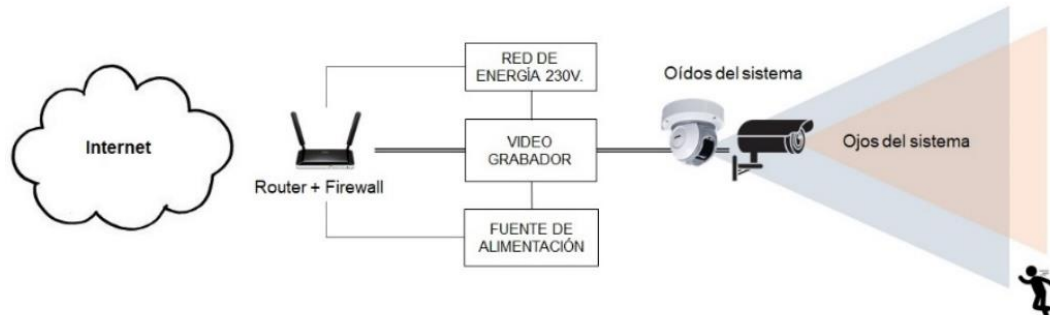


Figura 1. Sistema de seguridad. [10]

b. Sistema de vigilancia por video con detección de armas de fuego

Un detector de armas de fuego en un sistema de videovigilancia es un componente diseñado para identificar y generar alertas sobre la presencia de armas en imágenes o videos capturados por cámaras de seguridad. Utiliza algoritmos de procesamiento de imágenes y, en algunos casos, técnicas de machine learning, para analizar las características visuales de las armas y diferenciarlas de otros objetos en el entorno monitoreado. En el momento de detección de un arma de fuego, el sistema reproduce una alerta temprana, dando un aviso a las autoridades o a los encargados del orden para una pronta ante posibles amenazas.[11]

La Figura 2 muestra la detección de un arma corta tipo pistola en un sistema detector de armas de fuego.



Figura 2. Detección de arma de fuego. [11]

c. Alertas sonoras

Las alertas sonoras son dispositivos ubicados en zonas residenciales o comerciales que emiten avisos auditivos al detectar actividades sospechosas o situaciones de emergencia. Activadas por sensores de movimiento, sistemas de detección de intrusos u otros mecanismos de seguridad, su propósito es avisar a los residentes o al público circundante sobre posibles amenazas, promoviendo una respuesta rápida y colaborativa ante riesgos para la seguridad. [12]

La Figura 3 muestra una sirena comúnmente usada en sistemas de alertas comunitarias.



Figura 3. Sirena comunitaria. [13]

1.3.2 Sistemas CCTV

El propósito de este sistema es monitorear, gestionar y, cuando sea necesario, registrar la actividad física dentro de un espacio, local o entorno específico. Se conoce como circuito cerrado porque, a diferencia de la televisión convencional, su contenido es accesible únicamente para un grupo limitado y autorizado de usuarios, lo que garantiza un nivel superior de seguridad en su operación. [14]

Conformado de forma cerrada por cierto grupo de cámaras, conectadas en un nodo final denominado DVR, y siendo visualizado en un monitor para la visualización en tiempo real de las imágenes, tal y como lo ilustra en la Figura 4.

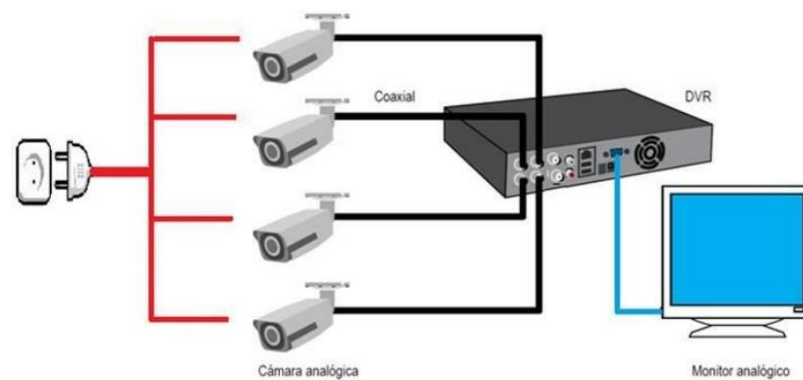


Figura 4. Sistema CCTV. [14]

a. Sistema CCTV con reconocimiento

Un sistema de circuito cerrado de televisión con alertas y reconocimiento es una solución de videovigilancia que combina tecnologías avanzadas de detección y reconocimiento para identificar en tiempo real eventos sospechosos o intrusos. [15]

La Figura 5 muestra la estructura básica de un sistema de video vigilancia integrando dispositivos y algoritmos de detección de personas, movimientos y hasta mascotas, esto mediante el acceso a internet del controlador y del DVR. [15]

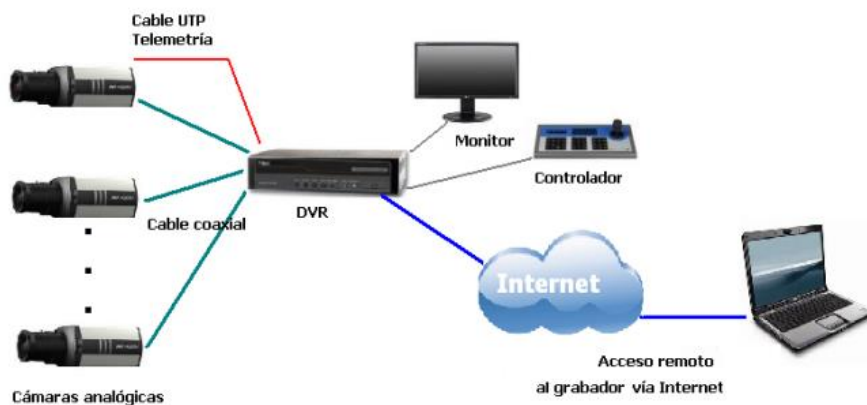


Figura 5. Estructura – Sistema CCTV con reconocimiento [15].

Este sistema emplea cámaras de seguridad conectadas a una unidad central de monitoreo que analiza constantemente las imágenes en busca de comportamientos o patrones predefinidos, como intrusiones o la presencia de armas. [15]

1.3.3 Tipos de sistemas CCTV

Los sistemas circuitos cerrados de televisión se clasifican en dos tipos según la tecnología empleada: analógicos y digitales. El sistema digital, también denominado CCTV IP, utiliza el protocolo TCP/IP para la transmisión de las señales de video. [16], [17]

a. CCTV digital

En los sistemas IP, al igual que en los analógicos, las cámaras convierten las variaciones de intensidad lumínica en señales eléctricas. Estas señales se digitalizan, se codifican mediante el protocolo Ethernet TCP/IP y se transmiten hacia un sistema de grabación en red (NVR) o un sistema de gestión de video. [17]

b. CCTV analógico

La señal de video analógica se produce al convertir las variaciones en la intensidad lumínica en señales eléctricas. Aunque la cámara digitaliza esta señal internamente, se reconvierte a formato analógico antes de su transmisión, permitiendo su recepción por un monitor o grabador de video mediante un cable físico. [17]

c. Componentes de un sistema CCTV

- ***DVR***

Un grabador de video digital (DVR) almacena las grabaciones localmente, generalmente en un disco duro, y admite tanto señales de video analógicas como fuentes digitales. Se conecta a cámaras analógicas mediante cables coaxiales y permite el acceso remoto al contenido. [18]

Los grabadores de video digital incluyen funciones avanzadas, como búsqueda de eventos en las grabaciones, organización por fecha y hora, y la capacidad de sobrescribir automáticamente las grabaciones más antiguas cuando el espacio de almacenamiento se agota. [18]

A continuación, en la Tabla 1 se observan las características principales y los modelos de grabadores de video digital que se encuentran disponibles en el mercado.

Tabla 1. Comparativa de los tipos de DVR. [19], [20], [21]

Marca	Características	Aspecto físico
HiLook 4 canales 104g-M1 1080p Lite Mini	- Clasificación de objetivos basado en aprendizaje profundo de Motion Detection 2.0 - Entrada de cámara IP de hasta 5 canales (hasta 2 MP) - Capacidad de codificación hasta 1080p Lite a 30 FPS.	
HiLook DVR-204G-K1(S) de 4 canales 120V	- Posee modos de grabación continuo y con detección de movimiento. - Resolución máxima de video de 1080p Lite - Capacidad de almacenamiento 6TB	
DVR Hikvision iDS-7208HQHI-M1/S/CHD de 8 canales	- Resolución Turbo HD, con resolución máxima de video 1080p y 8 canales disponibles. - Conectores de entrada AHD, CVBS, HDCVI, HDTVI, IP, RCA. - Capacidad de almacenamiento de 10TB.	

- **Cámaras de seguridad**

Las cámaras de seguridad cuentan con diversas funciones, como visión nocturna, detección de movimiento, grabación en alta definición y acceso remoto a través de internet, entre otras, para proporcionar una protección más eficiente. Pueden ser analógicas o digitales (IP), dependiendo de la tecnología utilizada para transmitir la señal. [22]

En la Tabla 2, se muestran modelos de cámaras de seguridad detallando sus características principales y precios.

Tabla 2. Modelos de cámaras de seguridad [23], [24], [25], [26]

Modelo	Características principales	Tipo de conexión	Aspecto físico
HiLook tipo tubo 2MP con audio color Vu	- Imágenes de alta calidad con resolución de 2 MP y 1920 × 1080. - Tecnología 3D DNR ofrece imágenes nítidas y limpias - Hasta 20 m de distancia de luz blanca para imágenes nocturnas brillantes	Análoga	

	- Un puerto para cuatro señales conmutables (TVI/AHD/CVI/CVBS)		
HiLook domo fija para interiores 2MP con audio color Vu	- Imágenes de alta calidad, con resolución de 2MP, por 1920x1080. - Tecnología 3D DNR, ofrece imágenes limpias y nítidas. - Lente focal fijo de 2.8 mm y 3.6 mm - Un puerto para cuatro señales conmutables (TVI/AHD/CVI/CVBS)	Análoga	
Ezviz cs-H9C 2k doble lente rotación horizontal y vertical	- Posee lentes duales 2k y 2k - Co-detección para seguimiento automático inteligente de detección humana y vehículos. - Cobertura panorámica de 360 grados más visión nocturna. - Seguridad automática	Wifi	
Xiaomi cámara de vigilancia AW300 MBC20	- Posee Ultra HD 2k, sensor Ultra HD 3 MPX, con un lente F2.0, para mejor detalle de imagen. - Modo de visión por infrarrojos integrado, visión nocturna. - Certificación IP66. - Seguridad automática	Análoga/Wifi	

• **Material de instalación**

La instalación de un circuito cerrado de televisión está dada por materiales adicionales para la conexión y funcionamiento de este.

La Tabla 3 presenta los materiales básicos para la instalación y funcionamiento de un sistema CCTV.

Tabla 3. Materiales usados en un sistema CCTV. [15], [17]

Materiales	Descripción	Característica principal	Aspecto físico
Video Baluns	Permite transmitir la señal de video a largas distancias mediante UTP. En muchos de los casos su uso también se da para el paso de energía.	Polaridad positiva y negativa para señal y voltaje.	

Fuentes de alimentación	Proveen energía eléctrica a las cámaras, DVR, y otros componentes del sistema	Voltajes de 12V	
Conectores macho/hembra	Facilita la conexión entre cámaras, baluns, y fuentes de poder.	Polaridad para fuente y señal	
Cajetín sobrepuesto	Sirve para alojar las conexiones de baluns y conectores.	Material plástico	
Tapa ciega	Utilizada para cerrar espacios no utilizados en cajetines o cajas de conexiones	Material plástico	
Disco duro	Dispositivo de almacenamiento que guarda las grabaciones del DVR.	Diferentes capacidades de almacenamiento	
Cable UTP Cat6	Cable de par trenzado, es utilizado para la transmisión de video y datos.	Categoría 6	
Tornillos, tacos	Componentes utilizados para asegurar elementos del sistema en superficies como paredes o techos.	Metal, plástico	

1.3.4 Inteligencia Artificial

La inteligencia artificial (IA) es una disciplina dentro de la informática que se enfoca en el diseño de sistemas y programas que pueden llevar a cabo tareas que normalmente requieren capacidades humanas, como razonamiento, aprendizaje, percepción visual, reconocimiento de voz y toma de decisiones. Estos sistemas utilizan algoritmos y modelos matemáticos para procesar grandes cantidades de datos, aprender de ellos y realizar tareas de forma autónoma, adaptándose de manera eficiente a nuevas situaciones. [27]

a. *Procesamiento Digital de Señales e Imágenes*

El Procesamiento Digital de Señales e Imágenes (DSPI) se enfoca en el análisis, manejo y transformación de señales y datos visuales mediante el uso de técnicas y algoritmos computacionales. Al realizarse de manera digital, permite realizar tareas como filtrado, mejora de calidad, extracción de características y reconocimiento de patrones en señales de audio, imágenes y videos. Las aplicaciones del DSPI son vastas, abarcando áreas como telecomunicaciones, medicina, procesamiento de imágenes médicas, seguridad y vigilancia, entre otras. [28]

De una forma resumida se muestra el procesamiento de una señal en la Figura 6, denotando las diferentes etapas para las cuales pasa.

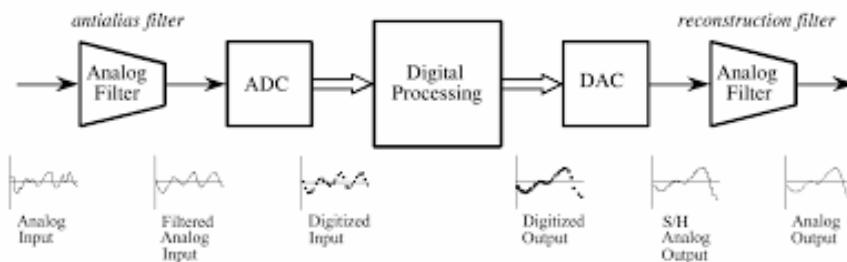


Figura 6. Diagrama del procesamiento de señales [28].

b. *Machine learning*

El Machine Learning, o Aprendizaje Automático, es una disciplina de la inteligencia artificial que se centra en el desarrollo de algoritmos y modelos computacionales capaces de aprender de datos previos para realizar tareas específicas sin una

programación explícita. Estos algoritmos son entrenados utilizando conjuntos de datos para identificar patrones, relaciones y regularidades, lo que les permite generalizar y tomar decisiones o hacer predicciones sobre nuevos datos no vistos anteriormente. El Machine Learning tiene aplicaciones en una amplia gama de campos, desde reconocimiento de voz y visión por computadora hasta análisis de datos y sistemas de recomendación. [29]

c. Técnicas de Machine Learning

Las técnicas de aprendizaje automático comprenden un conjunto diverso de herramientas y métodos diseñados para resolver una amplia variedad de problemas en este ámbito. Estas técnicas incluyen algoritmos especializados para tareas como clasificación, regresión, agrupamiento y detección de anomalías, entre otras. Asimismo, pueden adaptarse a distintos tipos de datos, ya sean estructurados, no estructurados o semiestructurados, y encuentran aplicaciones en múltiples campos, como la medicina, las finanzas, el marketing y la robótica. [29]

- ***Supervisado***

Esta es la modalidad más frecuente, donde los datos se presentan con etiquetas que guían al sistema para reconocer los patrones específicos que debe identificar. [29]

- ***No supervisado***

En este enfoque, los datos carecen de etiquetas, lo que exige que la máquina identifique y analice patrones de manera autónoma. Aunque es menos frecuente, su aplicabilidad es más limitada en comparación con otras técnicas. [29]

- ***Por refuerzo***

Un algoritmo adopta el enfoque de ensayo y error para aprender a cumplir un objetivo, evaluando diversas opciones y recibiendo recompensas o penalizaciones según qué tan bien se alineen sus decisiones con la meta establecida. [29]

- ***Aprendizaje profundo***

El deep learning permite a los modelos computacionales, compuestos por múltiples capas de procesamiento, adquirir representaciones de datos en distintos niveles de abstracción. Inspirado en la organización jerárquica del cerebro humano, este enfoque capacita a los modelos para identificar características complejas y llevar a cabo tareas avanzadas, como el reconocimiento de imágenes, el análisis de lenguaje natural y la predicción de patrones. Su habilidad para procesar grandes cantidades de datos lo ha posicionado como una tecnología fundamental en campos como la inteligencia artificial, el análisis de datos y la automatización. [30]

1.3.5 Métodos de aprendizaje

En el ámbito del aprendizaje automático, los métodos de aprendizaje abarcan algoritmos supervisados, no supervisados y de refuerzo. Entre las técnicas más comunes se encuentran los árboles de decisión, las redes neuronales, las máquinas de soporte vectorial, así como los métodos de agrupamiento y reducción de dimensionalidad. [31]

- a. Análisis del comportamiento del usuario***

Se centra en entender la forma en como interactúan las personas con productos o servicios digitales, evaluando los patrones y tendencias en sus acciones. [32]

- ***Identificación de patrones***

Implica la detección de regularidades y las estructuras en grandes volúmenes de datos, facilitando la predicción y comprensión de comportamientos futuros. [32]

- ***Algoritmos***

Es un conjunto de instrucciones definidas y finitas, diseñadas para resolver un problema específico o realizar una tarea particular. [32]

1.3.6 Modelos de detección

La detección de objetos es una pieza clave en la visión por computadora, porque permite no solo identificar sino también localizar objetos específicos dentro de imágenes o videos. A diferencia de la clasificación de imágenes, que se limita a asignar una etiqueta a la imagen completa, la detección de objetos proporciona información precisa sobre la posición y las dimensiones de cada objeto mediante cuadros delimitadores. [33]

En la Figura 7 se observa las capas por las cuales pasa la imagen de entrada para llegar a la detección final del objeto.

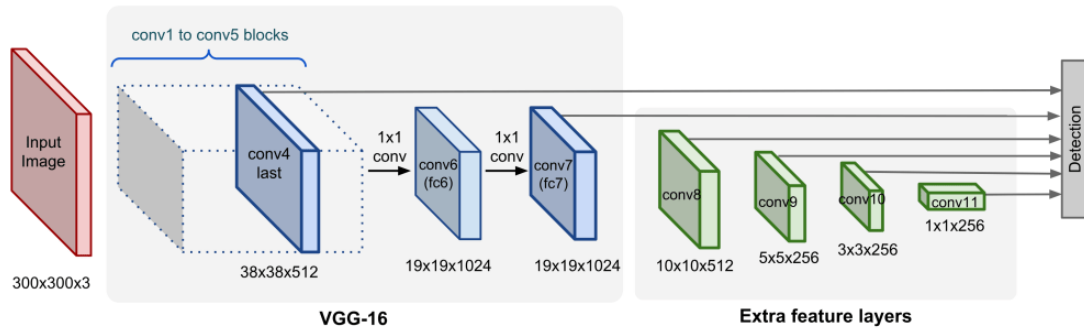


Figura 7. Capas de un modelo de detección.

En la Tabla 4 muestra los distintos modelos para la detección de objetos, detallando sus principales características.

Tabla 4. Comparativa de modelos de detección de objetos. [33], [34], [35]

Modelo	Descripción	Precisión (mAP)	Velocidad FPS	Recursos	Versiones Existentes
YOLO (You Only Look Once)	Realiza detección en una sola pasada sobre la imagen, prediciendo múltiples objetos a la vez.	~45-55%	30-60	Bajo (puede correr en GPU y CPU)	YOLOv3, YOLOv4, YOLOv5, YOLOv7, YOLOv8
Faster R-CNN (Region-based Convolutional Neural Networks)	Genera propuestas de regiones y luego clasifica y ajusta las regiones para detectar objetos.	~50-60%	5-7	Alto (requiere GPU potente)	Faster R-CNN (original), Mask R-CNN
SSD (Detector Multicaja de Disparo Único)	Similar a YOLO, pero predice objetos en múltiples escalas, haciendo más eficiente la detección en un solo paso.	~45-55%	20-30	Moderado (mejor en GPU)	SSD (original), SSD Mobilenet, SSD Inception
TensorFlow Object Detection API	Framework flexible para la detección de objetos que permite implementar diferentes arquitecturas (como SSD, Faster R-CNN, etc.).			Moderado a alto	Varios modelos como SSD, Faster R-CNN, EfficientDet, etc.

a. YOLO

YOLO es un modelo de detección de objetos en tiempo real que se basa en redes neuronales convolucionales (CNN). A diferencia de los métodos tradicionales, que segmentan la imagen en regiones o emplean múltiples etapas para identificar objetos, YOLO adopta un enfoque integral. Este modelo procesa la imagen de manera completa en una única pasada, lo que le permite detectar objetos y predecir su ubicación con gran velocidad y precisión. [36]

En la Tabla 5 se incluye un análisis comparativo de las diferentes versiones de YOLO, desde sus primeras implementaciones con YOLOv3 hasta las innovaciones más recientes, resaltando sus características clave, fortalezas y limitaciones.

Tabla 5. Versiones de Yolo. [36], [37]

Versión	Año	Características	Precisión (mAP)	Velocidad FPS	Tamaño del modelo	Arquitectura principal
YOLOv3	2018	Uso de FPN (Feature Pyramid Network). Detección multi-escala.	~57 % (COCO)	~30-50	~250 MB	Darknet-53
YOLOv4	2020	Implementación optimizada para hardware.	~65.7 % (COCO)	~60	~250 MB	CSPDarknet-53
YOLOv5	2020	Modelo ligero y rápido. Facilidades para entrenar y exportar.	~67 % (COCO)	~140	4-90 MB Según la variante	PyTorch, enfoque modular
YOLOv7	2022	Introducción de "Extended Efficient Layer Aggregation".	~56.8 % (COCO)	~160	~74 MB	E-ELAN, basándose en PyTorch
YOLOv8	2023	Mejora en arquitectura, facilidad de uso con ultralytics.	~57.5 % (COCO)	~170	20-100 MB	Basado en PyTorch

1.3.7 Softwares de etiquetado

El etiquetado de imágenes es un paso crucial en la creación de datasets para algoritmos de detección de objetos, puesto que proporciona los datos de entrenamiento indispensables para que el modelo aprenda a reconocer y localizar objetos en imágenes. Este proceso consiste en identificar y delimitar manualmente las áreas de interés en cada imagen, asignándoles una categoría o etiqueta que describa el objeto correspondiente.

a. LabelImg

Herramienta de código abierto diseñada para el etiquetado manual de imágenes. Destaca por ser ligera, intuitiva y eficiente, lo que la convierte en una opción ideal para generar anotaciones en formatos compatibles con YOLO y Pascal VOC. [38]

Principales características [38]:

- Compatibilidad con formatos XML (Pascal VOC) y TXT (YOLO).
- Interfaz gráfica sencilla y funcional.
- Disponible para Windows, macOS y Linux.

b. Roboflow

Plataforma basada en la nube que facilita la gestión integral de datos para proyectos de visión por computadora. Además de incluir herramientas para el etiquetado de imágenes, ofrece capacidades avanzadas de preprocesamiento, organización y exportación en una variedad de formatos. [39]

Principales características [39]:

- Soporte para formatos populares como YOLO, Pascal VOC y COCO.
- Posibilidad de etiquetado colaborativo en imágenes y videos.
- Funciones avanzadas de preprocesamiento, incluyendo ampliación de datos.

c. Labelme

Desarrollada por el MIT, es una herramienta de anotación diseñada para proyectos que requieren segmentación precisa de imágenes. Permite crear etiquetas poligonales que delimitan con exactitud las áreas de interés. [40]

Principales características [40]:

- Generación de anotaciones poligonales para segmentación semántica.
- Compatibilidad con formatos como JSON.
- Ideal para proyectos que demandan alta precisión en la definición de los contornos de los objetos.

1.3.8 Microcomputador

Es una computadora compacta que emplea un único microprocesador como unidad central de procesamiento (CPU). A diferencia de los mainframes o computadoras de gran tamaño, los microcomputadores son accesibles para uso personal o en pequeñas empresas. Su bajo costo, tamaño reducido y capacidad para realizar tareas complejas los convierten en la opción ideal para actividades cotidianas como navegación web, procesamiento de texto, programación y sistemas embebidos. [41]

a. NVIDIA Jetson Nano

Es una computadora diseñada para aplicaciones de Inteligencia Artificial, optimizada para ejecutar modelos entrenados de Deep Learning o Machine Learning en visión artificial. Con varios puertos y una GPU integrada, funciona como una PC, siendo ideal para proyectos de visión artificial. No incluye un sistema operativo, permitiendo al usuario elegir e instalar uno desde una memoria SD. Su arquitectura la hace perfecta para aplicaciones en automatización, domótica y más. [42]

Se puede observar en la Figura 8 se puede observar el microcomputador NVIDIA Jetson Nano sin su case ni componentes físicos.



Figura 8. NVIDIA Jetson Nano. [42]

b. Raspberry Pi 5

La Raspberry Pi 5, con arquitectura ARM Cortex-A76 y un procesador de cuatro núcleos a 2.4 GHz, ofrece conectividad avanzada, incluyendo puertos Gigabit

Ethernet, USB 3.0 y 2.0, y HDMI dual 4Kp60 con soporte HDR. Además, integra Wi-Fi de doble banda y Bluetooth 5.0. Con opciones de RAM de 4 GB y 8 GB, y expansión de almacenamiento mediante una tarjeta microSD de alta velocidad, esta versión es ideal para proyectos de desarrollo, IoT, educación e IA, soportando sistemas operativos completos como Linux. [43]

En la Figura 9 se muestra la forma física de una raspberry Pi 5, sus puertos y pines.



Figura 9. Raspberry Pi 5. [43]

c. Coral Dev Board

Desarrollado por Google, este dispositivo está optimizado para ejecutar modelos de aprendizaje automático, siendo ideal para TensorFlow y otros frameworks de inteligencia artificial. Equipado con un TPU especializado en IA, 1 GB de RAM y 8 GB de almacenamiento interno, también ofrece conectividad WiFi y Bluetooth. Compatible con diversos entornos de desarrollo, especialmente los de Google, destaca por su eficiencia energética y su capacidad para ejecutar modelos de alto rendimiento, lo que lo convierte en una opción excelente para aplicaciones de IA. [44]

La Figura 10 muestra el aspecto físico de una Coral Dev Board.



Figura 10. Coral Dev Board. [44]

d. Comparativa microcomputadores

La Tabla 6 muestra las diferentes características técnicas de cada uno de los microcomputadores que son aptos para modelos de inteligencia artificial.

Tabla 6. Comparativa de microcomputadores

Característica	NVIDIA Jetson Nano	Raspberry Pi 5	Coral Dev Board
CPU	Quad-core ARM Cortex-A57 1.43 GHz	Quad-core ARM Cortex-A76 2.4 GHz	Quad-core ARM Cortex-A53 1.5 GHz
GPU	NVIDIA Maxwell (128 núcleos CUDA)	VideoCore VII	Edge TPU integrado (procesador IA)
Acelerador de IA	No (solo GPU)	No (GPU optimizada)	Sí (Edge TPU dedicado, 4 TOPS)
RAM	4 GB LPDDR4 (64-bit, 25.6 GB/s)	4 GB u 8 GB LPDDR4X (opcional)	1 GB LPDDR4
Almacenamiento	Tarjeta microSD	Tarjeta microSD + eMMC opcional	Tarjeta microSD
Conectividad USB	4x USB 3.0 tipo A, 1x micro-USB	2x USB 3.0, 2x USB 2.0	1x USB tipo C, 1x USB 3.0 tipo A
Ethernet	Gigabit Ethernet (10/100/1000 Mbps)	Gigabit Ethernet (10/100/1000 Mbps)	Gigabit Ethernet
Video Salida	HDMI 2.0 y DisplayPort 1.2	2x micro-HDMI (4K a 60 fps)	HDMI 2.0 (4K a 60 fps)
Video Procesamiento	H.264/H.265 (codificación y decodificación)	H.264/H.265 (codificación y decodificación)	No especificado
I/O GPIO	40 Pines GPIO (compatible con Raspberry Pi)	40 Pines GPIO	40 Pines GPIO
Cámara	Interfaz CSI (MIPI-CSI, 15 pines)	2x Interfaz CSI (MIPI-CSI, 22 pines)	1x Interfaz CSI
Aceleración de Visión	No	No	Sí (Edge TPU optimizado para visión)
Alimentación	5V (conector de barril o micro-USB)	5V USB-C (3A mínimo recomendado)	5V USB-C
Dimensiones	100 x 80 x 29 mm	85.6 x 56.5 mm	88 x 60 mm

Peso	~140 g	~38 g	~45 g
Sistema Operativo	NVIDIA JetPack (Ubuntu Linux)	Raspberry Pi OS (Debian-based)	Mendel Linux (Debian-based)
IA y Visión por Computadora	GPU CUDA para TensorFlow, PyTorch	Compatible con TensorFlow y PyTorch	Edge TPU (4 TOPS, optimizado para TensorFlow Lite)
Uso Típico	IA, Visión por computadora, robótica	IoT, computación general, proyectos DIY	IA en dispositivos, visión, IoT

1.3.9 Plataformas de entrenamiento

Son servicios basados en Internet que permiten a los usuarios almacenar, procesar y gestionar datos sin necesidad de infraestructura física local. Proporcionan recursos de computación, almacenamiento y redes a través de la web, brindando acceso a herramientas y aplicaciones avanzadas sin depender de hardware dedicado. [45]

Estas plataformas eliminan las limitaciones físicas y económicas de mantener una infraestructura propia, promoviendo la colaboración entre equipos. Asimismo, aprovechan la escalabilidad, seguridad y eficiencia que proporciona la tecnología en la nube. [45]

En la Tabla 7 se detalla las versiones gratuitas de plataformas para entrenamiento en la nube, y la versión de paga, pero con más beneficios y limitaciones.

Tabla 7. Plataformas de entrenamiento basadas en la nube. [45], [46]

Plataforma	Descripción	Características
Google Colaboratory	Plataforma en la nube gratuita que permite ejecutar código Python en notebooks sin requerir configuración local. Diseñada para facilitar la creación y ejecución de proyectos de aprendizaje automático, ofreciendo acceso a GPUs y TPUs.	• Es compatible con frameworks populares como TensorFlow, Keras, PyTorch, entre otros. • Ofrece acceso a GPUs y TPUs para acelerar el entrenamiento de modelos. • Permite la ejecución de código en un entorno colaborativo, donde varios usuarios pueden trabajar en el mismo notebook en tiempo real.
Kaggle	Kaggle es una plataforma en línea que ofrece herramientas para análisis de datos y creación de modelos de aprendizaje automático, apoyada por una activa comunidad de científicos de datos. A través de Kaggle Kernels, los usuarios pueden ejecutar código en Python y R en la nube sin necesidad de configurar un entorno local.	• Kaggle ofrece acceso gratuito a GPUs y TPUs para entrenar modelos, junto con una extensa colección de datasets públicos. • Es compatible con frameworks como TensorFlow, PyTorch y Keras. • Permite la colaboración en competiciones de datos, así como la compartición de notebooks, integrando código y datasets en un entorno de trabajo común.
AWS (Amazon Web Services)	AWS SageMaker es una plataforma completamente gestionada que facilita el desarrollo, entrenamiento y despliegue de modelos de aprendizaje automático. Ofrece herramientas para entrenar modelos en la nube con recursos computacionales avanzados, como GPUs, y es altamente escalable.	• Es compatible con frameworks populares como TensorFlow, PyTorch y MXNet, entre otros. • Facilita el entrenamiento de modelos de detección de objetos con soporte para GPU. • Incluye herramientas para visualización, preprocesamiento de datos y gestión de modelos.

1.3.10 Software de programación

Un lenguaje de programación es un sistema estructurado que permite a los desarrolladores crear instrucciones y algoritmos comprensibles para las computadoras. Funciona como un vínculo entre humanos y máquinas, facilitando la ejecución de tareas específicas y la resolución de problemas mediante el procesamiento automatizado. [47]

Es esencial para desarrollar y entrenar modelos inteligentes. Permite implementar algoritmos de aprendizaje, manejar grandes volúmenes de datos y realizar ajustes iterativos en los modelos. Este proceso incluye etapas clave como el preprocesamiento

de datos, la selección de características, el entrenamiento, la validación y el despliegue, todo facilitado por las capacidades avanzadas de los lenguajes modernos. [47]

a. Comparativa de softwares de programación

La Tabla 8 detalla las características correspondientes para cada lenguaje de programación, enfocándose en 4 softwares, los cuales son Python, JavaScript, Matlab y Lisp, siendo estos los más usados en el ámbito de machine learning.

Tabla 8. Comparativa entre softwares de programación. [47], [48], [49]

Característica	Python	JavaScript	Matlab	Lisp
Sintaxis	Simple, clara, fácil de aprender; basada en indentación	Basada en C, requiere comprensión del DOM y manejo de eventos	Sintaxis matemática, orientada a matrices	Sintaxis basada en listas, orientada a recursión
Bibliotecas y frameworks	TensorFlow, Keras, PyTorch, Scikit-learn, XGBoost, etc.	TensorFlow.js, Brain.js, Synaptic (limitadas en comparación)	Toolboxes de Matlab como Statistics and Machine Learning Toolbox, Deep Learning Toolbox	Existen implementaciones como CLML, pero no tan populares
Facilidad de uso	Alta	Moderada	Moderada	Baja
Aplicaciones comunes	Ciencia de datos, IA, análisis de datos, automatización, desarrollo web.	Desarrollo web, aplicaciones interactivas en navegador, ML para la web.	Investigación científica, simulaciones numéricas, ingeniería.	Inteligencia artificial simbólica, investigación en computación.
Integración de sistemas	Integración con otras tecnologías, sistemas, y bases de datos.	Integración sencilla con otros lenguajes a través de APIs web.	Se integra bien con sistemas de ingeniería y científicos, pero más cerrada.	Integración limitada fuera de aplicaciones especializadas
Rendimiento	Bueno, con bibliotecas optimizadas (TensorFlow, PyTorch).	Menor rendimiento para ML, pero adecuado para aplicaciones web.	Muy bueno en cálculos numéricos, optimizado para álgebra matricial.	Bajo rendimiento comparado con lenguajes.
Desempeño en ML	Ampliamente utilizado para modelos complejos.	Enfocado a proyectos en el navegador o de pequeña escala.	Menos flexible para deep learning	No es una opción común para el ML.
Implementación de modelos	Rápido con APIs de alto nivel y bibliotecas como Scikit-learn y Keras.	Requiere comprender cómo trabajar con ML en la web y	Fácil en entornos científicos y académicos, pero limitado	Requiere construir muchos algoritmos desde cero sin bibliotecas modernas.

		bibliotecas limitadas.	para modelos más generales	
Soporte para deep learning	Bibliotecas como TensorFlow, Keras, y PyTorch dominan el campo.	TensorFlow.js permite crear y ejecutar modelos en el navegador.	Con Deep Learning Toolbox, pero menos popular.	Limitado, no tiene soporte moderno para deep learning.

1.3.11 Situación actual de seguridad

Ecuador atraviesa una grave crisis de seguridad que se ha agudizado en los últimos tres años. Aunque hace cinco años era considerado un referente en la región, en 2023 alcanzó niveles históricos de violencia, con una tasa de homicidios de 44,9 por cada 100.000 habitantes y un promedio diario de 21 asesinatos. [50], [51]

a. Robos a viviendas

Una de las mayores inquietudes de la comunidad es el aumento de la violencia por parte de los delincuentes, quienes emplean tácticas engañosas, como hacerse pasar por policías, para irrumpir en los hogares de sus víctimas. Esta situación genera un clima de inseguridad y temor entre los residentes, quienes se sienten cada vez más vulnerables ante estos actos delictivos. Además, la sofisticación de los métodos utilizados por los criminales dificulta la prevención y la protección efectiva de las viviendas. [52]

En la Figura 11 se observa la cantidad de robos suscitados a partir del año 2022, hasta el 2024, demostrando una reducción de este delito en casi un 5%



Figura 11. Cifras de robos a personas con armas de fuego 2022 – 2024

La Figura 12 muestra una reducción de cantidad con respecto a los robos a hogares, teniendo un índice alto en el año 2022, y reduciendo en aproximadamente 3000 casos para el año 2024.



Figura 12. Cifras de robos a hogares 2022 - 2024

1.3.12 Armas de fuego

La presencia de armas de fuego y su impacto en la sociedad es un tema ampliamente debatido. No obstante, la conexión entre estas armas y el crimen organizado influye de manera constante en la convivencia pacífica de la población. Este estudio tuvo como objetivo entender los factores y condiciones que llevaron a que las armas de fuego se convirtieran en el instrumento preferido para cometer actos violentos y delictivos en Ecuador [53].

a. Tipos de armas de fuego

Las armas de fuego, diseñadas para lanzar proyectiles mediante la combustión de pólvora, han evolucionado desde los antiguos arcabuces hasta las modernas armas automáticas. Si bien son utilizadas en actividades legítimas como defensa personal, deporte y caza, también están estrechamente vinculadas a la violencia en regiones con alta criminalidad. [54]

La Tabla 9 muestra la clasificación de las armas de fuego existentes y más usadas para el cometimiento de delitos.

Tabla 9. Tipos de armas de fuego. [54]

Tipos de armas					
Armas cortas			Armas largas		
Tipo	Característica	Figura	Tipo	Característica	Figura
Revolver	Estas armas tienen varias cámaras de disparo, también conocidas como "orificios de		Escopetas	Es un arma de fuego de cañón largo diseñada para disparar múltiples proyectiles a corta distancia.	

	carga", dispuestas en un cilindro giratorio.				
Pistola	Son armas de fuego pequeñas que se pueden manejar con una sola mano.		Fusiles	Arma de fuego de cañón largo que se usa para disparar a blancos lejanos con precisión.	
Artesanales	Son fabricadas de manera ilegal utilizando maquinaria industrial, sin estar registradas.		Metralletas	Es un arma automática que dispara ráfagas continuas de proyectiles con alta cadencia de fuego.	

b. Presencia de armas en el Ecuador

El decomiso de armas de fuego en el país ha demostrado el problema que atraviesa el país, dejando en evidencia las cantidades de armas que manejan grupos delincuenciales para cometer actos vandálicos dentro de hogares y personas en general [50].

En la Tabla 10 se muestra las cantidades de armas de fuego decomisadas por las fuerzas armadas en cada año.

Tabla 10. Cantidad de armas de fuego decomisadas. [55]

	Año	Cantidad
Armas decomisadas por las Fuerzas Armadas del Ecuador	2020	1372
	2021	2476
	2022	4612
	2023	8008

De las cuales, en su mayoría, aquellas armas decomisadas son de carácter letal (metralletas, escopetas, rifles).

c. Vulnerabilidad de los hogares

Los estándares económicos altos y un nivel de seguridad bajo o promedio, son factores que pueden hacer vulnerables a un hogar ante la crisis de delincuencia existente, enfocándose en el ámbito económico para un posible ataque de grupos delincuenciales.

Según el INEC, 4 de cada 10 hogares son víctimas de robo a la vivienda, la Tabla 11 detalla las ciudades con mayor índice de robo a viviendas en Ecuador.

Tabla 11. Ciudades con índice de robos a hogares más alto. [55]

Delito		%
Robos a viviendas	Morona Santiago	8.43
	Napo	8.25
	Pastaza	8.23
	Orellana	6.75

d. Vulnerabilidad de personas

Independientemente de la situación en la que se encuentre alguna persona, siempre estará expuesta a un atraco por parte de algún antisocial. Esto evidencia un número muy alto con respecto a robos a transeúntes, ya sea por zonas habitadas o zonas de escaso transitar.

En la Tabla 12 se muestra las ciudades donde las personas han sido víctimas de un robo o asalto.

Tabla 12. Ciudades con índice más alto de robos a personas [55].

Delito	Ciudad	%
Robo a personas	Guayas	18.29
	Pichincha	17.19
	Azuay	16.04
	Santo Domingo de los Tsáchilas	14.71
	Imbabura	13.82
	El Oro	13.29

Demostrando que 14 de cada 100 personas han sido víctimas del delito de robo a personas, según el INEC.

e. Confianza de las personas en entidades del orden

La cantidad de personas que presentan su confianza antes las autoridades del orden son baja considerando un 100% total, debido a que estas presentan temor por represalias por parte de aquellos antisociales que cometieron el delito. También influencia bastante el objeto que fue sustraído, independientemente de cómo haya sido perpetrado el delito.

En la Tabla 13 se muestra el porcentaje de personas que tienen la confianza de denunciar ciertos delitos con la espera de una respuesta pronta.

Tabla 13. Porcentaje de confianza de las personas para denunciar delitos [55].

Característica	Delito	Porcentaje
Confianza de las personas para denunciar delitos de los que fueron víctimas	Robo a viviendas	24.43%
	Robo a personas	15.68%

1.4 Objetivos

1.4.1 Objetivo general

Implementar un sistema detector de armas de fuego con alerta temprana aplicado a un sistema CCTV mediante machine learning.

1.4.2 Objetivos específicos

- Analizar la situación actual de los robos y asaltos con armas de fuego en hogares y locales comerciales.
- Determinar los componentes de hardware y software que formen parte del sistema detector de armas y alerta temprana.

- Seleccionar una técnica de machine learning para la detección de armas de fuego.
- Comprobar la funcionalidad del sistema CCTV con detección de armas de fuego en un entorno de prueba controlado.

CAPÍTULO II. METODOLOGÍA

2.1 Materiales

En el marco del proyecto, se desarrolló un sistema de detección de delitos en hogares utilizando técnicas de machine learning, basado en el lenguaje de programación Python. Para la detección de imágenes, se empleó la arquitectura YOLOv5, mientras que para el etiquetado de estas se utilizó la herramienta online Roboflow. El proceso de entrenamiento se realizó con un conjunto de datos propio compuesto por 8,100 imágenes en la plataforma en la nube Kaggle. Para el procesamiento de los datos, se utilizó una NVIDIA Jetson Nano y para la monitorización, un sistema CCTV. Para la alerta temprana se empleó una llamada telefónica mediante un módulo GSM800L y una alerta sonora que se desarrolla mediante una sirena.

2.2 Métodos

2.2.1 Modalidad de la investigación

A continuación, se presentan los diversos tipos de investigación utilizados en la realización del proyecto de investigación.

La presente investigación se definió como aplicada, porque se concentra en la aplicación práctica de técnicas de machine learning para el reconocimiento de armas de fuego en entornos vigilados por sistemas CCTV. Esta modalidad se centró en la recolección y análisis de información relevante sobre algoritmos de machine learning, así como su implementación y validación en la detección temprana de amenazas con armas de fuego en tiempo real.

Asimismo, se empleó la modalidad de investigación bibliográfica para revisar exhaustivamente artículos científicos, investigaciones previas, y otros recursos disponibles en repositorios universitarios y bases de datos, con el objetivo de establecer un fundamento teórico sólido y obtener datos relevantes sobre el reconocimiento de armas y la seguridad en sistemas de video vigilancia

El proyecto se llevó a cabo como una investigación de campo, debido a que implicó la recopilación de datos, la grabación de videos para la obtención de frames y la construcción de una base de datos eficiente, así como la implementación del sistema en un hogar que ha registrado antecedentes de asalto y robo con armas de fuego.

2.2.2 Recolección de información

Con el propósito de recopilar información pertinente para el progreso de la investigación, se emplearán diversas fuentes confiables y verificadas. Se llevó a cabo la búsqueda en repositorios universitarios tanto a nivel nacional como internacional, además de revisar libros y revistas científicas bajo la supervisión del tutor académico.

Tabla 14. Recolección de información

Criterio	Descripción
¿Para qué?	Identificar armas de fuego
¿De qué personas u objetos?	Personas propietaria s de hogares
¿Cómo?	Mediante técnicas de machine learning
¿Con qué?	A través de sistemas CCTV
¿Cuándo?	Durante el periodo académico septiembre – diciembre 2024
¿Dónde?	En la ciudad de Ambato, Ecuador

2.2.3 Procesamiento y análisis de datos

Para garantizar un desarrollo óptimo del sistema, es importante tener en cuenta aspectos fundamentales en la selección del hardware y software adecuados para el proceso y análisis de datos. A continuación, se presentan los puntos esenciales a considerar en este aspecto.

- Revisión de la información recopilada.
- Recopilación de información sobre los algoritmos de machine learning para el reconocimiento de armas de fuego.

- Pruebas piloto sobre algoritmos de detección de armas implementados en una cámara de seguridad.
- Pruebas de funcionamiento de la alerta temprana al momento de la detección de armas.
- Incorporación y pruebas de funcionamiento del algoritmo de detección y alerta temprana en el sistema CCTV.
- Presentación de los resultados mediante datos estadísticos y de utilidad para la investigación.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

3.1 Desarrollo de la propuesta

Para el desarrollo de este proyecto, se comenzó con la selección de elementos que pueden ser identificados a través de sistemas CCTV instalados en hogares. A continuación, se definen los componentes de hardware y software requeridos para el sistema. YoloV5 fue seleccionado para el entrenamiento de detección de armas de fuego. Roboflow fue la herramienta que se utilizó para la etiqueta de datos y creación del dataset correspondiente. Mediante la plataforma Google Colab con YOLOv5 Ultralytics, se realizó el entrenamiento del modelo de detección de armas de fuego.

3.1.1 Definición de los elementos para la propuesta

Mediante un análisis estadístico de los crímenes en hogares en Ecuador, se obtuvieron datos relevantes en los cuales se detalla que, hasta febrero del 2024, el índice más alto de crímenes dirigidos hacia hogares se dio en 8 provincias de las 24, con más indicio en la serranía ecuatoriana. [55]

Hasta 2023, se reportaron más de 8,000 robos a domicilios en Ecuador, según datos proporcionados por la Policía Nacional. En Quito, aproximadamente el 87% de estos robos ocurrieron en casas, mientras que el 13% afectaron a departamentos. A nivel nacional, los robos a hogares representaron cerca del 5.7% de los delitos denunciados relacionados con la propiedad. [55]

Este tipo de crímenes abarca en su mayoría los robos armados a casas y hogares, incluyendo departamentos, también formando parte de esta lista los secuestros y con menor porcentaje las extorsiones. Este tipo de actos en su mayoría son realizados mediante el uso de armas de fuego, siendo las armas de fuego cortas más utilizadas para robos, extorsiones, y armas largas para secuestros.

Este aumento en los robos refleja un preocupante caso de inseguridad creciente en el país, afectado por el uso de tecnología avanzada en algunos crímenes.

3.1.2 Elementos por detectar en el sistema

La creciente inseguridad en Ecuador, marcada por un aumento de robos y delitos violentos, destaca el papel predominante de las armas de fuego en estos actos. Las armas cortas, como pistolas y revólveres, son las más utilizadas en robos y extorsiones por su fácil ocultación y maniobrabilidad, mientras que las armas largas, como fusiles y escopetas, suelen asociarse a crímenes de mayor magnitud, como secuestros. [55]

Ante este panorama, resulta imprescindible implementar un sistema de detección capaz de identificar con precisión tanto armas cortas como largas, emitiendo alertas tempranas para prevenir y reducir estos delitos. Este enfoque aborda una problemática crítica al priorizar el impacto de las armas de fuego en la seguridad ciudadana y en los delitos más frecuentes, contribuyendo a la protección de los hogares en el país. [55]

a. Tipos de armas

Este proyecto al estar enfocado directamente a la detección de armas de fuego se ha determinado la identificación de tipos de armas, clasificándolas como armas cortas y largas. [54]

En la Tabla 15 se observa los elementos a detectar a partir del sistema detector de armas de fuego, identificando correspondiente el tipo de arma según su clasificación.

Tabla 15. Tipos y clasificación de armas. [54]

Tipos de armas	
Armas cortas	Armas largas
Es un arma de fuego compacta y ligera diseñada para dispararse con una sola mano, ampliamente utilizada en defensa personal y actividades delictivas debido a su portabilidad.	Es un arma de fuego diseñada para operarse con ambas manos y apoyarse en el hombro al disparar, destacándose por su mayor tamaño, peso, alcance y precisión.

En la Tabla 9 se puede observar con más detalle los distintos tipos de armas, ya sean cortas o largas, describiendo y ejemplificando cada una, aparte su concepto y su característica principal.

b. Rostros cubiertos

En la mayoría de los actos delictivos captados por cámaras de seguridad de hogares, se ha podido presenciar que muchos de estos individuos optan por cubrir su rostro antes del cometimiento del delito.

Otros de los puntos clave en el sistema es la detección de personas sospechosas, enfocándose principalmente en la detección de rostros cubiertos, en la Tabla 16 se detalla las diferentes formas en las que un atacante puede cubrir su rostro para cometer algún delito.

Tabla 16. Diferentes formas de cubrir el rostro

Formas de cubrir el rostro		
Elemento	Característica	Figura
Pasamontañas	Es una prenda que cubre el rostro, utilizada frecuentemente por delincuentes para ocultar su identidad durante la comisión de crímenes	
Otros	El uso de vestimentas para cubrir el rostro, como bufandas o capuchas, es una práctica común en la delincuencia para ocultar la identidad de los criminales.	

3.1.3 Esquema general del proyecto

En la Figura 13, se muestra la arquitectura general del sistema detector de armas de fuego y alerta temprana, el cual consta de 4 etapas, adquisición, entrenamiento, procesamiento y detección e interpretación de datos. Recalcando que cada una de esta cuenta con subetapas o está conformada por procesos adicionales.

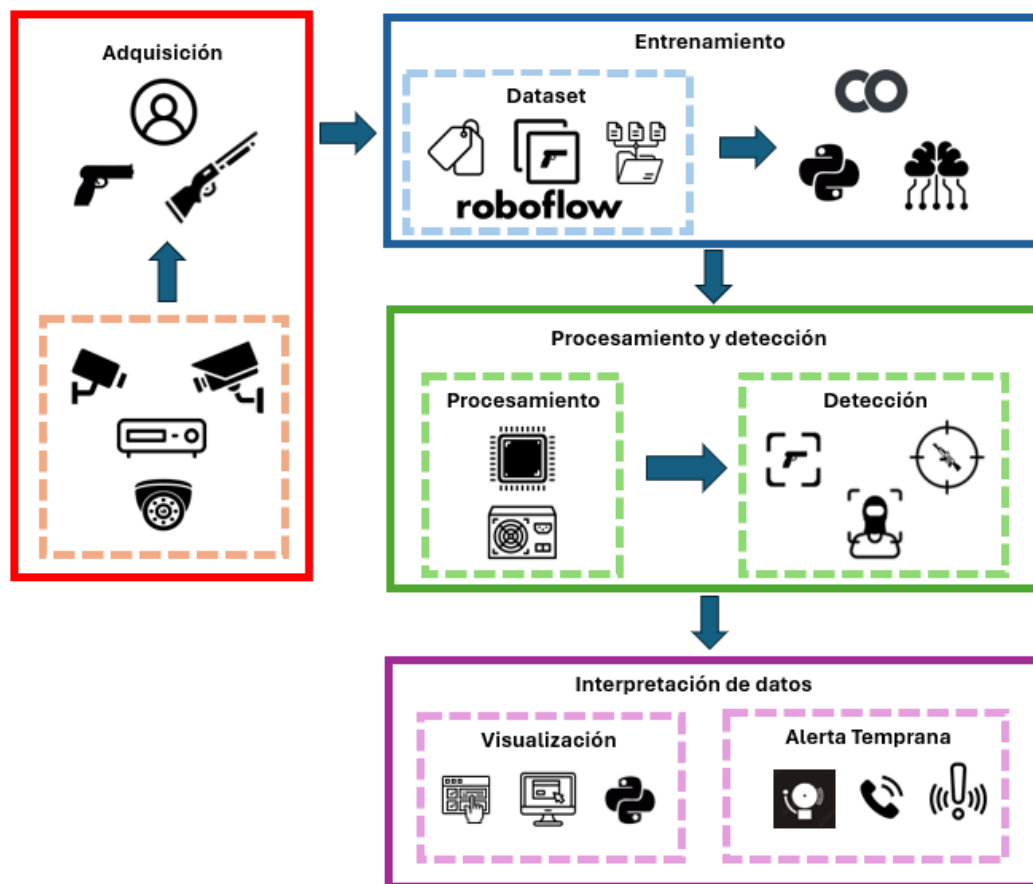


Figura 13. Arquitectura del sistema

a. Etapa 1: Adquisición

Para la obtención de imágenes necesarias para el sistema, se empleó un circuito cerrado de televisión o también denominado CCTV. Este sistema funciona en base a un conjunto de cámaras y a su fuente central llamada DVR para cámaras análogas o NVR para cámaras wifi, con grabaciones de largo tiempo prolongado y visualización en tiempo real de las áreas a monitorear, para que la adquisición de datos sea eficiente y efectiva.

b. Etapa 2: Entrenamiento

En la creación del Dataset, las imágenes que son obtenidas a través del sistema CCTV, son procesadas y etiquetadas para la posterior identificación de las armas, de tal manera que el conjunto de datos formado sea robusto para el entrenamiento del

modelo. Los recursos computacionales disponibles en la nube fueron utilizados para acelerar el proceso de entrenamiento de la red neuronal.

En este proceso, la red neuronal se optimiza y mejora al ser entrenada con un conjunto de datos amplio y representativo. Esto le permite desarrollar la capacidad de generalizar y detectar patrones complejos con gran precisión, lo que aumenta significativamente la eficiencia y confiabilidad del sistema.

La Figura 14 muestra la etapa de entrenamiento de forma más detallada, indicando la herramienta para la creación del dataset, la plataforma en la nube para el entrenamiento, y el modelo de entrenamiento resultante, en este caso YoloV5.

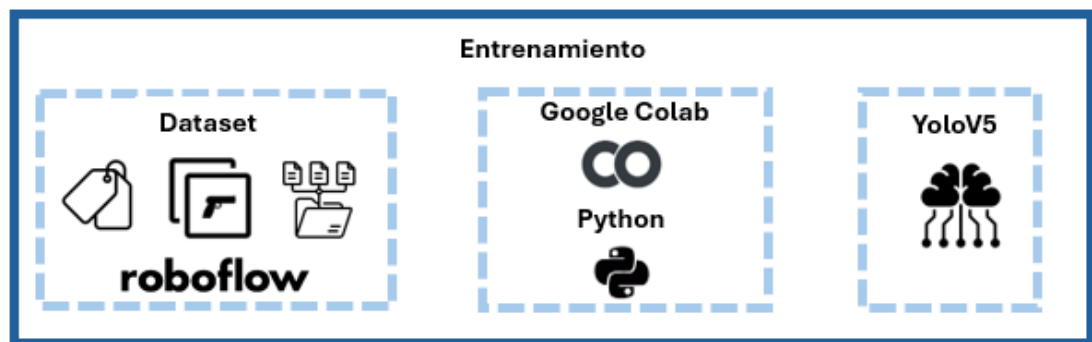


Figura 14. Etapa de entrenamiento detallada

c. Etapa 3: Procesamiento y detección

- **Procesamiento**

Una vez que se haya completado el entrenamiento, el modelo ya entrenado se desplegó en un microcomputador o también denominado dispositivo de computación de borde, el cual es adecuado esencialmente para aplicaciones relacionadas a la inteligencia artificial. Este dispositivo, con la capacidad de ejecutar algoritmos avanzados de aprendizaje profundo, facilita la realización autónoma y eficiente de tareas de visión por computadora. Garantiza un análisis en tiempo real, asegurando una respuesta ágil y dinámica en la detección de delitos, lo que refuerza la robustez y eficacia del sistema.

- ***Detección***

Cuando el modelo entrenado ya está implementado en el microcomputador, procede a la identificación de objetos en busca de aquellos similares a los que se utilizaron en el entrenamiento, en este caso armas de fuego, ya sean cortas o largas, de tal manera que al detectarlos resalta su ubicación en la imagen mediante un cuadro delimitador y una etiqueta la cual señala el objeto detectado, ya sea un arma corta, larga o un intruso. A partir de la detección, genera automáticamente una alerta que incluye una captura del cuadro con el arma detectada.

d. Etapa 4: Interpretación de datos

Esta etapa es la parte final y esencial del sistema, porque permite ver los datos obtenidos durante la detección en información práctica y aplicable.

- ***Visualización.***

El sistema presenta de manera clara y comprensible las detecciones realizadas, mostrando imágenes etiquetados con el tipo de armas identificadas, lo que facilita la comprensión inmediata de los eventos. Al ingresar con las credenciales correctas, el usuario tiene acceso a las cámaras de seguridad en tiempo real, al igual que la manipulación de las opciones que ofrece la interfaz.

- ***Alerta sonora***

Al detectar un arma de fuego o una persona con el rostro cubierto a través del sistema CCTV, se aprecia la activación de forma automática la alarma sonora, y también al mismo tiempo una llamada telefónica completamente automática, y a la vez spam de mensajes de texto, hasta que la alerta sea atendida.

3.1.4 Selección de tecnologías

a. Microcomputador

Para garantizar el correcto funcionamiento del sistema detector de armas de fuego, es fundamental elegir un microcomputador con las especificaciones adecuadas que

permitan procesar video a una velocidad óptima, considerando que la detección se realiza en tiempo real.

En la Tabla 6 se detallan las características y especificaciones técnicas de cada microcomputador que puede ser seleccionado casos de visión por computadora, por lo tanto, la NVIDIA Jetson Nano es una elección ideal para un sistema de detección de armas mediante CCTV porque combina potencia de procesamiento, compatibilidad con modelos de IA como YOLOv5, conectividad flexible, y bajo consumo energético, así como lo indica el Anexo B. Su GPU optimizada garantiza que el procesamiento en tiempo real se realice de manera eficiente, lo cual es crucial para un sistema de seguridad que requiere alta precisión y velocidad. Además, su costo y facilidad de integración la convierten en una solución práctica y económica.

b. Sistema CCTV

El sistema CCTV elegido incluye cámaras de seguridad conectadas a un DVR, lo que permite capturar y almacenar video en tiempo real. Este componente garantiza una cobertura visual integral, su implementación resulta esencial para la detección temprana de eventos críticos, como la presencia de armas de fuego, lo que contribuye a reforzar la seguridad en entornos vulnerables y a reaccionar de manera oportuna ante situaciones de riesgo.

En la Tabla 1 se muestra los distintos tipos de DVR que son aptos para un sistema CCTV, tomando en cuenta las diferentes características y beneficios que puede ofrecer cada uno. El DVR HiLook de 4 canales 104g-M1 1080p Lite Mini es una solución estratégica para sistemas CCTV visto en el Anexo B, se enfoca en la detección de armas de fuego. Su compatibilidad con cámaras estándar, funciones inteligentes, almacenamiento eficiente y costo accesible lo convierten en un componente versátil y efectivo, incluyendo su estructura física vista en el especialmente en el contexto de Ecuador, donde la videovigilancia es fundamental para enfrentar el aumento de la inseguridad.

Para complementar la selección del DVR, las cámaras de seguridad tienen que cumplir ciertas especificaciones técnicas, tanto para la compatibilidad con el DVR y también para una mejor respuesta en el momento de detección.

En la Tabla 2 se muestra la comparación entre las distintas cámaras de seguridad que pueden formar parte de un sistema CCTV, notando sus características principales, por tal motivo las cámaras HiLook, tanto el modelo tipo tubo de 2 MP con audio Color Vu como el modelo domo fijo de 2 MP para interiores mostradas, destacan por su resolución Full HD (1920x1080) y la incorporación de tecnología 3D DNR, que garantiza imágenes nítidas y libres de ruido. Su compatibilidad con múltiples señales (TVI/AHD/CVI/CVBS) a través de puertos conmutables las convierte en una opción versátil y fácil de integrar en distintos sistemas. El modelo tipo tubo es ideal para exteriores, gracias a su capacidad de iluminación nocturna de hasta 20 metros, mientras que el modelo domo fijo, equipado con lentes de 2.8 mm y 3.6 mm, ofrece un amplio campo de visión, siendo perfecto para interiores. Estas cualidades las posicionan como una solución confiable y eficiente para sistemas de videovigilancia enfocados en la detección de armas de fuego, donde la adaptabilidad y la precisión son esenciales.

c. Materiales de instalación

Aparte del DVR y las cámaras de seguridad, la instalación de un sistema CCTV viene dada por más materiales que son importantes para su correcto funcionamiento.

La Tabla 3 presenta los materiales seleccionados, los cuales garantizan la funcionalidad, confiabilidad y eficiencia del sistema CCTV. Los baluns permiten transmitir señales de video a largas distancias mediante cable UTP Cat6, optimizando costos y reduciendo interferencias. Las fuentes de alimentación aseguran un suministro eléctrico constante a cámaras y DVR, mientras que los conectores macho/hembra facilitan la instalación y mantenimiento del sistema. Los cajetines sobrepuestos y tapas ciegas protegen las conexiones, evitando daños o manipulaciones indebidas. El disco duro asegura el almacenamiento de grabaciones esenciales, y los tornillos y tacos garantizan una instalación segura y estable. En conjunto, estos materiales forman un sistema robusto, confiable y adaptado a las necesidades del proyecto.

d. Respaldo de energía

Un UPS (Sistema de Alimentación Ininterrumpida) es crucial para asegurar el funcionamiento continuo del sistema CCTV durante cortes o fluctuaciones de energía. Este dispositivo proporciona energía de respaldo temporal, permitiendo que cámaras,

DVR y otros equipos sigan operando por un tiempo determinado, asegurando la grabación ininterrumpida y la seguridad en todo momento. Además, el UPS protege el sistema contra picos de tensión y otros problemas eléctricos que podrían dañar los dispositivos. Su capacidad para mantener el suministro energético durante fallas es esencial en sistemas de seguridad 24/7, donde la pérdida de datos o el mal funcionamiento pueden tener consecuencias graves, especialmente en entornos inseguros.

En la Tabla 17 se observa las distintas marcas de UPS en el mercado, al igual que las características técnicas más relevantes de los mismos.

Tabla 17. Comparativa entre UPS

Características	Forza NT511	Hikvision DS-UPS1000	CDP R-AVR3008
Capacidad de potencia (VA)	1100 VA	1000 VA	3000 VA
Potencia en watts	660 W	600 W	2400 W
Tipo de dispositivo	UPS Interactivo	UPS Interactivo	Regulador de Voltaje
Entradas	1 Entrada AC 110V-120V	1 Entrada AC 110V-120V	1 Entrada AC 110V-120V
Salidas	3 salidas protegidas + 1 para batería	2 salidas protegidas	8 salidas protegidas
Tiempo de respaldo	Aproximadamente 10-15 minutos (dependiendo de la carga)	Aproximadamente 10-15 minutos (dependiendo de la carga)	No aplica (es regulador de voltaje)
Batería	Batería de 12V/7Ah	Batería de 12V/7Ah	No incluye batería

Se seleccionó el UPS Forza NT511, el cual es ideal para un sistema CCTV debido a su capacidad de 1100 VA y 660 W, suficiente para respaldar cámaras, DVR y dispositivos de red. Su diseño interactivo garantiza protección contra picos de voltaje y cortes de energía, asegurando hasta 2 horas de respaldo para mantener la vigilancia activa y evitar interrupciones críticas en la grabación.

e. Software de programación

La selección del software de programación adecuado se llevó a cabo mediante la comparación de las principales características de los lenguajes más empleados en el desarrollo de aplicaciones de inteligencia artificial.

En la Tabla 8 se muestra la evaluación de las características de estos lenguajes proporciona una base fundamental para medir sus capacidades y seleccionar el más

adecuado según los requisitos específicos del proyecto. Este análisis detallado garantiza una elección estratégica y eficiente al integrar inteligencia artificial en el sistema.

Tras evaluar las opciones, se optó por Python para el desarrollo del proyecto debido a su vasta oferta de bibliotecas, su activa comunidad y su simplicidad de uso ejecutándose en todo tipo de entorno, así como lo indica Anexo C. Estos aspectos convierten a Python en una herramienta potente, accesible y gratuita para implementar proyectos de inteligencia artificial.

f. Plataforma basada en la nube

Este proyecto requiere una plataforma en la nube para entrenar el modelo de inteligencia artificial enfocado en la detección de armas de fuego. Una plataforma en la nube proporciona una variedad de servicios y recursos informáticos accesibles a través de internet, eliminando la necesidad de infraestructura local, como servidores físicos. Utiliza la infraestructura de la nube para proporcionar acceso a aplicaciones, almacenamiento y otros servicios, lo que permite mayor flexibilidad y acceso remoto mediante conectividad en línea.

La Tabla 7 muestra un análisis de las plataformas basadas en la nube disponibles para entrenamientos de modelos de IA, denotando características de cada uno, y una breve descripción.

Después de evaluar todos los aspectos, se seleccionó Google Colab debido a su acceso gratuito a recursos de GPU, esenciales para el entrenamiento de modelos de inteligencia artificial. Además, facilita la integración con Google Drive, donde se almacena la base de datos, ofrece un entorno basado en Jupyter Notebooks, tiene bibliotecas populares de Python preinstaladas, permite la colaboración en tiempo real y proporciona acceso ágil a datos y conjuntos de datos públicos.

g. Arquitectura de visión artificial

Se refiere al diseño especializado de redes neuronales convolucionales y modelos de aprendizaje profundo, desarrollados para tareas de procesamiento de imágenes y

análisis visual. Estas estructuras son fundamentales en aplicaciones como detección de objetos, reconocimiento facial y clasificación de imágenes, entre otras.

En la Tabla 4 se observa las diferencias entre las distintas arquitecturas más usadas para la detección de objetos, enfocándose principalmente en la precisión, velocidad de detección, recursos que ocupa y versiones disponibles hasta la actualidad.

Luego de evaluar varias alternativas, se seleccionó YOLO debido a su arquitectura avanzada y eficiente puesto en el Anexo D, que asegura un rendimiento óptimo. Esta opción permite una detección precisa y veloz, ideal para aplicaciones que requieren procesamiento en tiempo real.

h. Versión de Yolo

En la Tabla 5 se observa las versiones de YOLO que han representado avances significativos en la detección de objetos en imágenes y videos en tiempo real. Desde su lanzamiento inicial, YOLO ha experimentado una evolución constante, mejorando su precisión, eficiencia y velocidad, con cada versión incorporando optimizaciones en su arquitectura y técnicas avanzadas de procesamiento.

Tabla 18. Comparativa YOLOv5 y YOLOv8. [56]

Características	YOLOv5	YOLOv8
Precisión de detección	Alta precisión, ideal para tareas de detección en imágenes de calidad variable. Capaz de identificar objetos de forma eficiente en imágenes con condiciones desafiantes.	Mejora en precisión, especialmente en imágenes complejas y de baja resolución, pero a costa de mayor tiempo de procesamiento.
Velocidad de procesamiento	Rápido en tiempo real, adecuado para sistemas CCTV donde se requiere análisis en vivo. Es eficiente en entornos con limitaciones de hardware.	Aunque es más preciso, puede ser más lento debido a su arquitectura más compleja y mejoras adicionales.
Uso de recursos (memoria, CPU/GPU)	Menor uso de memoria y recursos computacionales. Puede ser ejecutado con hardware más modesto y en dispositivos con recursos limitados.	Requiere más recursos (memoria y potencia de procesamiento), lo que puede ser un desafío para dispositivos con menos capacidad.
Rendimiento en hardware limitado	Eficiencia optimizada para funcionar bien incluso en plataformas de hardware limitado, como microcomputadores y cámaras de seguridad.	Optimizado para hardware más potente; puede no ser tan eficiente en dispositivos con pocos recursos como cámaras de seguridad o microcomputadores.
Detección en tiempo real	Excelente rendimiento en sistemas que requieren detección en tiempo	Detección en tiempo real, pero podría haber una ligera reducción en velocidad

	real, adecuado para aplicaciones de CCTV.	debido a la mayor complejidad del modelo.
Calidad en condiciones de baja resolución	Mantiene un rendimiento sólido incluso en imágenes de baja resolución o con mucho ruido. Ideal para sistemas CCTV con cámaras estándar.	Mejora en el manejo de imágenes de baja resolución, pero puede necesitar mayor potencia para lograr un desempeño óptimo.
Entrenamiento y tiempo de desarrollo	Rápido de entrenar, ideal para proyectos con plazos ajustados. Permite iteraciones rápidas y un buen rendimiento desde las primeras etapas.	El entrenamiento puede ser más lento y más exigente en recursos, lo que puede alargar el tiempo de desarrollo.
Size (píxeles)	Resolución de entrada flexible: 640x640 píxeles típicamente, ajustable según el modelo y el entorno.	Resolución de entrada flexible: 640x640 píxeles típicamente, ajustable. Algunos modelos avanzados pueden trabajar con resoluciones mayores para mejorar la precisión.
Speed CPU (ONNX) (ms)	En CPU, la velocidad con ONNX es aproximadamente 30-40 ms para imágenes de 640x640 en hardware modesto.	En CPU, la velocidad con ONNX es aproximadamente 40-60 ms en condiciones similares.

Tras evaluar las versiones de YOLO, se eligió YOLOv5 para el sistema de detección de armas en el proyecto CCTV, gracias a su balance entre precisión, velocidad y eficiencia en el uso de recursos. Aunque versiones más recientes como YOLOv8 mejoran la precisión, YOLOv5 destaca por su capacidad de procesamiento en tiempo real, un factor clave para aplicaciones de vigilancia que demandan análisis instantáneo. El análisis realizado en la Tabla 18 determina la elección de YOLOv5, sobre YOLOv8.

3.1.5 Descripción del sistema de detección

En la Figura 15 se presenta la propuesta del sistema detector de armas de fuego, detallando cada una de sus etapas, con un enfoque particular en su estructura física. Se muestra la conexión desde la fuente de alimentación hasta los dispositivos terminales, que incluyen las cámaras, el DVR, el router y el microcomputador NVIDIA Jetson Nano. Asimismo, se describe la comunicación que realiza el sistema, destacando como punto principal la conexión Wi-Fi entre las dos primeras etapas, junto con una comunicación especializada para la transmisión de imágenes en tiempo real, implementada mediante el protocolo RTSP.

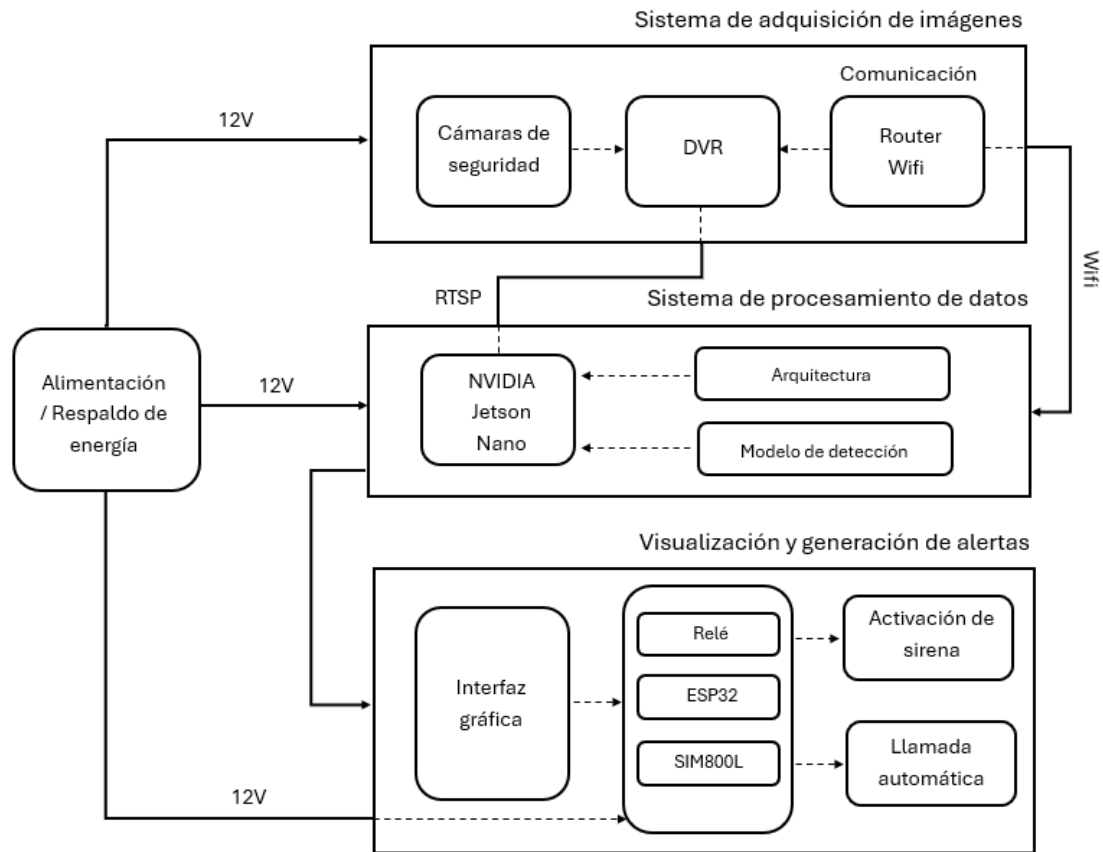


Figura 15. Esquema del sistema por bloques

3.1.6 Implementación sistema CCTV

La implementación de un sistema de circuito cerrado de televisión (CCTV) en un hogar es esencial para garantizar la seguridad y el monitoreo en tiempo real. Este proceso abarca la selección de componentes clave, como cámaras, DVR y dispositivos de red, así como la integración de estos elementos para asegurar una comunicación eficiente y una transmisión confiable de imágenes.

a. Instalación

La ubicación de los elementos es importante antes de proceder con la instalación. Por lo tanto, se realizó un estudio detallado del área a monitorear, identificando y descartando posibles ubicaciones donde las cámaras de seguridad no deberían instalarse. De igual manera, se definieron los sitios estratégicos donde su colocación resulta esencial para garantizar una cobertura óptima y efectiva, enfocando mayor

parte del espacio requerido, así como indica el Anexo E, y evitando dejar puntos ciegos o también conocidas como zonas muertas. [57]

La Figura 16 muestra las zonas muertas generadas a partir de una mala colocación de las cámaras de seguridad, considerando alturas superiores a los 3 metros.

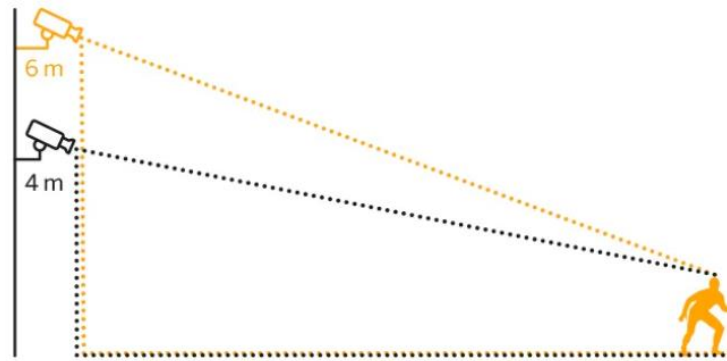


Figura 16. Zonas muertas. [57]

Considerando el aspecto de los puntos ciegos, y en base a la revisión del área, se instaló la cámara tipo tubo en la parte central de la sala del domicilio porque esta permite una movilidad en sentido horizontal y vertical más prolongado que las cámaras tipo domo, y también considerando la altura de la pared del domicilio (4m), de tal manera que generaría un punto ciego.

En la Figura 17 se puede observar el área total que abarca la cámara 1 tipo tubo instalada a una altura de 2 metros, evitando de esa forma los puntos ciegos, y cubriendo puntos vitales como las ventanas y la puerta principal, al igual que el área total de la sala y gran parte del comedor.

En cambio, en la Figura 18 se muestra la cámara 2 y se observa el área que cubre al ser una cámara tipo domo, enfocando principalmente la ventana y salida hacia el patio trasero.

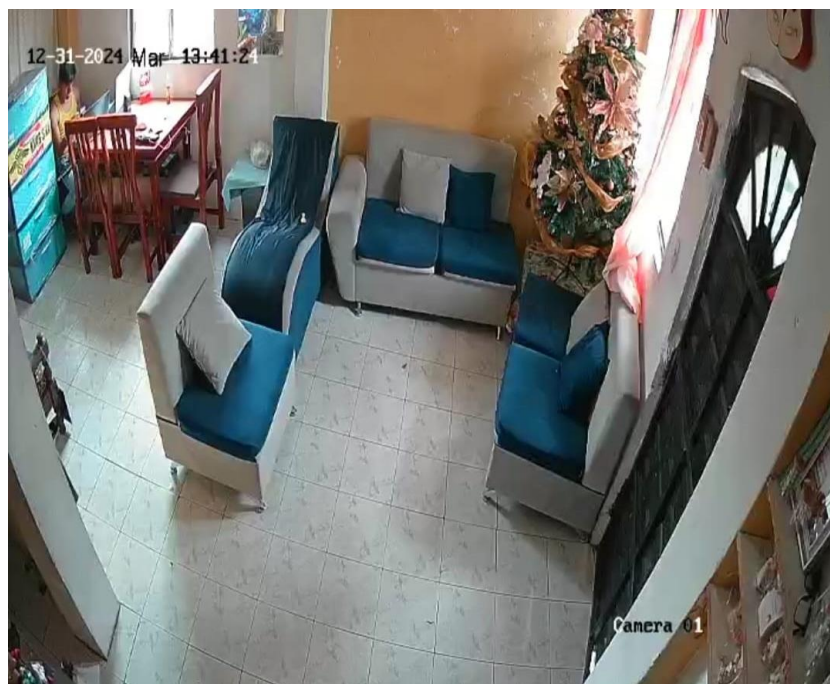


Figura 17. Vista de cámara 1 – sala

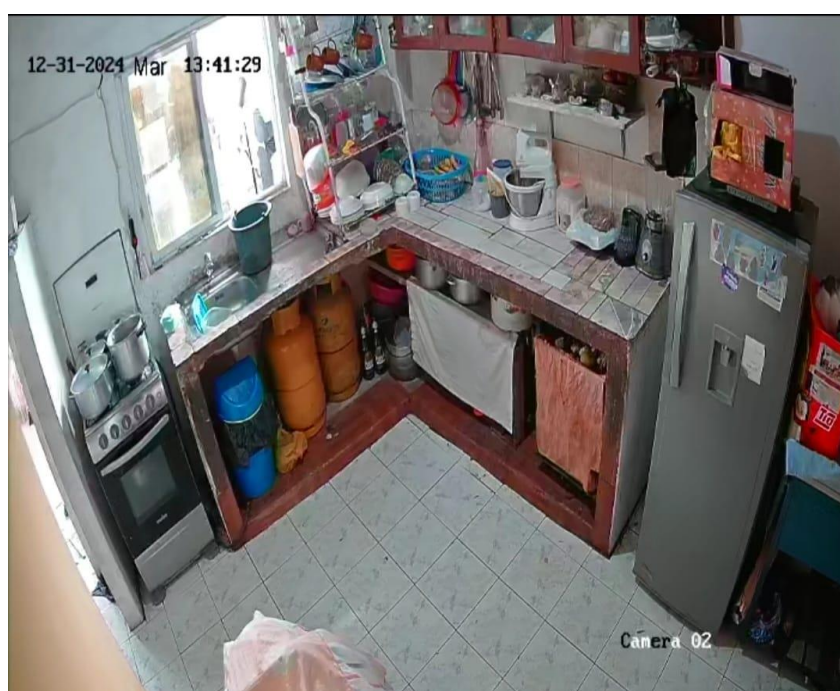


Figura 18. Vista de cámara 2 - cocina

b. Configuración

Este procedimiento incluye la conexión del DVR a los dispositivos periféricos, como cámaras, monitor y red, así como la definición de parámetros clave, como la calidad de grabación, zonas horarias, y ajustes de red para acceso remoto.

Con las cámaras estratégicamente posicionadas y el DVR conectado hacia el router para acceso a internet, se realizó las configuraciones necesarias para el funcionamiento del sistema CCTV.

En la Figura 19 se observa los parámetros correspondientes a la red, donde la activación del DHCP y la configuración del DNS fue importante, debido a que estos valores permiten al DVR tener acceso a internet, y por ende visualización en tiempo real.

The screenshot displays the HiLook web interface for configuring a DVR. The top navigation bar includes 'Vista en Directo', 'Reprod.', and 'Configuración'. The left sidebar shows a menu with 'Local', 'Sistema', 'Red', and 'Ajustes básicos' (selected). The main content area is titled 'TCP/IP' and shows settings for 'Lan1'. The 'Tipo de NIC' is set to '10M/100M/1000M Auto'. The 'DHCP' checkbox is checked. The 'Dirección Ipv4' is '192.168.101.2', 'Máscara de Subred Ipv4' is '255.255.255.0', and 'Dirección Ipv4 por defecto' is '192.168.101.1'. The 'Dirección IPv6' is 'fe80::de07:f8ff:fe72:95f', 'Longitud del prefijo de su...' is 'ffff:ffff:ffff:ffff::', 'Puerta de salida IPv6 por...' is empty, 'Dirección MAC' is 'dc:07:f8:72:09:5f', and 'MTU' is '1480'. The 'Servidor DNS' section has 'DNS automático' unchecked, 'Servidor DNS favorito' as '8.8.8.8', 'Servidor DNS alternativo' as '8.8.4.4', and 'Ruta por defecto' as 'Lan1'. A copyright notice at the bottom reads '©2024 Hikvision Digital Technology Co., Ltd. All Rights Reserved.'

Figura 19. Configuración de parámetros de red del DVR

La Figura 20 muestra los distintos puertos que posee el DVR, a través de los cuales se pueden determinar salidas para los distintos tipos de información, ya sea audio, video o imágenes de video.

HiLook Vista en Directo Reprod. Configuración

Local Sistema Red

Ajustes básicos Ajustes avanzados

Vídeo y audio Imagen Event Almacenamiento

TCP/IP DDNS **Puerto** NAT

Puerto HTTP 80

Puerto RTSP 554

Puerto HTTPS 443

Puerto del servidor 8000

Guardar

Figura 20. Asignación y apertura de puertos disponibles.

Es fundamental destacar que la activación del puerto 554, asociado al protocolo RTSP, resulta esencial para el sistema, pues mediante este se estableció la comunicación entre el sistema CCTV y el microcomputador.

Por último, la Figura 21 presenta los parámetros de salida recomendados para el sistema, se manejó estos parámetros porque de estos depende la imagen en tiempo real que se proyecte en el sistema, por ende, las imágenes en tiempo real que son evaluadas por el sistema.

HiLook Vista en Directo Reprod. Configuración

Local Sistema Red

Vídeo y audio Imagen Event Almacenamiento

Vídeo Mostrar información En línea

Cámara [A1] Camera 01

Resolución de entrada 960*1080(HD1080PLite)P30

Tipo de secuencia: Flujo principal(Normal)

Tipo de vídeo Vídeo y audio

Resolución 960*1080(1080P Lite)

Tipo de tasa de bits Variable

Calidad de vídeo El más alto

Velocidad de frames Fotograma completo fps

Veloc. máx. bits 1536 Kbps

Codificación de vídeo H.265

H.265+ APAGAR

Copiar en... Guardar

Figura 21. Configuración de las imágenes de video de salida

3.1.7 Desarrollo del dataset

a. Comparativa del dataset

Los datasets disponibles para entrenar un sistema de detección de armas de fuego son limitados, debido a que muchos de ellos han sido creados con propósitos específicos. Estos pueden incluir únicamente armas cortas, solo armas largas o la detección combinada de armas de fuego y armas blancas, dependiendo del propósito definido por el creador del dataset.

La Tabla 19 presenta los datasets más relevantes para la detección de armas de fuego, destacando como principal aporte sus características generales, la cantidad de imágenes que contiene cada uno y su correspondiente validación para su uso.

Tabla 19. Comparativa de datasets

Dataset	Sitio	Características	Cantidad imágenes	Clases	Validación
MAI-DAI	Roboflow Universe	Enfoque principal en una cantidad grande de imágenes relacionadas a armas de fuego, armas blancas y utensilios considerados de peligrosidad v para las personas.	5606	- Weapon - Knife	Roboflow
Detección de armas Open Data	Instituto Andaluz Interuniversitario en Data Science and Computational Intelligence	El conjunto de datos se centra en el desarrollo de sistemas automáticos de videovigilancia inteligente.	4710	Pistola	Universidad de Granada
Weapons_V2	Roboflow Universe	Contiene un conjunto amplio de armas cortas, enfocadas en diferentes posiciones y en diferentes situaciones delictivas.	3566	Gun	Universidad Panamericana
Gun Violent detection Dataset	Roboflow Universe	Datos destinados a la detección de armas de fuego cortas y largas, tomando como principal característica que su estructura viene dada de capturas de cámaras de video vigilancia.	5111	- Gun - Shotgun	Roboflow

Finalmente se decidió crear un nuevo dataset, debido a que las clases de los datasets presentados en la Tabla 19 no cumplían con los requisitos necesarios para el desarrollo del sistema de detección de armas de fuego. Este sistema está diseñado para detectar tres clases específicas: armas cortas, armas largas y rostros cubiertos, como se detalla en las Tabla 15 y Tabla 16.

b. Creación del dataset

La herramienta online Roboflow permite crear Datasets propios, con el fin de usarlos en trabajos relacionados a inteligencia artificial y detección de objetos. Esta herramienta también permite el uso de Datasets que ya estén creados, pero manteniendo el etiquetado que haya realizado el dueño.

Se crea un nuevo proyecto en el cual se guarda el dataset, este puede ser privado o público. Una vez que el proyecto esté creado, se comienza con la carga de las imágenes dentro del dataset.

En este caso, se tienen subidas 6884 imágenes dentro del dataset llamado Tipos_Armas tal y como lo muestra la Figura 22.

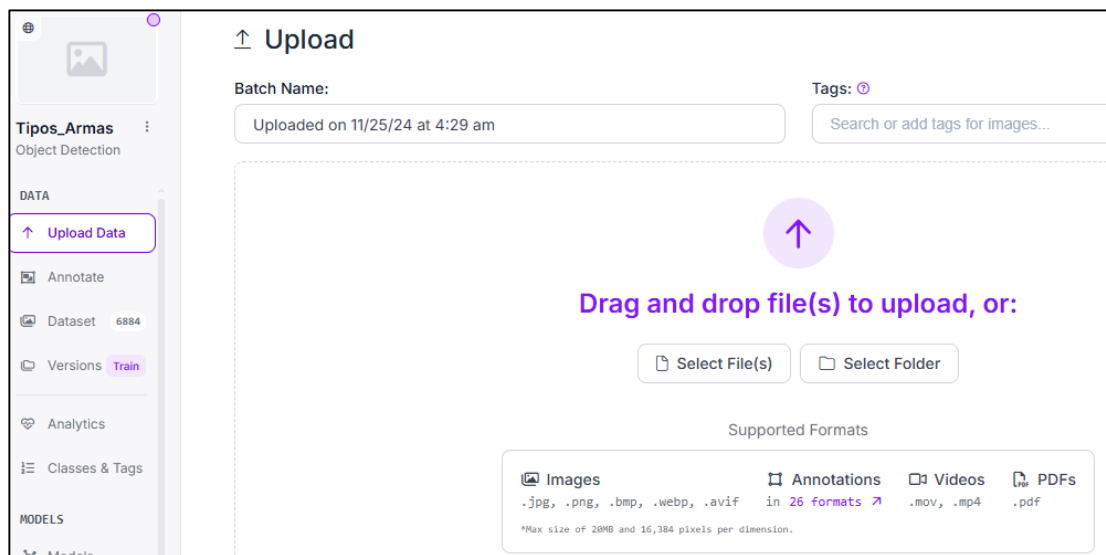


Figura 22. Ventana para cargar imágenes en Roboflow

Las etiquetas dentro de un dataset son las partes clave de lo que se quiere detectar, tomando como referencia formas de cada objeto, y diferenciándolo de otro por nombre y color de su recuadro.

En la Figura 23 se ilustra el proceso de etiquetado de un objeto a detectar utilizando la herramienta Roboflow. También se muestra la creación y asignación de las clases correspondientes al objeto, lo que permite definir las categorías a las cuales pertenece el objeto y facilita su detección en el proceso de entrenamiento del modelo

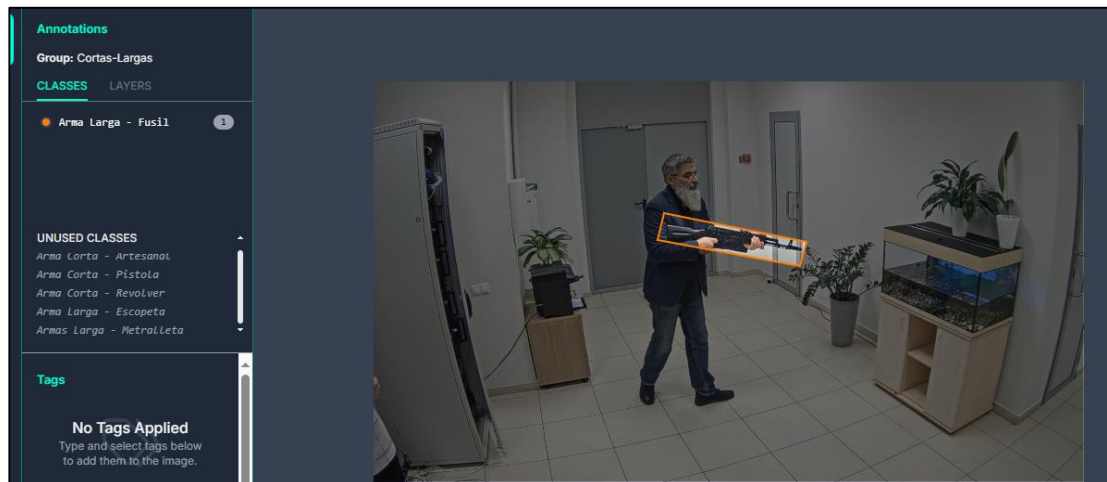


Figura 23. Proceso de etiquetado en Roboflow

Una vez finalizado el etiquetado de todas las imágenes, se procede a guardarlas en el dataset. Este proceso puede demorar algunos minutos, dependiendo de la cantidad de imágenes que se estén manejando.

En la Figura 24 se puede observar el dataset terminado completamente

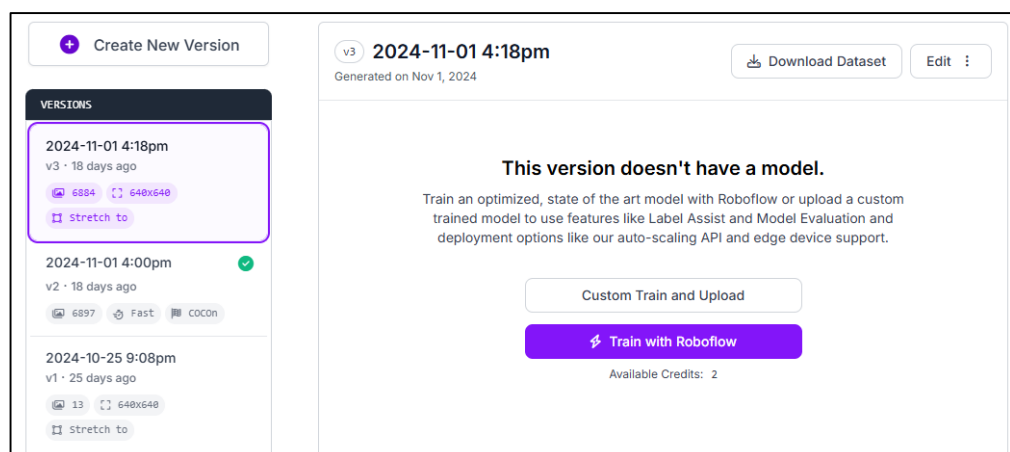


Figura 24. Dataset terminado

Roboflow permite descargar el dataset creado en formato ZIP, lo que facilita su uso adecuado. También ofrece la opción de obtener el dataset mediante un enlace URL, y

proporciona diversas versiones de salida, como YOLO, JSON, XML, TXT, CSV, entre otros.

La descarga fue directamente en la arquitectura YOLOv5 Pytorch, como lo muestra la Figura 25, pero en forma de URL para su posterior uso en el entrenamiento.

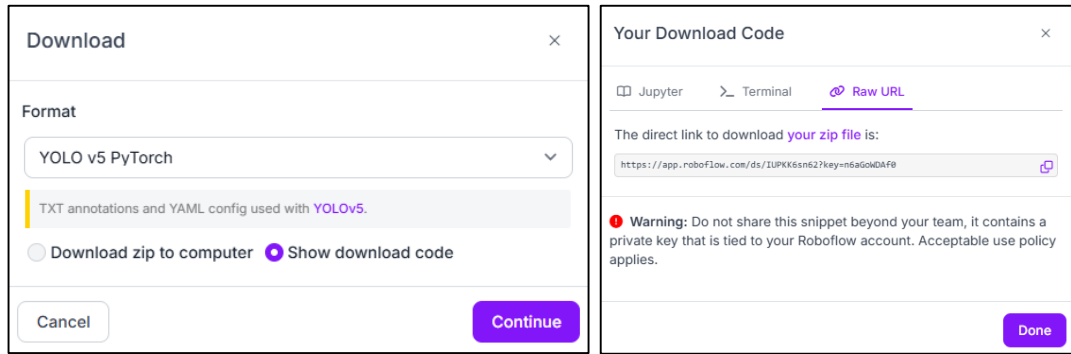


Figura 25. Dataset Descargado en Raw URL

c. Dataset terminado

La Tabla 20 presenta los datos principales del dataset creado, proporcionando una breve descripción de su contenido, el nombre asignado, la cantidad total de imágenes incluidas y, de manera destacada, las etiquetas que lo conforman. Además, esta información permite evaluar la estructura y cobertura del dataset, asegurando que cumple con los requisitos establecidos para el entrenamiento del modelo de detección de armas de fuego.

Tabla 20. Datos del dataset finalizado

Dataset	Sitio	Características	Cantidad imágenes	Clases	Validación
Tipos_Armas	Roboflow Universe	El conjunto de datos mostrado se enfoca en una cantidad promedio de imágenes, enfocándose en la detección de armas de fuego y personas sospechosas.	6884	- Arma Corta - Arma Larga - Intruso	Roboflow

Una vez completada la creación del dataset, se procede al entrenamiento del modelo de detección de armas de fuego. Este proceso utiliza las imágenes y etiquetas definidas en el dataset para optimizar la capacidad del modelo de identificar con precisión las clases objetivo.

3.1.8 Entrenamiento del sistema

Para entrenar el modelo, es importante considerar varios aspectos clave, los cuales se detallan en la Tabla 21.

Tabla 21. Parámetros por considerar en el entrenamiento.

Parámetros del entrenamiento		
Parámetros	Características	Detalle
E. Train	1430	Etiquetas de entrenamiento
E. Valid	723	Etiquetas de validación
E. Test	258	Etiquetas para prueba
Cant. Imágenes	2411	Imágenes totales
Cant. Etiquetas	5112	Total, de etiquetas existentes en las imágenes del dataset
Data	Data/Campo	Conjunto de datos
Épocas	65	Cantidad de épocas para el entrenamiento
Batch	16	Tamaño de las muestras que pasan por la red
Modelo	Yolov5 PyTorch	Modelo de entrenamiento
Clases	3	Etiquetas que lleva cada imagen del dataset.

Una vez considerados estos datos, se puede iniciar el entrenamiento. Cabe recalcar que, para su observación, se realizaron dos entrenamientos variando ciertos parámetros con el objetivo de obtener mejores resultados.

Se abrió un entorno de trabajo dentro de la plataforma Google Colab, la cual permite el uso de un GPU, lo que resulta muy útil para potenciar y acelerar el entrenamiento del modelo. A través de repositorios en GitHub, se ejecutaron las líneas de código correspondientes para la descarga e instalación de YOLOv5.

En la Figura 26 se presenta la línea de código necesaria para la descarga e instalación de YOLOv5 en Google Colab. Cabe recalcar que esta misma línea de código es aplicable en otros entornos como kaggle.

```
!git clone https://github.com/ultralytics/yolov5

Cloning into 'yolov5'...
remote: Enumerating objects: 17055, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (21/21), done.
remote: Total 17055 (delta 17), reused 24 (delta 12), pack-reused 17022 (from 1)
Receiving objects: 100% (17055/17055), 15.68 MiB | 18.24 MiB/s, done.
Resolving deltas: 100% (11711/11711), done.
```

Figura 26. Descarga de YOLOv5 en Google Colab

La instalación de los requisitos es fundamental para garantizar el correcto funcionamiento de YOLO en la detección de objetos, pues incorpora librerías y parámetros adicionales esenciales para su rendimiento.

En la Figura 27 se muestra el comando necesario para la instalación de los requerimientos de YOLOv5, este comando es aplicable tanto en Google Colab o Kaggle.

```
%pip install -r requirements.txt
```

Figura 27. Instalación de requerimientos de YOLOv5

Una vez instalados los requisitos, el siguiente paso fue seleccionar el dataset con el que se realizó el entrenamiento. En este caso, se utilizó un dataset creado en Roboflow. En la Figura 28 se presenta la línea de código correspondiente para la descarga del dataset en formato ZIP.

```
!wget https://universe.roboflow.com/ds/IUPKK6sn62?key=n6aGoWDAf0 -O roboflow.zip
```

Figura 28. Comando para descargar dataset desde Roboflow

Una vez que el archivo se haya descargado completamente, se procede a descomprimirlo. Este contiene las imágenes junto con sus respectivas etiquetas.

En la Figura 29 se muestran los diferentes parámetros utilizados para iniciar el entrenamiento del modelo de detección. Debido a las limitaciones de uso de Colab, se consideraron únicamente 45 épocas. Por otro lado, como se observa en la Figura 30, se utilizaron 65 épocas en Kaggle, en vista de que esta plataforma permite un mayor tiempo de uso del GPU.

```
!python train.py --img 416 --batch 16 --epochs 45 --data 'data.yaml' --cfg yolov5s.yaml
```

Figura 29. Línea de comando para comenzar con el entrenamiento – 45 épocas

```
!python train.py --img 416 --batch 16 --epochs 65 --data 'data.yaml' --cfg yolov5s.yaml
```

Figura 30. Línea de comando para comenzar con el entrenamiento – 65 épocas

3.1.9 Procesamiento y detección de armas en imágenes de video

El sistema cuenta con procesos individuales dentro de la codificación definitiva para su funcionamiento. En la Figura 31 se muestra la conexión entre los distintos procesos de inicio a fin.

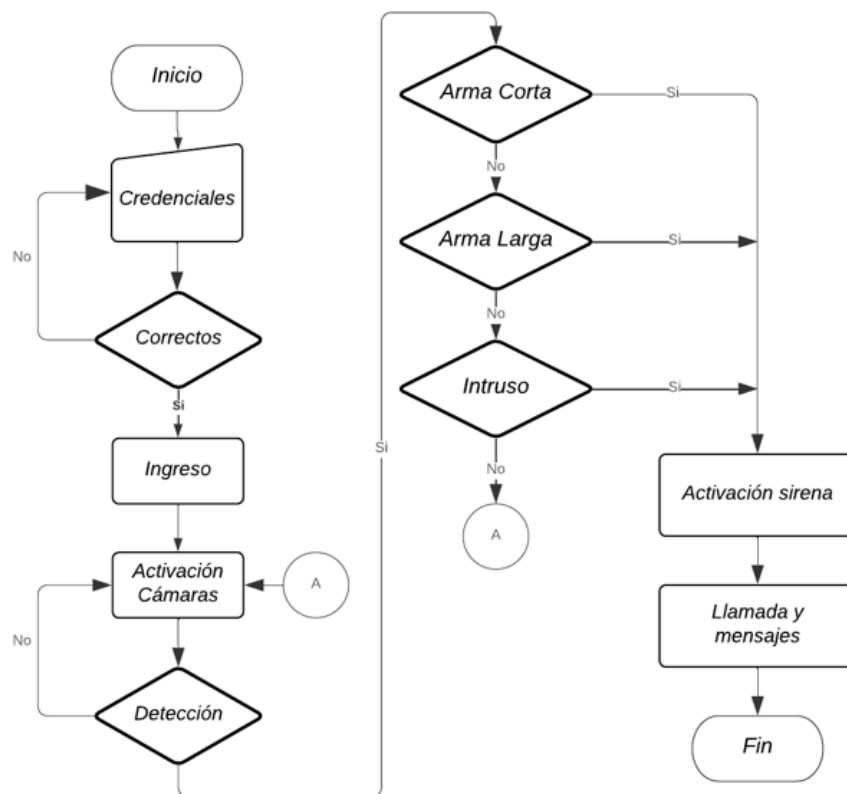


Figura 31. Diagrama de flujo general del sistema

a. Importación de librerías

Para desarrollar este sistema, se necesitó de varias bibliotecas nativas de Python, en la Tabla 22 se detallan el propósito en el sistema y sus características principales.

Tabla 22. Librerías utilizadas en el sistema. [58]

Librería	Función	Características
cv2 (OpenCV)	Procesamiento de imágenes y visión por computadora.	- Manipulación de imágenes y videos - Detección de objetos - Transformaciones geométricas - Soporte de cámaras.
torch (PyTorch)	Framework de aprendizaje profundo.	- Entrenamiento de redes neuronales - GPU/CPU acelerado - Soporte de modelos dinámicos - Amplia comunidad.
numpy	Manipulación de arreglos multidimensionales y operaciones matemáticas.	- Operaciones vectorizadas - Manipulación de matrices - Integración con otras librerías científicas.
pathlib	Manejo de rutas de archivos de manera orientada a objetos.	- Compatible con sistemas operativos - Métodos intuitivos para manipular rutas y archivos.
warnings	Manejador de advertencias en Python.	- Control y registro de advertencias - Filtrado de niveles de severidad.
Time	Manejo de operaciones relacionadas con tiempo.	- Manejo de fechas y horas. - Manejo de temporizadores. - Retrasos en ejecución (sleep).
tkinter	Creación de interfaces gráficas en Python.	- Widgets básicos (botones, etiquetas, etc.) - Fácil integración con código Python.
PIL (Pillow)	Procesamiento y manipulación de imágenes.	- Soporte de múltiples formatos de imágenes - Transformaciones y filtros de imágenes.
threading	Ejecución de hilos en Python.	- Concurrencia en procesos - Coordinación entre hilos.
pygame	Desarrollo de videojuegos y multimedia.	- Soporte para gráficos 2D - Control de audio - Soporte para entradas del usuario.
customtkinter	Extensión personalizada para interfaces de Tkinter.	- Diseños modernos y adaptables - Personalización avanzada de widgets.

Cada una de las librerías que se muestran en la Tabla 22 permite la ejecución de cada proceso en el cual sea necesaria su instancia, esto para el posterior funcionamiento total del sistema y su interfaz.

La Tabla 23 muestra la forma en la que las librerías están importadas dentro del código principal.

Tabla 23. Importación de librerías

<pre>import cv2 import torch import numpy as np import pathlib import warnings import time from tkinter import Label from PIL import Image, ImageTk import threading import pygame import customtkinter as ctk</pre>
--

b. Aplicación del modelo entrenado

El código carga un modelo YOLOv5 previamente entrenado (best_E5.pt) utilizando PyTorch, permitiendo realizar detección de objetos en imágenes o videos. Este modelo, personalizado para un conjunto de datos específico, puede identificar y clasificar objetos relevantes, como armas de fuego, en tiempo real o en imágenes estáticas. La carga se realiza a través de torch.hub desde el repositorio oficial de YOLOv5, configurado para detectar características definidas durante el entrenamiento, facilitando su integración en aplicaciones como sistemas de vigilancia, análisis de imágenes.

En la Tabla 24 se observa las líneas de código correspondientes para hacer uso de un modelo ya entrenado en un script de Python.

Tabla 24. Aplicación del modelo entrenado en el código

<pre>path = 'C:\\Users\\Administrator\\Desktop\\Tesis\\best_E5.pt' model = torch.hub.load('ultralytics/yolov5', 'custom', path, force_reload=True)</pre>
--

c. Detección de armas

Antes de iniciar el proceso de detección, es importante destacar las líneas correspondientes a la comunicación entre el sistema CCTV y el microcomputador, pues la detección de armas o intrusos requiere capturar imágenes en tiempo real provenientes del sistema.

- ***Comunicación con el sistema CCTV***

Para visualizar las imágenes en tiempo real provenientes del sistema CCTV, se usó el protocolo RTSP, debido a que la mayoría de DVR posee este protocolo para comunicaciones externas, sin depender de otro tipo de protocolos como el HTTP u OVNI.

En la Tabla 25 se observa las líneas correspondientes a la comunicación vía RTSP, tanto para la cámara 1 y para la cámara 2.

Tabla 25. Estructura básica de comunicación mediante RTSP

```
self.cap1 =
cv2.VideoCapture("rtsp://admin:12345@192.168.100.45:554/Streaming/Channels/102?transportmode=unicast&profile=Profile_1")
self.cap2 =
cv2.VideoCapture("rtsp://admin:12345@192.168.100.45:554/Streaming/Channels/202?transportmode=unicast&profile=Profile_2")
```

El código establece conexiones a dos flujos de video en tiempo real utilizando el protocolo RTSP (Real-Time Streaming Protocol) y la biblioteca OpenCV. Cada línea define un objeto videocapture, que permite acceder a transmisiones de video desde una cámara IP o un dispositivo similar. La URL RTSP contiene información clave:

- **Protocolo** (rtsp://): Define que se trata de un flujo de transmisión en tiempo real.
- **Credenciales** (admin:12345): Usuario y contraseña para autenticar el acceso al dispositivo.
- **Dirección IP** (192.168.100.45): Ubicación de la cámara o DVR en la red.
- **Puerto** (554): Puerto predeterminado para RTSP.
- **Ruta y parámetros**: Especifican el canal de video (Channels/102 y Channels/202), el modo de transporte (unicast) y el perfil de transmisión.

- **Detección de armas e intrusos**

Para la detección de armas e intrusos, la imagen captada por el sistema CCTV debe atravesar varias etapas de procesamiento. Sin embargo, las más importantes, que destacan desde el inicio del análisis, se detallan a continuación en la Tabla 26.

Tabla 26. Líneas del código para la parte de detección

Fragmento de Código	Descripción
with torch.no_grad(): results = model(frame)	Ejecuta el modelo YOLOv5 en un fotograma (frame), dentro del método process_frame, en donde realiza inferencias del modelo en el fotograma actual (frame).
detections = results.xyxy[0]	Extrae las detecciones del modelo en formato de coordenadas (x1, y1, x2, y2) confianza y clase detectada (0, 1, 2).

<pre> for det in detections: class_id = int(det[5]) if class_id in class_names: detected_arms.append(class_names[class_id]) print(f"Arma detectada: {class_names[class_id]}") </pre>	Itera sobre las detecciones, identifica las clases detectadas y las agrega a una lista si están en class_names, en este caso Arma Corta, Arma Larga, Intruso.
--	--

Se analizan las detecciones para identificar las clases relevantes, agregándolas a una lista y mostrando la información correspondiente, facilitando su uso en aplicaciones como la identificación de armas en sistemas de seguridad.

3.1.10 Visualización y alertas

Cuando el sistema detecta armas de fuego o personas con el rostro cubierto, de forma automática se genera una alerta sonora, y de forma seguida realiza una llamada telefónica a un número celular asignado. Estos procesos se llevan a cabo a partir de líneas de códigos especiales para su funcionamiento.

a. Interfaz

En la Tabla 27 Se observa las líneas básicas para la creación de la interfaz principal, es decir, la interfaz de inicio de sesión, en donde a través del correcto ingreso de las credenciales, se tiene acceso a la interfaz de visualización de las cámaras de seguridad en tiempo real.

Tabla 27. Codificación básica de la ventana principal

<pre> # Clase para la pantalla de inicio de sesión class LoginScreen: def __init__(self, root): self.root = root self.root.title("Inicio de Sesión") self.root.geometry("600x600") # Tamaño personalizado de la ventana # Cambiar el color de fondo self.root.configure(bg="#ffffff") # Color personalizado (oscuro) self.frame = ctk.CTkFrame(self.root, corner_radius=15) self.frame.pack(pady=20, padx=60, fill="both", expand=True) # Crear la ventana principal de Tkinter ctk.set_appearance_mode("System") # Modo de apariencia ("Light", "Dark", "System") ctk.set_default_color_theme("blue") # Tema de color root = ctk.CTk() LoginScreen(root) root.mainloop() </pre>
--

La estructura de la ventana principal es similar a la interfaz de visualización de las cámaras, enfocándose principalmente en el orden objeto – clase, y en mainloop() para la ejecución de la misma.

b. Llamada

La Tabla 28 muestra las partes más importantes para la ejecución de la llamada telefónica de forma automática, destacando dos partes principales, la elección del número destinatario para la llamada, y también el llamado de la función para el inicio de la llamada.

Tabla 28. Código para ejecución de la llamada

Código	Función
# Realizar la llamada telefónica self.make_call("+1805355938") # Reemplazar llamar	Se selecciona el destinatario al cual a través del módulo SIM800L se realiza la llamada.
self.sim800l.write(f'ATD{phone_number};\r'.encode()) time.sleep(2) # Esperar un momento para iniciar la llamada print(f"Llamada realizada al número {phone_number}.") except Exception as e: print(f"Error al realizar la llamada: {e}")	El código realiza una llamada utilizando el módulo SIM800L. El comando AT ATD{phone_number}; es enviado al módulo a través de la conexión serial (self.sim800l.write()), donde {phone_number} se reemplaza con el número de teléfono al que se desea llamar.

Se incluye un retraso de 2 segundos con time.sleep(2) para dar tiempo al módulo para iniciar la llamada. Si la llamada se realiza correctamente, se imprime un mensaje en consola confirmándolo; en caso de error, se captura la excepción y se informa del problema.

c. Sirena

La activación de la sirena se compone de dos procesos principales. Primero, en el aspecto físico, se genera un pulso desde la placa ESP32 hacia un relé, lo que activa la sirena. Sin embargo, para que este pulso sea enviado, es necesario que previamente se haya detectado un arma mediante el sistema de reconocimiento. Este segundo proceso, basado en la respuesta del sistema de detección de armas, asegura que la sirena solo se active en situaciones relevantes, así como se muestra en la Tabla 29.

Tabla 29. Codificación de alarma

Código	Función
<pre>def start_alarm(self): global alarm_active if not alarm_active: alarm_active = True self.alarm_thread = threading.Thread(target=self.play_alarm, daemon=True) self.alarm_thread.start()</pre>	La función <code>start_alarm</code> activa una alarma iniciando un hilo en segundo plano que ejecuta la función <code>play_alarm</code>. Verifica si la alarma ya está activa y, de no estarlo, la marca como activa y lanza el hilo, permitiendo que la alarma se ejecute sin interrumpir el flujo principal del programa.
<pre>def stop_alarm(self, event=None): global alarm_active if alarm_active: alarm_active = False pygame.mixer.music.stop()</pre>	La función <code>stop_alarm</code> detiene la alarma verificando si está activa. Si es así, cambia el estado de la variable global <code>alarm_active</code> a <code>False</code>. Esto garantiza que la alarma se apague de manera controlada.

Se implementó la cancelación de la alarma al presionar la tecla “C”, porque al momento de la detección, la alarma entra en un bucle donde finaliza únicamente al cerrar el programa completo, entonces, para evitar eso se implementó dicha funcionalidad.

3.2 Resultados

3.2.1 Resultados del entrenamiento

Durante el proceso de entrenamiento, el modelo fue evaluado a través de múltiples épocas, y se observó que el tiempo total de entrenamiento dependía directamente de la cantidad de épocas empleadas. A medida que aumentaba el número de épocas, el tiempo necesario para completar el entrenamiento también crecía, lo que permitió observar cómo las métricas de desempeño (precisión, recall, y F1-score) mejoraban progresivamente a lo largo del proceso.

En la Figura 32 se observa los resultados de entrenamiento realizados con 65 épocas, siendo este el mejor de los resultados obtenidos en el proceso de entrenamiento.

```

65 epochs completed in 0.803 hours.
Optimizer stripped from runs/train/exp/weights/last.pt, 14.3MB
Optimizer stripped from runs/train/exp/weights/best.pt, 14.3MB

Validating runs/train/exp/weights/best.pt...
Fusing layers...
YOLOv5s summary: 157 layers, 7018216 parameters, 0 gradients, 15.8 GFLOPs

```

Class	Images	Instances	P	R	mAP50	mAP50-95:
all	723	1520	0.911	0.902	0.942	0.68
Arma Corta	723	483	0.869	0.874	0.907	0.664
Arma Larga	723	652	0.938	0.908	0.954	0.692
Intruso	723	385	0.927	0.926	0.965	0.694

```

Results saved to runs/train/exp

```

Figura 32. Resultados de entrenamiento con 65 épocas.

Se realizaron entrenamientos del modelo durante 45, 40 y 65 épocas, tal y como se muestra en la Tabla 30.

Tabla 30. Comparativa de resultados de entrenamiento obtenidos

Épocas	Instancias	Precisión	Recall	mAP0.5	mAP0.5:0.95	Tiempo
45	1520	0.902	0.895	0.94	0.542	0.560h
60	1520	0.943	0.882	0.927	0.638	3.171h
65	1520	0.911	0.902	0.942	0.68	0.803h

El entrenamiento concluye cuando se hayan completado las épocas correspondientes, y sus resultados se visualizan como texto en la consola, al igual que varios archivos descargados, entre ellos los modelos entrenados llamados Best.pt y Last.pt. Un total de 0.803 horas, es decir, 48 minutos fue el tiempo total que tomó el servidor de Google Colab para el entrenamiento completo.

En la Figura 33 se detalla todos los archivos resultantes de entrenamiento, considerando las matrices de confusión, las curvas de precisión y las de recall.

```

confusion_matrix.png
events.out.tfevents.1734478020.e06a6f00261b.2654.0
F1_curve.png
hyp.yaml
labels_correlogram.jpg
labels.jpg
opt.yaml
P_curve.png
PR_curve.png
R_curve.png
results.csv
results.png
train_batch0.jpg
train_batch1.jpg
train_batch2.jpg
val_batch0_labels.jpg
val_batch1_labels.jpg
val_batch1_pred.jpg
val_batch2_labels.jpg
val_batch2_pred.jpg
weights/
val_batch0_labels.jpg

```

Figura 33. Resultados de entrenamiento

a. Matriz de confusión normalizada

A partir de los archivos obtenidos después del entrenamiento, se obtuvo la matriz de confusión normalizada, la cual evalúa el rendimiento del modelo en base a las clases presentadas “Arma Corta”, “Arma Larga” e “Intruso”. En el análisis resultante, se observó que el modelo fue capaz de identificar correctamente valores superiores al 90%, correspondientes a los valores presentados en la Tabla 31.

Tabla 31. Valores de predicción obtenidos

Clases	Valor	Porcentaje
Arma Corta	0.90	90%
Arma Larga	0.92	92%
Intruso	0.97	97%

En la Figura 34 se muestran los valores obtenidos con respecto a las predicciones realizadas durante el entrenamiento en comparación de cada una de las clases utilizadas en el modelo.

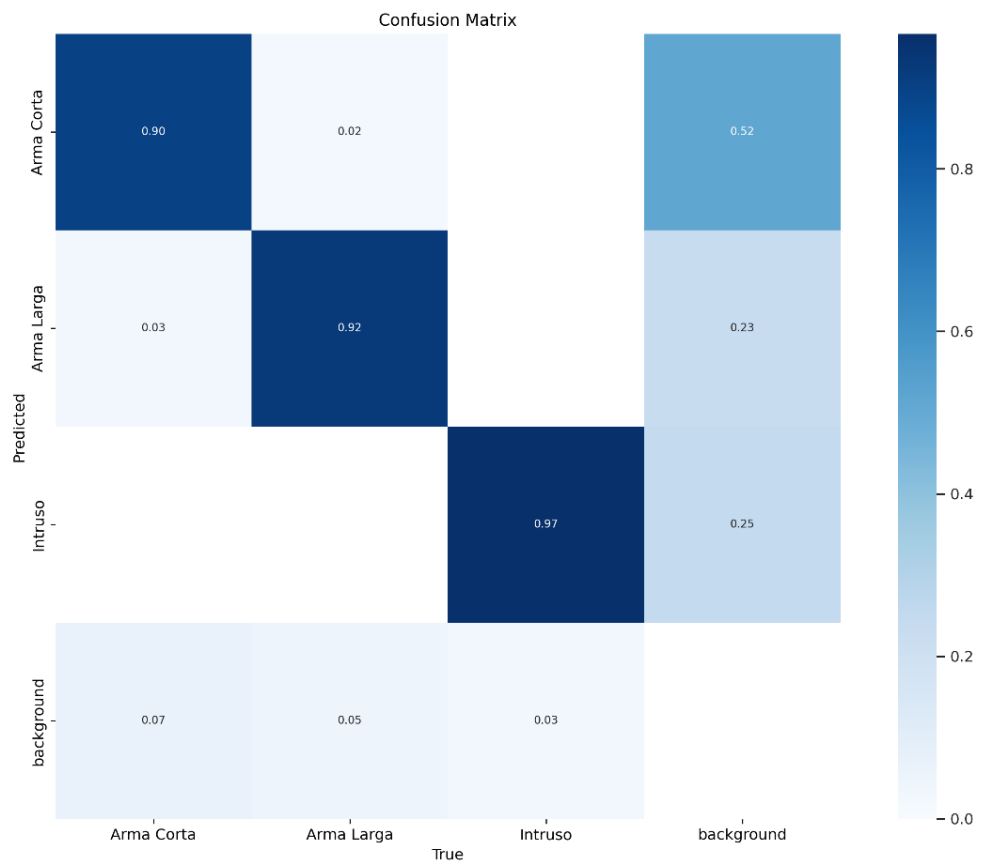


Figura 34. Matriz de confusión

b. Métricas de evaluación

- **Confianza**

La puntuación F1 se evalúa en diferentes niveles de confianza, donde un valor máximo refleja un desempeño óptimo del modelo. En la Figura 35 se muestra la curva de confianza de cada clase del modelo entrenado, siendo el pico mas alto para la clase “Intruso” con un valor de 0.91 en comparación al punto de inicio de 0.44, dando a entender que el modelo tiene una confianza entre la media armónica de precisión y recuperación del 91%.

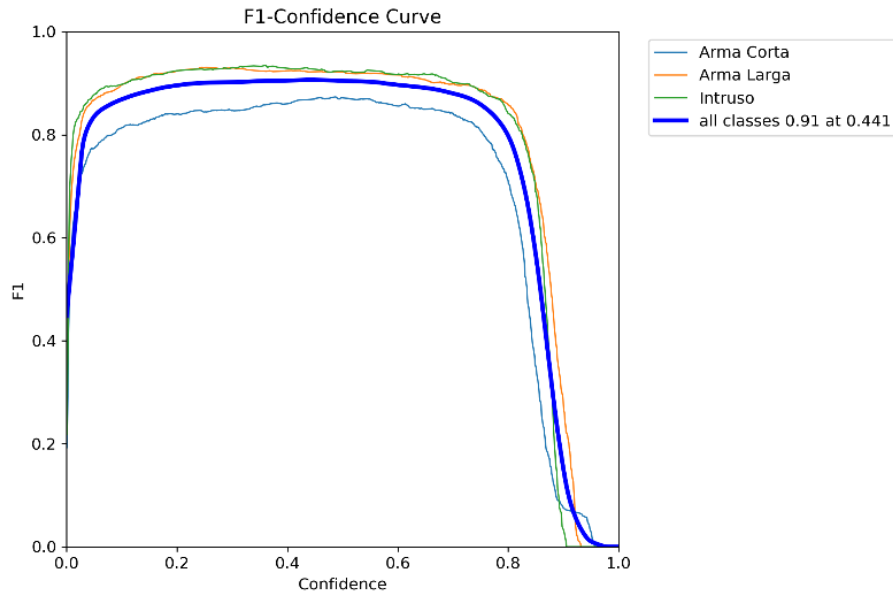


Figura 35. Curva de confianza - F1

- **Precisión**

Es la cantidad de predicciones correctas entre todas las realizadas para una clase determinada. Mide qué proporción de las predicciones positivas del modelo corresponde efectivamente a casos correctos, reflejando la relación entre los verdaderos positivos y el total de positivos predichos. En la Figura 36 correspondiente a la gráfica de precisión, el ultimo valor presentado es 0.911 (véase la Tabla 30), en la época 65, dando a entender que el modelo tiene buena precisión, siendo exactos en un 91% de las veces en las se realizaron las predicciones con otros objetos.

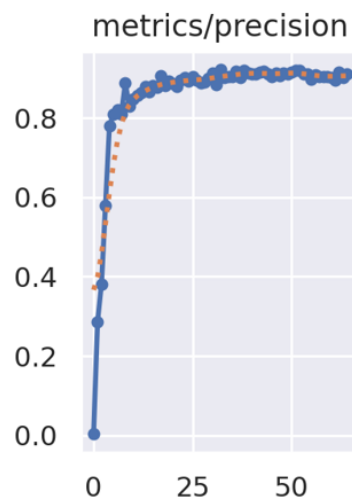


Figura 36. Métrica de precisión

En la Figura 37 se observa la curva de precisión y confianza, en donde el valor máximo alcanzado por el modelo es de 0.946 de precisión con respecto a la curva de confianza del modelo. Mientras más precisión haya en cada nivel de confianza, el modelo es mejor.

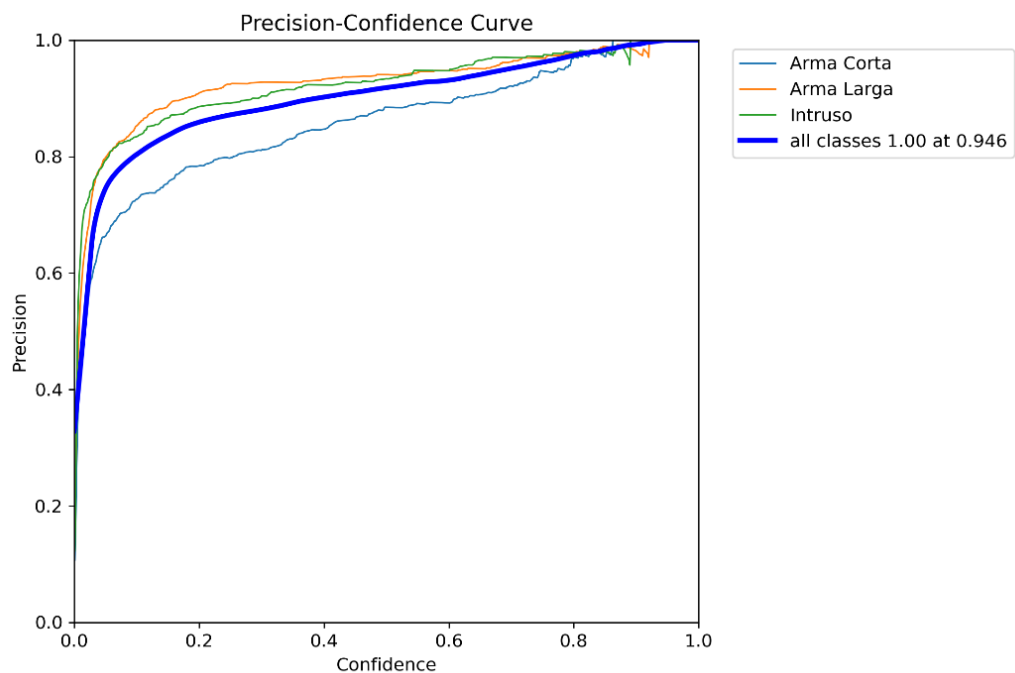


Figura 37. Curva de precisión y confianza

- **Sensibilidad (Recall)**

Es la proporción de objetos reales correctamente detectados frente al total de objetos presentes en la imagen, lo que mide la capacidad del modelo para reconocer exhaustivamente todos los objetos de una clase específica. En la Figura 38 se nota un incremento continuo que se nivela alrededor del 0.92 al finalizar las 65 épocas (véase la Tabla 30), por lo tanto, esto quiere decir que el modelo logra identificar aproximadamente un 92% de objetos de las diferentes clases en los datos usados para el entrenamiento.

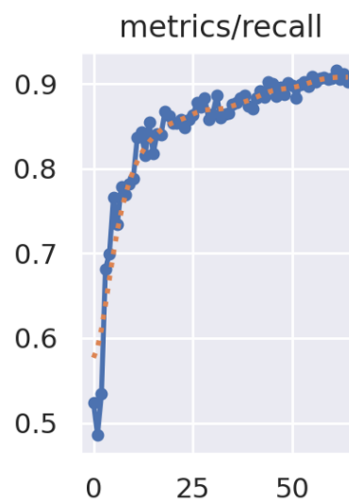


Figura 38. Métrica de recall

En la Figura 39 Se observa que el modelo alcanzó un recall de 98% para todas las clases combinadas en el modelo. Esto refleja que el modelo tiene una alta capacidad para identificar correctamente la mayoría de los objetos en el conjunto de datos.

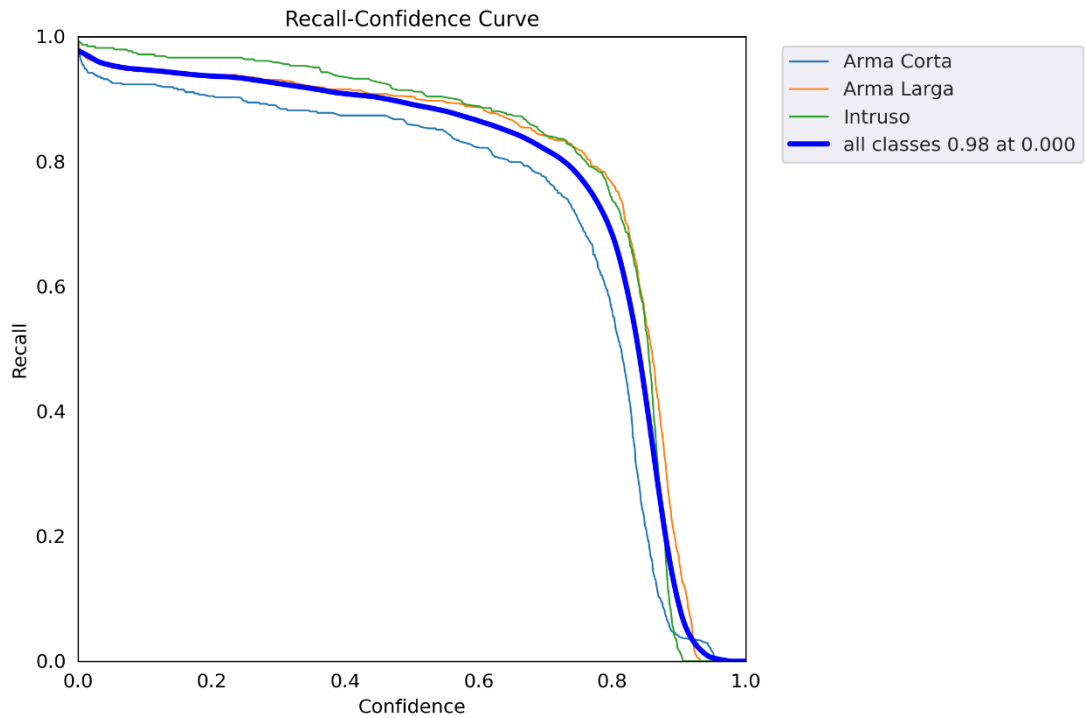


Figura 39. Curva de sensibilidad

- **Métrica mAP_{50}**

Es una métrica que evalúa la precisión promedio del modelo al identificar correctamente los objetos en un conjunto de datos, considerando todas las clases de manera integral. Con un resultado de 0.942 obtenido después de 65 épocas de entrenamiento (véase la Tabla 30) indica que el modelo tiene un alto rendimiento en la detección de los objetos, logrando un 94.2% de precisión promedio.

La Figura 40 muestra la evolución del mAP_{50} a lo largo de las épocas, donde se observa un incremento progresivo hasta estabilizarse cerca de este valor.

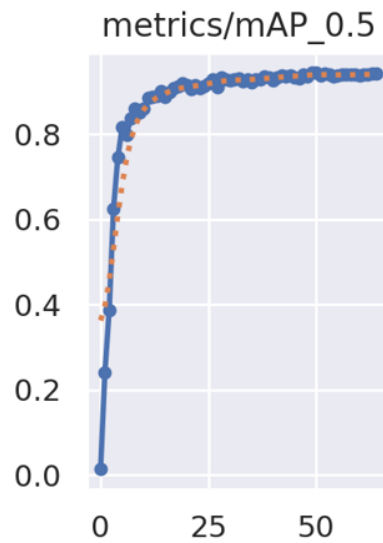


Figura 40. Métrica mAP50

- ***Métrica mAP_{50:95}***

Es una métrica más exigente que calcula la precisión promedio del modelo, incorporando diversos niveles de coincidencia entre las predicciones y las etiquetas reales. Un resultado de 0.55 tras 65 épocas de entrenamiento (véase la Tabla 30) sugiere que el modelo logra un 55% de precisión media en este rango de umbrales, indicando un rendimiento moderado en la detección de objetos bajo condiciones más rigurosas.

La Figura 41 muestra cómo el mAP50:95 evoluciona a lo largo de las épocas, generalmente comenzando con valores bajos y aumentando gradualmente hasta estabilizarse cerca del valor final.

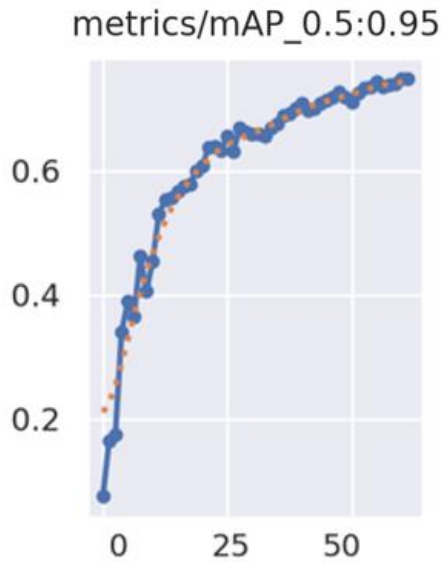


Figura 41. Métrica mAP50:95

- **Resultados generales**

En la Figura 42 se observa los resultados generales del entrenamiento, pero basado en las pérdidas, enfocado principalmente en los valores de train, val, y las métricas con respecto a los valores de precisión, recall, mAP05, y mAP05:0,95

En las gráficas de entrenamiento (train), se observó una disminución notable de las pérdidas a medida que el modelo aprendía.

Por otro lado, las gráficas de validación (val) mostró como el modelo se generaliza hacia los datos no vistos.

A lo largo de las épocas, es común observar una rápida disminución de las pérdidas en las etapas iniciales del entrenamiento, seguida de una estabilización progresiva a medida que el modelo se acerca a la convergencia.

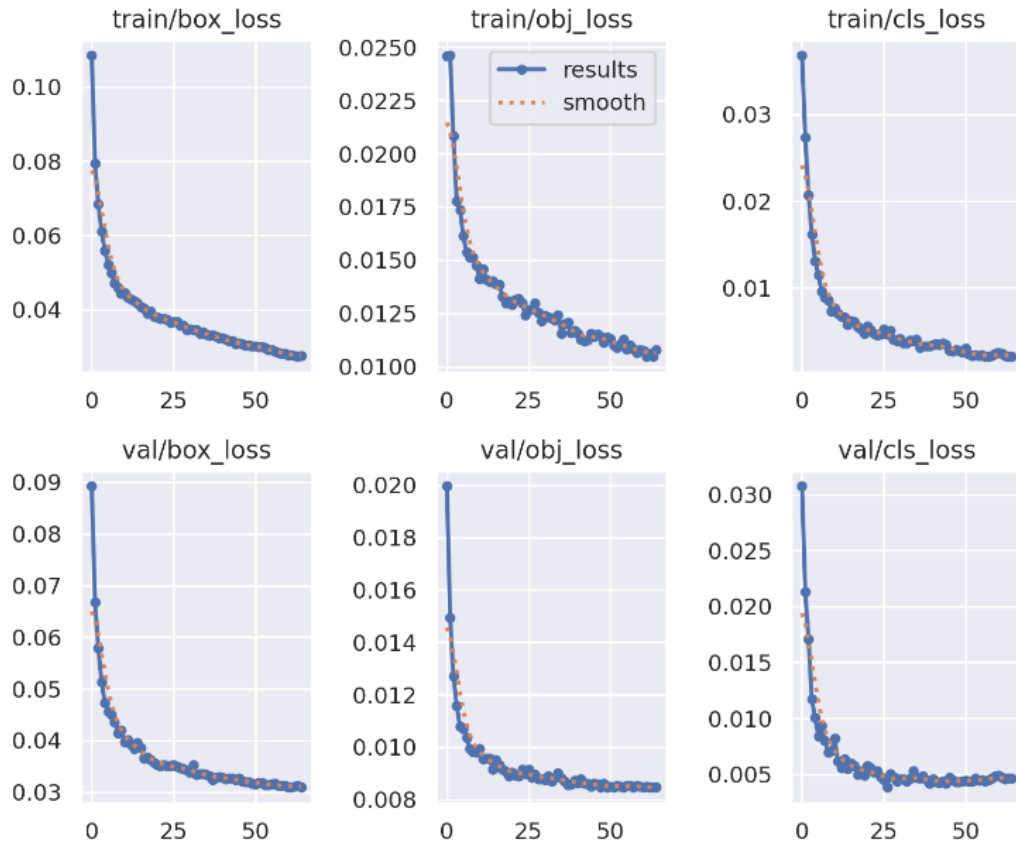


Figura 42. Resultados generales

c. Validación de resultados

Para la validación de resultados, es necesario utilizar los valores de la matriz de confusión normalizada (Figura 34), y el uso de la cantidad de datos utilizados en el entrenamiento (Tabla 20).

Mediante la Ecuación 1, se calcula la cantidad en números enteros de VP, VN, FP, y FN, esto en base a la cantidad de muestra puestas en el dataset.

$$TP, TN, FP, FN = Valor(M_{fn}) * Cant. Dataset \quad (1)$$

En donde:

TP = Verdaderos positivos

TN = Verdaderos Negativos

FP = Falsos Positivos

FN = Falsos Negativos

Mfn = Matriz de confusión normalizada

Por lo tanto, los resultados obtenidos en base a la matriz de confusión son los valores expresados en la Tabla 32.

Tabla 32. Matriz de confusión no normalizada

Clase	TP	FN	FP	TN
Arma corta	1549	912	173	4250
Arma larga	1584	448	120	4732
Intruso	1670	430	52	4732

Para conocer el rendimiento total del sistema, se consideró 4 métricas fundamentales en los temas de entrenamientos de inteligencia artificial, los cuales son: precisión, sensibilidad (recall), F1_score y exactitud.

La Ecuación 2 muestra cómo se obtiene el valor de Sensibilidad.

$$Precisión = \frac{TP}{FP+FN} \quad (2)$$

Para calcular el valor correspondiente a la Sensibilidad, se aplica la Ecuación 3.

$$Sensibilidad = \frac{TP}{TP + FN} \quad (3)$$

El valor correspondiente a F1_Score viene dado por la Ecuación 4.

$$Sensibilidad = 2 * \left(\frac{Precisión * Recall}{Precisión + Recall} \right) \quad (4)$$

Por último, el valor de exactitud se calcula mediante la Ecuación 5.

$$\frac{TP + VN}{TP + FP + TN + FN} \quad (5)$$

Reemplazando valores para cada clase determinada, se obtuvo los valores correspondientes para cada métrica, presentados en la Tabla 33.

Tabla 33. Valores de métricas obtenidos

Clase	Precisión	Sensibilidad	F1-score	Exactitud
Arma corta	0.899	0.729	0.739	0.842
Arma larga	0.930	0.779	0.848	0.918
Intruso	0.970	0.795	0.875	0.930
Total	0.923	0.759	0.820	0.896

Para la obtención de la eficiencia total del sistema, aplicamos la Ecuación 6.

$$\text{Eficiencia del sistema} = \left(\frac{\text{Precisión} + \text{Sensibilidad} + \text{F1} + \text{Exactitud}}{4} \right) * 100\% \quad (6)$$

$$\text{Eficiencia del sistema} = \left(\frac{0.923 + 0.759 + 0.820 + 0.896}{4} \right) * 100\%$$

$$\text{Eficiencia del sistema} = \left(\frac{3.398}{4} \right) * 100\%$$

$$\text{Eficiencia del sistema} = (0.8495) * 100\%$$

$$\text{Eficiencia del sistema} = 84.9\% \approx 85\%$$

$$\text{Eficiencia del sistema} = 85\%$$

Los cálculos totales indican que el desempeño o eficiencia total del sistema es del 85%, siendo un valor aceptable dentro del ámbito de la inteligencia artificial, siendo así un valor aceptable para la implementación directa en sistemas CCTV.

3.2.2 Pruebas de funcionamiento

a. Interfaz de visualización

Como resultado final, se tiene una interfaz principal de inicio, la cual, al digitar correctamente las credenciales de usuario, ejecuta la ventana principal de visualización de las cámaras en tiempo real.

En la Figura 43 se puede observar la interfaz de login o acceso de credenciales.



The image shows a web browser window titled 'Inicio de Sesión'. Inside the window, there is a light gray box containing the text 'Iniciar Sesión' at the top. Below this text are two input fields: the first is labeled 'Usuario' and the second is labeled 'Contraseña'. Below the input fields is a blue button with the text 'Login'.

Figura 43. Ingreso de credenciales

Cuando un inicio de sesión se da de forma incorrecta, ya sea en el usuario o contraseña, la interfaz despliega un mensaje de aviso en color rojo en la parte inferior, así como se muestra en la Figura 44.



The image shows the same web browser window as in Figure 43, but with updated content. The 'Usuario' input field now contains the text 'Jorge'. The 'Contraseña' input field contains a series of asterisks '*****'. Below the 'Login' button, there is a red error message that reads 'Usuario o contraseña incorrectos'.

Figura 44. Credenciales incorrectas

Con el usuario y contraseña correctos, el acceso hacia la interfaz de visualización de las cámaras de en tiempo real es inmediato, como se indica en la Figura 45.

Al momento de la ejecución de la interfaz, se da sin la visualización de las cámaras, pero con las opciones de activar cámara 1 o cámara 2, estas se ejecutan de forma fácil. Igualmente, también consta del botón cerrar, con el cual la finalización del programa se realiza de forma inmediata.

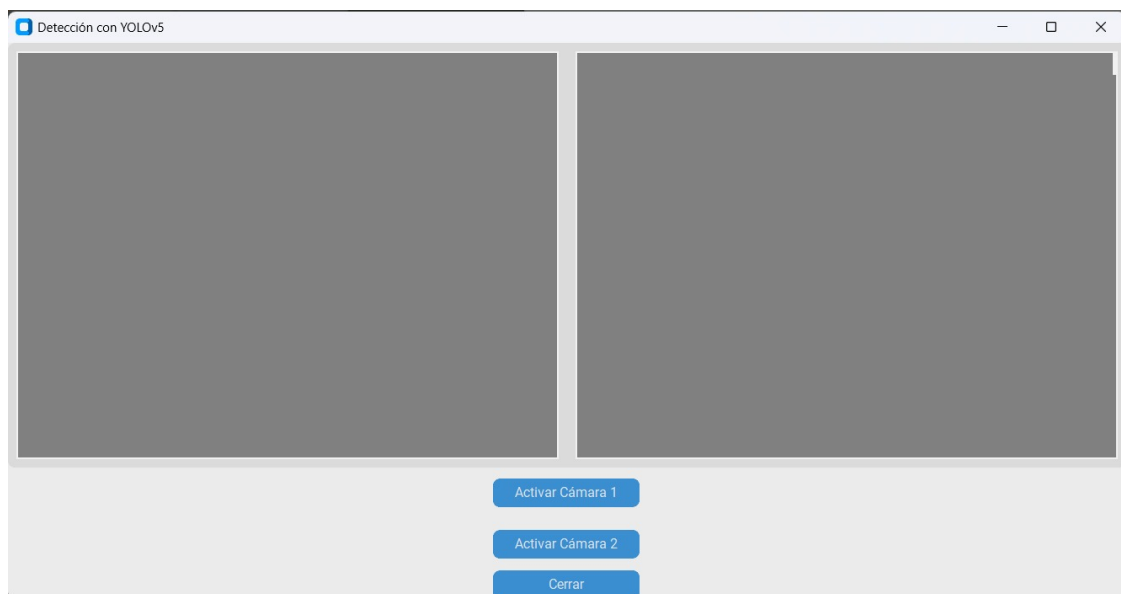


Figura 45. Interfaz de visualización de cámaras en tiempo real

Cabe recalcar que al momento de ingresar desde la interfaz de login, el programa está ejecutándose directamente ya con el modelo entrenado, entonces, únicamente estaría a la espera de la activación de alguna cámara para la detección de algún arma.

La Figura 46 muestra la cámara 1 activada, mientras que la cámara 2 permanece desactivada. En este caso, el algoritmo de detección de armas largas y cortas no detecta ningún objeto en el área de visión, puesto que no hay armas presentes en la escena. Es importante destacar que, como se mencionó previamente, el algoritmo de detección se aplica desde la ejecución de la interfaz de inicio de sesión, lo que garantiza que el proceso de monitoreo y detección esté en funcionamiento, pero en este escenario específico no se han detectado armas debido a la ausencia de estas en el campo de visión.

En la Figura 47 se muestra el mismo proceso, pero con la diferencia de que la cámara 1 está desactivada y la cámara 2 permanece activa. En este caso, el algoritmo de detección se activa en el momento en que se identifica un objeto de interés dentro del área de visión de la cámara 2, ya sea un arma de fuego o un intruso. La detección solo se activa cuando el sistema reconoce el objeto previsto, asegurando que el monitoreo se enfoque en la identificación de posibles amenazas, como armas, dentro de la escena capturada por la cámara activa.

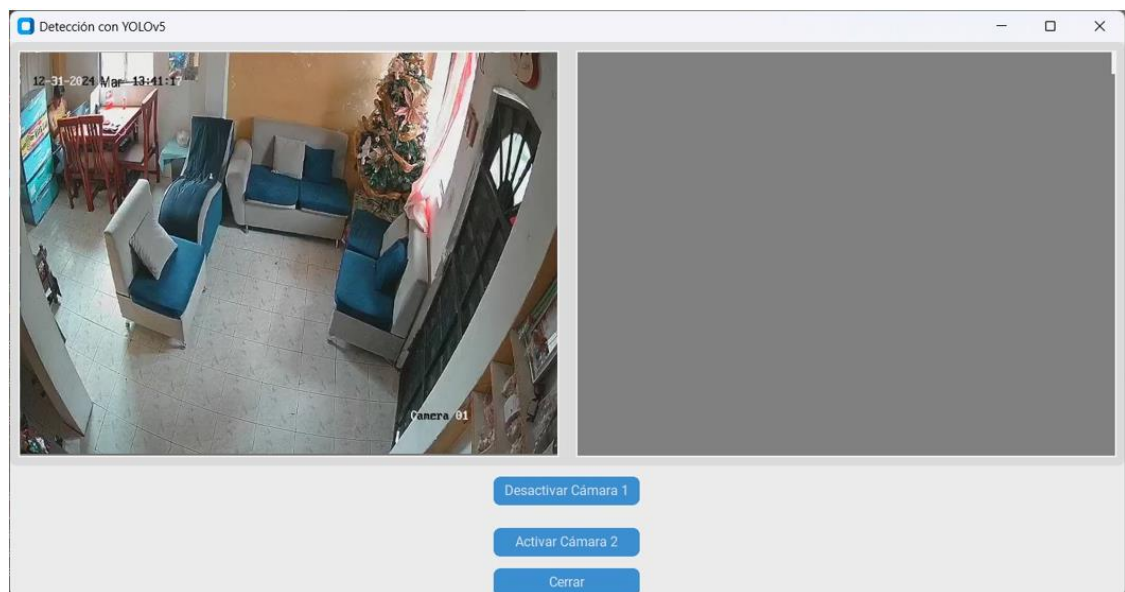


Figura 46. Activación cámara 1 – sin detección

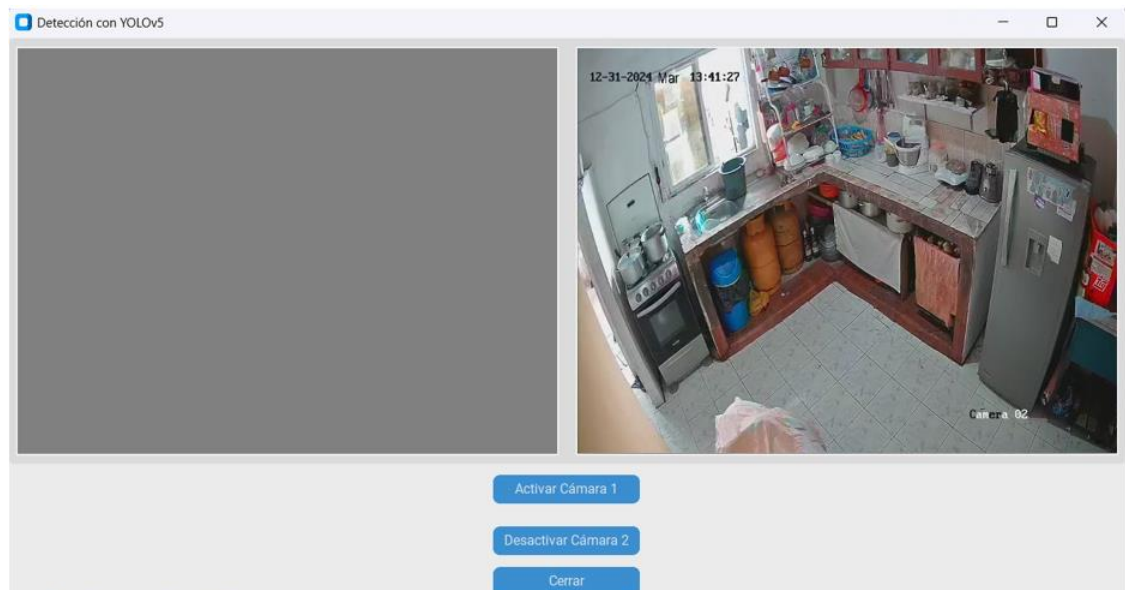


Figura 47. Activación cámara 2 – sin detección

En la Figura 48 muestra tanto la cámara 1 como la cámara 2 en estado activo, pero en este caso, no se realiza ninguna detección. Esto indica que, aunque ambos sistemas de cámaras están funcionando correctamente, no se ha identificado ninguna de las clases correspondientes al modelo de detección dentro del área de visión de cada dispositivo.

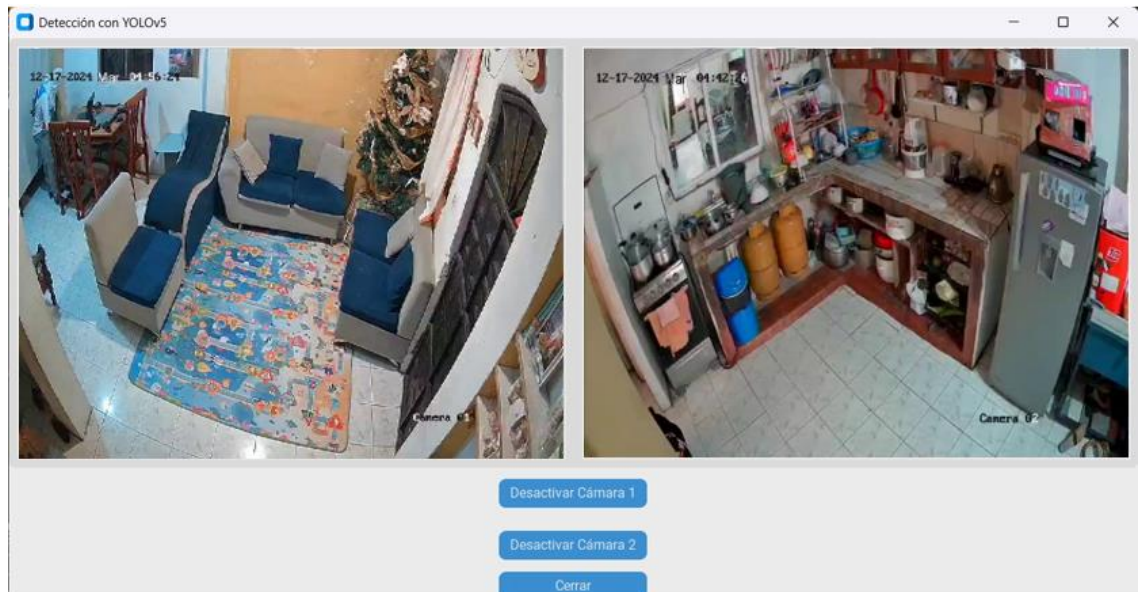


Figura 48. Cámara 1 y 2 activadas – sin detección

b. Interfaz con detecciones

Cuando las cámaras de vigilancia están en estado activo, el área que cubren está a la espera del enfoque de un objeto perteneciente a las 3 clases con las cuales se entrenó el modelo.

- ***Detección de armas cortas***

Cuando las clases a detectar ingresan dentro del área de visualización de las cámaras de vigilancia, el sistema detecta el objeto y lo membreta de acuerdo con la clase que corresponda.

En la Figura 49 se muestra la detección de un objeto correspondiente a la clase “Arma Corta”, a través de la cámara 1, presentado en la interfaz, donde la cámara 2 permanece desactivada.

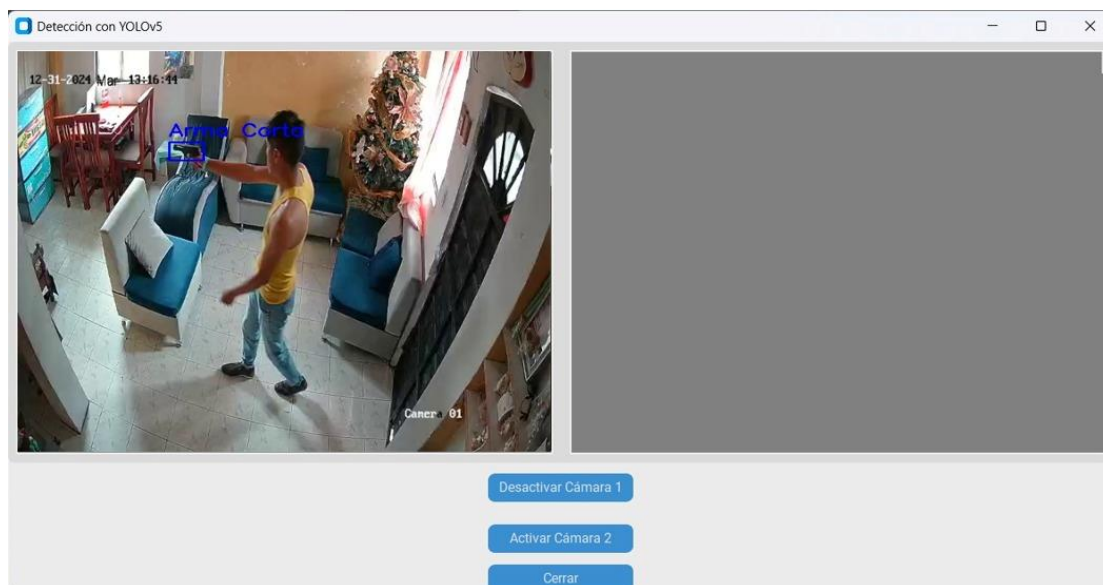


Figura 49. Detección de arma corta – cámara 1

Al cambiar de área de visualización, pero con el modelo implementado el resultado de detección es el mismo. En la Figura 50 se muestra igualmente la detección de un objeto correspondiente a la clase “Arma Corta” pero esta vez a través de la cámara 2, presentado en la interfaz, donde ahora la cámara 1 permanece desactivada.

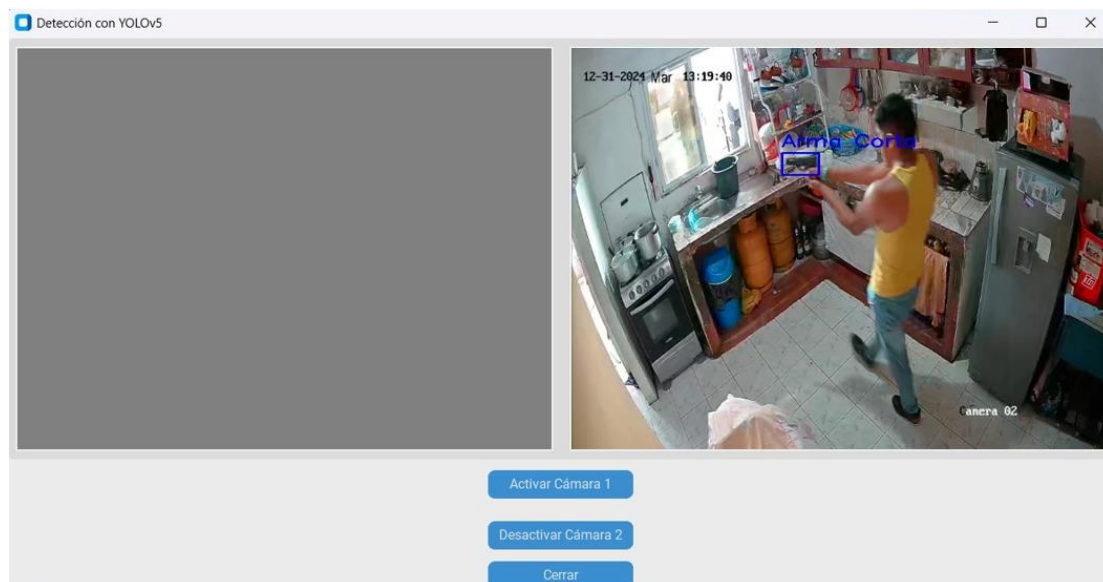


Figura 50. Detección de arma corta – cámara 2

- ***Detección de armas largas***

El modo de detección del objeto es el mismo para cualquiera de las tres clases definidas. Dentro del área de visualización, el enfoque principal del sistema es identificar los objetos presentes que correspondan a las clases configuradas.

La Figura 51 muestra la detección de un objeto correspondiente a la clase “Arma Larga”, a través de la cámara 1, presentado en la interfaz.

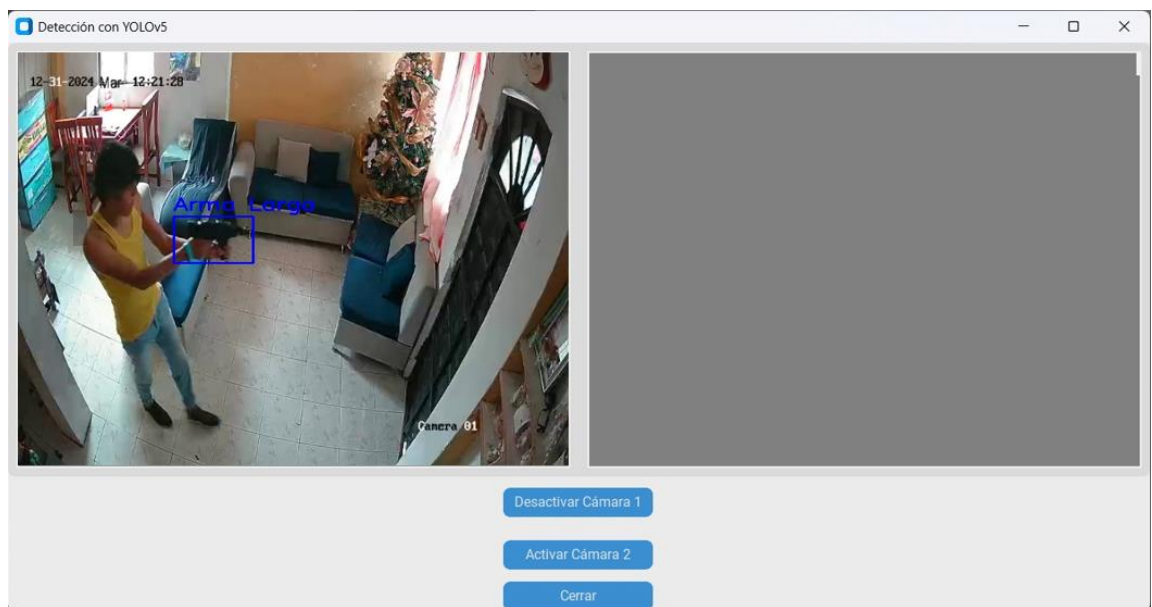


Figura 51. Detección de arma larga – cámara 1

En la Figura 52 se observa la detección de un objeto correspondiente a la clase “Arma larga”, esto en vista de que una metralleta, que es considerada un arma larga (véase la Tabla 9) entró en el área que cubre la cámara 2, de tal manera que el sistema de detección la clasificó como un arma larga.

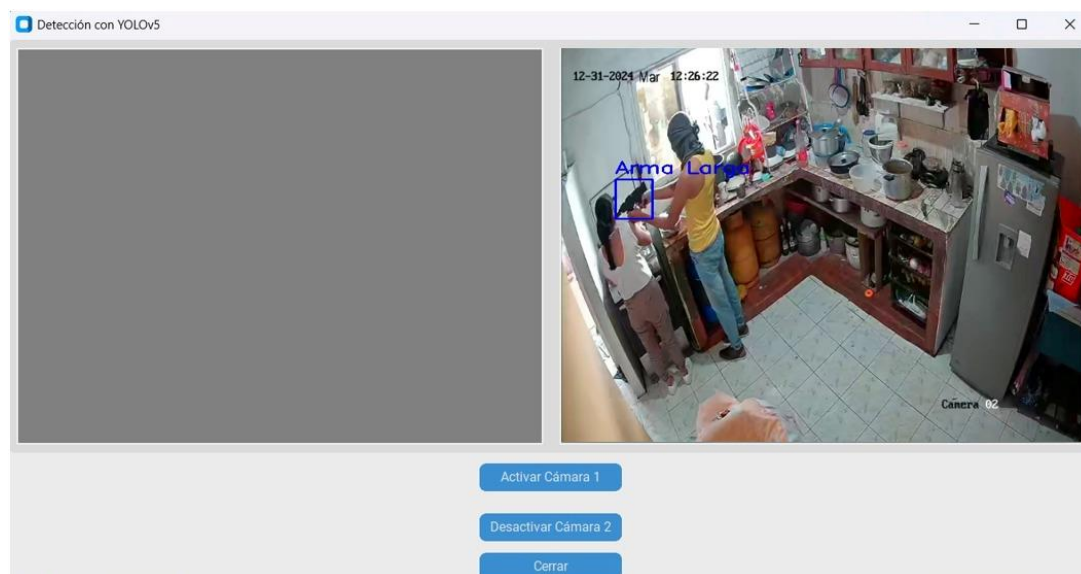


Figura 52. Detección de arma larga – cámara 2

- ***Detección de intrusos***

La clase “Intruso” viene detallada a partir de rostros cubiertos ya sea por camisetas, trapos, bufandas, pasamontañas, etc. En la Figura 53 se observa la detección de un objeto que ha sido etiquetado con el nombre intruso, esto en el área cubierta por la cámara 1.

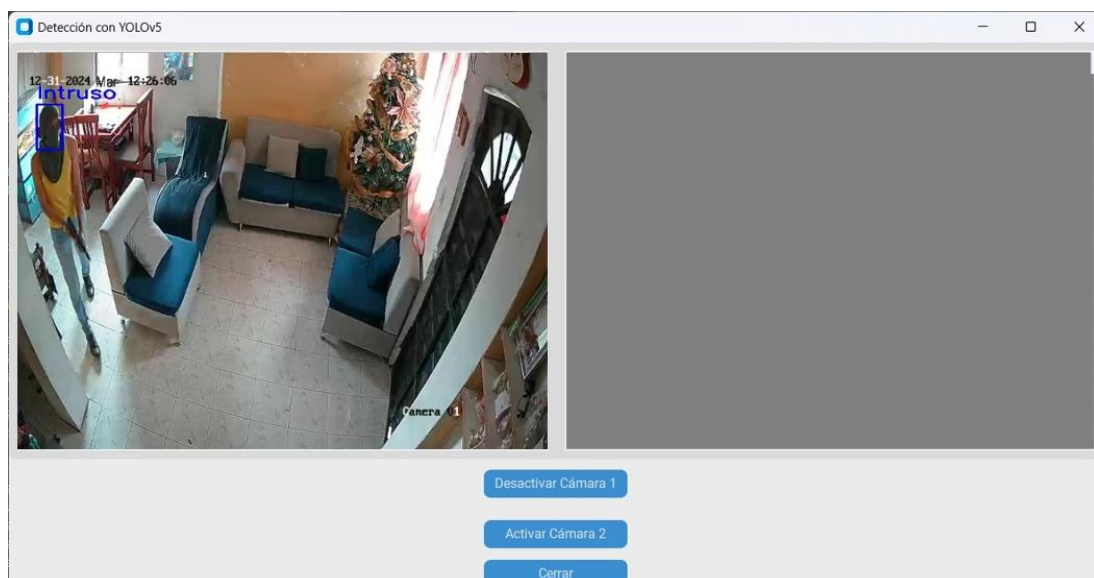


Figura 53. Detección de intruso – cámara 1

En la zona de visualización correspondiente a la cámara 2, se observa mediante la Figura 54 la detección de la clase “Intruso”, pero a su vez la detección de un objeto que corresponde a la clase “Arma Larga”.

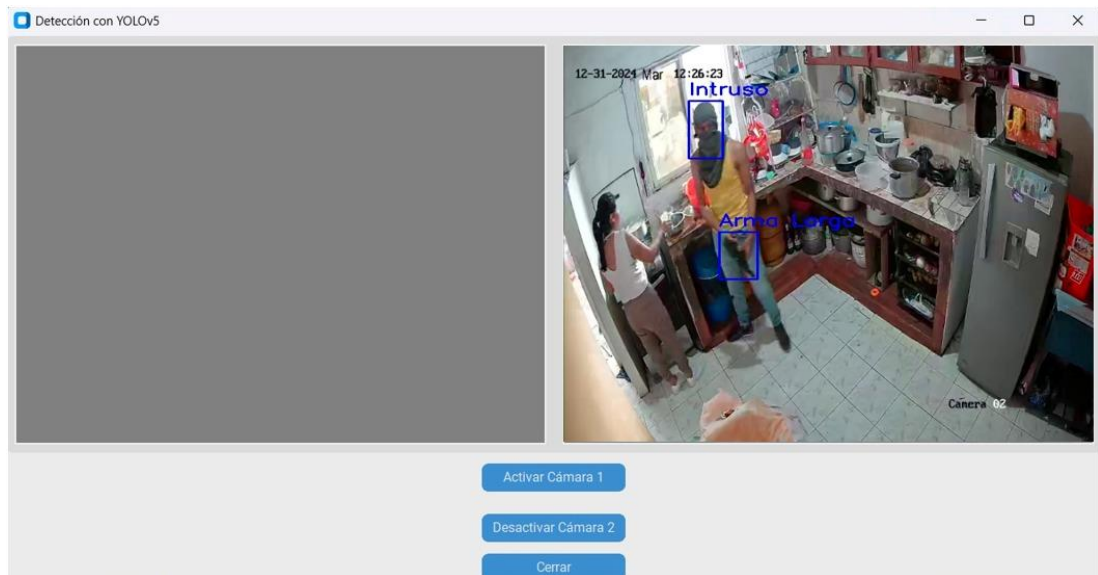


Figura 54. Detección de intruso – cámara 2

- ***Detección múltiple***

Las distintas áreas de visualización cuentan con la capacidad de identificar múltiples objetos pertenecientes a las clases previamente mencionadas. Como se muestra en la Figura 55, el sistema logró detectar simultáneamente un intruso y un arma larga dentro del área correspondiente a la cámara 1.

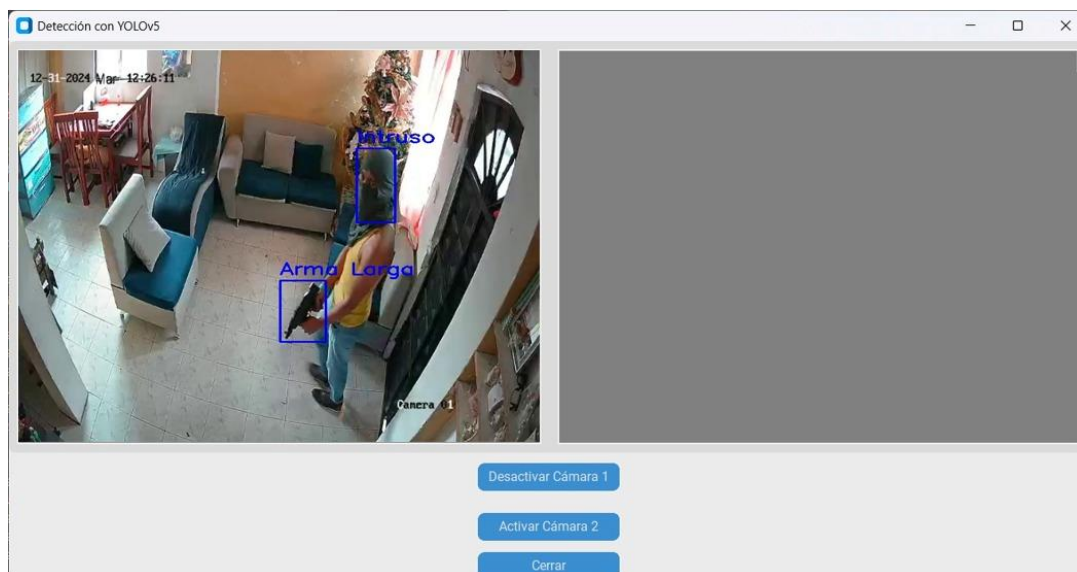


Figura 55. Detección de arma larga e intruso – cámara 1

En la Figura 56 se observa la detección de un arma larga y corta, pertenecientes a las clases “Arma Larga” y “Arma Corta” respectivamente, de la misma forma en el área cubierta por la cámara 1.

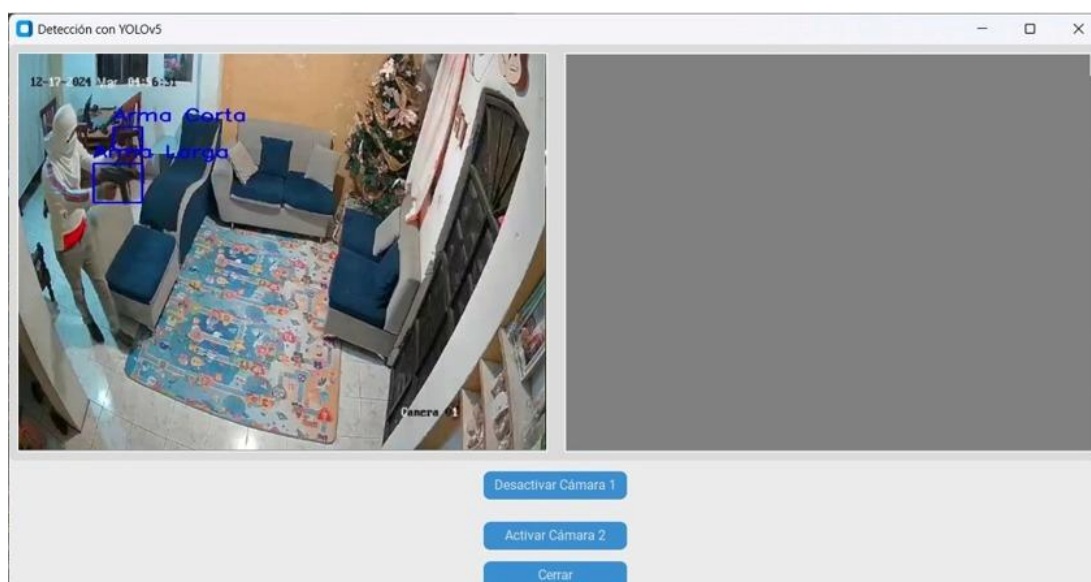


Figura 56. Detección de un arma larga y corta – cámara 1

c. Almacenamiento de imágenes detectadas

A medida que el sistema vaya detectando armas de fuego, ya sean largas o cortas, y también intrusos, el programa las guarda en una carpeta especificada en el servidor, así como se observa en la Figura 57.

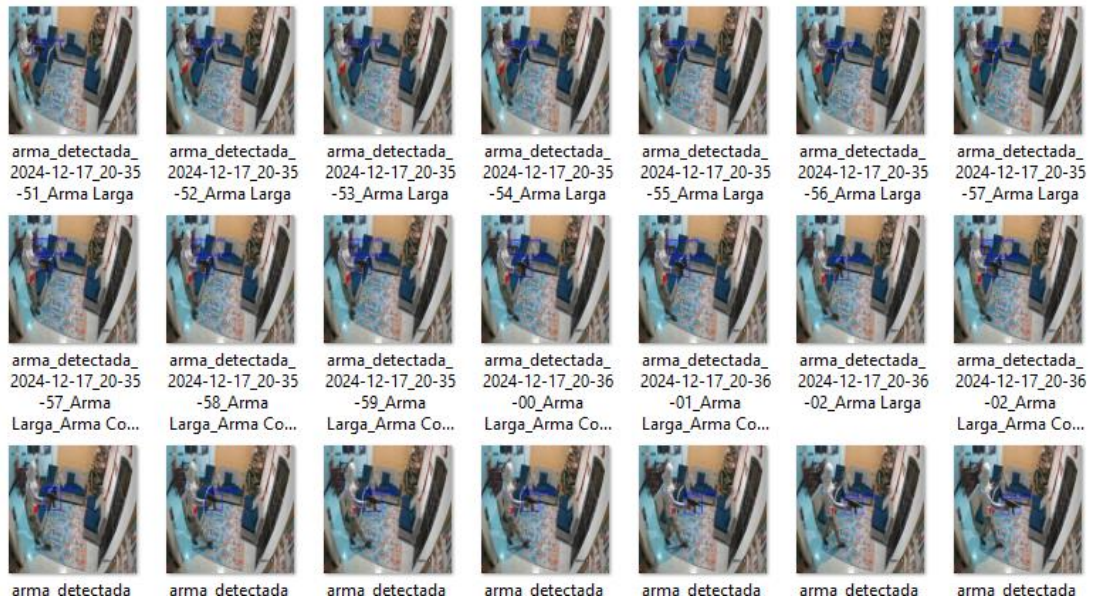


Figura 57. Imágenes de detección guardadas

El almacenamiento de estas imágenes está dado a partir de la detección de objeto. El aviso de “arma detectada”, la fecha, el número de imagen y la clase correspondiente son las características con las cuales se almacenan cada una de estas imágenes, así como se muestra en la Figura 58.

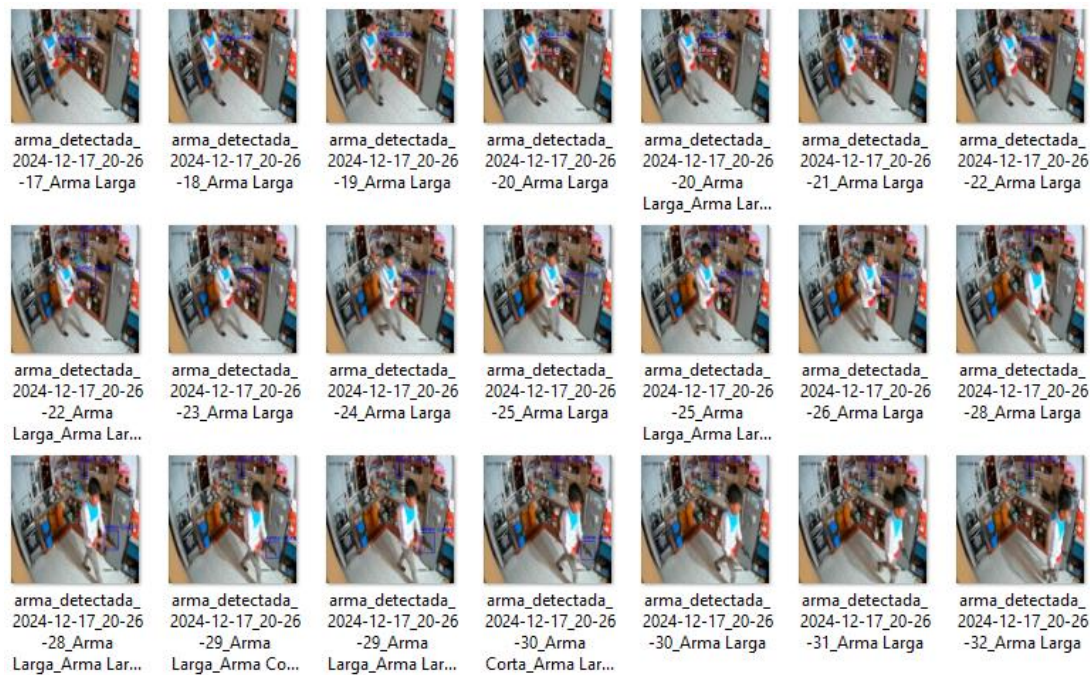


Figura 58. Membretado de imágenes almacenadas

3.2.3 Pruebas de generación de alertas

Cuando el sistema identifica un objeto perteneciente a las clases “Arma Larga,” “Arma Corta,” o “Intruso,” genera alertas de manera inmediata con el objetivo de advertir y asistir a las posibles víctimas del delito que pudiera estar ocurriendo en ese momento.

a. Activación de sirena

Una sirena es un artefacto que emite un sonido de largo alcance, esta es usada normalmente en sistemas de alarmas para avisos, advertencias hasta prevenciones.

El sistema detector de armas de fuego activa una alerta sonora la cual está dada por la sirena cuando las cámaras de seguridad detectan un arma o un intruso. En el momento de activación de la sirena, la consola del programa emite diferentes datos, los cuales se pueden observar en la Figura 59.

```
Detectado: Arma Larga
Alarma sonora generada
Llamada a +593987584107 realizada
Detectado: Arma Larga
Alarma sonora generada
Llamada a +593987584107 realizada
Detectado: Arma Larga
Alarma sonora generada
Llamada a +593987584107 realizada
Detectado: Arma Larga
Alarma sonora generada
Llamada a +593987584107 realizada
Detectado: Arma Larga
```

Figura 59. Alerta activada

Al momento de la impresión de resultados en la consola, se observa que se detalla otro dato, el cual es el tipo de objeto detectado, y este corresponde al tipo de arma de fuego y al intruso. En la Figura 60 se puede observar la clase detectada, en ese caso “Arma Larga”

```
Detectado: Arma Larga
Alarma sonora generada
Llamada a +593987584107 realizada
Detectado: Arma Larga
Alarma sonora generada
Llamada a +593987584107 realizada
Detectado: Arma Larga
Alarma sonora generada
Llamada a +593987584107 realizada
Detectado: Arma Larga
Alarma sonora generada
Llamada a +593987584107 realizada
Detectado: Arma Larga
```

Figura 60. Detección de clases

b. Llamada telefónica resultante

Después de la activación de la sirena, la llamada telefónica viene dada a partir del módulo SIM800L que al recibir una de las 3 clases detectadas, procede a comunicarse mediante vía serial con la esp32, iniciando la activación del comando AT para posteriormente realizar la llamada al número telefónico configurado.

En la Figura 61 se observa el aviso que hace el programa a través de consola, esto indicando que se ha realizado una llamada hacia el número de teléfono indicado.

```
Alarma sonora generada  
Llamada a +593987584107 realizada  
Detectado: Arma Corta  
Alarma sonora generada  
Llamada a +593987584107 realizada
```

Figura 61. Generación de llamada telefónica

Cabe recalcar que las llamadas se generan a partir de una tarjeta SIM que está incrustada en el módulo SIM800L, por tal motivo las llamadas de emergencia que recibe el destinatario vienen dadas con el nombre del dicho contacto, en este caso se nombró como “Emergencia” a la tarjeta SIM emisora de la llamada. En la Figura 62 se observa el recibimiento de la llamada telefónica proveniente del sistema detector de armas de fuego.



Figura 62. Llamada de emergencia generada a partir de la detección

3.3 Costos

En esta sección se detallan los costos asociados a los recursos económicos y la mano de obra que fue necesaria para realizar el estudio.

3.3.1 Costos de insumos del proyecto

En la Tabla 34 se expresan los costos de cada elemento que se utilizó a lo largo del estudio.

Tabla 34. Costos de insumos del proyecto

Material	Cantidad	Valor Unitario	Valor Total
NVIDIA Jetson Nano	1	\$200,00	\$200,00
Grabador de video digital (DVR)	1	\$45,00	\$45,00
Disco Duro para DVR marca WD PURPLE de 1TB	1	\$75,00	\$75,00
Cámara domo	1	\$22,00	\$22,00
Cámara tipo tubo	1	\$25,00	\$25,00
UPS	1	\$85,00	\$85,00
Fuentes 12v	2	\$5,00	\$10,00
Video Baluns	4	\$2,00	\$8,00
Conectores	4	\$1,00	\$4,00
Cable UTP CAT6	30 metros	\$0.30	\$9,00
Materiales de instalación	1	\$30,00	\$30,00
Material de oficina	1	\$20,00	\$20,00
Total			\$533,00

Como resultado, se obtuvo que en materiales e insumos el gasto total fue 533,00 dólares americanos.

3.3.2 Costos de desarrollo técnico

Para calcular este valor, se toma en cuenta el número de horas laborales dedicadas al desarrollo del proyecto, así como lo muestra la Tabla 35.

Tabla 35.Total de horas laborales utilizadas en el proyecto

Descripción	Cantidad
Tiempo empleado en el proyecto, desde investigación, desarrollo y ejecución de pruebas de funcionamiento	30 días
Horas laborales	8 horas
Horas totales	240 horas

El sueldo de un ingeniero en Telecomunicaciones enfocado directamente al área de programación, dado por el ministerio de trabajo en el 2024, establece que un ingeniero en telecomunicaciones recibe un sueldo básico de 494\$, con esta información se obtuvo el valor por hora de lo que gana un ingeniero en telecomunicaciones, considerando 20 días laborales de los 30 que tiene normalmente un mes, y en cada día 8 horas de trabajo, así como se muestra en la Tabla 36.

Tabla 36. Salario por hora de un ingeniero en Telecomunicaciones

Descripción	Cantidad
Valor mensual	494
Valor diario	24,7
Valor hora	3,08

Ya con estos valores obtenidos, el costo del desarrollo técnico viene dado por la Ecuación 7.

$$\text{Costo desarrollo técnico} = \text{Valor hora} * \text{Horas totales} \quad (7)$$

$$\text{Costo desarrollo técnico} = 3,08 * 240$$

$$\text{Costo desarrollo técnico} = 739,20$$

3.3.3 Costo total

La Ecuación 8 muestra el costo total de la investigación.

$$\text{Costo total} = \text{Costo desarrollo técnico} + \text{Costo insumos} \quad (8)$$

$$\text{Costo total} = 739,20 + 533,00$$

$$\text{Costo total} = 1272,20$$

Por lo tanto, la totalidad del proyecto está valorizada en 1272,20 dólares americanos.

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- La situación actual de robos y asaltos con armas de fuego en Ecuador refleja una grave crisis de seguridad marcada por el aumento de la violencia, la sofisticación de los métodos delictivos y el uso generalizado de armas letales por parte de grupos organizados. A pesar de una leve reducción en los índices de robos a viviendas y personas entre 2022 y 2024, la percepción de inseguridad persiste debido a la vulnerabilidad de hogares y transeúntes, así como a la desconfianza hacia las autoridades, con menos del 25% de las víctimas dispuestas a denunciar. Este panorama evidencia la necesidad urgente de estrategias integrales que fortalezcan la seguridad ciudadana, limiten el acceso a armas ilegales y mejoren la eficacia y confianza en las instituciones encargadas del orden.
- Tras un análisis exhaustivo de las características técnicas y los factores económicos, se seleccionó la NVIDIA Jetson Nano como microordenador principal, destacando por su sobresaliente capacidad de procesamiento y eficiencia en aplicaciones de inteligencia artificial. Para la captura de imágenes en tiempo real, se optó por un sistema CCTV compuesto por dos cámaras de 2MP y un DVR HiLook, reconocido por su alta resolución. Asimismo, se integró un UPS de 450W, capaz de garantizar la continuidad operativa del sistema durante interrupciones eléctricas, proporcionando una autonomía de hasta 4.5 horas. Durante la fase de entrenamiento del modelo, se empleó el algoritmo YOLOv5 en la plataforma Google Colab, aprovechando su alto rendimiento y velocidad para optimizar los tiempos de desarrollo y maximizar la eficiencia del sistema.
- Considerando las múltiples técnicas de machine learning enfocadas en la detección de armas de fuego, se optó por emplear redes neuronales convolucionales (CNNs) debido a su destacada capacidad para resolver problemas de visión por computadora. Dentro de las alternativas disponibles,

YOLOv5 fue seleccionado como el modelo más adecuado gracias a su habilidad para realizar detecciones en tiempo real con elevada precisión y rapidez. Este enfoque de deep learning se ajusta perfectamente a los requerimientos del sistema, permitiendo identificar de manera eficiente distintos tipos de armas de fuego en imágenes y videos. La implementación de YOLOv5 garantiza no solo un alto nivel de precisión en la detección, sino también una óptima velocidad y desempeño, factores esenciales para la eficacia de un sistema de alerta temprana.

- Luego de 65 épocas de entrenamiento intensivo, el modelo demostró una notable efectividad en su proceso de aprendizaje. Se realizaron pruebas operativas para evaluar la capacidad del sistema en la detección de diferentes tipos de armas, como cortas, largas, así como intrusos. Los resultados obtenidos fueron altamente satisfactorios, alcanzando una tasa de aceptabilidad general del 85%. Este desempeño valida la eficacia y fiabilidad del sistema para identificar amenazas potenciales, confirmando su buen rendimiento en condiciones reales de operación.

4.2 Recomendaciones

- El aumento de datos del dataset utilizado durante el proceso de entrenamiento resulta importante para mejorar la precisión del sistema de detección de armas de fuego. Al incrementar tanto la cantidad como la diversidad de las imágenes, el modelo tiene acceso a una mayor variedad de ejemplos, lo cual le permite generalizar mejor y aprender a reconocer distintos tipos de armas bajo diversas condiciones. Este enfoque no solo aumenta la capacidad del sistema para identificar amenazas de manera más precisa, sino que también fortalece su rendimiento en situaciones del mundo real. Es importante tener en cuenta que la efectividad de la detección será mayor si se utiliza la misma cámara con la cual se entrenó el modelo. Esto se debe a que cualquier cambio en la resolución o los megapíxeles puede llevar a que el sistema funcione de manera menos precisa.

- Optar por un sistema que utilice Tensor Processing Units (TPU) en lugar de Graphics Processing Units (GPU) ofrece una ventaja significativa en términos de velocidad y eficiencia en tareas de visión por computadora. Las TPUs están diseñadas para manejar operaciones tensoriales de manera más optimizada, lo que las hace ideales para el procesamiento intensivo de datos que requieren los modelos de aprendizaje automático. Al emplear esta tecnología especializada, se logra una mejora considerable en los tiempos de respuesta y en la eficacia general del sistema, especialmente en aplicaciones que demandan un alto rendimiento en la manipulación de grandes volúmenes de datos.
- El posicionamiento estratégico de las cámaras es fundamental para la calidad del conjunto de datos utilizado en el entrenamiento del modelo. Es importante asegurar que las cámaras estén ubicadas de manera que cubran todas las áreas relevantes y estén orientadas para capturar imágenes con la mayor claridad posible. Además, las características de las cámaras tienen un impacto directo en la calidad de las imágenes. Equipos con mejores especificaciones contribuyen a una captura más precisa de las escenas, lo que permite al modelo aprender patrones más detallados y robustos. Esto, a su vez, mejora el rendimiento en las fases de detección y clasificación, garantizando una mayor efectividad del sistema.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Instituto Nacional de Estadísticas y Censos, «Estadísticas de Seguridad Integral,» Quito, Quito, 2024.
- [2] J. Tamayo, «Delitos contra la propiedad,» Universidad Técnica de Ambato, Ambato, 2020.
- [3] A. Sanchez, T. Vayas, F. Mayorga y C. Freire, «Delincuencia e inseguridad ciudadana en Ambato,» Observatorio Economico y Social de Tungurahua, Ambato, 2022.
- [4] D. Romero, «Desarrollo de un sistema de detección de armas de fuego cortas,» Universidad Politécnica Salesiana de Cuenca, Cuenca, 2019.
- [5] F. Oñate, «Diseño y construcción de nodos inteligentes para detección de armas dentro de una red de videovigilancia utilizando visión artificial,» Escuela Superior Politécnica de Chimborazo, Riobamba, 2020.
- [6] A. Durán, «Detector de personas con armas de fuego a partir de un sistema de visión artificial basado en el análisis de posturas corporales,» Universidad Santo Tomás, Pasto, 2022.
- [7] A. Schcolnik, «Detección de armas tipo pistola mediante el uso de redes convolucionales con una arquitectura tipo yolo y estereoscopía,» Instituto Tecnológico de la Paz, Baja California Sur, 2023.

- [8] E. Sandoval, «Sistema de detección de situaciones delictivas en establecimientos comerciales usando inteligencia artificial,» Universidad Técnica de Ambato, Ambato, 2024.
- [9] J. Muñoz, *Sistemas de Seguridad*, Barcelona: Ediciones Parainfo. S.A., 2000.
- [10] F. Garcia, «Video vigilancia: CCTV usando videos IP,» ELEARNING, Madrid, 2010.
- [11] R. Olmos, S. Tabik, A. Lamas y F. Perez, «A binocular image fusion approach for minimizing false positives in handgun detection with deep learning,» *Information Fusion*, Granada, 2019.
- [12] Llagua y Evelyn, «Sistema de monitoreo de alarmas mediante mensajería SMS e interface hacia un computador para la recepción de eventos de emergencia,» Universidad Técnica de Ambato, Ambato, 2017.
- [13] C. Paguay, «Alarmas comunitarias, una forma de combatir la delincuencia en los barrios,» *La prensa Chimborazo*, pp. 1-2, 03 Noviembre 2021.
- [14] J. M. Merchán, «Diseño e instalación de sistemas de videovigilancia cctv digitales,» EDITOR ANTONIO MADRID VICENTE, Madrid, 2013.
- [15] P. Pazmiño, «Implementación de sistema de CCTV con cámaras IP en las sucursales de una empresa farmacéutica a nivel nacional,» Escuela Politecnica Nacional, Quito, 2019.
- [16] A. Taborda, «Sistema de alertas comunitarias para la ciudad de Barranquilla,» DuraSpace, Barranquilla, 2017.

- [17] W. Correa, C. Nuñez y M. Corredor, Funcionamiento de los sistemas CCTV en la era digital, Santa Marta: Universidad Cooperativa de Colombia, 2018.
- [18] J. García, Diseño e implementación de herramientas de configuración para sistemas transmisores y grabadores de video digital, Barcelona: Universidad Politécnica de Cataluña, 2010.
- [19] «Hikvision,» 2024. [En línea]. Available: <https://www.hikvision.com/es-la/products/HiLook-Turbo-HD-Product/DVR/1080P-Lite/dvr-104g-m1/>. [Último acceso: Enero 2025].
- [20] «Hikvision,» 2022. [En línea]. Available: <https://www.hikvision.com/es/products/Turbo-HD-Products/DVR/Value-Series/ids-7104hghi-m1/>. [Último acceso: Enero 2025].
- [21] «Hivision,» 2023. [En línea]. Available: <https://www.hikvision.com/es-la/products/Turbo-HD-Products/DVR/AcuSense-Series/iDS-7208HQHI-M1-S/>. [Último acceso: Enero 2025].
- [22] «Hiraoka,» Importaciones Hiraoka S.A.C, 2023. [En línea]. [Último acceso: Enero 2025].
- [23] «Comercial Benavides,» 2023. [En línea]. Available: <https://www.comercialbenavides.net/p/hiloo/CAMARAHILOOK28317>. [Último acceso: Enero 2025].
- [24] «Comercial Benavides,» 2023. [En línea]. Available: <https://www.comercialbenavides.net/p/camaras-vigilancia/28500>. [Último acceso: Enero 2025].

- [25] «Comercial Benavides,» 2024. [En línea]. Available: <https://www.comercialbenavides.net/p/ezviz/camaraezvizh9c>. [Último acceso: Enero 2025].
- [26] «Comercial Benavides,» 2023. [En línea]. Available: <https://www.comercialbenavides.net/p/xiaomi/CAMARAXIAOMIAW300>. [Último acceso: Enero 2025].
- [27] F. Ruiz, M. Quevedo, M. Alfonso, O. Pardo y M. Ortega, «Inteligencia artificial: Modelos, técnicas y áreas de aplicación,» Thomson Reuters, España, 2003.
- [28] E. Bertran, «Procesado digital de señales : fundamentos para comunicaciones y control I,» Edicions UPC, Barcelona, 2006.
- [29] Z.-H. Zhou, «Machine Learning,» Universidad de Nanjing, Nanjing, 2016.
- [30] I. Goodfeow, Y. Bengio y A. Courville, Deep Learning, London: The MIT Press, 2016.
- [31] A. Jimenez y J. Ospina, Revisión sistemática de literatura: Técnicas de aprendizaje automático (Machine Learning), Medellín: Cuaderno Activa 2021, 2021.
- [32] C. Huyen y M. Torres, Diseño de sistemas de Machine Learning, Barcelona: Marcombo, 2022.
- [33] L. Bográn y J. Martínez, «Comparativa de modelos de detección de objetos y personas en espacios cerrados de acceso público,» 66, nº 1, p. 12, 2023.

- [34] J. Bagnato, «Aprende Machine Learning,» 21 Agosto 2020. [En línea]. Available: <https://www.aprendemachinelearning.com/modelos-de-deteccion-de-objetos/>. [Último acceso: Enero 2025].
- [35] «IBM documentation,» 25 Junio 2024. [En línea]. [Último acceso: Enero 2025].
- [36] J. Peiyuan, E. Daji, L. Fangyao y C. Ying, «A Review of Yolo Algorithm Developments,» *ScienceDirect*, p. 8, 2022.
- [37] «Ultralytics,» 2024. [En línea]. [Último acceso: Enero 2025].
- [38] Tzutalin, «GitHub,» 2015. [En línea]. Available: <https://github.com/HumanSignal/labelImg>. [Último acceso: Enero 2025].
- [39] «Roboflow Inc,» 2024. [En línea]. Available: <https://roboflow.com/about>. [Último acceso: Enero 2025].
- [40] «Labelme.io,» 2024. [En línea]. Available: <https://www.labelme.io/>. [Último acceso: Enero 2025].
- [41] F. Saez, «Microprocesadores y microcomputadoras,» 2019. [En línea]. Available: https://oa.upm.es/22244/1/Microprocesadores_y_Microcomputadoras.pdf. [Último acceso: Enero 2025].
- [42] «Sigma Electronica,» 2024. [En línea]. Available: <https://www.sigmaelectronica.net/producto/jetson-nano-dev-kit/>. [Último acceso: Enero 2025].

- [43] «Raspberry Pi,» Enero 2023. [En línea]. Available: <https://www.raspberrypi.com/products/raspberry-pi-5/>. [Último acceso: Enero 2025].
- [44] «TiendaTec,» 2024. [En línea]. Available: <https://www.tiendatec.es/raspberry-pi/ai-raspberry-pi/1545-google-coral-dev-board-mini-0193575011295.html>. [Último acceso: Enero 2025].
- [45] T. Gunawan, A. Ashraf, B. Subhan y E. Haryanto, «Development of video-based emotion recognition using deep learning with Google Colab,» *Telkomnika Telecommunication*, vol. I, n° 9, pp. 4-5, 2018.
- [46] «A Dataset of Python Jupyter Notebooks from Kaggle,» *Quaranta, Luigi; Calefato, Fabio; Lanubile, Filippo*, pp. 3-4, 2021.
- [47] A. Sierra, «Departamento de Ingeniería Telemática,» 17 Noviembre 2017. [En línea]. Available: [http://trajano.us.es/~antonio/FdeT/curso2010_11/Tema_1\[1\]\[1\].pdf](http://trajano.us.es/~antonio/FdeT/curso2010_11/Tema_1[1][1].pdf). [Último acceso: Enero 2025].
- [48] B. Kernighan y D. Ritchie, *El lenguaje de programación C*, Ciudad de México: Pearson Education, 2005.
- [49] Á. Hinojosa, *PYTHON Paso a paso*, Madrid: RA-MA, 2016.
- [50] E. Paucar, «Ecuador lidera estadística de robos y asaltos en América Latina, según encuesta de CID Gallup,» *Radio Pichincha*, pp. 1-2, 20 Septiembre 2024.

- [51] Prosegur, «Prosegur Research,» 11 Enero 2023. [En línea]. Available: <https://www.prosegurresearch.com/blog/insights/amenazas-seguridad-ecuador#:~:text=Ecuador%20sufre%20una%20crisis%20de,media%20de%2021%20asesinatos%20diarios..> [Último acceso: Enero 2025].
- [52] M. Gonzáles, «Primicias,» 30 Diciembre 2024. [En línea]. Available: <https://www.primicias.ec/seguridad/cifras-violencia-inseguridad-robos-ecuador-2024-86371/#:~:text=En%202024%2C%20entre%20enero%20y,10.399%20personas%20ligadas%20al%20narcotr%C3%A1fico..> [Último acceso: Enero 2025].
- [53] G. Leiva, C. Romero y O. Alvarado, «Las armas de fuego y su impacto en la inseguridad ciudadana en Ecuador,» *Revista de Investigación en Seguridad Ciudadana y Orden Público*, Cuenca, 2020.
- [54] «UNODC,» Septiembre 2021. [En línea]. Available: <https://www.unodc.org/e4j/es/firearms/module-2/key-issues/common-firearms-types.html>. [Último acceso: Septiembre 2024].
- [55] Instituto Nacional de Estadísticas y Censos, «Estadísticas de delitos de mayor connotación,» Quito - Ecuador, Quito, 2023.
- [56] M. Hussain, «YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection,» *University of Huddersfield*, pp. 9-10, 2023.
- [57] «DAVANTIS,» Applus ISO 9001, 25 mayo 2021. [En línea]. Available: <https://www.davantis.com/es/blog/tecnologia/como-resolver-zonas-muertas-videovigilancia/>. [Último acceso: enero 2025].

- [58] I. Challenger, Y. Díaz y R. Becerra, «El lenguaje de programación Python,» *Ciencias Holguín*, vol. XX, nº 2, pp. 1-13, 2014.
- [59] J. De Ávila y C. Teherán, «SISTEMA HARDWARE-SOFTWARE PARA EL SEGUIMIENTO Y ANÁLISIS DE LA TEMPERATURA CORPORAL DEL USUARIO MEDIANTE EL USO DE CÁMARAS TERMOGRÁFICAS DE BAJO COSTO DENTRO DE UN ENTORNO CONTROLADO PARA PRUEBAS DE USABILIDAD,» Universidad de Cartagena, Cartagena de Indias, 2022.
- [60] L. Culquichicón, «Análisis comparativo de metodos de reconocimiento facial en plataforma android apra resolver problemas de seguridad,» Universidad Señor del Sipán, Pimentel, 2018.

ANEXOS

Anexo A. Especificaciones técnicas de la Jetson Nano en video



Jetson Nano System-on-Module
Maxwell GPU + ARM Cortex-A57 + 4GB LPDDR4 + 16GB eMMC

- Duplicate copies of the L1 data cache directories for coherency support
- Hardware pre-fetch support
- ECC support

1.2.2 Performance Monitoring

The performance monitoring unit (part of MPCore non-CPU logic) provides six counters, each of which can count any of the events in the processor. The unit gathers various statistics on the operation of the processor and memory system during runtime, based on ARM PMUv3 architecture.

1.3 High-Definition Audio-Video Subsystem

The audio-video subsystem off-loads audio and video processing activities from the CPU subsystem resulting in faster, fully concurrent, highly efficient operation.

1.3.1 Multi-Standard Video Decoder

The video decoder accelerates video decode, supporting low resolution content, Standard Definition (SD), High Definition (HD) and UltraHD (2160p, or 4k video) profiles. The video decoder is designed to be extremely power efficient without sacrificing performance.

The video decoder communicates with the memory controller through the video DMA which supports a variety of memory format output options. For low power operations, the video decoder can operate at the lowest possible frequency while maintaining real-time decoding using dynamic frequency scaling techniques.

Video standards supported:

- H.265: Main10, Main
- WEBM VP9 and VP8
- H.264: Baseline (no FMO/ASO support), Main, High, Stereo SEI (half-res)
- VC-1: Simple, Main, Advanced
- MPEG-4: Simple (with B frames, interlaced; no DP and RVLC)
- H.263: Profile 0
- DivX: 4/5/6
- XviD Home Theater
- MPEG-2: MP

Figura A1. Características de video procedentes de la Jetson nano

Anexo B. Especificaciones técnicas DVR HiLook



The image shows a product page for the HiLook DVR-104G-M1 TURBO HD DVR. The background features a large, stylized circular graphic with concentric rings. The HiLook logo is prominently displayed in the top right corner. A blue banner on the left side contains the product name. Below this, the 'Key Feature' section lists several technical specifications. A small image of the DVR unit is shown to the right of the key features. Below the key features, there is a row of six icons representing various capabilities: motion detection, video compression, network connectivity, XVR support, audio input, and mobile access. The 'Smart Function' section highlights deep learning-based motion detection. The 'Compression and Recording' section describes H.265 Pro+ video compression and 1080p Lite support. The 'Storage and Playback' section mentions the SATA interface and smart search functionality. The 'Network & Ethernet Access' section notes HiLookVision and DDNS support.

HiLook

**DVR-104G-M1
TURBO HD DVR**

Key Feature

- Deep learning based human and vehicle targets classification of Motion Detection 2.0
- H.265 Pro+/H.265 Pro/H.265/H.264+/H.264 video compression
- HDTV/AHD/CVI/TVBS/IP video input
- Audio via coaxial cable
- Up to 5-ch IP camera inputs (up to 1080p)
- Up to 1080p Lite@15 fps encoding capability
- Max. 1200 m for 720p HDTV signal

Smart Function

- Deep learning-based motion detection 2.0

Compression and Recording

- H.265 Pro+ video compression improves encoding efficiency, and reduces data storage costs
- Support 1080p Lite of all channels. When 1080p Lite is on, the image effect is similar with 1080p

Storage and Playback

- 1 SATA interface (up to 6 TB capacity per HDD)
- Smart search for efficient playback

Network & Ethernet Access

- HiLookVision & DDNS (Dynamic Domain Name System) for easy network management

Figura B1. Consideraciones técnicas importantes del DVR HiLook

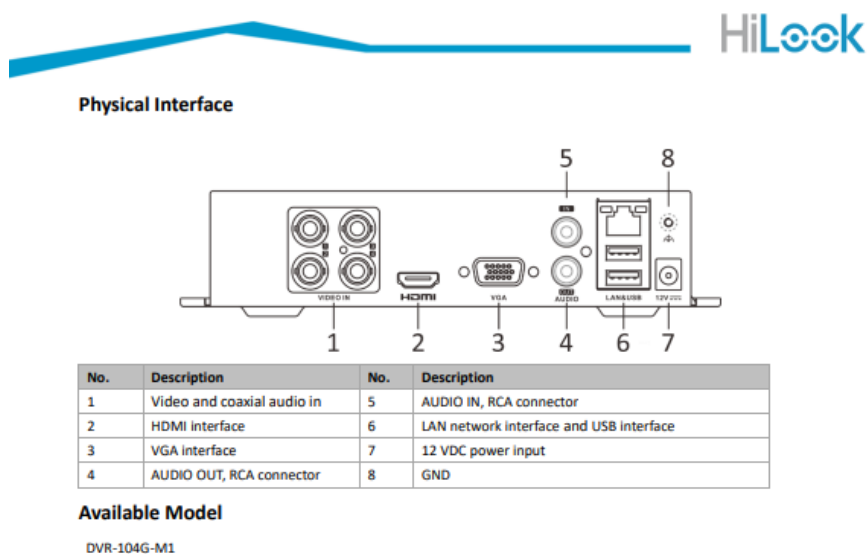


Figura B2. Interfaz física del DVR HiLook

Anexo C. Elección de Python

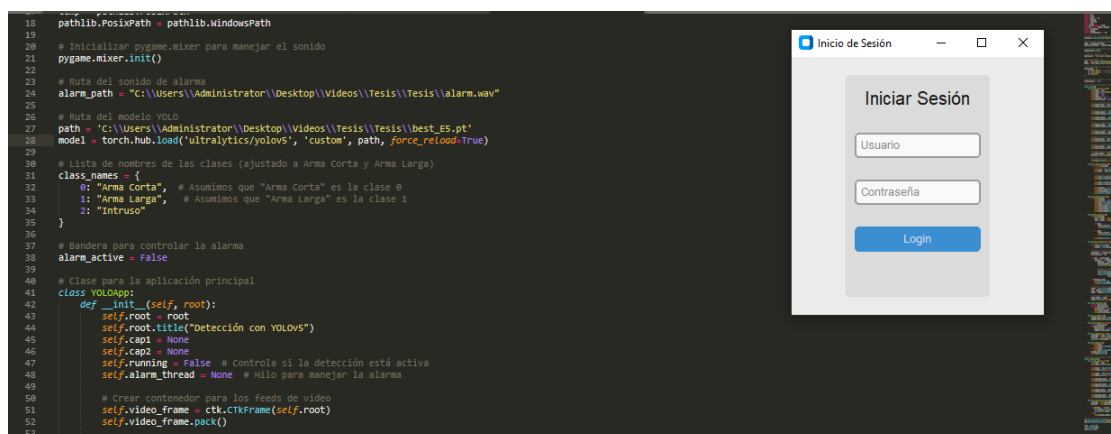


Figura C1. Ejecución del script final mediante el lenguaje de programación Python.

Anexo D. Arquitectura de Yolov5

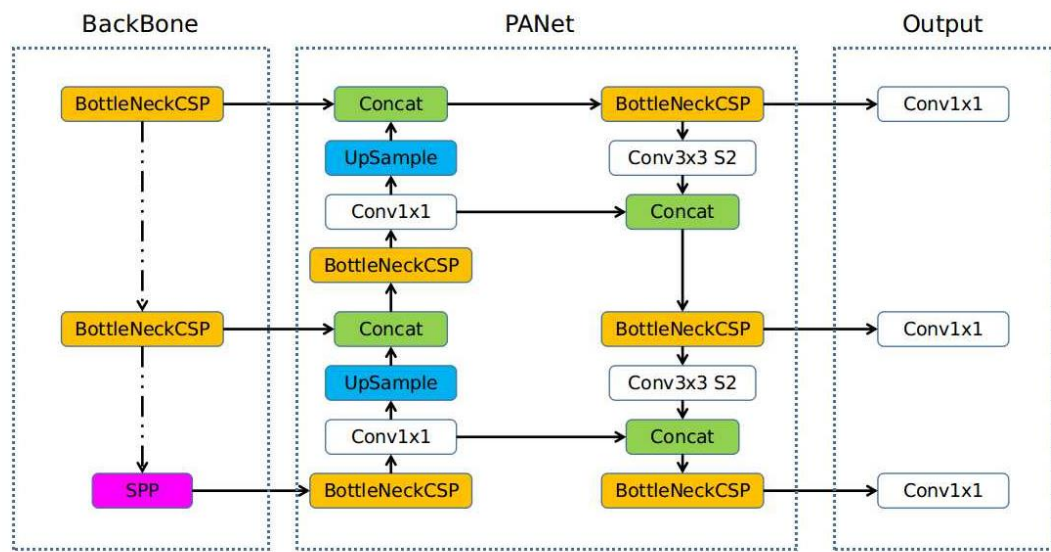


Figura D1. Descripción general de la arquitectura en capas de Yolov5

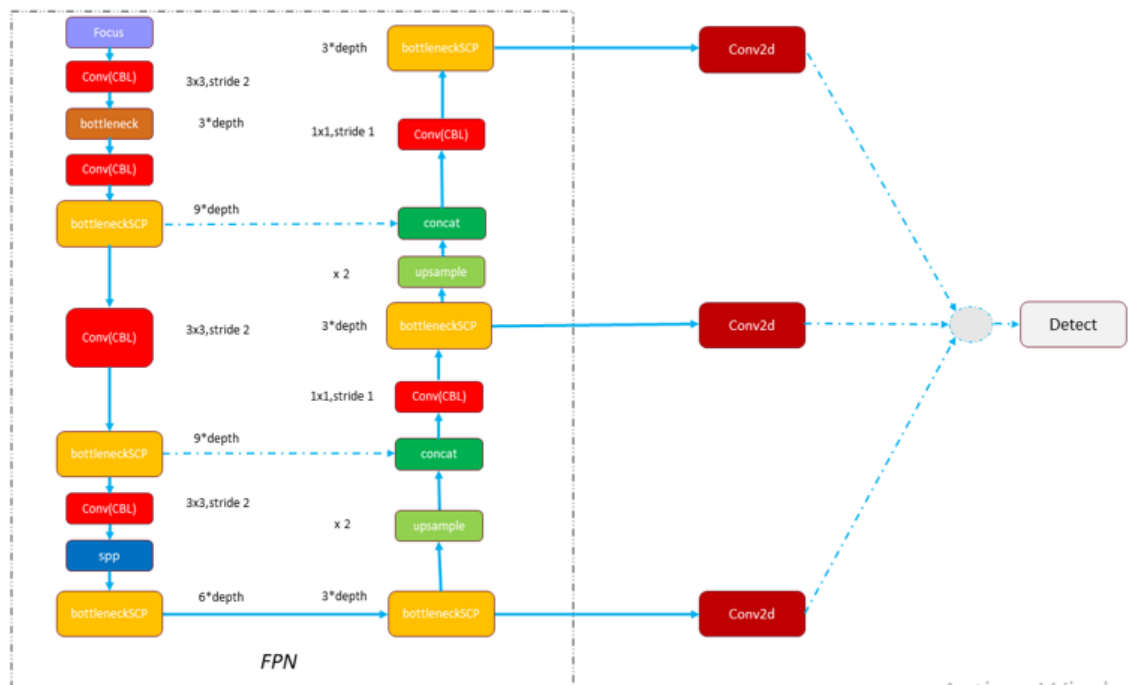


Figura D2. Arquitectura de detección de objetos de Yolov5

Anexo E. Montaje del sistema CCTV



Figura E1. Instalación de cámara tipo tubo en la sala



Figura E2. Instalación cámara domo en la cocina



Figura E3. Visualización a través de las cámaras de seguridad ya instaladas



Figura E4. Instalación y conexión del DVR con las cámaras