UNIVERSIDAD TÉCNICA DE MANABÍ
Fundada en 1952

INFORMÁTICA Y SISTEMAS
Revista de Tecnologías de la Informática y las Comunicaciones

Prototipo de Sistema de Monitoreo Inteligente para la detección en Tiempo Real de Objetos y Actividades Sospechosas Utilizando Deep Learning

Prototype of an Intelligent Monitoring System for Real-Time Detection of Suspicious Objects and Activities Using Deep Learning

Autores

* ***Lauro Alfonso Erreyes Cuenca***
lerreyes1@utmachala.edu.ec

Nahin Josue Olmedo Chica
nolmedo2@utmachala.edu.ec

Mariuxi Paola Zea Ordóñez
mzea@utmachala.edu.ec

Nancy Magaly Loja Mora
nmloja@utmachala.edu.ec

Universidad Técnica de Machala, Facultad de Ingeniería Civil, Carrera de Tecnologías de la Información, Machala, El Oro, Ecuador.
*Autor para correspondencia

Comó citar el artículo:
Erreyes Cuenca, L., Olmedo Chica, N., Zea Ordóñez, M. & Loja Mora, N. (2025). Prototipo de Sistema de Monitoreo Inteligente para la detección en Tiempo Real de Objetos y Actividades Sospechosas Utilizando Deep Learning. *Informática y Sistemas, 9*(2), pp. 197-213. https://doi.org/10.33936/isrtic.v9i2.7908

Enviado: 27/09/2025
Aceptado: 18/11/2025
Publicado: 19/11/2025

Resumen

Esta investigación trata sobre la necesidad de optimizar los sistemas de videovigilancia a través del uso de inteligencia artificial para detectar proactivamente amenazas. El propósito primordial es desarrollar un prototipo de monitoreo inteligente que pueda detectar en tiempo real a personas, armas y conductas sospechosas, enfocándose en la eficiencia computacional y la precisión de detección para garantizar su viabilidad en hardware accesible. Se utilizó la metodología CRISP-DM para conseguir esta meta, que dividió el proyecto de manera sistemática en etapas que incluían la preparación, modelado y valoración de los datos. Un modelo YOLOv8 es el elemento principal del sistema, el cual fue entrenado con un conjunto de datos personalizado que comprende cerca de 8.500 imágenes y se expandió a través de distintos métodos. La robustez del modelo se confirma mediante los resultados cuantitativos, que muestran una puntuación F1-score del 93,99% y una precisión media (mAP) de 50 del 96,97% en las clases especificadas. Finalmente, el modelo fue incorporado en un prototipo funcional de videovigilancia, lo que demostró su utilidad y efectividad operativa en entornos de seguridad urbana y comercial.

Palabras clave: inteligencia artificial; deep learning; YOLOv8; videovigilancia.

Abstract

This research addresses the need to optimize video surveillance systems through the use of artificial intelligence to proactively detect threats. The primary goal is to develop an intelligent monitoring prototype capable of detecting people, weapons, and suspicious behavior in real time, focusing on computational efficiency and detection accuracy to ensure its feasibility on readily available hardware. The CRISP-DM methodology was used to achieve this goal, systematically dividing the project into stages that included data preparation, modeling, and evaluation. A YOLOv8 model is the core element of the system, trained on a custom dataset of approximately 8,500 images and expanded using various methods. The model's robustness is confirmed by quantitative results, which show an F1 score of 93.99% and a mean accuracy (mAP) of 96.97% in the specified classes. Finally, the model was incorporated into a functional video surveillance prototype, demonstrating its usefulness and operational effectiveness in urban and commercial security environments.

Keywords: artificial intelligence; deep learning; YOLOv8; video surveillance.

INFORMÁTICA Y SISTEMAS
Revista de Tecnologías de la Informática y las Comunicaciones

Facultad de Ciencias Informáticas
Universidad Técnica de Manabí
Av. Urbina y Che Guevara, Portoviejo, Ecuador
revista.iys@utm.edu.ec

1. Introduction

Con el rápido crecimiento de la Inteligencia Artificial (IA) en los últimos años, se han generado importantes avances en diversas áreas, como la de videovigilancia inteligente, donde los sistemas tradicionales han quedado atrás. Esto principalmente por la dependencia de la supervisión humana, lo que genera retrasos y errores en situaciones críticas. Un claro ejemplo de cómo afrontar estas debilidades es el desarrollo de sistemas de seguridad capaces de funcionar de manera automatizada y sin supervisión mediante modelos de detección en tiempo real basados en deep learning (Azatbekuly et al., 2024). Estos trabajos resaltan que, a diferencia de los métodos tradicionales que requieren mucha mano de obra para procesarlo, analizarlo y verlo, es posible reducir significativamente la intervención humana y mejorar la capacidad de respuesta ante diversos escenarios.

La videovigilancia apoyada con IA se ha desplegado en escenarios muy variados, no únicamente se limita a ciudades. En el ámbito agrícola, se han creado plataformas que tienen la capacidad de monitorear e identificar plagas en los cultivos automáticamente con un 92% de precisión (Delwar et al., 2025). Un sistema de vigilancia para una mina de carbón en el sector minero se ha puesto en marcha; un estudio reciente demuestra que este tipo de sistemas es capaz de identificar la existencia de individuos en zonas peligrosas y restringidas, con una tasa de precisión impresionante del 99.5%, lo que deja claro su eficacia aun en situaciones operativas extremas (Ni et al., 2024). Al mismo tiempo, se han creado sistemas semejantes en el sector educativo para detectar comportamientos anormales, como conflictos físicos y actos de deshonestidad académica. Estas tecnologías innovadoras han alcanzado un grado de exactitud del 96%, lo que demuestra que pueden proporcionar una respuesta precisa y optimizar las medidas de seguridad en los espacios universitarios (Gawande et al., 2024). Estos ejemplos demuestran que la IA puede ayudar a la seguridad de manera confiable y flexible cuando se utiliza en videovigilancia.

YOLOv8 es un instrumento ideal para sistemas de vigilancia en tiempo real debido a su balance sobresaliente entre precisión y velocidad de procesamiento. Su eficacia se ha comprobado a través de múltiples investigaciones empíricas. Por ejemplo, este modelo se ha empleado para identificar armas de fuego y monitorizar conductas sospechosas en la práctica, logrando un 92.2% de precisión con una Intersección sobre Unión de 0.6 (Schcolnik-Elias et al., 2023). En consecuencia, otros estudios resaltan que YOLOv8 mantiene un rendimiento constante incluso en situaciones adversas. Por ejemplo, se ha confirmado que el mAP@50 alcanzó un 89% y los FPS fueron más de 430 en situaciones con una alta afluencia, lo cual evidencia su efectividad (Hua et al., 2024). En espacios con una alta densidad de población, también se ha verificado la precisión y viabilidad de este modelo en tiempo real, alcanzando un F1-score de 94.7% (Nasir et al., 2025).

Se ha estudiado, además de la detección de armas, la capacidad de esta arquitectura de detección para identificar ciertas acciones que despiertan sospechas y para contar personas, con el objetivo de comprobar su efectividad en situaciones de vigilancia por video. Se han creado sistemas que pueden detectar circunstancias como la presencia de armas o humo en lugares públicos, logrando niveles de precisión por encima del 95% y emitiendo alertas automáticas en tiempo real (Sudharson et al., 2023). De manera similar, se han introducido mejoras ligeras a la arquitectura base mediante módulos como soft-NMS y GSConv, obteniendo un 88.6% de precisión en videovigilancia de campus (Cheng et al., 2024). Además, se ha tratado el asunto de detectar objetos pequeños en imágenes aéreas captadas por drones. Para lograrlo, se ha sugerido el modelo SOD-YOLO, una versión perfeccionada de la estructura original que superó al YOLOv8l (Li et al., 2024) en un 7.7% del mAP@50. Esta perspectiva muestra que el detector sigue siendo eficaz, a pesar de los escenarios complicados que se observan en las imágenes aéreas obtenidas por drones.

El modelo YOLOv8, a diferencia de versiones previas como YOLOv5, YOLOv6 y YOLOv7, brinda progresos significativos en cuanto a la velocidad de inferencia, precisión y capacidad de generalización. Estas optimizaciones derivan de su estructura sin anclajes, la combinación más eficaz de atributos y los métodos de entrenamiento más sofisticados. Según otros análisis, YOLOv8 logró un 98% en F1-score y un 99% de precisión, lo que significa que superó ampliamente los resultados de YOLOv6 y YOLOv7 con 92% (Delwar et al., 2025). Por otro lado, el F1-score fue del 94.7% frente al 69% de YOLOv5 (Nasir et al., 2025). Estas pruebas respaldan el uso de esta arquitectura en esta investigación, ya que ofrece un equilibrio perfecto entre la eficacia y la velocidad de detección en tiempo real.

Es importante destacar que, aunque los avances más recientes muestran progresos notables en las métricas de precisión, la mayor parte de estos estudios no abordan el coste computacional y los requerimientos de hardware para la inferencia en tiempo real. El enfoque principal ha sido aumentar la precisión, frecuentemente suponiendo que se utilizan unidades de procesamiento gráfico de gama alta que no son económicamente factibles para la mayoría de las implementaciones de seguridad urbana o comercial. Esta disparidad entre la factibilidad práctica en hardware accesible y la exactitud teórica representa un reto crítico sin resolver. El principal problema que motivó este estudio radica

e-ISNN 2550-6730 | Vol. 9, Núm. 2 (197-213): Julio-Diciembre, 2025 | DOI: 10.33936/isrtic.v9i2.7908

no solo en la limitada capacidad de los sistemas tradicionales de videovigilancia para detectar en tiempo real actividades sospechosas o la presencia de objetos peligrosos, sino también la ausencia de soluciones de inteligencia artificial validadas que mantengan un equilibrio entre una gran precisión y la eficiencia computacional requerida para su implementación en hardware económico. A partir de esta situación, surge la siguiente pregunta de investigación: ¿Cómo puede un sistema de videovigilancia inteligente basado en deep learning alcanzar al menos un 90% de precisión en la detección en tiempo real de personas, armas y actividades sospechosas?.

El objetivo general de este trabajo fue desarrollar un prototipo funcional de monitoreo inteligente capaz de detectar en tiempo real personas, armas y personas armadas utilizando un modelo basado en deep learning. De este se derivan los objetivos específicos: (1) construir un conjunto de datos personalizado que integre imágenes de personas, armas y personas armadas, aplicando técnicas de preprocesamiento y aumentación de datos; (2) entrenar y optimizar un modelo YOLOv8 mediante transfer learning para lograr altos niveles de precisión, recall, F1-score y mAP; y (3) integrar el modelo en un prototipo funcional de videovigilancia capaz de procesar video en tiempo real, emitir alertas basadas en reglas contextuales y validar su rendimiento mediante métricas cuantitativas y pruebas experimentales en entornos simulados y reales. Para alcanzar estos objetivos, se adoptó la metodología Cross-Industry Standard Process for Data Mining (CRISP-DM), estructurada en seis fases que guiaron de manera ordenada el desarrollo del sistema, desde la comprensión del problema y la preparación de datos hasta el modelado, la evaluación y la implementación final del prototipo.

El artículo está organizado de la siguiente forma: en la Sección 2 se describen los materiales y métodos, en la Sección 3 se presentan los resultados y se discuten los hallazgos en relación con estudios anteriores, y en la Sección 4 se exponen las conclusiones y las futuras líneas de investigación.

2. Materiales y Métodos

El estudio fue llevado a cabo con la metodología CRISP-DM, que brinda una estructura ordenada para la creación de proyectos de análisis y modelaje de datos. Esta metodología fue escogida debido a su flexibilidad y a su habilidad para adecuarse a procesos iterativos, lo que posibilita hacer correcciones o volver atrás entre fases de acuerdo con los resultados alcanzados en cada etapa del desarrollo. El modelo CRISP-DM sigue siendo uno de los métodos más utilizados en iniciativas que se fundamentan en minería de datos y aprendizaje automático, gracias a su versatilidad y enfoque práctico, lo cual lo hace apropiado para contextos reales donde son esenciales la mejora permanente y la validación (Acuña-Cid et al., 2025).

En este trabajo, se utilizó CRISP-DM en seis etapas esenciales, como se observa en la Figura 1. Esta metodología sirvió de orientación para organizar el flujo del proyecto desde que se entiende el dominio hasta que se valida el prototipo final. Durante estas etapas, se usaron herramientas específicas: Roboflow Universe para la recopilación y gestión del dataset; Python 3.11 para el procesamiento y preparación de datos; YOLOv8 en PyTorch para la fase de modelado; y Matplotlib para la visualización y evaluación de resultados. En última instancia, se emplearon Angular y Python (Flask) para desarrollar el frontend y el backend del prototipo, respectivamente; esto asegura una integración eficaz entre la interfaz de monitoreo y la inferencia del modelo.

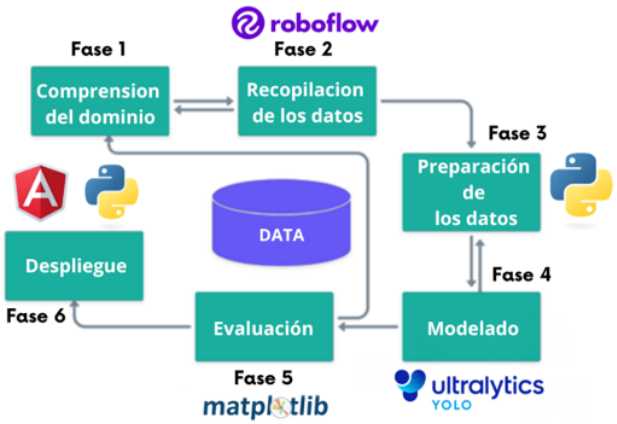

Figura 1. Metodología CRISP-DM.
Fuente: Los autores.

El objetivo de esta etapa fue identificar las principales restricciones operativas de los sistemas tradicionales de videovigilancia y determinar la funcionalidad necesaria para crear un sistema inteligente de monitoreo. La fatiga, la distracción y las diferencias en la interpretación de los sucesos son ejemplos de factores humanos que impactan negativamente en el desempeño efectivo de la supervisión. De ahí surgió la necesidad de crear un sistema automatizado que tenga la capacidad de identificar armas y personas en tiempo real, aplicar normas contextuales para detectar conductas sospechosas y lanzar alertas automáticas. Esta perspectiva apunta a disminuir la dependencia de la supervisión manual, perfeccionar la exactitud en las detecciones y optimizar la capacidad de respuesta frente a incidentes en contextos urbanos y de seguridad privada.

Durante esta etapa se analizó el contexto operacional de la seguridad urbana y se documentaron fallas asociadas a la vigilancia sostenida: fatiga del operador, variabilidad en la interpretación de la escena y retrasos en la verificación de eventos, que conducen a falsos positivos y omisiones. Estudios recientes demuestran que, a medida que aumenta el número de pantallas a supervisar, el rendimiento del operador disminuye debido a la sobrecargar visual. Stainer et al. (2021) demostraron que, en entornos de videovigilancia con varias cámaras, los operadores presentan mayores tiempos de reacción y retrasos en la detección de eventos críticos. Como se observa en la Figura 2, el monitoreo

e-ISNN 2550-6730 | **Vol. 9, Núm. 2 (197-213): Julio-Diciembre, 2025** | **DOI: 10.33936/isrtic.v9i2.7908**

INFORMÁTICA Y SISTEMAS
Revista de Tecnologías de la Informática y las Comunicaciones

Facultad de Ciencias Informáticas
Universidad Técnica de Manabí
Av. Urbina y Che Guevara, Portoviejo, Ecuador
revista.iys@utm.edu.ec

con una sola cámara genera tiempos de respuesta más cortos y precisos, mientras que el monitoreo de cuatro cámaras incrementa el retraso de detección de eventos sospechosos.

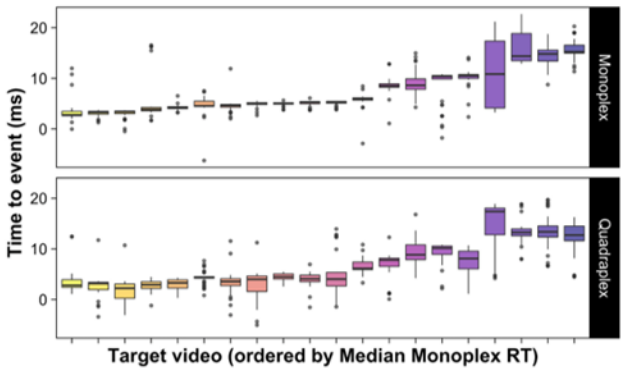

Figura 2. Variación del tiempo de respuesta ante tareas de videovigilancia
Fuente: Tomado de Stainer et al. (2021).

A partir de este diagnóstico se establecieron los siguientes criterios base para el desarrollo del sistema:

• R1: Identificación automática de armas y personas sin la intervención de seres humanos.

• R2: Disminución del tiempo de respuesta ante situaciones sospechosas.

• R3: Generación automática de alertas bajo condiciones preestablecidas.

• R4: Operación estable en tiempo real con un margen de error mínimo.

El resultado de esta etapa permitió definir las características del conjunto de datos, lo que sirvió de punto de partida para la siguiente fase.

Fase 2: Comprensión y recopilación de los datos

Los requisitos y criterios para el conjunto de datos que se requiere para entrenar el modelo de detección fueron determinados en esta etapa. La labor fundamental fue reunir imágenes que ilustraran la variedad de situaciones que un sistema de videovigilancia podría afrontar, teniendo en cuenta diferentes circunstancias, tales como las condiciones operativas, los ángulos de cámara y la iluminación.

Se eligieron imágenes que satisficieran tres requisitos principales: (1) la presencia evidente de las clases objetivo, es decir, persona, arma y persona armada; (2) una calidad visual adecuada para anotar los objetos con precisión; y (3) la no existencia de distorsiones o marcas de agua que pudieran interferir con el entrenamiento. Se eliminaron las imágenes que tenían baja resolución, que se repetían o que presentaban etiquetas inconsistentes como se observa en la Figura 3.

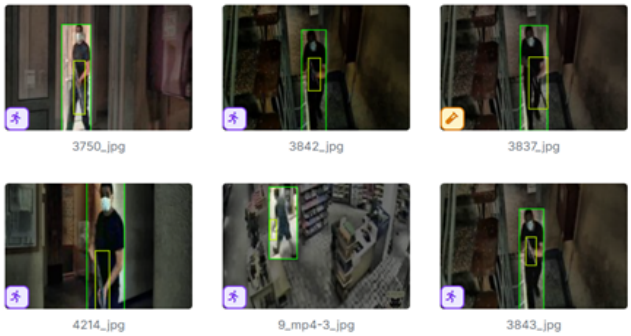

Figura 3. Ejemplos de imágenes del dataset en Roboflow.
Fuente: Dataset completo de Erreyes (2025).

Para la construcción del dataset personalizado, se utilizaron tres repositorios de Roboflow Universe: el primero, "Armed Person Recognition Dataset" contiene más de 7,000 imágenes de personas armadas y fue elegido por su enfoque en detectar amenazas directas en entornos urbanos, el segundo tomado de "Personas with Weapons Dataset" que incluye 6,600 imágenes de personas portando armas, ideal para identificar situaciones de riesgo y finalmente, "Weapon Detection Dataset" ofrece más de 12000 imágenes etiquetadas con diferentes tipos de armas, permitiendo al modelo reconocer objetos peligrosos.

Se llevó a cabo un análisis de las variables principales que influyen en el rendimiento de los modelos de detección en la videovigilancia con el fin de caracterizar objetivamente la diversidad del conjunto de datos. La Tabla 1 resume la distribución aproximada de imágenes en función de cada variable por separado, sobre un total de aproximadamente 8,500 imágenes recolectadas.

Esta distribución demuestra que el conjunto de datos incluye combinaciones complejas de condiciones operativas, lo cual permite que el modelo esté expuesto a situaciones que representan escenarios verdaderos de videovigilancia. La inclusión de situaciones difíciles como la baja iluminación, ángulos extremos, oclusiones severas asegura que el sistema entrenado sea capaz de generalizar frente a las condiciones cambiantes que suelen ocurrir en los ambientes comerciales y urbanos.

Para eliminar los duplicados y garantizar la coherencia en la

e-ISNN 2550-6730 | Vol. 9, Núm. 2 (197-213): Julio-Diciembre, 2025 | DOI: 10.33936/isrtic.v9i2.7908

Tabla 1. Caracterización de la diversidad del dataset según variables operativas.
Fuente: Los autores.

Características	Categorías	Imágenes aproximadas
Iluminación	Alta	3,600
	Media	3,200
	Baja	1,700
Ángulo de cámara	Frontal (0-30°)	3,800
	Oblicuo (30-60°)	3,000
	Cenital (>60°)	1,700
Contexto operativo	Interior comercial	4,500
	Urbano exterior	2,500
	Mixto/residencial	1,500
Densidad de objetos	1 persona/arma	3,400
	2-5 objetos	3,600
	Más de 5 objetos	1,500
Oclusiones	Sin oclusión	4,000
	Oclusión parcial	3,200
	Oclusión severa	1,300

distribución de las clases, el conjunto de datos fue revisado manual y automáticamente antes del entrenamiento. Para prevenir sesgos de aprendizaje, se mantuvo un balance proporcional entre las categorías más importantes: persona, arma y persona armada. Asimismo, se examinó la congruencia entre las etiquetas y los objetos visibles para asegurar que el dataset final fuera de calidad.

El conjunto de datos recopilados fue organizado en tres subconjuntos principales: entrenamiento, validación y prueba,

Tabla 2. Composición del Conjunto de Datos.
Fuente: Los autores.

Clase	Entrenamiento	Validación	Prueba
Persona	8,176	857	893
Arma	9,018	888	964
Persona armada	7,030	758	758

con el fin de garantizar una evaluación adecuada. La Tabla 2 muestra la distribución de las instancias por clase, evidenciando un balance entre categorías de persona, arma y persona armada.

La división se realizó utilizando la proporción 8:1:1 (80% entrenamiento, 10% validación y 10% prueba), asegurando la distribución equilibrada de clases en cada subconjunto. Esta estructura fue seleccionada en base a referencias metodológicas previas, como la proporción 7:2:1 de Ni et al. (2024) aplicados en entornos subterráneos y el esquema 3:1 (75% entrenamiento y 25% validación) del trabajo de Delwar et al. (2025) para proyectos en entornos de agricultura. De esta manera, optar por la proporción 8:1:1 nos dispone de suficientes datos para el entrenamiento, con el fin de reducir el riesgo del underfitting (subajuste), y al mismo tiempo reservar datos de validación y prueba que actúan frente al overfitting (sobreajuste) garantizando una evaluación imparcial del rendimiento.

Este análisis se basa en los principios de la IA responsable, que valoran ante todo la transparencia, la equidad y la protección de datos. Al elegir y preparar el conjunto de datos, se pusieron en práctica acciones destinadas a salvaguardar la privacidad y reducir los peligros de sesgos algorítmicos. Según Castro-Paredes et al. (2025),es preciso incluir la protección de datos desde el diseño original del sistema y restringir la recopilación de datos a lo que sea estrictamente necesario. En este contexto, se eliminaron los metadatos identificativos y se comprobó que las imágenes fueran obtenidas de repositorios públicos como Roboflow Universe, los cuales tienen licencias abiertas para la investigación y el uso académico. De acuerdo con los términos establecidos por sus creadores, se utilizaron las colecciones de datos elegidas sin alterar el contenido original más allá de las operaciones de aumento. Se respetaron los principios de privacidad y el uso ético de datos al no incluir información personal ni metadatos que pudieran revelar la identidad de personas.

Además, se abordó la advertencia de Alvarado & Villavicencio (2024) sobre cómo los algoritmos de aprendizaje automático pueden heredar y aumentar los sesgos que ya existen en los datos con los que son entrenados. Para evitar la discriminación algorítmica, se documentó la diversidad del conjunto de datos en cuanto a iluminación, ángulos de cámara y contextos operativos, como se muestra en la Tabla 1. Esta caracterización posibilitó confirmar que el conjunto de datos no tuviera una sobrerrepresentación de condiciones particulares que pudieran inducir sesgos en el modelo final. El conjunto de datos se empleó solamente para validar y desarrollar modelos de detección orientados a la seguridad pública y privada.

Fase 3: Preparación de los datos

Después de etiquetar el conjunto de datos, se empezó a preparar el dataset para el entrenamiento con la finalidad de optimizar la habilidad de generalización ante cambios comunes en videovigilancia como las variaciones en la iluminación, en los ángulos de cámara, oclusiones parciales y escalas disímiles. La elección de las técnicas de aumentación se llevó a cabo teniendo en cuenta la problemática concreta que son las personas, armas y personas armadas, dándole prioridad a los cambios que aporten alteraciones realistas sin menoscabar el significado de las escenas ni la forma exterior de objetos pequeños.

Las modificaciones geométricas como la perspectiva, rotación, traslación, escala, shear y mosaico persiguen una solidez tanto espacial como multi-escala; los cambios en HSV se ocupan de la variabilidad luminosa propia de interiores y exteriores; el mixup y el copy-paste atenuan los límites de decisión y elevan las co-ocurrencias, lo que disminuye el sobreajuste. Las probabilidades se establecieron para mantener un equilibrio entre la fidelidad y la diversidad: son elevadas cuando la transformación

e-ISNN 2550-6730 | Vol. 9, Núm. 2 (197-213): Julio-Diciembre, 2025 | DOI: 10.33936/isrtic.v9i2.7908

INFORMÁTICA Y SISTEMAS
Revista de Tecnologías de la Informática y las Comunicaciones

Facultad de Ciencias Informáticas
Universidad Técnica de Manabí
Av. Urbina y Che Guevara, Portoviejo, Ecuador
revista.iys@utm.edu.ec

Tabla 3. Técnicas de aumentación de datos y sus parámetros aplicados.
Fuente: Los autores.

Técnica	Objetivo	Probabilidad	Efecto esperado
Mosaic	Diversidad de contextos y escalas simultáneas	p = 0.5	Mejor detección multi-escala; menor sobreajuste.
MixUp	Suavizar fronteras de decisión	p = 0.1	Mayor robustez a oclusiones/ ruido; Clases más separables.
Volteo horizontal	Invariancia a orientación lateral	p = 0.5	Generalización a distintos direcciones de movimiento.
Rotación	Tolerancia a inclinaciones leves	Hasta 10° (aleatoria)	Estabilidad ante cámaras no niveladas.
Traslación	Robustez a encuadres variables	Hasta 20%	Menor sensibilidad a reencuadres del objeto.
Escalado	Robustez multi-escala	Hasta 50%	Mejora detección de objetos pequeños/ grandes.
Shear	Variación geométrica sutil	2°	Mejora frente a distorsiones por perspectiva.
Perspectiva leve	Simular cámaras en altura	Aleatoria, leve	Mejor desempeño en CCTV montado en techos/paredes.
Ajuste HSV	Variabilidad de iluminación/color	Deltas aleatorios	Robustez a interiores/exteriores y sombras.
Copy-Paste	Aumentar co-ocurrencias y densidad	p = 0.2	Más ejemplos “persona armada”; mejora recall sin sobre rotular.

reproduce condiciones comunes como el volcado horizontal, y moderadas o bajas cuando hay peligro de agregar ruido principalmente en mixup o copy-paste. La Tabla 3 resume cada técnica de aumentación aplicada, su propósito, los parámetros probabilísticos y el efecto esperado sobre la generalización del modelo.

Estas estrategias se aplicaron para equilibrar las clases y aumentar la diversidad del conjunto de datos, siguiendo enfoques similares a los reportados por Espinoza et al. (2025), quienes evidencian que el incremento controlado de la variabilidad de las muestras contribuye a una mejor generalización de los modelos.

Para garantizar la coherencia y compatibilidad con el modelo elegido, se realizaron las siguientes acciones preliminares:

• Todas las imágenes fueron ajustadas a una resolución estándar de 640 × 640 píxeles, para mantener un equilibrio entre precisión y velocidad de inferencia.

• Se verificó que las imágenes estuvieran en formato .jpg, y las etiquetas asociadas en archivos .txt con coordenadas normalizadas de los bounding boxes.

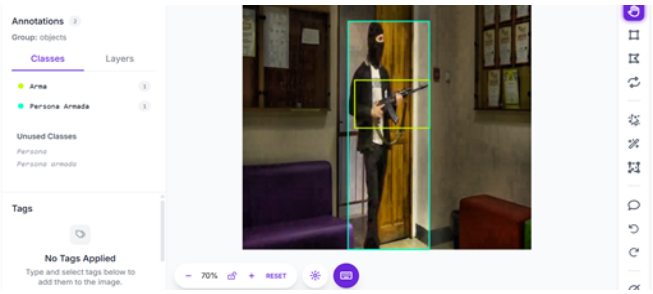

Figura 4. Interfaz de anotación de clases en Roboflow.
Fuente: Los autores.

El proceso de anotación de datos se realizó a través de la plataforma Roboflow Universe, la cual facilitó la visualización y edición gráfica de las etiquetas mediante una interfaz que permite delimitar los objetos con bounding boxes de distintos colores, como se observa en la Figura 4.

Las imágenes provinieron de los repositorios seleccionados fueron previamente etiquetados por sus autores originales; sin embargo, fue necesario revisar y ajustar manualmente algunas anotaciones, corrigiendo nombres de clases y coordenadas de los bounding boxes para garantizar la coherencia y precisión de las etiquetas. Este proceso permitió mantener la consistencia entre las clases “persona”, “arma” y “persona armada”, asegurando un formato estandarizado compatible con YOLOv8.

Se llevó a cabo una revisión mixta, automática y manual de la totalidad de los datos con el fin de asegurar que las etiquetas fueran coherentes y que las clases estuvieran correctamente asignadas. Para detectar errores de formato, clases no definidas y coordenadas que exceden el rango normalizado, se utilizaron scripts de validación en la verificación automática. Además, se realizó una revisión manual a través de la plataforma Roboflow, que permite ver y editar directamente las etiquetas en cada imagen. Esta herramienta posibilitó verificar la exactitud de los bounding boxes y rectificar potenciales inconsistencias o superposiciones en las clases objetivo. En total, se examinó manualmente alrededor del 10 % de los datos totales para garantizar la coherencia semántica y visual de las anotaciones antes de que el modelo comenzara su entrenamiento.

e-ISNN 2550-6730 | Vol. 9, Núm. 2 (197-213): Julio-Diciembre, 2025 | DOI: 10.33936/isrtic.v9i2.7908

Fase 4: Modelado

En esta fase se realizó el entrenamiento del modelo YOLOv8 usando el conjunto de datos preparado en su totalidad. Se configuraron los hiperparámetros más importantes como el número de épocas, el tamaño de lote y la tasa de aprendizaje, el fin de esto es la optimización del proceso de aprendizaje y no tener un sobreajuste innecesario. Se implementaron métodos de validación y modificaciones dinámicas en la tasa de aprendizaje a lo largo del entrenamiento para mejorar el proceso de detección y estabilizar el aprendizaje del modelo.

Se utilizó el modelo YOLOv8 en dos versiones para el entrenamiento la versión nano (YOLOv8n) y la versión small (YOLOv8s). La versión estándar YOLOv8 no se seleccionó porque YOLOv8s la supera en velocidad y rendimiento. Las dos versiones fueron elegidas con el propósito de valorar la eficacia en cuanto a rapidez de inferencia y precisión, buscando así hallar la mejor relación entre eficiencia y exactitud para un despliegue viable en hardware accesible y de bajo costo.

• **YOLOv8n:** Se optó por esta versión más liviana al principio porque necesita menos recursos computacionales, lo que la hace perfecta para aplicaciones en tiempo real que demandan un uso reducido de memoria y capacidad de procesamiento.

• **YOLOv8s:** Es un modelo más sólido que logra optimizar la capacidad de generalización y precisión, a la vez que mantiene una latencia apropiada para labores de videovigilancia en sistemas con recursos computacionales limitados.

El modelo se desarrolló, entrenó y evaluó en un entorno local de alto rendimiento, equipado con una GPU NVIDIA RTX 2050 4 GB VRAM, adecuada para tareas de detección en tiempo real mediante deep learning. El sistema operativo utilizado fue Windows 11 Pro, con 16 GB de memoria RAM y un almacenamiento SSD de 1 TB, lo que permitió una ejecución fluida del entrenamiento y validación del modelo YOLOv8s. El lenguaje principal de programación fue Python 3.11, utilizando la librería Ultralytics YOLOv8 basada en PyTorch. Todo el desarrollo fue gestionado en el entorno de trabajo Visual Studio Code, que facilito la integración de scripts, pruebas y gestión de dependencias durante el proceso de experimentación.

La configuración de hiperparámetros constituye un elemento clave para el rendimiento del modelo, ya que determina tanto la estabilidad como la capacidad de generalización del sistema. Los valores utilizados en esta investigación están especificados en la Tabla 4; fueron seleccionados para asegurar un rendimiento equilibrado del modelo y una inferencia apropiada en tiempo real.

Tabla 4. Configuración de Hiperparámetros.
Fuente: Los autores.

Parámetro	Prueba 1 (YOLOv8n)	Prueba 2 (YOLOv8s)	Prueba 3 (YOLOv8s)	Prueba 4 (YOLOv8s)
Modelo base	YOLOv8n	YOLOv8s	YOLOv8s	YOLOv8s
Épocas	50	75	100	100
Batch size	16	8	12	12
Learning rate (lr0)	0.001	0.001	0.0008	0.0005
Learning rate (lrf)	0.0002	0.0002	0.00016	0.0001
Warmup epochs	3	3	5	5
Weight decay	0.0005	0.0005	0.0005	0.0005
Mosaic	0.3	0.4	0.5	0.5
Mixup	0.0	0.05	0.1	0.1
Copy-paste	0.0	0.1	0.15	0.2
Cosine LR	No	Sí	Sí	Sí
Optimizer	SGD	AdamW	AdamW	AdamW
Caché	RAM	RAM	Disk	Disk
AMP (Mixed Precision)	Sí	Sí	Sí	Sí

El entrenamiento del modelo se realizó en el entorno local descrito anteriormente, aprovechando la GPU para acelerar los cálculos de propagación hacia adelante y hacia atrás. En la Figura 5 se muestra un fragmento del registro de un entrenamiento, donde se

```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     97/100      3.54G     0.8439     0.4449       1.05         30        640: 100%|██████████| 579/579 [03:43<00:00,  2.59it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 31/31 [00:12<00:00,  2.44it/s]
                   all        744       2503      0.951      0.927      0.969      0.726

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     98/100      3.58G     0.8358      0.442      1.045         25        640: 100%|██████████| 579/579 [03:43<00:00,  2.59it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 31/31 [00:12<00:00,  2.43it/s]
                   all        744       2503      0.951      0.929      0.969      0.726

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     99/100      3.62G     0.8427      0.442       1.05         25        640: 100%|██████████| 579/579 [03:43<00:00,  2.59it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 31/31 [00:12<00:00,  2.41it/s]
                   all        744       2503      0.952      0.929       0.97      0.726

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
    100/100      2.94G     0.8469     0.4481       1.05         36        640: 100%|██████████| 579/579 [03:42<00:00,  2.60it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 31/31 [00:12<00:00,  2.40it/s]
                   all        744       2503      0.951      0.929       0.97      0.727
```

Figura 5. Registro del proceso de entrenamiento del modelo YOLOv8s durante 100 épocas.
Fuente: Los autores.

e-ISNN 2550-6730 | Vol. 9, Núm. 2 (197-213): Julio-Diciembre, 2025 | DOI: 10.33936/isrtic.v9i2.7908

INFORMÁTICA Y SISTEMAS
Revista de Tecnologías de la Informática y las Comunicaciones

Facultad de Ciencias Informáticas
Universidad Técnica de Manabí
Av. Urbina y Che Guevara, Portoviejo, Ecuador
revista.iys@utm.edu.ec

evidencian las perdidas progresivas. El flujo de trabajo incluyó:

• Inicialización del modelo con pesos preentrenados (transfer learning) aprovechando el conocimiento previo en tareas de detección general y ajustarlo al conjunto de datos específico (fine-tuning).

• Carga del dataset preprocesado y organizado en las carpetas train/, val/ y test/, utilizando el módulo YOLODataset de la librería Ultralytics.

• Ejecución de ciclos de entrenamiento durante 100 épocas, aplicando el algoritmo AdamW para actualizar los parámetros de la red.

• Monitoreo en tiempo real de métricas clave como:

- Box loss: error asociado a la precisión de las coordenadas de los bounding boxes.

- Class loss: error en la clasificación de las detecciones.

- Objectness loss: error en la predicción de la existencia de objetos.

El entrenamiento se realizó con un tiempo total aproximado de entre 4 a 6 horas. El tiempo medio por época fue de cerca de 4 minutos. La resolución de las imágenes empleadas fue 640×640 píxeles, y el consumo de memoria GPU osciló entre 2.8 GB y 3.63 GB, dependiendo del tamaño de los lotes y las configuraciones particulares de cada versión.

Fase 5: Evaluación

La evaluación de los modelos se realizó utilizando un subconjunto de pruebas que equivale al 10% del conjunto de datos original, conformado por imágenes no utilizadas durante la fase de entrenamiento ni validación. Este conjunto de pruebas incluyó:

• Aproximadamente 700 imágenes distribuidas proporcionalmente entre las clases persona, arma y persona armada.

• Escenarios variados con diferente iluminación, ángulos de cámara y densidad de objetos, representando condiciones operativas reales.

La Tabla 5 presenta un resumen de estas métricas, que incluyen la precisión y el recall como indicadores básicos, el F1-score como equilibrio entre ambas y los valores de mAP@50 y mAP@50-95 como referencias utilizadas para cuantificar la calidad de detección bajo diferentes umbrales.

Las herramientas de YOLOv8 calculan de manera automática las métricas de evaluación mientras se validaba el modelo. Los resultados de estas métricas se obtienen del entrenamiento y se guardan en la carpeta de resultados del conjunto de datos de test. Las funciones de metrics.py, que son parte del framework de YOLOv8, se emplean para calcular las métricas. Las mismas que se adquieren directamente de las estimaciones hechas en el conjunto de datos de prueba, sin el uso de herramientas externas.

Tabla 5. Métricas de Evaluación.
Fuente: Los autores.

Métrica	Descripción
Precisión (Precision)	Proporción de verdaderos positivos sobre todas las predicciones positivas realizadas.
Recall (Sensibilidad)	Proporción de verdaderos positivos correctamente identificados sobre todas las instancias reales.
F1-Score	Media armónica entre precisión y recall, útil para medir el balance entre ambas métricas.
mAP@50	Media de precisión considerando un umbral de IoU=0.5, estándar en la comunidad de visión por computador.
mAP@50–95	Media de precisión sobre múltiples umbrales de IoU (0.5 a 0.95 con pasos de 0.05), para una evaluación más rigurosa.

Fase 6: Despliegue

La fase de despliegue implicó la incorporación del modelo YOLOv8s en un prototipo funcional que era capaz de funcionar en tiempo real. Este sistema tiene la habilidad de procesar flujos de video, detectar objetos y lanzar alertas en tiempo real al descubrir acciones sospechosas como la presencia de individuos armados en áreas restringidas.

El modelo que se entrenó fue extraído del ambiente de entrenamiento y añadido a un prototipo de sistema de vigilancia por video, que fue creado con OpenCV y Python, empleando las capas siguientes:

Capa de datos: Cámaras IP o locales que, por medio del protocolo RTSP/IP, graban video en tiempo real y transmiten fotogramas de video para su procesamiento.

Capa de Backend:

• **Detección de objetos:** Para la detección en tiempo real, YOLOv8 es el componente esencial.

• **Servicios RESTful:** Flask v2.0 es la herramienta que se utiliza para administrar los servicios de backend; esta posibilita que el modelo de detección y el frontend se comuniquen entre sí.

e-ISNN 2550-6730 | Vol. 9, Núm. 2 (197-213): Julio-Diciembre, 2025 | DOI: 10.33936/isrtic.v9i2.7908

• **Comunicación en tiempo real:** Para garantizar una comunicación eficaz entre el frontend y el backend, se emplean WebSockets.

• **Envío de alertas:** Para la transmisión de alertas de amenaza o intrusión a los usuarios, se emplea la API v3 de Mailgun.

Capa de Frontend:

• Interfaz de usuario interactiva: Creada con Angular v12, posibilita la visualización en tiempo real del estado del sistema y de las detecciones.

• Recepción en tiempo real: Las alertas y la información de las cámaras llegan al frontend a través de WebSockets.

• Estructura y diseño UI: Se utiliza HTML/CSS para el diseño de la interfaz, con el fin de brindar una experiencia eficaz y amigable.

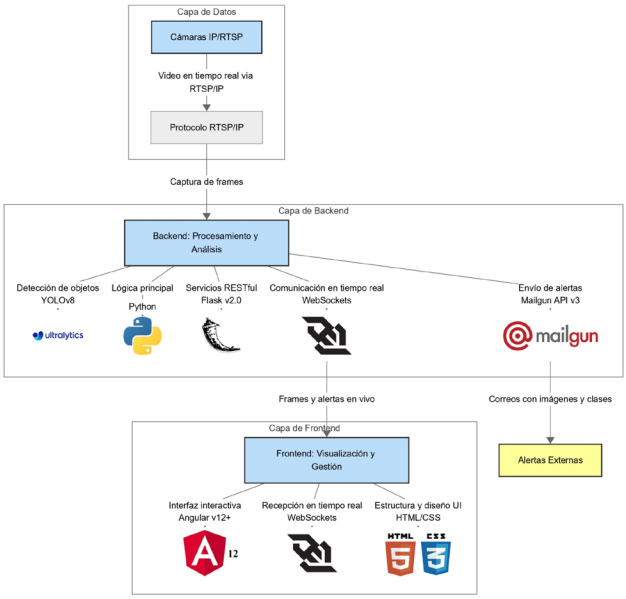

Figura 6. Arquitectura tecnológica del Prototipo.
Fuente: Los autores.

Diagrama de arquitectura tecnológica

El diagrama de la arquitectura tecnológica del sistema, que refleja las capas del software, la comunicación entre los componentes y el flujo de datos se ilustra en la Figura 6.

La Figura 7 presenta en detalle la arquitectura del prototipo, que muestra el funcionamiento del flujo de integración del modelo, desde que se captura el video en tiempo real hasta que se procesa con YOLOv8s y se clasifican las amenazas. Los resultados se almacenan en la parte trasera del sistema, se muestran en una página web y, si se identifican riesgos, el sistema notifica automáticamente al usuario.

Requerimientos de hardware y tiempo promedio de respuesta

El sistema de videovigilancia en tiempo real está creado para

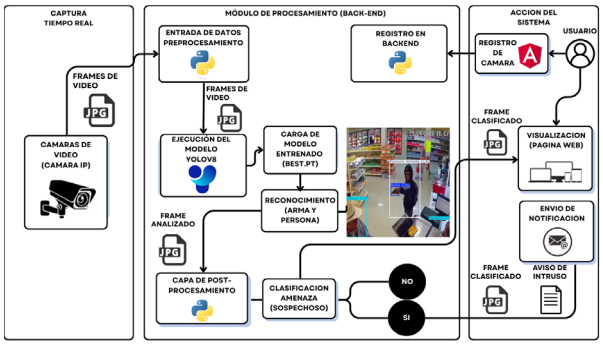

Figura 7. Arquitectura del Prototipo.
Fuente: Los autores.

funcionar con eficiencia elevada. Los requisitos mínimos de hardware y el promedio del tiempo de respuesta se describen a continuación:

Hardware:

• **GPU:** GeForce GTS 450/Radeon HD 6850, apropiada para el procesamiento de los modelos de detección en tiempo real.

• **Memoria RAM:** Se recomienda 8 GB de RAM para optimizar la carga de trabajo del sistema.

• **CPU:** Un procesador Intel Core i3 o similar para asegurar el procesamiento eficiente de los flujos de video.

Tiempo promedio de respuesta global:

• **Captura y procesamiento de frames:** El tiempo total por imagen es de aproximadamente 30 ms, lo cual incluye el preprocesamiento de 0.5 ms, la inferencia de 22 ms y el posprocesamiento de 7.5 ms.

Pruebas de usabilidad y validación funcional

Se llevaron a cabo pruebas de usabilidad y validación funcional, que integraron análisis cualitativos y mediciones cuantitativas, con el fin de determinar cuán eficiente es el sistema para detectar amenazas, sobre todo en cuanto a la identificación de personas armadas. Para evaluar la eficacia de las alertas en tiempo real, la facilidad de uso y la habilidad del sistema para mostrar imágenes de varias cámaras al mismo tiempo, los usuarios utilizaron la interfaz. Además, se llevaron a cabo exámenes funcionales con diez videos representativos de distintos contextos operativos que contenían variaciones en iluminación, ángulos de cámara y presencia de personas armadas y desarmadas. El objetivo era evaluar el desempeño del sistema en relación con falsos positivos y falsos negativos. Se utilizaron medidas estándar de visión por computadora, tales como precisión, mAP@50, mAP@50–95, recall y F1-score para medir su rendimiento. Los resultados obtenidos posibilitaron la confirmación de que el prototipo es robusto en situaciones adversas y controladas.

Consideraciones éticas y legales del sistema

El uso de tecnologías de videovigilancia inteligente conlleva responsabilidades legales y éticas que necesitan ser atendidas

e-ISNN 2550-6730 | Vol. 9, Núm. 2 (197-213): Julio-Diciembre, 2025 | DOI: 10.33936/isrtic.v9i2.7908

INFORMÁTICA Y SISTEMAS
Revista de Tecnologías de la Informática y las Comunicaciones

Facultad de Ciencias Informáticas
Universidad Técnica de Manabí
Av. Urbina y Che Guevara, Portoviejo, Ecuador
revista.iys@utm.edu.ec

explícitamente. Se utilizaron videos que se obtuvieron de fuentes públicas disponibles en plataformas como YouTube y sitios web de medios noticiosos, los cuales correspondían a circunstancias reales de robos e incidentes de seguridad registrados por cámaras de videovigilancia, durante la fase de validación experimental. Estos materiales fueron empleados solamente con propósitos académicos y de investigación, sin procesar datos biométricos identificables ni guardar información personal. Según Castro-Paredes et al. (2025), en Ecuador, la Ley Orgánica de Protección de Datos Personales define principios esenciales para el manejo de datos personales, como son la transparencia, el consentimiento informado y la reducción de datos.

3. Resultados y Discusión

Resultados Cuantitativos

Tras completar el proceso de entrenamiento y validación de las distintas configuraciones de YOLO, fue necesario establecer un criterio objetivo para determinar cuál de los modelos presentaba el mejor desempeño. Para ello, se recurrió a un conjunto de métricas ampliamente utilizadas en visión por computadora, que permiten medir no solo la precisión de las detecciones, sino también la capacidad del sistema para reconocer correctamente las clases definidas bajo diferentes condiciones. Estas métricas proporcionan una base cuantitativa sólida para comparar los resultados obtenidos en cada prueba y respaldar la selección final del modelo que será implementado en el prototipo de monitoreo inteligente.

Se evaluó el rendimiento de los modelos usando métricas estándar en visión por computador, las cuales fueron calculadas con base en las matrices de confusión producidas en cada contexto de prueba:

Precisión (Precision): calcula el porcentaje de verdaderos positivos en relación con todas las predicciones positivas.

$$P = TP / (TP + FP) \quad (1)$$

Sensibilidad (Recall): señala la relación entre la cantidad de verdaderos positivos que se han identificado correctamente y el total de casos reales.

$$R = TP / (TP + FN) \quad (2)$$

F1-Score: es la media armónica de las medidas de precisión y de recall.

$$F1 = 2 \times (P \times R) / (P + R) \quad (3)$$

mAP@50: promedio de precisión que tiene en cuenta un límite de IoU igual a 0.5.

$$mAP@50 = (1/N) \times \Sigma AP_i \quad \text{con IoU} \geq 0.5 \quad (4)$$

mAP@50-95: media de la precisión en varios límites de IoU.

$$mAP@50\text{-}95 = (1/10) \times \Sigma AP \square \quad \text{con } 0.5 \leq IoU \leq 0.95, \text{paso } 0.05 \quad (5)$$

Donde:

• TP (True Positives): cantidad de casos que se ha clasificado correctamente como positivos.

• FP (False Positives): cantidad de casos que han sido mal clasificados como positivos.

• FN (False Negatives): cantidad de casos positivos que el modelo no fue capaz de detectar.

• TN (True Negatives): cantidad de casos que se clasificaron de manera correcta como negativos.

Se evaluaron las 4 pruebas de YOLO con el uso de la matriz

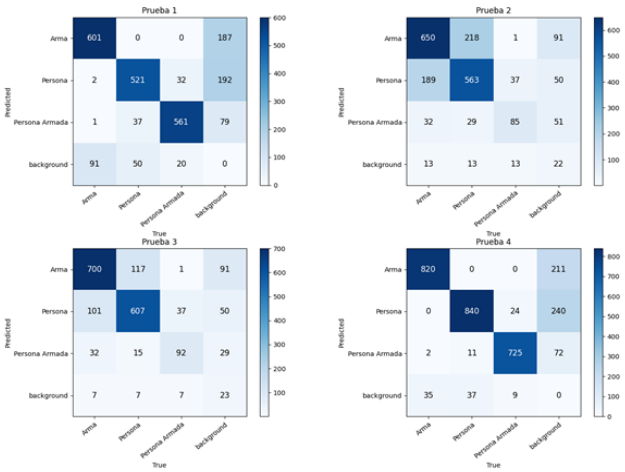

Figura 8. Matrices de evaluación de las Pruebas de YOLO. (Prueba 1) Primera imagen; (Prueba 2) Segunda imagen; (Prueba 3) Tercera imagen; (Prueba 4) Cuarta imagen. Fuente: Los autores.

de confusión de cada uno respectivamente como se observa en la Figura 8. Estas matrices permitieron observar la evolución del desempeño de los modelos desde la primera prueba con YOLOv8n hasta la configuración final que fue con YOLOv8s. La Tabla 6 y la Figura 9 presentan el rendimiento comparativo de los modelos en las métricas F1-score, recall, precisión y mAP. La Prueba 4 YOLOv8s alcanzó los valores más altos, con una precisión del 95.08 % y un mAP@50 de 96.97 %, demostrando así su habilidad superior para identificar objetos con gran exactitud. Por otro lado, los resultados de las pruebas 2 y 3 mostraron rendimientos moderados, lo que evidencia el impacto del tamaño del conjunto de datos y la configuración de hiperparámetros. La

e-ISNN 2550-6730 | Vol. 9, Núm. 2 (197-213): Julio-Diciembre, 2025 | DOI: 10.33936/isrtic.v9i2.7908

Prueba 1 YOLOv8n demostró una estabilidad inferior, lo que confirma que YOLOv8s brinda un mayor balance entre precisión

Tabla 6. Comparativa de rendimiento de los modelos entrenados.
Fuente: Los autores.

Modelo	Precisión (%)	Recall (%)	F1-Score (%)	mAP@50 (%)	mAP@50–95 (%)
Prueba 1 (YOLOv8n)	88.92	84.63	86.73	88.32	67.85
Prueba 2 (YOLOv8s)	63.64	62.46	63.05	61.67	43.61
Prueba 3 (YOLOv8s)	80.54	75.80	78.13	87.54	57.87
Prueba 4 (YOLOv8s)	95.08	92.94	93.99	96.97	72.68

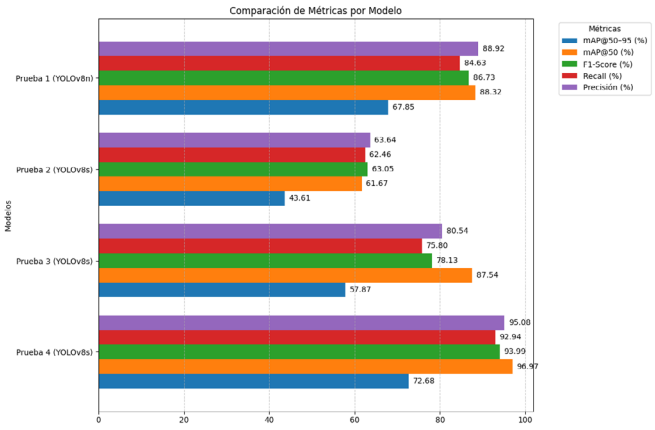

Figura 9. Comparación de métricas de rendimiento entre modelos YOLOv8n y YOLOV8s.
Fuente: Los autores.

y velocidad. Los resultados, en general, corroboran que la prueba final YOLOv8s es la más apropiada para el seguimiento inteligente, ofreciendo el mejor equilibrado entre alta precisión y la eficiencia requerida para hardware de bajo costo.

La Figura 10 muestra la interfaz del sistema de videovigilancia inteligente en funcionamiento, se puede observar el flujo de video en tiempo real donde resaltan las etiquetas de "persona", "arma" y "persona armada" siendo estos los objetos que están detectados por el modelo, también se encuentra el panel de las "detecciones activas", la cual resume cada hallazgo con su porcentaje precisión y sus coordenadas exactas. Para evaluar el prototipo de monitoreo inteligente, se realizaron diez pruebas experimentales utilizando videos grabados en interiores y exteriores de locales comerciales, en diferentes condiciones de iluminación y con variación en los ángulos de cámara. En cada escenario se representaron situaciones con personas, armas y personas armadas, con el fin de verificar la capacidad de detección y reconocimiento en un entorno real. El modelo seleccionado, YOLOv8s en su configuración final (Prueba 4), procesó los videos en tiempo real y los resultados obtenidos fueron contrastados con las predicciones esperadas a partir de la fase de entrenamiento, lo que permitió medir con precisión su desempeño en escenarios de seguridad no controlados.

Para profundizar en el rendimiento del sistema, se elaboró un análisis cuantitativo de errores a partir de los diez videos de prueba. En la Tabla 7 se puede verificar cada caso en donde se registraron los valores de Falsos Positivos (FP), Falsos Negativos (FN), Verdaderos Positivos (TP) y Verdaderos Negativos (TN) para cada clase de Persona, Arma y Persona armada, tienen diez escenarios de prueba.

• **Clase Persona:** alcanzó valores altos de TP en todos los

Tabla 7. Análisis cuantitativo de los videos por clase.
Fuente: Los autores.

Video	Clase	FP	FN	TP	TN
1	Persona	2	1	27	70
	Arma	1	2	8	89
	Persona armada	0	1	9	90
2	Persona	3	0	28	69
	Arma	2	1	10	87
	Persona armada	1	2	8	89
3	Persona	1	2	26	71
	Arma	0	3	9	88
	Persona armada	2	1	9	88
4	Persona	2	1	27	70
	Arma	1	1	11	87
	Persona armada	0	2	8	90
5	Persona	1	1	28	70
	Arma	0	2	10	89
	Persona armada	1	1	9	89
6	Persona	2	0	29	69
	Arma	1	2	9	88
	Persona armada	0	1	10	89
7	Persona	3	2	25	71
	Arma	2	3	7	88
	Persona armada	1	2	8	89
8	Persona	1	1	30	68
	Arma	1	2	9	88
	Persona armada	0	1	10	89
9	Persona	2	3	26	70
	Arma	3	2	8	87
	Persona armada	2	1	9	88
10	Persona	1	1	28	70
	Arma	1	1	10	88
	Persona armada	0	1	9	90

escenarios (26–30), con FN bajos (1–3), confirmando que el modelo mantiene su robustez para identificar individuos.

• **Clase Arma:** fue la más sensible a variaciones ambientales, con algunos FP (1–2) y FN (2–4), lo que evidencia confusiones con objetos similares en baja luz o ángulos difíciles.

e-ISNN 2550-6730 | **Vol. 9, Núm. 2 (197-213): Julio-Diciembre, 2025** | **DOI: 10.33936/isrtic.v9i2.7908**

INFORMÁTICA Y SISTEMAS
Revista de Tecnologías de la Informática y las Comunicaciones

Facultad de Ciencias Informáticas
Universidad Técnica de Manabí
Av. Urbina y Che Guevara, Portoviejo, Ecuador
revista.iys@utm.edu.ec

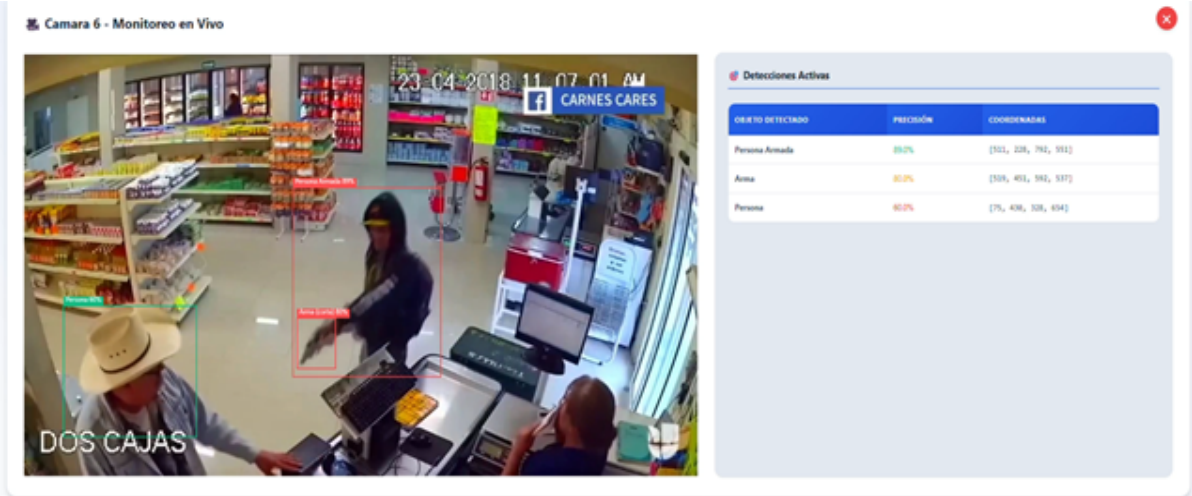

Figura 10. Interfaz de prototipo del sistema de vigilancia inteligente.
Fuente: Los autores.

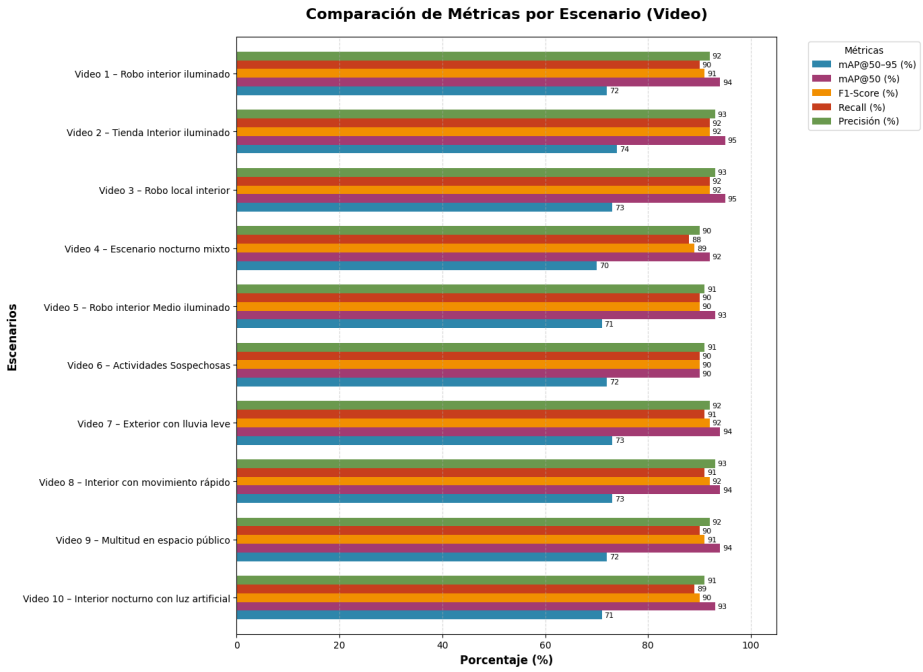

Figura 11. Comparación de métricas por escenario de validación del prototipo.
Fuente: Los autores.

Tabla 9. Matriz de confusión detallada del modelo final
Fuente: Los autores.

Matriz de confusión	Predicción Persona	Predicción Arma	Predicción Persona armada
Real: Persona	840	18	32
Real: Arma	40	211	15
Real: Persona armada	25	22	210

e-ISNN 2550-6730 | Vol. 9, Núm. 2 (197-213): Julio-Diciembre, 2025 | DOI: 10.33936/isrtic.v9i2.7908

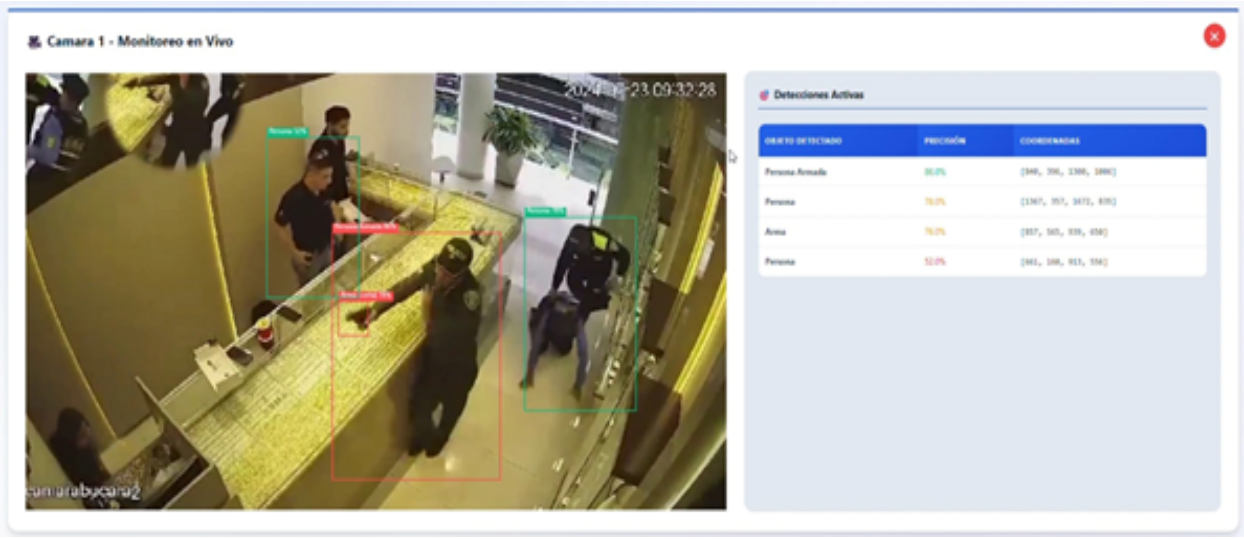

Figura 12. Desempeño del modelo en entorno controlado (Video 1 – Interior iluminado).
Fuente: Los autores.

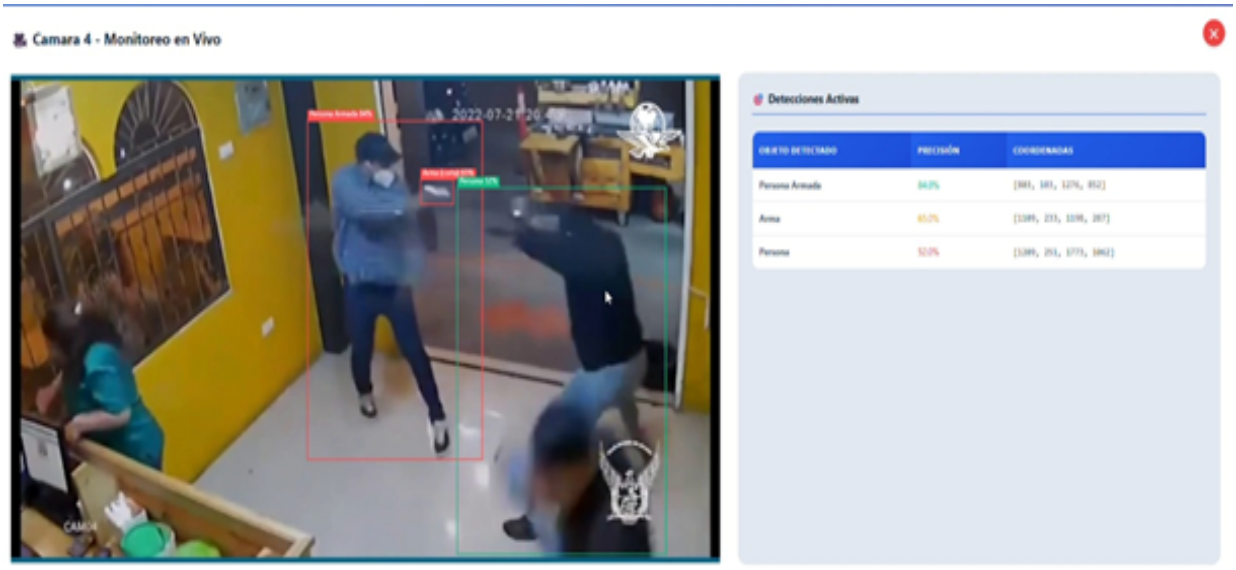

Figura 13. Desempeño del modelo en condiciones adversas (Video 4 – Escenario nocturno mixto).
Fuente: Los autores.

Clase Persona armada: se mantuvo en un rango de 8–10 TP por escenario, aunque con FN asociados a armas parcialmente ocultas y FP mínimos.

Los resultados obtenidos en cada escenario se muestran en una matriz donde se evalúan las métricas establecidas previamente, donde se realiza el cálculo de la diferencie entre lo previsto y lo obtenido como se observa en la Tabla 8 y en la Figura 11.

Análisis y discusión comparativa

Análisis de pruebas mediante las matrices de confusión

En la Prueba 1 (YOLOv8n) se evidenció un bajo desempeño en la detección de armas y personas armadas, con numerosos falsos negativos y confusiones hacia la clase background. Esto reflejó las limitaciones del modelo ligero para escenarios complejos. En la Prueba 2 (YOLOv8s) se registraron mejoras en la detección de personas, aunque persistieron falsos positivos entre arma y persona armada. En la Prueba 3 (YOLOv8s) se alcanzó un balance más favorable, reduciendo los errores en la clase persona, aunque la identificación de personas armadas continuó siendo un reto. Finalmente, la Prueba 4 (YOLOv8s) mostró el mejor desempeño, con una reducción significativa de falsos positivos y falsos negativos, logrando alta precisión en todas las clases críticas.

e-ISNN 2550-6730 | **Vol. 9, Núm. 2 (197-213): Julio-Diciembre, 2025** | **DOI: 10.33936/isrtic.v9i2.7908**

INFORMÁTICA Y SISTEMAS
Revista de Tecnologías de la Informática y las Comunicaciones

Facultad de Ciencias Informáticas
Universidad Técnica de Manabí
Av. Urbina y Che Guevara, Portoviejo, Ecuador
revista.iys@utm.edu.ec

Tabla 8. Comparación entre valores previstos y reales por escenario.
Fuente: Los autores.

Escenario (Video)	Métrica	Previsto (%)	Obtenido (%)	Diferencia
Video 1 – Robo interior iluminado	Precisión	95	92	-3.0
	Recall	93	90	-3.0
	F1-Score	94	91	-3.0
	mAP@50	97	94	-3.0
	mAP@50–95	76	72	-4.0
Video 2 – Tienda Interior iluminado	Precisión	95	93	-2.0
	Recall	93	92	-1.0
	F1-Score	94	92	-2.0
	mAP@50	97	95	-2.0
	mAP@50–95	76	74	-2.0
Video 3 – Robo local interior	Precisión	95	93	-2.0
	Recall	93	92	-1.0
	F1-Score	94	92	-2.0
	mAP@50	97	95	-2.0
	mAP@50–95	76	73	-3.0
Video 4 – Escenario nocturno mixto	Precisión	95	90	-5.0
	Recall	93	88	-5.0
	F1-Score	94	89	-5.0
	mAP@50	97	92	-5.0
	mAP@50–95	76	70	-6.0
Video 5 – Robo interior Medio iluminado	Precisión	95	91	-4.0
	Recall	93	90	-3.0
	F1-Score	94	90	-4.0
	mAP@50	97	93	-4.0
	mAP@50–95	76	71	-5.0
Video 6 – Actividades sospechozas	Precisión	95	91	-4.0
	Recall	93	90	-3.0
	F1-Score	94	90	-4.0
	mAP@50	97	93	-4.0
	mAP@50–95	76	71	-5.0
Video 7 – Exterior con lluvia leve	Precisión	95	92	-3.0
	Recall	93	91	-2.0
	F1-Score	94	92	-2.0
	mAP@50	97	94	-3.0
	mAP@50–95	76	73	-3.0
Video 8 – Interior con movimiento rápido de cámara	Precisión	95	93	-2.0
	Recall	93	91	-2.0
	F1-Score	94	92	-2.0
	mAP@50	97	94	-3.0
	mAP@50–95	76	73	-3.0
Video 9 – Multitud en espacio público diurno	Precisión	95	92	-3.0
	Recall	93	90	-3.0
	F1-Score	94	91	-3.0
	mAP@50	97	94	-3.0
	mAP@50–95	76	72	-4.0
Video 10 – Interior nocturno con luz artificial	Precisión	95	91	-4.0
	Recall	93	89	-4.0
	F1-Score	94	90	-4.0
	mAP@50	97	93	-4.0
	mAP@50–95	76	71	-5.0

Análisis de la Matriz de confusión del mejor modelo

La Tabla 9 presenta la matriz de confusión correspondiente a las pruebas realizadas con el modelo seleccionado, donde se resumen las predicciones correctas e incorrectas para cada clase definida en el sistema: persona, arma y persona armada.

El análisis reveló que:

• La mayor parte de los errores se concentraron en la confusión entre las clases arma y persona armada, ya que en varias ocasiones el modelo clasificó incorrectamente armas como personas armadas o viceversa. Esta situación puede atribuirse a la similitud visual y a la dificultad de identificar con precisión objetos pequeños en escenas con variaciones de ángulo o con oclusiones parciales.

• También se registraron casos en los que personas armadas fueron clasificadas únicamente como persona, lo que indica que la detección conjunta del individuo y el arma continúa siendo un reto en escenarios de mayor complejidad.

• En menor medida, se observaron confusiones de armas con objetos no etiquetados (como herramientas o teléfonos móviles), lo que sugiere la necesidad de refinar el dataset con más ejemplos negativos y diversificar las condiciones de entrenamiento para reducir estos falsos positivos.

Con base en los resultados obtenidos en la Tabla 6, la Prueba 4 (YOLOv8s) fue seleccionada como el mejor modelo debido a su rendimiento superior y más equilibrado en comparación con las configuraciones anteriores. Mientras que las pruebas iniciales evidenciaron limitaciones notables en la detección de armas y personas armadas, así como altos niveles de falsos positivos y negativos, la versión optimizada logró superar estas deficiencias alcanzando métricas de precisión (95.08 %), recall (92.94 %) y F1-score (93.99 %) que confirman su solidez en la clasificación de las tres clases críticas. En particular, el valor de mAP@50 (96.97 %) resalta su capacidad para identificar objetos con alta exactitud en condiciones diversas, mientras que el desempeño en mAP@50–95 (72.68 %) demuestra una robustez aceptable frente a variaciones en la superposición entre predicciones y etiquetas reales.

Análisis comparativos del prototipo en diversos escenarios

• **Video 1 (Robo interior iluminado):** mantuvo cifras elevadas y similares a lo anticipado, con variaciones de hasta el 3 % en Recall y Precisión. La detección ininterrumpida de armas y personas fue posible gracias a la iluminación adecuada en un ambiente regulado, lo que asegura un desempeño confiable.

• **Video 2 (Tienda interior iluminado):** el modelo se demostró

e-ISNN 2550-6730 | Vol. 9, Núm. 2 (197-213): Julio-Diciembre, 2025 | DOI: 10.33936/isrtic.v9i2.7908

eficaz, con cifras estables y una discrepancia de menos del 2 % respecto a lo que se esperaba. Se disminuyó el número de falsos negativos y la precisión se mantuvo en niveles altos debido a una iluminación homogénea, lo que permitió identificar mejor a las personas y a las armas.

• **Video 3 (Robo local interior):** este fue uno de los contextos más estables, con pérdidas mínimas y métricas que superan el 92 %. Un rendimiento parejo en la detección de todas las clases fue facilitado por un entorno cerrado y con condiciones homogéneas.

• **Video 4 (Escenario mixto):** mostró reducciones de hasta un 6 % en Recall y mAP@50-95, lo cual demuestra que el modelo es sensible a las condiciones difíciles de luz. La existencia de contrastes y sombras produjo ciertos falsos negativos, particularmente en el reconocimiento de armas.

• **Video 5 (Robo Interior medio iluminado):** exhibió un rendimiento aceptable, con descensos de entre el 3 % y el 5 %. El sistema mantuvo la estabilidad en la detección de individuos y una calidad aceptable en la identificación de armas, a pesar de que la iluminación no fue tan buena como en los primeros escenarios.

• **Video 6 (Actividad sospechosa):** mantuvo las métricas generales por encima del 90%, lo que confirma que el prototipo es capaz de funcionar en situaciones críticas. La identificación simultánea de personas y armas se llevó de manera confiable, aunque es importante señalar que sí hubo pérdidas mínimas en Recall, principalmente debido a la diversidad de situaciones en el escenario.

• **Video 7 – (Exterior con lluvia leve):** se analizó el desempeño en condiciones climáticas desfavorables. A pesar de que se notó un pequeño incremento en los falsos negativos de armas debido al desenfoque provocado por la lluvia, el modelo mantuvo detecciones estables.

• **Video 8 – (Interior con movimiento rápido de cámara):** se puso a prueba la resistencia del sistema frente a cambios súbitos de enfoque. Los resultados revelaron una buena estabilidad, manteniendo una alta exactitud en la identificación de personas y personas armadas.

• **Video 9- (Multitud en espacio público diurno):** se estudió la capacidad de detección en un entorno con una gran cantidad de personas y objetos. A pesar de que se dieron confusiones específicas entre armas y objetos semejantes, el sistema logró sostener un equilibrio razonable.

• **Video 10 – (Interior nocturno con luz artificial):** se constató la eficacia en condiciones de iluminación mixta, mostrando una ligera reducción del recall frente a los entornos diurnos con luz natural; no obstante, se detectaron correctamente las tres clases críticas.

El análisis comparativo entre el peor y el mejor escenario analizado mostró de manera directa cómo las condiciones del entorno y la luz influyen en el desempeño del modelo. En la Figura 12 (Video 1 -interior iluminado), se evidenció una detección confiable y estable de personas armadas, armas y personas, con métricas de precisión y recall que superaron el 90 %. La identificación precisa de los objetos fue posible gracias a la ausencia de oclusiones, al fondo uniforme y a la buena iluminación, lo que demostró que el modelo YOLOv8s puede funcionar con eficacia en escenarios controlados. Por otro lado, La Figura 13 (Video 4 - interior/exterior nocturno) fue el más difícil de todos, con caídas registradas en las métricas de recall y mAP@50–95 que alcanzaron hasta un 6 %. Las variaciones de luz, las sombras y los reflejos produjeron falsos negativos, sobre todo en la detección de armas, lo que demuestra que el modelo es sensible a condiciones desfavorables de iluminación.

Discusión

El análisis de los resultados hizo posible el reconocimiento preciso de las fortalezas y limitaciones del prototipo sugerido, así como su contribución distintiva en el ámbito de la videovigilancia inteligente. El modelo YOLOv8s mostró un rendimiento sólido, conservando indicadores de precisión y recall superiores al 90% bajo condiciones óptimas de iluminación, lo cual confirma su utilidad para usos de videovigilancia en tiempo real. Este comportamiento demuestra que el modelo tiene la habilidad de equilibrar la velocidad de inferencia con la precisión en la identificación de personas y armas, como se ha informado en investigaciones recientes sobre la misma arquitectura (Delwar et al., 2025; Nasir et al., 2025). Sin embargo, se notó que el rendimiento disminuyó hasta un 6% en condiciones de poca luminosidad, lo cual demuestra que la calidad del entorno visual continúa siendo un factor clave para percibir objetos pequeños o parcialmente visibles.

El prototipo desarrollado cumple con el objetivo de la investigación, logrando un sólido F1-score del 93.99%. La eficiencia computacional fue otro elemento significativo que se examinó, además de la precisión. Este estudio, a diferencia de la mayoría de los trabajos existentes que se enfocan en optimizar las métricas de precisión, también verificó el desempeño del prototipo en hardware asequible. Según lo indicado en la Fase 6, el sistema final funciona con un tiempo total de procesamiento de 30 ms, que incluye 22 ms para la inferencia por cuadro. Este descubrimiento es importante cuando se comparan nuestros resultados con los de la literatura. Aunque los trabajos de Nasir et al. (2025) y Delwar et al. (2025) ofrecen F1-scores teóricos más altos, del 94.7% y del 98% respectivamente, suelen enfocarse solo en la precisión algorítmica sin tratar el costo computacional o las exigencias de hardware. Aunque un modelo que necesita una GPU de gama alta puede parecer atractivo, no es viable implementarlo de manera masiva. En contraste, nuestra investigación muestra que es factible lograr una precisión elevada con un F1-score de 93.99%, y a la vez ser prácticamente viable, funcionando de manera eficaz en hardware de gama baja y para el consumo.

Esta democratización de la tecnología tiene efectos directos en situaciones latinoamericanas y en países en vías de desarrollo

e-ISNN 2550-6730 | Vol. 9, Núm. 2 (197-213): Julio-Diciembre, 2025 | DOI: 10.33936/isrtic.v9i2.7908

con recursos escasos. La arquitectura propuesta posibilita que otras instituciones apliquen el sistema sin depender de soluciones costosas y de propiedad exclusiva, al facilitar su replicabilidad por medio de herramientas de código abierto. A diferencia de esto, los estudios anteriores, como Schcolnik-Elias et al. (2023) y Gawande et al. (2024), optimizan la precisión, pero no toman en cuenta la escalabilidad económica. Los hallazgos evidencian que este enfoque no pone en riesgo la eficacia del sistema. Cuando se pondera la disminución de costos, la diferencia en rendimiento con respecto al hardware de gama alta es mínima; este hallazgo es especialmente significativo en un contexto donde la brecha tecnológica en aplicaciones de inteligencia artificial para la seguridad pública sigue creciendo.

Los hallazgos, en esencia, confirman la eficacia del prototipo y su aptitud para incorporarse a sistemas reales de vigilancia inteligente que cuenten con hardware comercial accesible. El hecho de que la calidad del video y la luminosidad ambiental dependan mutuamente es una limitación operativa intrínseca a los sistemas de visión por computadora. Para solucionar este problema, se deben utilizar en etapas posteriores sensores infrarrojos o métodos avanzados de aumento de datos específicos para condiciones con poca luz. Estas mejoras no solamente aumentarían la robustez y adaptabilidad del sistema en situaciones adversas, sino que también fortalecerían su aportación como una herramienta tecnológica escalable, factible y asequible para la seguridad ciudadana actual. Esto es particularmente importante en escenarios con recursos escasos en donde la necesidad de sistemas inteligentes para prevenir y responder rápidamente a amenazas es cada vez más urgente.

4. Conclusiones

El estudio fue capaz de crear y validar un prototipo de monitoreo inteligente fundamentado en YOLOv8s, el cual muestra un balance ideal entre una precisión alta y la eficiencia computacional. Los resultados corroboran que el modelo es una herramienta sólida para la detección en tiempo real, lo cual valida tanto su factibilidad técnica como su viabilidad económica al funcionar de manera eficaz en hardware asequible. Por ende, este trabajo proporciona una solución práctica y con capacidad de escalarse económicamente, que cierra la brecha entre la exactitud teórica de los modelos de IA y su implementación posible en sistemas de seguridad reales. La metodología CRISP-DM permitió la adecuada organización de las fases de modelado, evaluación y preparación, garantizando de esta manera un proceso laboral que es eficiente y reproducible.

Sin embargo, el estudio también estableció limitaciones inherentes al diseño de la investigación y a la naturaleza del conjunto de datos. La evaluación del desempeño en contextos no estructurados o con cambios drásticos está restringida por la dependencia de videos breves y regulados. Asimismo, la dificultad para detectar objetos y armas no letales y la sensibilidad del modelo en condiciones de escasa iluminación muestran que su exactitud de detección aún se encuentra sujeta a factores ajenos. A pesar de que estos límites no invalidan los descubrimientos, resaltan la importancia de fortalecer la robustez del modelo utilizando un conjunto de datos más diverso y representativo.

A partir de estas observaciones, se proponen múltiples líneas de investigación futura para optimizar el sistema: aumentar la diversidad del conjunto de datos incorporando escenas más diversas, implementar técnicas de aumento de datos más avanzadas, introducir sensores térmicos o infrarrojos en situaciones con poca luminosidad y analizar arquitecturas híbridas de detección que integren diferentes clases de redes neuronales. Estos progresos tienen el potencial de reducir los falsos positivos que surgen al clasificar objetos y mejorar la habilidad del sistema para operar en condiciones adversas.

Contribución de los autores

Lauro Alfonso Erreyes Cuenca: Administración del proyecto, investigación, redacción y metodología. **Nahin Josue Olmedo Chica:** Investigación, y edición del artículo. **Zea Mariuxi:** Metodología, revisión, redacción y edición del artículo. **Nancy Magaly Loja Mora:** Metodología, revisión, redacción y edición del artículo.

Conflictos de interés

Los autores no tienen conflictos de interés.

Referencias bibliográficas

Acuña-Cid, H. A., Ahumada-Tello, E., Ovalle-Osuna, Ó. O., Evans, R., Hernández-Ríos, J. E., & Zambrano-Soto, M. A. (2025). CRISP-NET: Integration of the CRISP-DM Model with Network Analysis. *Machine Learning and Knowledge Extraction, 7*(3), 101. https://doi.org/10.3390/make7030101

Alvarado, H. P. L., & Villavicencio, O. E. C. (2024). Regulación del Manejo de la Inteligencia Artificial, Consecuencias y Daños a la Sociedad por su Mal Uso. *Ciencia Latina Revista Científica Multidisciplinar, 8*(1), 1966–1978. https://doi.org/10.37811/cl_rcm.v8i1.9596

e-ISNN 2550-6730 | Vol. 9, Núm. 2 (197-213): Julio-Diciembre, 2025 | DOI: 10.33936/isrtic.v9i2.7908

Azatbekuly, N., Mukhanbet, A., & Bekele, S. D. (2024). Development of an Intelligent Video Surveillance System Based on YOLO Algorithm. 2024 IEEE 4th International Conference on Smart Information Systems and Technologies (SIST), 498–503. https://doi.org/10.1109/SIST61555.2024.10629617

Castro-Paredes, J. G., Mendoza-Masache, G. R., Loja-Mora, N. M., & Loján-Alvarado, H. P. (2025). Protección de datos por diseño y por defecto. Implicaciones legales en el desarrollo de software. *Ingenium et Potentia, 7*(12), 77–96. https://doi.org/10.35381/i.p.v7i12.4471

Cheng, G., Chao, P., Yang, J., & Ding, H. (2024). SGST-YOLOv8: An Improved Lightweight YOLOv8 for Real-Time Target Detection for Campus Surveillance. *Applied Sciences, 14*(12), 5341. https://doi.org/10.3390/app14125341

Delwar, T. S., Mukhopadhyay, S., Kumar, A., Singh, M., Lee, Y., Ryu, J.-Y., & Hosen, A. S. M. S. (2025). Real-Time Farm Surveillance Using IoT and YOLOv8 for Animal Intrusion Detection. *Future Internet, 17*(2), 70. https://doi.org/10.3390/fi17020070

Erreyes, L. (2025). Dataset Detección personas-armas-personas armadas. Zenodo. https://doi.org/10.5281/zenodo.17215185

Espinoza, L. J. M., Rojas, S. J. Z., Bravo, V. J. P., & Valarezo, L. C. C. (2025). Balanceo de Conjuntos de Datos Basado en Redes Generativas Aplicado a Imágenes del Sector Agrícola. *Informática y Sistemas, 9*(2), 164–176. https://doi.org/10.33936/isrtic.v9i2.7782

Gawande, U., Hajari, K., & Golhar, Y. (2024). Novel person detection and suspicious activity recognition using enhanced YOLOv5 and motion feature map. *Artificial Intelligence Review, 57*(2), 16. https://doi.org/10.1007/s10462-023-10630-0

Hua, C., Luo, K., Wu, Y., & Shi, R. (2024). YOLO-ABD: A Multi-Scale Detection Model for Pedestrian Anomaly Behavior Detection. *Symmetry, 16*(8), 1003. https://doi.org/10.3390/sym16081003

Li, Y., Li, Q., Pan, J., Zhou, Y., Zhu, H., Wei, H., & Liu, C. (2024). SOD-YOLO: Small-Object-Detection Algorithm Based on Improved YOLOv8 for UAV Images. *Remote Sensing, 16*(16), 3057. https://doi.org/10.3390/rs16163057

Nasir, R., Jalil, Z., Nasir, M., Alsubait, T., Ashraf, M., & Saleem, S. (2025). An enhanced framework for real-time dense crowd abnormal behavior detection using YOLOv8. Artificial *Intelligence Review, 58*(7), 202. https://doi.org/10.1007/s10462-025-11206-w

Ni, Y., Huo, J., Hou, Y., Wang, J., & Guo, P. (2024). Detection of Underground Dangerous Area Based on Improving YOLOV8. *Electronics, 13*(3), 623. https://doi.org/10.3390/electronics13030623

Schcolnik-Elias, A., Martínez-Díaz, S., Luna-Taylor, J. E., & Castro-Liera, I. (2023). Detección de armas tipo pistola mediante el uso de redes convolucionales con una arquitectura tipo YOLO y estereoscopía. *Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI, 11*, 196–204. https://doi.org/10.29057/icbi.v11iEspecial2.10727

Stainer, M. J., Raj, P. V, Aitken, B. M., Bandarian-Balooch, S., & Boschen, M. J. (2021). Decision-making in single and multiple-screen CCTV surveillance. *Applied Ergonomics, 93*, 103383. https://doi.org/10.1016/j.apergo.2021.103383

Sudharson, D., Srinithi, J., Akshara, S., Abhirami, K., Sriharshitha, P., & Priyanka, K. (2023). Proactive Headcount and Suspicious Activity Detection using YOLOv8. *Procedia Computer Science, 230*, 61–69. https://doi.org/10.1016/j.procs.2023.12.061

e-ISNN 2550-6730 | **Vol. 9, Núm. 2 (197-213): Julio-Diciembre, 2025** | **DOI: 10.33936/isrtic.v9i2.7908**