



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO
CENTRO UNIVERSITARIO UAEM ATLACOMULCO

“Análisis comparativo de clasificadores densos y
detectores de una etapa para la identificación automatizada de
violencia en secuencias de imágenes”

T E S I S

Que para obtener el Título de:
Ingeniero en Computación

Presenta:
Yessenia Mora Esquivel

Director de Tesis:
Dr. Everardo Efrén Granda Gutiérrez

Atlacomulco, México; agosto del 2026

Resumen

El presente trabajo de investigación propone evaluar la viabilidad operativa de diversas arquitecturas de *Deep Learning* aplicadas a la detección automatizada de violencia en sistemas de videovigilancia. El problema central aborda la deficiencia técnica de los modelos tradicionales de clasificación, que presentan alta exactitud en entornos controlados, pero fallan al enfrentarse a escenarios visuales desconocidos. Asimismo, la alta complejidad computacional de estos algoritmos genera latencias que imposibilitan el procesamiento de video en tiempo real y limitan su aplicación efectiva en la seguridad ciudadana.

Para abordar esta problemática, se propone una metodología comparativa basada en criterios de confiabilidad propios de la Ingeniería en Computación. Se contrastan redes neuronales convolucionales de múltiples etapas frente a un modelo de detección de objetos de una sola etapa, denominado YOLOv8. El enfoque principal consiste en someter a los algoritmos a pruebas de estrés computacional con conjuntos de datos externos altamente desequilibrados en clases, priorizando la sensibilidad de detección sobre la exactitud global y realizando evaluaciones de eficiencia para medir la latencia por cuadro de imagen.

Los resultados experimentales demostraron una degradación severa en la detección de violencia física en entornos desconocidos y una velocidad de detección inviable, a excepción del modelo de una sola etapa, convirtiéndolo en la mejor opción operativa.

Considerando lo descrito anteriormente, esta investigación realiza una aportación técnica a la disciplina al comprobar empíricamente que, para el diseño de sistemas de alerta temprana en tiempo real, es imperativo priorizar la capacidad de reacción y la resiliencia sobre la precisión matemática aislada. La aplicación de arquitecturas de detección espacial para resolver cuellos de botella en el análisis de video establece un estándar de diseño escalable y altamente eficiente para la automatización futura de la seguridad mediante sistemas de videovigilancia.

Palabras clave: desplazamiento de dominio, clasificadores densos, detectores de una etapa, eficacia, eficiencia.

Abstract

This research evaluates the operational feasibility of various Deep Learning architectures for automated detection of violent events in video surveillance systems. The study addresses the technical limitations of traditional classification models, which often achieve high accuracy in controlled conditions but degrade significantly when exposed to unfamiliar visual scenarios. Furthermore, the high computational complexity of these algorithms introduces latency that prevents real-time video processing and limits their effective application in public safety.

To address these challenges, this study proposes a comparative method based on reliability criteria in Computer Engineering. Multi-stage convolutional neural network architectures are compared with the one-stage object detection model YOLOv8. The methodology subjects both approaches to computational stress tests using external datasets with significant class imbalance, prioritizes detection sensitivity over overall accuracy, and evaluates computational efficiency by measuring latency per video frame.

The experimental results show a significant drop in violent-event detection when exposed to unfamiliar visual scenarios, along with infeasible processing speed, except for the one-stage model, which emerged as the best operational choice.

The findings provide empirical evidence that the design of real-time early warning systems should prioritize responsiveness and resilience over isolated mathematical precision. By proving the advantages of spatial object detection architectures for mitigating computational bottlenecks in video analysis, this research shows a scalable, computationally efficient framework for further development of intelligent video surveillance systems.

Keywords: Domain Shift, dense classifiers, one-stage detectors, effectiveness, efficiency.

Índice

Resumen	2
Abstract.....	3
Índice	4
Índice de tablas	7
Índice de algoritmos	8
Índice de figuras	9
1 INTRODUCCIÓN	11
2 PLANTEAMIENTO DEL PROBLEMA	13
2.1 Definición del problema	13
2.2 Objetivos de investigación.....	14
2.3 Preguntas de investigación	15
2.4 Hipótesis	16
2.5 Justificación.....	16
2.6 Impactos.....	17
2.7 Limitaciones	18
3 ESTADO DEL ARTE	20
3.1 Inteligencia artificial para la detección de violencia	20
3.2 Desafíos computacionales y de detección	21
3.3 Comparativa entre modelos de detección.....	24
3.4 Detección de objetos.....	25
3.5 Métodos alternativos para la detección de violencia	27
3.6 Algoritmos de visión por computadora	31
3.7 Algoritmos para la detección de violencia	32
3.8 Síntesis de las brechas identificadas	38

4	MARCO CONCEPTUAL.....	44
4.1	Secuencias de imágenes y video.....	44
4.2	Inteligencia artificial y visión por computadora.....	47
4.3	Métricas para la evaluación de modelos de IA.....	51
5	MÉTODOS	54
5.1	Tipo y diseño de la investigación	55
5.2	Requerimientos y herramientas	56
5.2.1	Conjunto de datos de prueba	56
5.2.2	Herramientas de desarrollo.....	57
5.2.3	Lenguaje de programación	58
5.2.4	Librerías y paquetes de software	59
5.3	Diseño e implementación	60
5.3.1	Algoritmos de entrenamiento y validación.....	61
5.4	Experimentación.....	67
5.4.1	Configuración inicial	67
5.4.2	Entrenamiento.....	67
5.4.3	Evaluación de las métricas	69
5.4.4	Prueba de latencia.....	69
6	RESULTADOS.....	71
6.1	Exploración de los conjuntos de datos	71
6.1.1	Real Life Violence Dataset (RLVS).....	71
6.1.2	Smart City CCTV Violence Detection (SCVD).....	74
6.1.3	AIRTLAB.....	76
6.1.4	Extracción de frames	78
6.2	Desempeño durante la fase de entrenamiento	79

6.2.1	Hiperparámetros	80
6.2.2	Entrenamiento de EfficientNetB0	81
6.2.3	Entrenamiento de ResNet50	82
6.2.4	Entrenamiento de YOLOv8.....	83
6.2.5	Exactitud del entrenamiento de los modelos	84
6.3	Desempeño durante la fase de evaluación.....	85
6.3.1	Pruebas de Generalización.....	85
6.3.2	Análisis del área bajo la curva.....	90
6.4	Pruebas de Robustez.....	91
6.5	Evaluación de eficiencia computacional	97
6.6	Síntesis y comparativa global	100
CONCLUSIONES.....		102
Referencias		104
Glosario de acrónimos		114

Índice de tablas

Tabla 1. Matriz de referencias para determinar el estado del arte.	39
Tabla 2. Elección de modelos pre-entrenados.	61
Tabla 3. Características de los conjuntos de datos seleccionados.	78
Tabla 4. Balance de los conjuntos de datos en función del número de frames extraídos.	79
Tabla 5. Exactitud del entrenamiento de los modelos.	84
Tabla 6. Exactitud de la evaluación de los modelos frente a la prueba de generalización. ..	85
Tabla 7. Comparativa entre la exactitud del entrenamiento contra la evaluación de generalización.	86
Tabla 8. Métricas de la evaluación de generalización para EfficientNetB0.	87
Tabla 9. Métricas de la evaluación de generalización para ResNet50.	88
Tabla 10. Métricas de la evaluación de generalización para YOLOv8.	89
Tabla 11. Comparativa entre los modelos a partir de sus matrices de confusión.	90
Tabla 12. Exactitud de la evaluación de los modelos frente a la prueba de robustez.	91
Tabla 13. Comparativa sobre la exactitud en las diferentes etapas.	92
Tabla 14. Métricas de la evaluación de robustez para EfficientNetB0.	93
Tabla 15. Métricas de la evaluación de robustez para ResNet50.	94
Tabla 16. Métricas de la evaluación de robustez para YOLOv8.	95
Tabla 17. Comparativa entre los modelos a partir de sus matrices de confusión.	96
Tabla 18. Comparativa de latencia y velocidad de inferencia.	99

Índice de algoritmos

Algoritmo 1 Primera etapa: configuración general y extracción de frames	62
Algoritmo 2 Segunda etapa: entrenamiento de los modelos	64
Algoritmo 3 Tercera etapa: testeo y métricas	65

Índice de figuras

Figura 1. Visor de frames de un video de 6 FPS.	45
Figura 2. Comparación entre diferentes tasas de fotogramas por segundo.	46
Figura 3. Secuencia de video sin Frame Dropping.	47
Figura 4. Secuencia de video con Frame Dropping.	47
Figura 5. Contraste entre el balance y el desbalance de clases.	50
Figura 6. Dominios.	51
Figura 7. Matriz de confusión.	52
Figura 8. Curvas AUC-ROC.	53
Figura 9. Diagrama de flujo de la configuración inicial.	68
Figura 10. Diagrama de flujo del entrenamiento de los modelos.	68
Figura 11. Diagrama de flujo de la evaluación con SCVD.	69
Figura 12. Diagrama de flujo de la evaluación con AIRTLAB.	70
Figura 13. Diagrama de flujo de la prueba de latencia de los modelos.	70
Figura 14. Escenas violentas del conjunto de datos RLVS.	72
Figura 15. Escenas pacíficas del conjunto de datos RLVS.	73
Figura 16. Ejemplos de imágenes del conjunto de datos RLVS que plantean un desafío de clasificación.	73
Figura 17. Escenas violentas del conjunto de datos SCVD.	74
Figura 18. Escenas pacíficas del conjunto de datos SCVD.	75
Figura 19. Ejemplos de imágenes del conjunto de datos SCVD que presentan un desafío de clasificación.	75
Figura 20. Escenas violentas del conjunto de datos AIRTLAB.	76
Figura 21. Escenas pacíficas del conjunto de datos AIRTLAB.	77
Figura 22. Ejemplos de imágenes del conjunto de datos AIRTLAB que presentan un desafío de clasificación.	77
Figura 23. Curva de la función de pérdida del entrenamiento de EfficientNetB0.	82
Figura 24. Curva de la función de pérdida del entrenamiento de ResNet50.	83
Figura 25. Curva de la función de pérdida del entrenamiento de YOLOv8.	84

Figura 26. Matriz de confusión de la evaluación de EfficientNetB0 en las pruebas de generalización.....	87
Figura 27. Matriz de confusión de la evaluación de ResNet50 en las pruebas de generalización.....	88
Figura 28. Matriz de confusión de la evaluación de YOLOv8 en las pruebas de generalización.....	89
Figura 29. Área bajo la curva de la evaluación para pruebas de generalización.....	91
Figura 30. Matriz de confusión de la evaluación de EfficientNetB0 en las pruebas de robustez.....	93
Figura 31. Matriz de confusión de la evaluación de ResNet50 en las pruebas de robustez.	94
Figura 32. Matriz de confusión de la evaluación de YOLOv8 en las pruebas de robustez..	95
Figura 33. Área bajo la curva de la evaluación para pruebas de robustez.....	97
Figura 34. Representación de los FPS.	98
Figura 35. Comparativa de velocidad (FPS).	99

1 INTRODUCCIÓN

Actualmente se utilizan miles de cámaras de videovigilancia que registran lo que ocurre en las calles, escuelas y centros de trabajo. Sin embargo, esta abundancia de información visual ha superado por completo la capacidad de monitoreo humano. Un operador de seguridad no puede mantener su nivel de atención al observar decenas de pantallas simultáneamente durante horas sin sufrir fatiga ni distracciones. Como consecuencia, la mayoría de las cámaras funcionan como herramientas de grabación pasiva: no previenen incidentes, sino que sus videos solo se revisan una vez que un acto de violencia ya ha ocurrido. A pesar de ello, el juicio humano sobre un acontecimiento puede verse afectado por sus creencias o por conveniencia.

Los sistemas de videovigilancia evolucionan para adaptarse al cambio de las variables del entorno y a nuevas necesidades. El primer sistema de videovigilancia estaba compuesto únicamente por una cámara y una radio de onda corta. Con los años, comenzaron a desarrollarse sistemas de almacenamiento de grabaciones; alrededor de los años 70, estas se almacenaban en casetes, pequeñas cajas de plástico que contenían un rollo de cinta magnética que debía ser reemplazado manualmente por uno nuevo una vez que se llenara. Asimismo, en ocasiones se optaba por sobrescribir la misma cinta, provocando la pérdida de los datos e imágenes de la grabación anterior (2M Technology, 2024). A partir de los 90, comenzaron a emplearse herramientas digitales en los sistemas de videovigilancia, trascendiendo de imágenes en blanco y negro, de baja resolución y falta de claridad en el audio a imágenes a color, alta resolución, claridad de audio, modalidad de grabación nocturna, almacenamiento de las grabaciones en dispositivos locales más seguros o en servidores en la nube y grabaciones en tiempo real.

No fue hasta la época de la pandemia que los sistemas de seguridad se vieron potenciados con la llegada de mejores herramientas de video, como la integración del *Deep Learning* (Cásares, 2022). Lo más destacable, en un principio, fue la detección del movimiento o de la silueta de las personas; posteriormente, los sistemas de videovigilancia incorporaron redes neuronales convolucionales para la detección de objetos, personas o eventos de acuerdo con una serie de reglas o condiciones previamente establecidas.

En este sentido, la motivación principal para desarrollar este proyecto de investigación proviene de la necesidad de transformar estos sistemas pasivos de videovigilancia en herramientas activas de prevención y alerta temprana. Para lograrlo, la inteligencia artificial y la visión por computadora entrenan sistemas computacionales para identificar agresiones físicas de forma automática y sin intervención humana.

No obstante, llevar esta tecnología de un entorno controlado de laboratorio al mundo real supone desafíos significativos. Muchos programas o algoritmos enfrentan dos grandes problemas: primero, tienden a confundirse o fallar cuando cambian las condiciones del entorno, como la iluminación, el fondo o el ángulo de la cámara (fenómeno conocido como desplazamiento de dominio). Segundo, los modelos matemáticos más precisos suelen ser tan complejos y pesados que analizan el video con demasiada lentitud, perdiendo segundos significativos de información, lo cual es inaceptable en una situación de emergencia, donde cada instante cuenta (Parlamentario, Instituto de Derecho, 2024) (Lenovo, 2024).

Bajo este contexto, esta investigación se enfoca en evaluar diferentes modelos pre-entrenados de inteligencia artificial (redes neuronales convolucionales) para determinar si son capaces de operar en un sistema de seguridad en vivo. El enfoque no se limita a buscar el que tiene la mejor calificación teórica, sino aquel que logra un equilibrio entre ser lo suficientemente rápido para procesar video en tiempo real y lo bastante robusto para detectar amenazas reales en escenarios nuevos y desconocidos.

Al finalizar la experimentación, los descubrimientos más importantes revelaron que los modelos tradicionales más densos y complejos, aunque son eficientes en pruebas controladas por su alta exactitud de detección, fracasan en entornos desconocidos o resultan demasiado lentos para el video en vivo. En contraste, un modelo enfocado en la detección ágil y directa superó ampliamente las expectativas operativas en términos de eficacia y eficiencia. Este modelo fue capaz de analizar flujos de video a velocidades sobresalientes (más de 260 cuadros por segundo), manteniendo una buena sensibilidad para capturar actos de violencia, posicionándose como la solución tecnológica más segura y viable para integrarse en cámaras de seguridad reales, a pesar de generar advertencias ocasionales ante movimientos bruscos, que no necesariamente constituyen actos de violencia.

2 PLANTEAMIENTO DEL PROBLEMA

La inmensa cantidad de imágenes recolectadas a partir de grabaciones obtenidas por sistemas de videovigilancia frecuentemente pasa desapercibida, sin un análisis concreto ni una detección en tiempo real, permitiendo que las grabaciones solo tengan como objetivo la justificación o el testimonio frente a una incidencia de violencia física consumada.

En este apartado se analiza esta problemática, abordando su detección, las repercusiones que conlleva y la posibilidad de hallar una solución viable. A su vez, se presentan tanto el objetivo general como los objetivos específicos. En los que se explica el procedimiento a seguir y los resultados que se desean obtener en cada caso, así como sus limitaciones individuales. Igualmente, se exponen las preguntas de investigación que guiarán todo el proceso metodológico, teniendo presente el enfoque del proyecto de investigación y sus limitaciones. Por otro lado, se expone la necesidad e importancia de la problemática planteada considerando los puntos anteriores. Finalmente, se manifiestan los impactos que tendrá la investigación y sus alcances, así como sus posibles usos y los beneficios de aplicar los resultados teóricos en un entorno real.

2.1 Definición del problema

En la actualidad, el incremento del número de cámaras de videovigilancia (CCTV, por sus siglas en inglés, Closed-Circuit Television) ha generado un volumen de datos visuales que supera la capacidad de monitoreo humano, resultando en sistemas puramente reactivos frente a incidentes de violencia. Aunque existen soluciones de visión por computadora basadas en *Deep Learning* para automatizar esta tarea, la mayoría enfrenta un problema crítico: operan de manera deficiente fuera de los entornos controlados en los que fueron entrenadas (fenómeno de desplazamiento de dominio o *Domain Shift*) (Taboga, 2021). Además, muchos de los modelos clásicos de clasificación de video exigen altos recursos computacionales, lo que genera una latencia inaceptable (bajos cuadros por segundo) que imposibilita la detección de agresiones físicas rápidas en tiempo real (SharkYun, 2024). Por lo tanto, el problema radica en la dificultad de encontrar un modelo computacional que logre un equilibrio viable entre una alta sensibilidad para detectar la amenaza en entornos desconocidos y la velocidad necesaria para operar en tiempo real.

Dicho esto, el alcance de la presente investigación se delimita a la evaluación del rendimiento de tres modelos preexistentes de redes neuronales convolucionales (CNNs, por sus siglas en inglés, Convolutional Neural Networks): dos clasificadores de múltiples etapas (EfficientNetB0 y ResNet50) y un detector de una sola etapa (YOLOv8). El proyecto se centra exclusivamente en el análisis de secuencias de video pregrabadas provenientes de tres conjuntos de datos públicos: para el entrenamiento, RLVS (Mohamed, Mohamad, & Abd El Massih, 2019), y para las pruebas de generalización y robustez, SCVD (Aremu, Li, & Alameeri, 2024) y AIRTLAB (airtlab, 2020).

La investigación abarca desde el preprocesamiento de los datos y la aplicación de *Transfer Learning*, hasta la evaluación de métricas de eficacia (*Accuracy*, *Recall*, *F1-Score*) y eficiencia (Latencia y FPS, del inglés, *Frames Per Second*) en un entorno de hardware controlado (GPU T4, del inglés, *Graphics Processing Unit Tesla 4*). Cabe resaltar que el alcance de este trabajo no incluye la implementación física del modelo en hardware de cámara, el desarrollo de una arquitectura o de un modelo de red neuronal desde cero, ni el análisis de audio asociado a los videos.

La brecha existente entre el conocimiento actual y el nuevo radica en el enfoque de evaluación de estos sistemas. La literatura actual suele priorizar el uso de modelos densos, que presentan alta exactitud global en conjuntos de datos homogéneos. El nuevo conocimiento aportado por esta investigación demuestra empíricamente que, para el nicho específico de la seguridad y bajo condiciones de desequilibrio de clases y de desplazamiento de dominio, dichos modelos fallan en su funcionamiento. La investigación comprueba que es más viable utilizar detectores de una sola etapa que, aunque sacrifican la especificidad estadística al generar falsas alarmas ocasionales, garantizan la resiliencia en la recuperación de eventos violentos a velocidades superiores a los 260 FPS, un estándar de eficiencia no documentado en comparativas clásicas de clasificación binaria de la violencia.

2.2 Objetivos de investigación

Objetivo general

Evaluar el desempeño de tres modelos de *Deep Learning* pre-entrenados (EfficientNetB0, ResNet50 y YOLOv8) aplicados a la detección automatizada de violencia

en secuencias de video, con el propósito de determinar el modelo que ofrezca el mejor equilibrio entre la resiliencia ante el desplazamiento de dominio (eficacia) y el procesamiento en tiempo real (eficiencia).

Objetivos específicos

1. Acondicionar los conjuntos de datos de video mediante técnicas de extracción de cuadros (*frames*) a una frecuencia estandarizada, estableciendo los dominios base para el entrenamiento y las pruebas ciegas.
2. Entrenar los modelos base seleccionados mediante técnicas de transferencia de aprendizaje (*Transfer Learning*) y mecanismos de parada temprana (*Early Stopping*) para adaptar la extracción de características visuales a la clasificación binaria de la violencia.
3. Cuantificar la eficiencia computacional de cada modelo mediante pruebas de estrés, midiendo la latencia de inferencia y la tasa de procesamiento en un entorno de hardware controlado.
4. Evaluar la eficacia y la capacidad de generalización de los modelos frente al fenómeno de desplazamiento de dominio y al desbalance de clases, priorizando el análisis de la sensibilidad sobre la exactitud global en escenarios desconocidos.

2.3 Preguntas de investigación

Derivado de la definición técnica anterior, la pregunta central que guiará esta investigación es:

¿De qué manera el uso de un modelo de detección de objetos de una sola etapa, en comparación con los clasificadores convolucionales densos, optimiza el equilibrio entre la sensibilidad de detección frente al desplazamiento de dominio y la eficiencia computacional en sistemas de videovigilancia en tiempo real?

En el mismo sentido, de esta interrogante central se desprenden las siguientes preguntas específicas:

- ¿Cuál es el impacto del fenómeno de desplazamiento de dominio y del desbalance de clases en la capacidad de generalización de los modelos evaluados al pasar del conjunto de entrenamiento base a entornos de prueba externos y desconocidos?

- ¿En qué medida la complejidad estructural de los modelos analizados determina su viabilidad operativa, medida por la latencia de inferencia y la tasa de procesamiento, para el análisis continuo de video en hardware estándar?
- ¿Qué modelo ofrece el mejor compromiso técnico-operativo al priorizar la máxima captura de eventos violentos frente a la pérdida de especificidad en escenarios de seguridad?

2.4 Hipótesis

La implementación de un modelo de *Deep Learning* de una sola etapa orientado a la detección presentará una resiliencia superior frente al fenómeno de desplazamiento de dominio en secuencias de videovigilancia, manteniendo una alta sensibilidad para la detección de violencia y logrando una eficiencia computacional en tiempo real (>30 FPS). En contraste, los clasificadores convolucionales densos de múltiples etapas sufrirán una degradación crítica de su capacidad de generalización por sobreajuste al fondo o presentarán una latencia inoperante, lo que demuestra que, para sistemas de seguridad en vivo, el diseño arquitectónico de una sola etapa es el único que satisface el compromiso entre eficacia y tiempo de reacción.

2.5 Justificación

Gran parte de la literatura científica actual enfocada en la detección de violencia prioriza el diseño de arquitecturas sumamente profundas y complejas para alcanzar una exactitud casi perfecta en conjuntos de datos homogéneos y controlados. Sin embargo, en el ámbito de la videovigilancia, las condiciones de iluminación, los ángulos de cámara y los entornos cambian drásticamente. Un modelo que es matemáticamente preciso en el laboratorio resulta inútil si presenta una latencia alta que le impide procesar video en tiempo real o si colapsa ante el fenómeno de desplazamiento de dominio.

La principal contribución técnica de este proyecto radica en ofrecer una comparativa empírica que desafía el paradigma tradicional de clasificación. Al someter arquitecturas dispares a pruebas de estrés de inferencia computacional y a escenarios visuales desconocidos y desbalanceados, esta investigación resolvió una interrogante fundamental en la ingeniería de visión por computadora aplicada a la seguridad: determinar qué tipo de

modelo ofrece el mejor compromiso entre el tiempo de reacción y la sensibilidad de detección. Los hallazgos obtenidos aportan certidumbre técnica, demostrando que los detectores de una sola etapa superan operativamente a los clasificadores de múltiples etapas al priorizar la captura de amenazas a velocidades superiores a los 260 FPS.

Si bien este proyecto se desarrolló en un entorno controlado de experimentación computacional, sus beneficios directos impactan a la comunidad de integradores y desarrolladores de sistemas de seguridad automatizados. Al culminar esta investigación, se proporciona un marco de referencia validado que facilita la toma de decisiones tecnológicas y evita la inversión de recursos en arquitecturas ineficientes para el procesamiento de flujos de video. En un plano proyectado, la aplicación de estos hallazgos en sistemas de circuito cerrado (CCTV) del sector público o privado tiene el potencial de beneficiar a la sociedad civil al transformar el monitoreo pasivo en un sistema de alertas tempranas y reducir significativamente los tiempos de respuesta de las autoridades ante agresiones físicas.

2.6 Impactos

El impacto científico de este trabajo radica en la aportación de evidencia empírica sobre los efectos del desplazamiento de dominio (*Domain Shift*) en arquitecturas de *Deep Learning* aplicadas a la videovigilancia. La investigación desmitifica la dependencia exclusiva de la exactitud global (*Accuracy*) como métrica de éxito, demostrando matemáticamente que, en escenarios desbalanceados, los clasificadores densos (como ResNet50) sufren sobreajuste de fondo (*Background Overfitting*). Este análisis aporta a la comunidad científica una validación empírica de que la sensibilidad (*Recall*) debe ser el estándar de evaluación primario para la detección de violencia.

Desde la perspectiva tecnológica, el proyecto establece un marco de referencia de viabilidad operativa. Se comprobó que la adopción de modelos de una sola etapa, específicamente YOLOv8, resuelve el cuello de botella de la latencia en el procesamiento de video. Al pasar de los 14 FPS de los modelos tradicionales a más de 260 FPS.

En el ámbito económico, la alta eficiencia computacional demostrada por YOLOv8 permite proyectar una reducción de costos significativa en la infraestructura de seguridad. Dado su bajo consumo de milisegundos por *frame*, un solo servidor equipado con hardware

de procesamiento gráfico de gama media podría analizar flujos de video de múltiples cámaras simultáneamente. Socialmente, el uso potencial de esta tecnología en centros de monitoreo promete reducir la dependencia de la vigilancia humana pasiva, traducándose en respuestas policiales o médicas más rápidas y, en última instancia, en entornos más seguros para la ciudadanía.

Como resultado tangible del desarrollo metodológico y la experimentación de esta tesis, se generaron los siguientes entregables y herramientas computacionales:

1. Modelos de detección optimizados: archivos de pesos entrenados (.keras y .pt) correspondientes a los modelos pre-entrenados, ajustados para la discriminación binaria de violencia, preparados para ser desplegados en entornos de inferencia.
2. Algoritmos de preprocesamiento y evaluación: scripts y libretas computacionales que automatizan la extracción de cuadros de video a frecuencias específicas y la generación de métricas de rendimiento.
3. Base de resultados comparativa: registros documentados (.csv) que detallan el comportamiento exacto de los modelos frente a escenarios desconocidos, los cuales pueden ser utilizados por futuras investigaciones como punto de partida o comparación.

2.7 Limitaciones

La presente investigación se limita a la evaluación experimental del desempeño de los modelos EfficientNetB0, ResNet50 y YOLOv8 en la detección automatizada de violencia en secuencias de video. En consecuencia, los resultados obtenidos corresponden únicamente a las condiciones experimentales definidas. Además, el estudio se restringe a los tres modelos seleccionados, por lo que no se incluyen comparaciones con otras arquitecturas de *Deep Learning* o enfoques alternativos.

Es importante aclarar que no se contempla el desarrollo ni la implementación de un sistema de detección de violencia en tiempo real. La eficiencia computacional se evalúa exclusivamente mediante métricas como la latencia de inferencia y la tasa de procesamiento en un entorno de hardware controlado.

La investigación utiliza únicamente los conjuntos de datos considerados en el estudio; por tanto, no se desarrolla un conjunto de datos propio ni se incorpora información adicional para el entrenamiento ni para la evaluación.

Finalmente, el preprocesamiento de los videos se limita a la extracción de fotogramas a una frecuencia de muestreo estandarizada. No se analizan métodos que exploten explícitamente la información temporal, como el *Optical Flow* u otras técnicas de modelado espaciotemporal.

3 ESTADO DEL ARTE

Para fundamentar el desarrollo de la presente investigación, se realizó una revisión minuciosa de la literatura científica y tecnológica reciente, priorizando artículos científicos y tesis de grado afines al uso de la visión por computadora y de las redes neuronales convolucionales para la detección de incidencias de violencia física o similares. Este análisis permite establecer un punto de partida, identificar las brechas tecnológicas actuales y justificar la innovación de la solución propuesta.

3.1 Inteligencia artificial para la detección de violencia

En la literatura “Literature Review of Deep-Learning-Based Detection of Violence in Video” desarrollada por (Negre, S. Alonso, González Briones, Prieto, & Rodríguez González, 2024) se presenta el análisis de violencia física por medio de videgrabaciones implementando 16 tipos de entradas de algoritmos de inteligencia artificial con la finalidad de capturar diferentes resultados. En relación con este objetivo, se pusieron en práctica 21 métodos de extracción de fotogramas clave a partir de 28 conjuntos de datos creados específicamente durante un periodo no especificado de años. De igual manera, esta literatura recaudó diferentes artículos orientados a la violencia física referentes a las videgrabaciones, los retos e importancia que implica la detección o análisis de violencia física mediante el uso de técnicas de inteligencia artificial y visión por computadora.

Dicha literatura funge como un precursor significativo del presente proyecto de tesis al brindar una diversa fuente de artículos e investigaciones orientados al mismo fin, una comparativa entre las diferentes técnicas utilizadas en los artículos e investigaciones, un panorama acerca de la violencia física en entornos abiertos y la importancia de la detección de este tipo de incidencias en videos capturados. Teniendo esto último como distintivo del presente proyecto de investigación.

El artículo titulado “Inteligencia Artificial Fiable para la Detección de Violencia en Video” desarrollada por (Rodríguez, 2025) puntualiza acerca de la fiabilidad que los algoritmos deben brindar al momento de realizar la detección de la violencia física, siempre respetando los marcos éticos y legales para la implementación de este tipo de sistemas. Retomando el punto de interés, en este artículo la detección de incidentes de violencia física

actualmente se realiza mediante diferentes algoritmos o tecnologías de inteligencia artificial, entre los que predomina el uso de redes neuronales convolucionales densas y redes de memoria a corto y largo plazo (LSTM, por sus siglas en inglés, *Long Short-Term Memory*). En este se investigaron, diseñaron, desarrollaron y validaron tres arquitecturas que utilizan arquitectura VGG-19 (del inglés, *Visual Geometry Group*) pre-entrenada, una red neuronal convolucional que destaca por su habilidad en la extracción de características visuales. Además, posee capas de LSTM y BiLSTM (por sus siglas en inglés, *Bidirectional Long Short-Term Memory*). Asimismo, para alinearse a los marcos éticos y legales, se implementó GradCAM (por sus siglas en inglés, *Gradient-weighted Class Activation Mapping*), una técnica de inteligencia artificial explicable.

Al igual que en el presente, se utilizan redes neuronales convolucionales por su alta fidelidad en la detección de imágenes, y se aborda la detección de violencia física. Sin embargo, presenta el mismo patrón de complejidad matemática que la mayoría de los artículos consultados, con el fin de lograr una alta exactitud teórica, dejando de lado su viabilidad para su implementación en tiempo real.

3.2 Desafíos computacionales y de detección

De la misma manera, se analizó el trabajo titulado "Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future directions", elaborada por (Alzubaidi, y otros, 2021). Este documento proporciona un mapeo detallado de la evolución de las redes neuronales convolucionales y define las barreras estructurales que limitan su implementación práctica. En el trabajo analizado se traza el desarrollo histórico desde arquitecturas tempranas hasta modelos sumamente profundos como ResNet, los cuales introdujeron conexiones residuales para resolver el problema de la desaparición del gradiente al entrenar redes de hasta 152 capas. Sin embargo, el estudio subraya una problemática crítica: los modelos de *Deep Learning* poseen requerimientos intensivos de memoria y cómputo debido a su enorme complejidad y a la vasta cantidad de parámetros que los componen. Esta demanda de recursos reduce drásticamente la viabilidad de desplegar estas arquitecturas en máquinas con poder computacional limitado o en sistemas embebidos. Para mitigar este problema, se señala que el procesamiento paralelo a través de unidades de procesamiento gráfico se ha vuelto indispensable, ya que son extremadamente efectivas para

ejecutar las multiplicaciones de matrices y convoluciones que exigen las redes densas. Asimismo, este trabajo destaca que la necesidad de compresión de modelos es actualmente un área de investigación para reducir la redundancia de parámetros sin afectar el rendimiento.

La revisión de (Alzubaidi, y otros, 2021) proporciona el marco teórico que justifica el enfoque técnico de esta tesis. Al documentar formalmente que la profundidad de redes como ResNet conlleva un costo computacional y de memoria inalcanzable para sistemas limitados, se fundamenta teóricamente la causa subyacente del fenómeno de pérdida de cuadros en la videovigilancia. Dado que los sistemas de circuito cerrado de televisión del mundo real rara vez cuentan con granjas de servidores GPU de alta gama dedicadas a una sola cámara, la implementación de modelos de clasificación densos resulta inviable. Legitimando la transición propuesta en esta investigación: adoptar arquitecturas ágiles orientadas a la detección espacial directa que, por diseño, eluden la complejidad paramétrica de los clasificadores densos tradicionales, garantizando un rendimiento fluido en tiempo real.

Por otra parte, en la bibliografía titulada "Violence Detection Using Computer Vision Approaches", desarrollada por (Raihan Talha, Bandapadya, & Monirujjaman Khan, 2022). Se expone el objetivo de los investigadores respecto a la construcción de un sistema de monitoreo en tiempo real capaz de detectar la velocidad de los movimientos y la proximidad entre personas en áreas públicas para prevenir actos delictivos.

Para lograrlo, los investigadores propusieron una arquitectura híbrida que extrae secuencias de 10 fotogramas consecutivos de un video. En la primera fase, cada fotograma es procesado por una red neuronal convolucional para extraer las características espaciales del entorno. Posteriormente, estos datos alimentan una capa de memoria a corto y largo plazo bidireccional (BiLSTM), la cual analiza los factores temporales comparando la información del fotograma actual tanto con los anteriores como con los siguientes. Al someter este modelo a experimentación, los investigadores reportaron exactitudes en la fase de entrenamiento del 94.9% para el conjunto de datos de hockey y del 92.9% para secuencias de películas. Sin embargo, al evaluar el sistema frente al conjunto de datos de multitudes, el rendimiento del modelo experimentó una degradación crítica, cayendo a una exactitud del 77.3%.

El análisis de esta bibliografía aporta una evidencia empírica invaluable para justificar el planteamiento central de la presente tesis. En primer lugar, la drástica caída de rendimiento

(del 94.9% al 77.3%) reportada por (Raihan Talha, Bandapadya, & Monirujjaman Khan, 2022) al cambiar de un entorno controlado a uno no estructurado, es un caso de estudio perfecto del fenómeno de desplazamiento de dominio. Esto demuestra que las arquitecturas recurrentes tienden a sobreajustarse a secuencias de movimiento muy específicas, perdiendo capacidad de generalización en el mundo real.

En segundo lugar, la metodología de requerir la captura y almacenamiento en búfer de 10 fotogramas consecutivos para que la red BiLSTM pueda operar, introduce una latencia sistémica que contradice el principio de reacción inmediata. Este antecedente respalda la decisión metodológica de esta investigación: prescindir por completo del costoso análisis de memoria temporal. Al optar por un detector espacial directo de una sola etapa, la presente tesis demuestra que es posible analizar cada fotograma de manera veloz e independiente. De este modo, no solo se evita el cuello de botella computacional que propicia la pérdida de cuadros, sino que se logra una mayor resiliencia ante el *Domain Shift* al no depender de secuencias de movimiento pre-memorizadas.

En otra instancia, se consultó el proyecto titulado "Fast Violence Detection in Video", desarrollado por (Deniz, Serrano, Bueno, & Kim, 2015). Este proyecto es particularmente relevante porque expone una crítica temprana y directa a los cuellos de botella computacionales en la visión por computadora aplicada a la videovigilancia. En su planteamiento, los investigadores señalaron que, si bien los marcos de trabajo tradicionales para el reconocimiento de acciones lograban tasas de exactitud alentadoras cercanas al 90%, el costo computacional derivado de la extracción de características espaciotemporales resultaba imposible para aplicaciones prácticas en el mundo real. Ante esta limitación, y fundamentándose en estudios psicológicos que sugieren que las características puramente cinemáticas son suficientes para discriminar acciones agresivas, los investigadores propusieron un enfoque orientado en agilizar drásticamente la detección explotando únicamente estos patrones de movimiento rápidos.

La aportación de este proyecto es de suma importancia para la presente tesis, ya que demuestra que la latencia y los altos costos computacionales no son un problema exclusivo del *Deep Learning* moderno, sino una barrera histórica y persistente en la ingeniería de visión

por computadora. Los investigadores de este proyecto identificaron hace años que un sistema de seguridad pierde su utilidad práctica si su lentitud de procesamiento lo vuelve inviable.

3.3 Comparativa entre modelos de detección

La investigación de (Bilous, Malko, Frohme, & Nechyporenko, 2024), titulada "Comparison of CNN-Based Architectures for Detection of Different Object Classes", aporta un análisis comparativo exhaustivo. El objetivo de este estudio consistió en evaluar el rendimiento de diversas redes neuronales para la detección en tiempo real de personas y objetos técnicos en situaciones de emergencia, como desastres naturales y operaciones militares.

Metodológicamente, los autores contrastaron el desempeño de modelos de detección de una sola etapa (las versiones de YOLOv4 a YOLOv8, SSD y EfficientDet) frente a clasificadores de múltiples etapas basados en propuestas de región (Faster R-CNN). Para garantizar la robustez del análisis, las arquitecturas fueron entrenadas y evaluadas utilizando conjuntos de datos altamente diversos y complejos, incluidos COCO2017 (del inglés, *Common Objects in Context*), imágenes marítimas y secuencias capturadas por vehículos aéreos no tripulados (VisDrone2019, del inglés, *Vision Meets Drone*). Los resultados experimentales revelaron que, si bien modelos como Faster R-CNN logran una alta exactitud, sufren un cuello de botella crítico en la velocidad de procesamiento, alcanzando apenas 10 fotogramas por segundo. Por el contrario, la arquitectura YOLOv8 superó a todos los modelos evaluados de forma integral, demostrando ser la solución más equilibrada al obtener la mayor exactitud media (mAP de 0.88), el mejor $FI-Score$ (0.88) y el procesamiento más ágil (48 fotogramas por segundo).

Esta investigación constituye una justificación arquitectónica directa para la presente tesis. El análisis presentado por (Bilous, Malko, Frohme, & Nechyporenko, 2024) demuestra empíricamente el colapso operativo que sufren los modelos de múltiples etapas al operar a escasos 10 fotogramas por segundo, lo que los inhabilita para tareas de monitoreo continuo. Este hallazgo fundamenta de manera irrefutable la decisión de este proyecto de descartar clasificadores densos o secuenciales en favor de YOLOv8 para la detección de violencia. Al evidenciar que YOLOv8 no solo es sustancialmente más rápido, sino que también mantiene niveles superiores de exactitud en entornos dinámicos y complejos (similares a los que

propician el *Domain Shift*), la investigación consultada respalda la hipótesis central de esta investigación: la adopción de detectores de una sola etapa es el enfoque técnico más viable para erradicar la pérdida de cuadros y garantizar la operatividad de los sistemas de videovigilancia en tiempo real estricto.

3.4 Detección de objetos

Se consultó la tesis doctoral titulada “Visión Artificial en la Gestión y Apoyo a la Seguridad de los Trabajadores” elaborada por (Massiris Fernández, 2021) en la Universidad Nacional del Sur en Bahía Blanca, Argentina. En esta se aborda la evaluación de factores de riesgo ergonómicos en el Equipo de Protección Personal (por sus siglas, EPP) y ofrece un método basado en visión por computadora para determinar la utilización de equipo de protección personal, enfocándose primero en la detección del trabajador y luego en la posterior detección de presencia o ausencia del EPP recomendado. En dicho trabajo se implementan redes neuronales convolucionales tales como; YOLOv2, OpenPose 18 y ResNet 18. Donde YOLOv2 se encarga de la detección de objetos en tiempo real, OpenPose 18 (por sus siglas, OP18) estima la postura corporal a partir de la detección de la ubicación de las articulaciones y cómo se conectan entre ellas, y ResNet 18 (por sus siglas, RN18) permite el entrenamiento de redes neuronales convolucionales de manera rápida para la experimentación en una detección precisa, veloz y simple (Soni, 2023) (Roggio, Trovato, Sortino, & Musumeci, 2024) (roboflow, 2025). Además, se usaron herramientas como cámaras RGB (por sus siglas en inglés, Red, Green, Blue), conjuntos de videos obtenidos de la plataforma de YouTube, el lenguaje de programación Python, y librerías como Keras, TensorFlow y OpenCV.

Finalmente, los resultados obtenidos involucraron la detección parcial de los trabajadores a través del uso de la red neuronal; a su vez, la detección del equipo de protección personal usando YOLOv2, OpenPose 18 y ResNet 18; y se obtuvo una precisión del 62% para el casco y el chaleco con la ayuda de YOLOv2, mientras que para la detección del trabajador se obtuvo una precisión del 99% usando OpenPose 18 y ResNet 18.

Para efectos de esta investigación, el trabajo de (Massiris Fernández, 2021) resulta un antecedente crucial, ya que valida la viabilidad de combinar arquitecturas de detección de una sola etapa (YOLO, por sus siglas en inglés, *You Only Look Once*) con clasificadores

densos (ResNet, por sus siglas en inglés, Residual Network) para el monitoreo y seguimiento de sujetos humanos en secuencias de video, sentando las bases operativas que aquí se extrapolan hacia la detección de agresiones físicas.

Se examino un documento similar nombrado “Implementación de un Sistema de Detección de Equipos de Protección Personal Mediante Visión Artificial para Trabajadores del Sector Industrial”, desarrollado por el bachiller (Agramonte, 2024) en la Universidad Católica de Santa María, en la Facultad de Ciencias e Ingenierías Físicas y Formales, Escuela Profesional de Ingeniería Electrónica en Perú.

En este documento se tuvieron como objetivos implementar y proponer un sistema de detección de equipos de protección personal mediante ciencia aplicada de visión por computadora y *Machine Learning* para supervisión de trabajadores del sector industrial, identificar métodos de visión por computadora, identificar y extraer algoritmos de procesamiento de imágenes, instituir algoritmos en lenguaje de programación para la detección de EPP, evaluar los algoritmos y técnicas de visión por computadora, y obtener un sistema que permita realizar la inspección de equipos de protección personal con una perspectiva a integrar más elementos al mismo sistema.

Al igual que el primer trabajo de grado presentado, se usó el lenguaje de programación Python y la librería de OpenCV, añadiendo las librerías de NumPy y Mediapipe. Sin embargo, un distintivo de esta tesis parte del uso del algoritmo LBPH (del inglés, *Local Binary Pattern Histograms*) dejando de lado las redes neuronales y arquitecturas de detección y clasificación.

Ambos trabajos se orientan a la detección del equipo de protección personal de los trabajadores del sector industrial.

Como resultados presento la implementación de un sistema de detección de equipos de protección personal mediante visión por computadora y *Machine Learning* que se presenta como una solución innovadora que tiene el potencial de reducir significativamente la incidencia de accidentes laborales al garantizar el cumplimiento del uso de equipos de protección personal. Donde la clase casco obtuvo una exactitud del 100%, la clase lentes una exactitud del 86% y la clase mascarilla una exactitud del 92%.

Este antecedente es de particular relevancia para el presente proyecto, ya que permite establecer un punto de contraste metodológico entre la extracción de características locales y los modelos de *Deep Learning* aquí empleados. Mientras que el enfoque de (Agramonte, 2024) demuestra una alta eficacia en la detección de objetos con estructuras geométricas definidas y estáticas, la presente investigación aborda el desafío de la detección de violencia física. Dicha tarea, al ser intrínsecamente dinámica y semánticamente compleja, justifica la transición hacia modelos de una y múltiples etapas, las cuales superan las limitaciones de los algoritmos clásicos al generalizar patrones de movimiento y acción en entornos no controlados.

3.5 Métodos alternativos para la detección de violencia

En el ámbito de la extracción y análisis de múltiples modalidades para la detección de violencia, destaca la investigación de (Gu, Wu, & Wang, 2020), titulada "Violent Video Detection Based on Semantic Correspondence". Este estudio identifica y aborda una problemática frecuentemente ignorada en la literatura: la falta de correspondencia semántica entre los datos visuales y auditivos dentro de un mismo video, como ocurre cuando un material audiovisual presenta una agresión física acompañada de música relajante de fondo. Los autores argumentan que la fusión a ciegas de estas características no aporta beneficios reales e incluso resulta potencialmente perjudicial para la exactitud del modelo.

En esta investigación se emplean redes neuronales profundas para la extracción independiente de tres modalidades: apariencia y movimiento (mediante redes Pseudo-3D y LSTM operando sobre secuencias de cuadros y *Optical Flow*) y audio (utilizando la arquitectura VGGish). La innovación central del sistema radica en fusionar estas características a nivel de subespacios compartidos, guiando a la red neuronal mediante información explícita de correspondencia semántica. Esto se logra integrando el aprendizaje multitarea y el aprendizaje de incrustación semántica. Los resultados obtenidos demostraron que este enfoque supera sustancialmente a los métodos de fusión tradicionales, optimizando la capacidad discriminativa del sistema en conjuntos de datos públicos como VSD2015 (del inglés, *Violent Scene Detection*) y colecciones autoconstruidas.

La relevancia de esta investigación para el actual trabajo de tesis radica en su demostración empírica sobre los riesgos operativos que conlleva la heterogeneidad de los

datos y la contradicción de información. Mientras que (Gu, Wu, & Wang, 2020) buscan mitigar la brecha de heterogeneidad entre modalidades distintas (audio y video) enseñándole a la red a identificar correspondencias, el presente proyecto aborda un desafío conceptualmente análogo, pero estrictamente visual: la extrema variabilidad de los fondos y escenarios, formalizada como desplazamiento de dominio.

En la misma línea de investigación enfocada en la seguridad automatizada, destaca el artículo científico "Campus Violence Detection Based on Artificial Intelligent Interpretation of Surveillance Video Sequences", desarrollado por (Ye, y otros, 2021). Este estudio se centra específicamente en el entorno escolar y aborda la detección de agresiones físicas mediante un enfoque bimodal, combinando el análisis de secuencias de video con el reconocimiento de emociones a través de señales acústicas.

En este se optó por extraer características espaciotemporales de los videos utilizando una red neuronal convolucional tridimensional (C3D, del inglés, Convolutional 3D). Este modelo procesa bloques de 16 fotogramas simultáneamente para capturar el movimiento. De manera paralela, se extrajeron coeficientes cepstrales de frecuencia Mel (MFCC, por sus siglas en inglés, Mel-Frequency Cepstral Coefficients) del audio para clasificar emociones asociadas al acoso. El principal desafío fue el conflicto de evidencias. Para resolver dicho desafío, la innovación central del estudio consistió en proponer un algoritmo de fusión mejorado basado en la teoría de la evidencia de Dempster-Shafer, logrando reconciliar ambas modalidades para alcanzar una exactitud final del 97%.

La aportación de este artículo al desarrollo de la presente tesis radica en la justificación técnica sobre la eficiencia y la simplificación arquitectónica. El trabajo de (Ye, y otros, 2021) evidencia el alto costo computacional que conlleva el análisis de video volumétrico (procesamiento múltiple de fotogramas) sumado a la necesidad de ejecutar complejos algoritmos matemáticos para resolver las contradicciones entre el audio y el video.

Este antecedente valida directamente la ruta metodológica adoptada en este proyecto. Al demostrar lo demandante que resulta un sistema bimodal y tridimensional, se justifica de manera contundente la decisión de esta tesis de utilizar arquitecturas de una sola etapa que operan fotograma a fotograma. Al prescindir del procesamiento de audio y de los costosos algoritmos de fusión, el presente trabajo logra erradicar el problema de la pérdida de cuadros,

demostrando que un enfoque puramente visual y simplificado, siendo una solución mucho más viable y ágil para entornos de videovigilancia en estricto tiempo real.

En la investigación de (Durães, Veloso, & Novais, 2023), titulada "Violence Detection in Audio: Evaluating the Effectiveness of Deep Learning Models and Data Augmentation", el estudio se enfoca en la detección de situaciones de violencia en el interior de vehículos utilizando exclusivamente señales de audio. En esta se argumenta que, si bien el análisis de video es predominante en la vigilancia, el audio presenta ventajas operativas significativas al demandar menor capacidad de procesamiento, ancho de banda y almacenamiento, además de ser resistente a problemas de iluminación y temperatura.

Para llevar a cabo la experimentación, se construyó y preprocesó un corpus propio que abarca diversos escenarios violentos y pacíficos. El audio fue transformado en representaciones visuales bidimensionales, permitiendo el procesamiento del sonido empleando arquitecturas de redes neuronales de visión. Derivado de la limitación en el volumen de datos originales, se implementaron técnicas de *Data Augmentation*, específicamente el *flipping* de los espectrogramas, lo que duplicó el tamaño del conjunto de entrenamiento y mejoró la generalización del modelo. Finalmente, se evaluaron diversas arquitecturas de *Deep Learning* (tales como ResNet50, MobileNet y la familia EfficientNet), demostrando que el modelo EfficientNetB1 alcanzó la mayor exactitud con un 95.06%.

Esta investigación resalta la constante preocupación de la comunidad científica por la eficiencia computacional y los costos de procesamiento. Mientras que (Durães, Veloso, & Novais, 2023) optaron por prescindir del video y utilizar únicamente audio convertido a Mel-espectrogramas para reducir la carga de hardware en espacios confinados, esta tesis aborda exactamente el mismo problema de optimización computacional, pero demostrando que el análisis visual sigue siendo viable mediante la correcta elección de una arquitectura.

El trabajo desarrollado por (Bakhshi, García Gómez, Gil Pita, & Chalup, 2023), titulado "Violence Detection in Real-Life Audio Signals Using Lightweight Deep Neural Networks". Este estudio aborda la detección de anomalías conductuales, centrándose en el análisis de señales acústicas provenientes de entornos reales. Los autores proponen esta aproximación como una alternativa robusta para sistemas de vigilancia donde los datos visuales no están disponibles, son de baja calidad, o presentan restricciones de privacidad.

En cuanto a la metodología, la investigación resuelve el desafío de procesar datos temporales convirtiendo las señales de audio en representaciones visuales bidimensionales conocidas como imágenes de Mel-espectrogramas. Estas imágenes bidimensionales permiten alimentar directamente arquitecturas de redes neuronales convolucionales pre-entrenadas. El diseño experimental dividió las evaluaciones en dos categorías: modelos profundos convencionales (tales como ResNet50 y ResNet152) y redes neuronales profundas ligeras (como EfficientNetv2s, MobileNetv2 y ShuffleNetv2). Durante el entrenamiento, se implementó el optimizador Adam y se aplicaron diversas transformaciones de *Data Augmentation* (recortes, rotaciones y reflejo de espejo en horizontal) para evitar el *Overfitting*. Las pruebas se ejecutaron sobre conjuntos de datos altamente heterogéneos, incluyendo *Real-Life Violence Situations Dataset* (por sus siglas en inglés, RLVS).

Los resultados exhibieron que los modelos de arquitectura ligera lograron clasificar el comportamiento violento con una exactitud que superó en aproximadamente un 8% al estado del arte previo en las mismas bases de datos. Se concluyó que estas arquitecturas optimizadas son capaces de competir, e incluso superar, a los modelos tradicionales más profundos, con la ventaja crítica de requerir una cantidad considerablemente menor de parámetros y recursos computacionales.

La relevancia de este trabajo para la presente tesis se basa en dos aportaciones principales. En primera instancia, legitima la utilización del conjunto de datos RLVS como un estándar fundamentado para el entrenamiento en tareas de detección de violencia, validando la decisión de utilizar esta misma fuente de datos como el dominio base de este proyecto para el entrenamiento de los modelos.

En segunda instancia, el trabajo de (Bakhshi, García Gómez, Gil Pita, & Chalup, 2023) aborda el problema de los altos requerimientos de procesamiento, estableciendo un antecedente directo para la evaluación de la eficiencia computacional que se persigue en este documento. Mientras que los autores demuestran que reducir la complejidad de la red es imperativo para implementaciones en dispositivos móviles o de borde, el presente proyecto extrapola esta filosofía hacia la optimización de los cuadros por segundo. Es decir, la demostración empírica de que modelos menos densos pueden mantener un alto rendimiento respalda directamente la transición propuesta en esta investigación: alejarse de los

clasificadores densos tradicionales para adoptar arquitecturas ágiles de una sola etapa con el fin de erradicar la pérdida de cuadros en la videovigilancia en tiempo real.

3.6 Algoritmos de visión por computadora

Se examinó el trabajo de grado de (Sánchez, 2021), titulado "Detección de Víctimas mediante Redes Neuronales e Imágenes Térmicas", desarrollado en la Universidad Politécnica de Madrid. El propósito de este proyecto consistió en implementar modelos de visión por computadora capaces de identificar personas en situaciones de catástrofe mediante el procesamiento de imágenes infrarrojas, con miras a integrar el sistema en un robot móvil de búsqueda y rescate. (Sánchez, 2021) estructuró un análisis comparativo entre tres arquitecturas de redes neuronales convolucionales: un modelo de múltiples etapas (Faster R-CNN) y dos detectores de una sola etapa (SSD y YOLOv3). Para evaluar la robustez de los modelos ante cambios en el contraste térmico, se generaron conjuntos de datos en distintas condiciones ambientales, categorizados en escenarios de día, de noche y combinados. Los resultados experimentales expusieron deficiencias operativas significativas en los modelos tradicionales: la arquitectura SSD (del inglés, *Single Shot Detector*) arrojó altos valores de pérdida derivados de severos errores de clasificación entre extremidades del cuerpo, mientras que Faster R-CNN, a pesar de exhibir una buena eficacia teórica, operó a tasas insuficientes (alrededor de 7 FPS) para el tiempo real. En contraste, YOLOv3 demostró una superioridad técnica integral, logrando una exactitud media superior al 85% y una velocidad de inferencia ágil (45 FPS), validando su integración exitosa en el hardware del robot.

La aportación de este trabajo representa un respaldo empírico innegable para la fundamentación metodológica de la presente tesis. En primer lugar, la decisión de (Sánchez, 2021) de someter a evaluación las redes bajo condiciones ambientales opuestas legitima el estándar actual de la disciplina: un sistema de seguridad no puede considerarse robusto si no es capaz de generalizar ante variaciones extremas del entorno visual. Esto valida directamente el enfoque de esta investigación de forzar el desplazamiento de dominio evaluando los modelos en conjuntos de datos ajenos a su entrenamiento.

En segundo lugar, y de manera fundamental, este antecedente proporciona la justificación definitiva para descartar arquitecturas densas en la detección de violencia. Al confirmarse que los modelos basados en propuestas de región generan cuellos de botella

computacionales inoperables, se demuestra que el uso de estas redes en la videovigilancia propiciaría un colapso del sistema por pérdida de cuadros. El presente proyecto adopta y evoluciona esta premisa al implementar la iteración más reciente de la familia de detectores de una sola etapa.

3.7 Algoritmos para la detección de violencia

Se examinó el artículo científico de acceso abierto titulado “Threshold Active Learning Approach for Physical Violence Detection on Images Obtained from Video (Frame-Level) Using Pre-Trained Deep Learning Neural Network Models” (Abundez, y otros, 2024). El objetivo central de dicho estudio radica en mejorar la exactitud, eficiencia y confiabilidad de los sistemas de videovigilancia mediante la evaluación de diversos modelos pre-entrenados para la identificación de comportamientos violentos. La metodología que emplean los autores se basó en el análisis a nivel de fotograma (*frame-level*) utilizando el lenguaje de programación, Python y bibliotecas de ciencia de datos como TensorFlow, Scikit-learn y Pandas, aplicando arquitecturas convolucionales como DenseNet121, MobileNetV2 y EfficientNetB0. Los resultados experimentales obtenidos por los investigadores demostraron la efectividad de un enfoque de aprendizaje activo basado en umbrales (*Threshold Active Learning*), logrando clasificar escenas de violencia en entornos visuales desconocidos. Su principal contribución fue la formulación de un umbral métrico (μ) a partir de la salida de la red neuronal, permitiendo la intervención de expertos humanos para retroalimentar y robustecer el modelo.

Este artículo científico constituye un pilar del presente trabajo de grado por dos razones principales. En un primer momento, legitima el uso de arquitecturas como EfficientNetB0 para la extracción de características de violencia a nivel de fotograma. En un segundo momento, aborda directamente el reto de la clasificación ante entornos desconocidos, validando la necesidad de investigar el fenómeno de desplazamiento de dominio (del inglés, *Domain Shift*). Sin embargo, mientras que (Abundez, y otros, 2024) proponen la intervención humana continua para mitigar este problema, la presente tesis aporta un enfoque orientado a la autonomía total del sistema, demostrando que la transición hacia arquitecturas de una sola etapa puede ofrecer una resiliencia superior ante estos cambios de dominio de manera automatizada y en tiempo real.

Ahora bien, se evaluó la literatura recientemente publicada que lleva el nombre de "Strategic Sample Selection in Deep Learning: A Case Study on Violence Detection Using Confidence-Based Subsets", desarrollada por diferentes personalidades de distintas instituciones educativas (Primero Primero, y otros, 2025). La finalidad principal de este estudio consistió en abordar las ineficiencias computacionales y la heterogeneidad de los datos en los sistemas de detección de violencia física, proponiendo un enfoque metodológico para la identificación automática de muestras relevantes. Los autores argumentan que los métodos tradicionales tratan todas las muestras de entrenamiento de manera homogénea, ignorando las diferencias en la complejidad de la clasificación, lo que genera un alto costo computacional, especialmente en entornos con recursos limitados. Para contrarrestar esta problemática, la investigación estructuró un fuerte protocolo experimental utilizando bases de datos públicas enfocadas en escenarios violentos y pacíficos. La arquitectura base seleccionada fue DenseNet121, elegida por su capacidad para extraer patrones visuales sutiles mediante conexiones de intercapas densas. La innovación primordial de la metodología radicó en segmentar el conjunto de datos de entrenamiento en tres subconjuntos especializados, basados en los niveles de confianza predictiva de la red neuronal: Subconjunto seguro (Safe): compuesto por imágenes con alta certeza predictiva (probabilidades extremas cercanas a 0.1 o 0.9), representando escenarios visuales sin ambigüedad. Subconjunto fronterizo (Border): agrupó las muestras con la máxima incertidumbre, ubicadas simétricamente alrededor del límite de decisión de la red (probabilidad cercana a 0.5). Subconjunto promedio (*Average*): integró casos de certeza moderada (probabilidades entre 0.4 y 0.6). Además de DenseNet121, los investigadores extendieron su evaluación a arquitecturas como MobileNetV2 y *Vision Transformer* (ViT, por sus siglas en inglés) para comprobar la generalización de su estrategia de selección de muestras. Los resultados experimentales demostraron que el subconjunto Average ofreció el mejor equilibrio técnico, logrando un *F1-Score* de 0.89 y una media geométrica (g-mean, por sus siglas en inglés) de 0.93 al utilizar únicamente el 20% del volumen total de datos. Asimismo, se reportó una reducción significativa en los tiempos de entrenamiento computacional, confirmando que la selección estratégica de muestras de confianza moderada permite un aprendizaje profundo altamente efectivo sin requerir bases de datos masivas.

La aportación de esta literatura es de suma trascendencia para el desarrollo de la presente tesis, pues fundamenta científicamente la necesidad de optimizar los recursos computacionales en tareas de clasificación visual compleja. Mientras que (Primero Primero, y otros, 2025) abordan la optimización desde la perspectiva de los datos (reduciendo la cantidad de imágenes mediante umbrales de confianza antes del entrenamiento), el presente proyecto de investigación aborda el mismo problema desde la perspectiva arquitectónica. Al contrastar clasificadores densos de múltiples etapas con detectores de una sola etapa, esta tesis busca mitigar el fenómeno de pérdida de cuadros (*Frame Dropping*) directamente en la latencia de inferencia en tiempo real. Adicionalmente, la diversidad de conjuntos de datos empleados en dicha literatura valida la metodología aquí empleada para forzar el desplazamiento de dominio, lo que demuestra que la heterogeneidad de los escenarios visuales constituye el estándar metodológico actual para evaluar la verdadera robustez de un sistema de videovigilancia.

En el trabajo de investigación nombrado "Machine Learning Software for the Detection of Violence from CCTV Live Footage", presentado por (Nithin Sai & Siva Prasad, 2023). La meta de este trabajo consistió en el desarrollo de un software analítico capaz de procesar flujos de video para identificar situaciones de violencia de manera instantánea y generar notificaciones automatizadas a las agencias de seguridad y personal médico. La propuesta de los autores destaca por la flexibilidad en la captura de datos, permitiendo el análisis simultáneo de transmisiones en vivo desde enlaces externos, videos pregrabados y señales de cámaras de seguridad en tiempo real.

El sistema se fundamentó en la extracción de características espaciotemporales mediante el uso de redes neuronales convolucionales tridimensionales. Esta arquitectura emplea tamaños de *kernel* temporal más grandes, lo que permite al modelo capturar la dinámica del movimiento a lo largo del tiempo cruzando múltiples fotogramas. Gracias a esto, el sistema logra discriminar eficazmente entre patrones de movimiento humano pacíficos y acciones intrínsecamente agresivas.

Mientras que (Nithin Sai & Siva Prasad, 2023) confían en el uso de convoluciones tridimensionales para capturar el contexto temporal completo de las agresiones, la presente

investigación cuestiona la escalabilidad y la viabilidad operativa de este enfoque en entornos de producción con restricciones computacionales.

La literatura de (Bermejo Nievas, Deniz Suarez, Bueno García , & Sukthankar, 2011), titulada "Violence Detection in Video Using Computer Vision Techniques". Este estudio representa uno de los pilares en la detección de violencia utilizando visión por computadora clásica, previo a la adopción masiva del *Deep Learning*. En la literatura se señala que, aunque la comunidad científica se había centrado en detectar acciones simples, el reconocimiento de comportamientos agresivos había sido comparativamente menos estudiado.

La literatura se fundamentó en el marco de trabajo de bolsa de palabras visuales (Bag of Words), una técnica tradicional que requiere la extracción manual de características. Para ello, evaluaron dos de los descriptores de acción más avanzados de su época: los Puntos de Interés Espaciotemporales (STIP, del inglés, *Spatio-Temporal Interest Points*) y MoSIFT (por sus siglas en inglés, *Motion Scale-Invariant Feature Transform*). Adicionalmente, ante la carencia de conjuntos de datos estandarizados, los autores introdujeron una base de datos propia compuesta por 1000 secuencias de video, divididas equitativamente entre escenarios de peleas y situaciones pacíficas. Sus experimentos, que incluyeron secuencias de películas de acción, demostraron que era posible detectar peleas con una exactitud cercana al 90% utilizando estos descriptores clásicos.

La inclusión de esta investigación aporta una perspectiva histórica a la presente tesis, ya que ilustra el paradigma clásico de la visión por computadora y establece el punto de partida desde el cual la tecnología ha evolucionado. El enfoque de (Bermejo Nievas, Deniz Suarez, Bueno García , & Sukthankar, 2011) dependía de la formulación algorítmica manual para extraer el movimiento y la forma. Esta extracción manual de características es altamente susceptible a fallar cuando cambian las condiciones de iluminación, el ángulo de la cámara o el entorno visual, haciéndola vulnerable ante el fenómeno de desplazamiento de dominio.

Este antecedente justifica la decisión de esta investigación de omitir los algoritmos clásicos y adoptar directamente arquitecturas de *Deep Learning*. Al emplear modelos basados en redes neuronales convolucionales, se delega la extracción de características a la propia red, permitiendo que el sistema descubra automáticamente los patrones visuales subyacentes de la violencia. De este modo, se demuestra que la transición hacia detectores de una sola

etapa no solo busca resolver el problema moderno de la latencia y la pérdida de cuadros, sino que también representa la superación de las limitaciones de generalización inherentes a los modelos matemáticos tradicionales de la década pasada.

Se analizó el artículo denominado "AI-based Violent Incident Detection in Surveillance Videos to Enhance Public Safety", desarrollado por (Merit & Beladgham, 2025). El objetivo de este estudio se enfoca en la automatización del monitoreo de actividades humanas para garantizar la seguridad pública, proponiendo un marco de trabajo capaz de detectar violencia y emitir alertas tempranas en áreas densamente pobladas.

En la investigación se propone un enfoque híbrido secuencial. En una primera etapa, se emplearon modelos pre-entrenados de transferencia de aprendizaje (*Transfer Learning*) basados en la familia ResNet (ResNet50, ResNet101 y ResNet152) para realizar la extracción de características espaciales de cada secuencia de fotogramas del video. En una segunda etapa, estas características fueron procesadas y clasificadas mediante un algoritmo de unidad recurrente cerrada bidireccional (BiGRU, del inglés, *Bidirectional Gated Recurrent Unit*), diseñado para capturar la dependencia temporal de las acciones. La experimentación se llevó a cabo utilizando tres conjuntos de datos públicos correspondientes a películas y multitudes. Los resultados experimentales indicaron que la combinación de ResNet152 junto con el modelo BiGRU ofreció el mejor rendimiento, superando a enfoques tradicionales y a otros modelos de clasificación temporal.

El impacto de tal artículo en la presente tesis radica en la capacidad de la arquitectura ResNet como extractor de características espaciales altamente competente en el dominio de la violencia visual. Sin embargo, el enfoque de (Merit & Beladgham, 2025) ejemplifica la tendencia clásica de la literatura por incrementar drásticamente la profundidad computacional para maximizar la exactitud teórica. En esta investigación se aborda el problema desde la perspectiva diametralmente opuesta: la viabilidad operativa. Añadir redes recurrentes a modelos convolucionales densos agrava severamente la latencia de inferencia, propiciando la pérdida de *frames* al procesar transmisiones en vivo. En este sentido, este antecedente justifica de manera contundente la decisión de esta tesis de contrastar a ResNet50 con una arquitectura de detección de una sola etapa. Se busca demostrar empíricamente que prescindir del costoso análisis temporal complejo es una alternativa superior si se logra una

velocidad de reacción extremadamente veloz, garantizando una detección de amenazas en estricto tiempo real para la videovigilancia.

De otro modo, se consultó el trabajo de investigación denominado "Deep Learning-Enhanced Video Surveillance System for Detecting Violent Behavior Using Anomaly Detection Techniques", elaborado por (Rajini, Deekshitha, Kashish, Yoganarasimhan, & Sinchana, 2026) . Aborda la implementación de técnicas de detección de anomalías mediante *Machine Learning* para identificar comportamientos violentos en entornos de videovigilancia. El análisis de acciones en tiempo real ha demostrado ser eficaz para elevar la calidad de los sistemas de seguridad, superando las limitaciones de los métodos tradicionales en escenas con multitudes o condiciones adversas como la oscuridad y las oclusiones visuales.

En el trabajo de investigación se evalúa el impacto de la extracción de características espaciotemporales mediante una amplia variedad de redes neuronales, incluyendo arquitecturas convolucionales, redes recurrentes, redes híbridas, redes neuronales tridimensionales y Transformers. Pese a que los autores reportan que estas metodologías logran rendimientos teóricos superiores al 90%, exponen una problemática crítica para su implementación en el mundo real: los problemas de escalabilidad, la necesidad de diversidad en los conjuntos de datos y los elevados costos computacionales requeridos para la ejecución de las operaciones. Para mitigar estas barreras, el trabajo de investigación subraya la necesidad de transitar hacia arquitecturas alternativas ligeras que permitan el despliegue de estas aplicaciones directamente en dispositivos de borde en entornos que carecen de una infraestructura de servidores robusta.

(Rajini, Deekshitha, Kashish, Yoganarasimhan, & Sinchana, 2026) exponen explícitamente que la diversidad de los conjuntos de datos sigue siendo una preocupación vigente, lo cual respalda la iniciativa de someter a los modelos de la presente a un análisis de desplazamiento de dominio utilizando conjuntos de datos ajenos al entrenamiento original. Por otro lado, el impacto más significativo de esta investigación es su diagnóstico sobre los cuellos de botella del hardware. Al confirmar que los modelos espaciotemporales densos enfrentan severos problemas de escalabilidad y costos operacionales.

El estudio de (Salman, Abbas, ur Rahman, Al Alshaikh, & Saudagar, 2026), titulado "An Embedded Deep Learning Framework for Real-Time Violence Detection and Alert Generation". Este estudio aborda el desafío de implementar sistemas automatizados de detección de violencia directamente en dispositivos de borde con recursos limitados, buscando optimizar la conciencia situacional y reducir la dependencia del monitoreo humano continuo.

La metodología propuesta en dicho trabajo consiste en un modelo híbrido espaciotemporal que integra una red neuronal convolucional ligera para la extracción de características espaciales, acoplada a unidades recurrentes cerradas para el modelado de las secuencias temporales. Para lograr la viabilidad de despliegue en hardware restringido como una Raspberry Pi 5, la red fue sometida a rigurosas estrategias de optimización de nivel de hardware y software, incluyendo cuantización, poda estructural (*Structure Pruning*) y aceleración mediante TensorRT (del inglés, *Tensor RunTime*). Los resultados de esta implementación embebida demostraron que es posible alcanzar una alta velocidad de inferencia y un bajo consumo energético sin comprometer la precisión del sistema.

La relevancia de este trabajo para la presente investigación radica en su confirmación de la pesada carga computacional que exigen los modelos espaciotemporales tradicionales. El trabajo de (Salman, Abbas, ur Rahman, Al Alshaikh, & Saudagar, 2026) demuestra que, para procesar secuencias de video en tiempo real mediante arquitecturas que combinan redes neuronales convolucionales y recurrentes, es obligatorio someter al modelo a procesos complejos de compresión matemática y poda estructural.

Este hallazgo justifica la aportación de la metodología de la actual tesis. En lugar de aplicar técnicas de reducción a posteriori a arquitecturas pesadas o dependientes del tiempo, la presente propone solucionar el problema de la latencia desde la raíz arquitectónica, adoptando un detector espacial de una sola etapa.

3.8 Síntesis de las brechas identificadas

A partir de la literatura consultada, resumida en la *Tabla 1*, es innegable la existencia de una tendencia a la priorización de la exactitud teórica mediante la implementación de arquitecturas densas, espaciotemporales o híbridas sumamente profundas e incluso, tales

como redes tridimensionales, recurrentes y fisiones multimodales. Sin embargo, la evidencia recolectada muestra que esta creciente complejidad estructural conlleva dos debilidades sistémicas que inhabilitan su despliegue en entornos reales de videovigilancia. En primera instancia, la alta carga paramétrica y la dependencia de secuencias temporales en el búfer generan cuellos de botella computacionales que provocan una pérdida inevitable de cuadros (*Frame Dropping*). Y, en segunda instancia, se constata una vulnerabilidad crítica ante la heterogeneidad visual, provocando que modelos matemáticamente precisos en laboratorio colapsen operativamente al enfrentarse a entornos no controlados (desplazamiento de dominio).

Frente a las limitaciones documentadas, resulta necesaria la transición hacia enfoques de ingeniería más ágiles. Los antecedentes confirman que prescindir de la costosa extracción temporal en favor de detectores espaciales directos de una sola etapa resuelve la latencia de inferencia en su raíz arquitectónica. Al delegar el reconocimiento de amenazas a la discriminación fotograma a fotograma, se logra un equilibrio en el que la agilidad de procesamiento erradica los tiempos muertos, garantizando la viabilidad operativa necesaria para transformar el monitoreo pasivo en una herramienta eficaz de prevención y alerta.

Tabla 1. Matriz de referencias para determinar el estado del arte.

Fuente	Resumen	Metodología	Áreas de oportunidad
“Visión artificial en la gestión y apoyo a la seguridad de los trabajadores” (Massiris Fernández, 2021)	El autor nos plantea la importancia de la detección del correcto uso del EPP por parte de los trabajadores con la finalidad de evitar incidencias que puedan perjudicar la integridad y bienestar de estos. Con la ayuda de tres arquitecturas de CNN logró un 62% de precisión en la detección del EPP y un 99% en la detección de la presencia del trabajador.	La detección se dividió en la presencia del trabajador y del equipo de protección personal. De esta manera, se utilizó YOLOv2 para la detección del EPP, OP18 y RN18 para la detección del trabajador. La primera es una arquitectura orientada a la clasificación de objetos en tiempo real. Por otro lado, OP18 está orientada a la detección de puntos clave de la anatomía humana para	Entre sus principales ventajas se encuentran: la detección del EPP en tiempo real para prevenir incidentes laborales, la clasificación individual de los elementos del EPP para determinar el cumplimiento completo de este, y no se requiere de supervisión humana. Ahora con respecto a sus desventajas: la iluminación, la lejanía o ángulo desde el que se realiza la detección pueden afectar a la

		determinar su posición y RN18 se orientó a la rapidez y sencillez de la detección.	precisión. Además de la posibilidad de falsos positivos frente a prendas similares al EPP.
"Threshold Active Learning Approach for Physical Violence Detection on Images Obtained from Video (Frame-Level) Using Pre-Trained Deep Learning Neural Network Models" (Abundez, y otros, 2024)	El artículo consta de un estudio que radica en mejorar la precisión, eficiencia y confiabilidad de los sistemas de videovigilancia mediante la evaluación de diversos modelos pre-entrenados para la identificación de comportamientos agresivos.	Se realizó un análisis de videos a nivel de fotograma usando Python y redes neuronales como DenseNet121, MobileNetV2 y EfficientNetB0.	Una posible aplicación de este trabajo consistiría en su implementación en sistemas de videovigilancia, la cual es mencionada en el artículo. Principalmente en entornos reducidos. Si el sistema se vuelve robusto, podría incluso implementarse en entornos masivos.
"Strategic Sample Selection in Deep Learning: A Case Study on Violence Detection Using Confidence-Based Subsets" (Primero Primero, y otros, 2025)	El estudio consistió en abordar las ineficiencias computacionales y la heterogeneidad de los datos en los sistemas de detección de violencia física, proponiendo un enfoque metodológico para la identificación automática de muestras relevantes.	Se usaron conjuntos de datos de origen público que contuvieran escenas violentas y pacíficas. Estos se segmentaron bajo tres subconjuntos. Y con estos se entrenaron arquitecturas como DenseNet121, MobileNetV2 y ViT.	A través de dicho artículo, se da a conocer de una alternativa viable para el ahorro de recursos computacionales en investigaciones que impliquen el entrenamiento de arquitecturas por medio de conjuntos de datos visuales. Por lo tanto, su aplicación radica en investigaciones posteriores.
"AI-based Violent Incident Detection in Surveillance Videos to Enhance Public Safety" (Merit & Beladgham, 2025)	El estudio se enfoca en la automatización del monitoreo de actividades humanas para garantizar la seguridad pública, proponiendo un marco de trabajo capaz de detectar violencia y emitir alertas tempranas en áreas densamente pobladas.	Se usaron modelos pre-entrenados de <i>Transfer Learning</i> ResNet50, ResNet101 y ResNet152 para la extracción de características de los <i>frames</i> . Mismas que fueron procesadas y clasificadas posteriormente a través del uso de un algoritmo BiGRU.	Si bien, al experimentar con diferentes modelos pre-entrenados de <i>Transfer Learning</i> y combinarlos con un algoritmo BiGRU se obtuvieron resultados destacables. Es importante resaltar la complejidad matemática al combinar dos algoritmos, de esta manera incrementa la precisión en la detección, pero decrementa la latencia, lo que resulta en

			la pérdida de cuadros y convierte a este algoritmo híbrido en uno inviable para procesar datos en tiempo real.
"An embedded Deep Learning framework for real-time violence detection and alert generation" (Salman, Abbas, ur Rahman, Al Alshaikh, & Saudagar, 2026)	En este estudio se aborda el desafío de implementar sistemas automatizados de detección de violencia directamente en dispositivos de borde con recursos limitados, buscando optimizar la conciencia situacional y reducir la dependencia del monitoreo humano continuo.	Consiste en un modelo híbrido espaciotemporal que integra una red neuronal convolucional ligera para la extracción de características espaciales, acoplada a unidades recurrentes cerradas para el modelado de las secuencias temporales.	Los resultados de esta implementación embebida demostraron que es posible alcanzar una alta velocidad de inferencia y un bajo consumo energético sin comprometer la precisión del sistema. A costa de un mayor recurso computacional y una latencia inoperable en situaciones de tiempo real.
"Deep Learning-Enhanced Video Surveillance System for Detecting Violent Behavior Using Anomaly Detection Techniques" (Rajini, Deekshitha, Kashish, Yoganarasimhan, & Sinchana, 2026)	Aborda la implementación de técnicas de detección de anomalías mediante <i>Machine Learning</i> para identificar comportamientos violentos en entornos de videovigilancia.	Se usaron redes neuronales, incluyendo arquitecturas convolucionales, recurrentes, redes híbridas, redes neuronales tridimensionales y Transformers.	Pese a haber logrado resultados teóricos superiores al 90%. Su principal desventaja radica en la falta de escalabilidad, la necesidad de diversidad en los conjuntos de datos y los altos costos computacionales requeridos para ejecutar las operaciones.
"Machine Learning Software for the Detection of Violence from CCTV Live Footage" (Nithin Sai & Siva Prasad, 2023)	Consistió en el desarrollo de un software analítico capaz de procesar flujos de video para identificar situaciones de violencia de manera instantánea y generar notificaciones automatizadas a las agencias de seguridad y personal médico.	El sistema se basa en la extracción de características espaciotemporales mediante redes neuronales convolucionales tridimensionales. Capturando la dinámica del movimiento a lo largo del tiempo mediante múltiples fotogramas.	El sistema discrimina eficazmente entre patrones de movimiento humano pacíficos y acciones intrínsecamente agresivas. Sin embargo, presenta problemas de escalabilidad y viabilidad operativa en entornos de producción con restricciones computacionales.
"Comparison of CNN-Based Architectures for	Consistió en evaluar el rendimiento de diversas redes neuronales para la	Se compararon modelos de detección de una sola etapa (YOLOv4 a	Un área de oportunidad para este estudio radica en el punto de partida

Detection of Different Object Classes" (Bilous, Malko, Frohme, & Nechyporenko, 2024)	detección en tiempo real de personas y objetos técnicos en situaciones de emergencia, como desastres naturales y operaciones militares.	YOLOv8, SSD y EfficientDet) frente a clasificadores de múltiples etapas basados en propuestas de región (Faster R-CNN). Las arquitecturas fueron entrenadas y evaluadas utilizando conjuntos de datos altamente diversos y complejos, incluyendo COCO2017, imágenes marítimas y secuencias capturadas por vehículos aéreos no tripulados.	para futuras investigaciones, pues al comparar diferentes arquitecturas haciendo uso de una gran diversidad de conjuntos de datos será más fácil para otros investigadores definir el tipo de arquitectura que utilizaran para sus investigaciones según sus requerimientos. Además, YOLOv8 demostró los mejores resultados en relación eficacia-eficiencia, dando soporte al modelo de una etapa electo para esta tesis.
"Violence Detection Using Computer Vision Approaches" (Raihan Talha, Bandapadya, & Monirujjaman Khan, 2022)	Se construyó un sistema de monitoreo en tiempo real capaz de detectar la velocidad de los movimientos y la proximidad entre personas en áreas públicas para prevenir actos delictivos.	Se propuso una arquitectura híbrida que extrae secuencias de 10 fotogramas consecutivos de un video. Después, cada fotograma fue procesado para extraer las características espaciales del entorno. Posteriormente, estos datos alimentaron una capa BiLSTM, misma que analizó los factores temporales comparando la información del fotograma actual tanto con los anteriores como con los siguientes.	Se obtuvieron precisiones que superaron el 90% en entornos controlados que no presentaron variaciones. Sin embargo, al momento de evaluar entornos diversos y fluctuantes, la precisión cayó drásticamente por debajo de un 80%. Por lo tanto, el sistema se vuelve inoperante en entornos no controlados y, por ende, no es recomendable para sistemas de videovigilancia en tiempo real.
"Detección de víctimas mediante redes neuronales e imágenes térmicas" (Sánchez, 2021)	Este proyecto consistió en implementar modelos de visión por computadora capaces de identificar personas en situaciones de catástrofe mediante el procesamiento de imágenes infrarrojas, con miras a integrar el	Se compararon tres arquitecturas de redes neuronales convolucionales: un modelo de múltiples etapas (Faster R-CNN) y dos detectores de una sola etapa (SSD y YOLOv3). Cada una se evaluó a partir de un	De acuerdo con los resultados presentados, SSD arrojó altos valores de pérdida derivados de severos errores de clasificación entre extremidades del cuerpo, mientras que Faster R-CNN, a pesar de exhibir una buena eficacia

sistema en un robot móvil de búsqueda y rescate.	conjunto de datos frente a distintas condiciones ambientales.	teórica, operó a tasas insuficientes para el tiempo real (7 FPS aproximadamente). En contraste, YOLOv3 demostró una superioridad técnica integral, logrando una precisión media superior al 85% y una velocidad de inferencia ágil (45 FPS), validando su integración exitosa en el hardware del robot.
--	---	---

4 MARCO CONCEPTUAL

En esta sección se presentan los conceptos técnicos más relevantes para comprender el contenido del trabajo de investigación. Se han organizado en tres ejes temáticos fundamentales, avanzando desde definiciones más amplias hasta las más específicas, con el fin de establecer una base teórica para la detección de violencia en secuencias de imágenes mediante modelos de inteligencia artificial.

4.1 Secuencias de imágenes y video

Este apartado conceptualiza la naturaleza de los datos de entrada y el entorno físico en el que se generan. La progresión va desde el macrosistema de captura hasta los componentes individuales del video y, finalmente, las limitaciones técnicas y artefactos que pueden comprometer la calidad de la información para el análisis algorítmico.

Sistema de videovigilancia: es el conjunto de dispositivos de grabación (principalmente cámaras) que se encuentran conectados entre sí, almacenan las grabaciones y pueden visualizarse a través de monitores.

Secuencia de imágenes (video): conjunto ordenado y temporal de fotogramas que, al reproducirse a una velocidad determinada, genera la ilusión de movimiento. En el contexto de la detección de violencia, el análisis puede ser espacial (lo que hay en un frame) y temporal (cómo cambian las interacciones entre personas u objetos a lo largo del tiempo).

Fotograma (frame): es una imagen estática e individual que forma parte de un conjunto de imágenes en un video.

Debajo se exhibe una ejemplificación visual en la *Figura 1* donde se demuestra una secuencia de un total de seis *frames* (imágenes) que componen un video en el que se anima de forma simple el rebote de una pelota.

Resolución espacial y temporal: la resolución espacial se refiere a la cantidad de píxeles en cada fotograma (por ejemplo, 1920x1080 px), que es útil para detectar detalles pequeños. La resolución temporal se refiere a la frecuencia de muestreo (FPS, *Frames Per Second*), determinante para capturar movimientos rápidos propios de agresiones físicas sin perder información crítica.

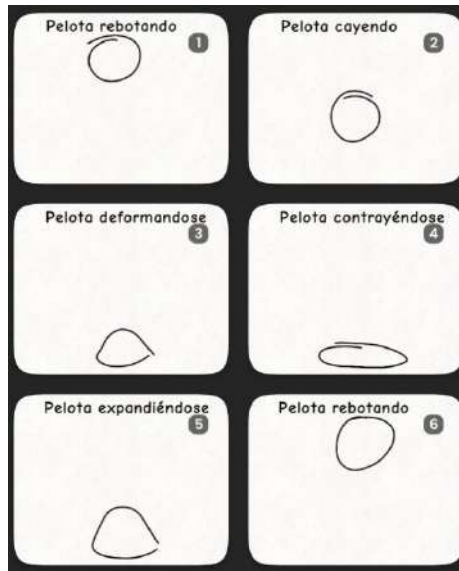


Figura 1. Visor de frames de un video de 6 FPS.

Cuadros por segundo: es una métrica que cuantifica el número de imágenes proyectadas por segundo. Es un término que suele utilizarse en animaciones, videojuegos, televisión y, en general, en secuencias de imágenes. Una mayor cantidad de fotogramas por segundo implica una mayor fluidez del video. Se suele usar en su idioma original, abreviado como FPS (*Frames Per Second*).

A modo de ejemplo, la *Figura 2* muestra una comparación entre diferentes tasas de fotogramas por segundo de la misma animación. A 30 FPS la animación de una “pelota” siguiendo una trayectoria de bajada, ha culminado. Mientras tanto, a 14 FPS se encuentra casi a mitad de trayecto. Por último, a 4 FPS se aprecia únicamente el inicio de la animación.

En el caso que se presenta en este trabajo de tesis, es indispensable contar con un *frame rate* (velocidad de fotogramas) de al menos 30 FPS. Este es el estándar para una fluidez considerable, pues una menor cantidad que esta puede tener como consecuencia el retraso o la descomposición de imágenes. Lo que en cámaras de videovigilancia en tiempo real se traduciría en la pérdida de *frames*, y de esta forma podría comprometer la adecuada detección de incidencias de violencia física.

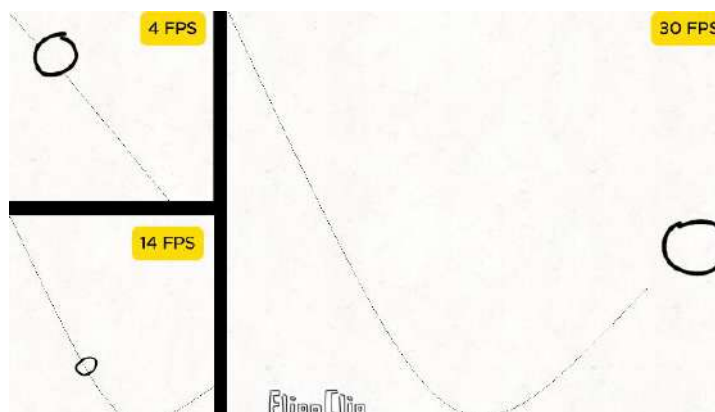


Figura 2. Comparación entre diferentes tasas de fotogramas por segundo.

Codificación y Compresión de Video (Codecs): algoritmos (como H.264 o H.265) que reducen el tamaño de los archivos de video eliminando redundancias. Una compresión excesiva puede introducir artefactos visuales (pixelación, imagen borrosa), mientras que una compresión ligera exige mayor ancho de banda y capacidad de almacenamiento.

Búfer: es un espacio de almacenamiento temporal en la memoria de uno de los dispositivos mientras se transmiten datos entre ambos. Esto tiene como finalidad permitir la fluidez de la transferencia y evitar posibles interrupciones, haciéndolo funcional a pesar de las diferencias de velocidad entre dispositivos (Cloudflare, 2026).

Latencia: retardo de tiempo entre la captura de un evento en el mundo real y su procesamiento o visualización en el sistema. En la detección de violencia, una latencia alta puede hacer que la alerta llegue demasiado tarde para prevenir o mitigar el daño.

Cuello de botella computacional: es un conflicto que se presenta cuando existe un desequilibrio en la capacidad y la velocidad de procesamiento entre los diferentes componentes que integran un sistema o una computadora. Limitando el rendimiento de la ejecución de tareas en función de las características del componente más desequilibrado (PC Componentes, 2022).

Perdida de cuadros (Frame Dropping): es la interrupción de la fluidez de video causada por el retraso de un *frame* provocando la repetición del *frame* perdido y retardando la aparición del siguiente *frame*. La Figura 3 muestra el flujo normal y deseado de una secuencia de video sin pérdida de cuadros, mostrando cada fotograma que la compone. Por

otra parte, la *Figura 4* representa la pérdida de cuadros en video, donde se repite al menos uno de los fotogramas, alterando el flujo normal del video. En sistemas de videovigilancia en tiempo real, el *Frame Dropping* causaría la pérdida de imágenes que podrían ser críticas para la detección de incidencias de violencia física.

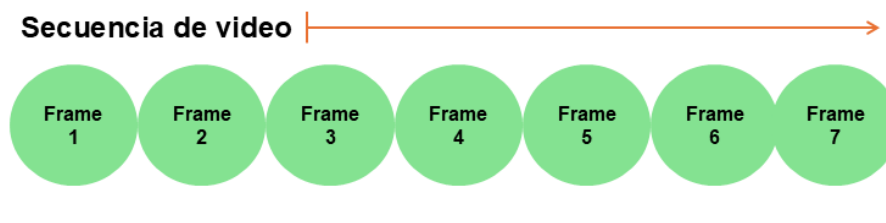


Figura 3. Secuencia de video sin Frame Dropping.

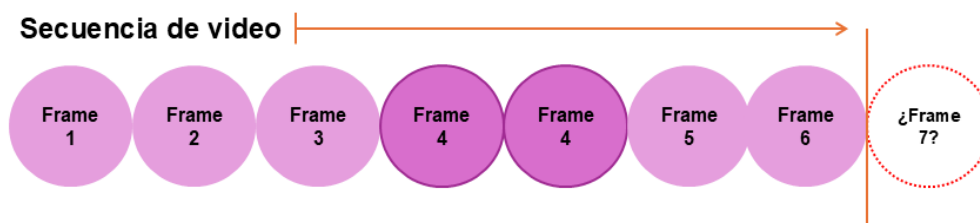


Figura 4. Secuencia de video con Frame Dropping.

4.2 Inteligencia artificial y visión por computadora

Este eje define el marco tecnológico de la solución. Comienza con las disciplinas madre (IA y sus subcampos), avanza hacia las tareas específicas de análisis de video, detalla las arquitecturas de redes neuronales empleadas y concluye con los fenómenos y desafíos críticos que surgen durante el entrenamiento y el despliegue de estos modelos.

Inteligencia Artificial (IA): es un término global usado para describir el conjunto de técnicas que, integradas, simulan la inteligencia humana y funciones cognitivas a través de sistemas capaces de comprender el lenguaje natural, reconocer patrones o visualizar objetos (IBM, 2024). Con ello se busca automatizar o asistir en el desarrollo de una gran diversidad de tareas y centrarse esencialmente en la clasificación, la predicción y la optimización.

Machine Learning: es una subdisciplina de la Inteligencia Artificial que, a partir de datos estructurados, es capaz de reconocer patrones y realizar predicciones en base a ellos, sin supervisión humana.

Deep Learning: es un subcampo del *Machine Learning* que no requiere datos estructurados para aprender, haciendo uso principalmente de redes neuronales inspiradas en el funcionamiento de las neuronas biológicas, que llevan a cabo el intercambio de información a través de impulsos eléctricos y químicos, mientras que las neuronas artificiales lo hacen por medio de cálculos matemáticos (Red Hat, 2026).

Visión por computadora: es una técnica de extracción de características de imágenes o imágenes de video a partir del reconocimiento de patrones y el procesamiento de imágenes (Unity, 2025) aplicando algunos procedimientos del *Machine Learning*. Dicha técnica se inspira en la visión humana.

Reconocimiento de Actividades Humanas: subcampo de la visión por computadora enfocado en identificar y clasificar acciones o comportamientos humanos a partir de secuencias de video. La detección de violencia es un caso de uso específico dentro de este ámbito, que requiere analizar la interacción temporal y/o espacial entre múltiples entidades.

Detección de Objetos: tarea de visión por computadora que consiste en identificar la presencia, la categoría y la ubicación (mediante cuadros delimitadores o *bounding boxes*) de objetos específicos en una imagen.

Red Neuronal Convolutacional: es un tipo de red neuronal artificial y un modelo de aprendizaje basado en las redes neuronales biológicas que, a partir de capas convolucionales y de agrupación, es capaz de reconocer patrones, realizar detecciones y clasificar imágenes (NVIDIA, 2017).

Modelo Pre-entrenado: es un modelo basado en redes neuronales artificiales que ha sido previamente entrenado con una amplia variedad de datos para fines de generalización.

Transfer Learning (Aprendizaje por Transferencia): es una técnica de inteligencia artificial que utiliza el aprendizaje previo de un modelo para entrenarlo con una nueva tarea similar.

Modelo de Múltiples Etapas: se trata de sistemas basados en redes neuronales artificiales que poseen una alta precisión, ya que las neuronas de salida actúan como neuronas de entrada para otras neuronas, y así sucesivamente, siguiendo una serie de pasos. A diferencia de los modelos de una etapa, estos generan propuestas de regiones de detección y,

posteriormente se realiza la clasificación de objetos dentro de los cuadros delimitadores (SharkYun, 2024).

Modelo de una etapa: son sistemas de redes neuronales artificiales reconocidos por su velocidad, puesto que, la detección de objetos la realizan directamente en un solo proceso y dividen la imagen y realizan la asignación de los cuadros delimitadores de manera simultánea (SharkYun, 2024).

EfficientNetB0: es un modelo multi-etapa de *Machine Learning* pre-entrenado a partir de un conjunto de imágenes diverso llamado *ImageNet*. Este modelo se basa en una red neuronal convolucional. Se utiliza principalmente para la clasificación de imágenes (MathWorks, 2019).

ResNet50: es un modelo multi-etapa de *Machine Learning* pre-entrenado que hace uso de conexiones residuales, lo que le permite dar saltos entre pasos del proceso de aprendizaje. Tiende a ocuparse en la clasificación de imágenes (Ultralytics, 2026).

YOLOv8: es un modelo de visión por computadora utilizado ampliamente en la detección de objetos, clasificación y segmentación de imágenes (Ultralytics, 2023).

Desbalance de clases: los conjuntos de datos para el entrenamiento de modelos de *Machine Learning* generalmente se clasifican bajo una etiqueta favorable y una desfavorable (Google, 2025). Un conjunto de datos adecuado para el entrenamiento debería poseer una relación entre las cantidades de ambas etiquetas de más o menos la misma cifra; a esto se le conoce como balance de clases. El balance correcto permitirá la correcta discriminación del modelo frente a los escenarios o condiciones para los que ha sido entrenado.

Sin embargo, en caso de presentar una diferencia notable entre la cantidad de datos de la etiqueta favorable y la desfavorable (lo que se conoce como desbalance de clases), ello daría lugar a una inclinación predictiva hacia la clase mayoritaria, sesgando los resultados.

En la *Figura 5* se muestra un ejemplo visual entre el balance y el desbalance de clases. La clase balanceada cuenta con la misma cantidad (o similar) entre caritas felices y caritas tristes, tratándose así de un conjunto de datos homogéneo. Por otro lado, la clase desbalanceada muestra una cantidad reducida de caritas tristes, predominando las caritas felices; en este caso, se interpretaría como un conjunto de datos heterogéneo.

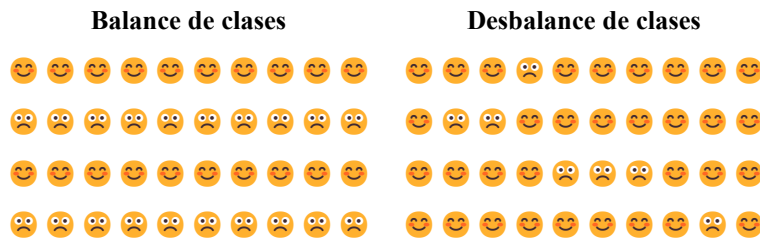


Figura 5. Contraste entre el balance y el desbalance de clases.

Data Augmentation (Aumento de Datos): es una técnica con la que, a partir de pequeñas modificaciones en los datos originales, es posible generar nuevos datos. Suele usarse en modelos de *Machine Learning* que aplican técnicas de visión por computadora, cuando los conjuntos de datos no son diversos o presentan un alto desequilibrio de clases, para evitar que el modelo memorice los patrones en lugar de aprenderlos y lograr realizar predicciones reales es que se aplica la técnica del aumento de datos a partir de las rotaciones, recortes, reflejos y cambios en el color (contraste, brillo, sombras, bordes) de las imágenes de entrada (Amazon Web Services, 2026).

Desplazamiento de dominio (Domain Shift): es el conflicto que se presenta ante el cambio de escenarios en los que fue entrenado el modelo, dando como resultado una predicción incorrecta o incluso la falla del modelo.

De manera ilustrativa, la *Figura 6* muestra dos conjuntos de datos del mismo rubro, donde uno se utiliza como conjunto de entrenamiento y el otro para la validación de un modelo. Se obtiene, como resultado, una clasificación acertada en el primer caso. A la vez, se obtiene una clasificación errónea en el segundo caso. Concretamente con el número “0”. Probablemente el error se originó por la diferencia en la iluminación que muestra cada imagen.

Sobreajuste de fondo (Background Overfitting): es la memorización de un modelo de los datos con los que ha sido entrenado, repercutiendo en su incapacidad para generalizar y realizar nuevas predicciones.

Early Stopping: es una técnica de control que, al definirse o incluirse en un algoritmo de aprendizaje, se activa cuando la validación del modelo no muestra mejora en cada época,

como medida de prevención de la memorización del modelo, puesto que esta afectaría la precisión del modelo y le impediría generalizar sobre los datos.



Figura 6. Dominios.

4.3 Métricas para la evaluación de modelos de IA

Este último grupo establece las herramientas matemáticas para validar el rendimiento del modelo. El orden va desde la herramienta base que organiza los resultados, pasando por las métricas de proporción derivadas de ella, hasta llegar a las métricas avanzadas diseñadas específicamente para evaluar tareas complejas como la detección de objetos y los datos desbalanceados.

Accuracy: es una métrica que promedia el número de clasificaciones acertadas entre el total de clasificaciones (Google, 2026). Su fórmula se ve de la siguiente manera:

$$Accuracy = \frac{Clasificaciones\ acertadas}{Total\ de\ clasificaciones} = \frac{VP + VN}{VP + VN + FP + FN}$$

donde:

$VP = Verdaderos\ Positivos$

$VN = Verdaderos\ Negativos$

$FP = Falsos\ Positivos$

$FN = Falsos\ Negativos$

F1-Score: es una métrica de rendimiento de clasificación binaria, aplicable para modelos de *Machine Learning* que fueron entrenados o evaluados a partir de conjuntos de datos desbalanceados. Y se calcula como la media armónica de la precisión y la exhaustividad (*Recall*) (C3.ai, 2026).

$$F_1 = 2 \times \left(\frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \right)$$

Recall: es una métrica que promedia, de manera ponderada, los positivos reales clasificados correctamente entre todos los positivos (verdaderos positivos y falsos negativos) (Google, 2026).

$$\text{Recall} = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Negativos}}$$

Matriz de Confusión: es una herramienta visual que sirve de apoyo para la interpretación del rendimiento que ha obtenido un modelo durante su entrenamiento entre los valores reales (verdaderos positivos y negativos) y los valores predichos (falsos positivos y negativos). De esta forma, es posible analizar la capacidad discriminativa de un modelo frente a la clasificación de un fenómeno específico.

A continuación, en la *Figura 7* se demuestra visualmente una matriz de confusión. Misma que se compone por cuatro cuadrantes a los que se les asigna un valor del total de datos con los que ha sido entrenado el modelo, mismos que son marcados de acuerdo con una escala de calor según su valor dentro de un rango previamente determinado.



Figura 7. Matriz de confusión.

Curvas AUC-ROC (Área Bajo la Curva): se trata de una representación visual que expone gráficamente la capacidad matemática de clasificación binaria de un modelo

entrenado entre sus resultados correspondientes a los falsos positivos y los verdaderos positivos. Esta gráfica se encuentra dividida a la mitad por una línea punteada con un valor AUC de 0.5 que indica un funcionamiento al azar, mientras que las líneas que se encuentren por encima (un AUC mayor a 0.5 y menor o igual a 1) señalan un adecuado aprendizaje de discriminación, y por otro lado, todas aquellas líneas que se encuentren por debajo (un AUC menor a 0.5) manifiestan un funcionamiento degradante en comparación con el azar, es decir, el modelo presenta memorización e incluso una confusión sistemática severa.

La *Figura 8* ilustra una Curva UAC-ROC en la que se presentan los tres casos posibles antes mencionados. Pues para este trabajo de investigación, la Curva AUC-ROC demuestra los resultados obtenidos de tres modelos pre-entrenados al finalizar su fase de entrenamiento. Algo adicional que no se presenta en la figura de ejemplo, es que en las gráficas de la Curva AUC-ROC se exponen las leyendas para cada modelo y el puntaje AUC que ha obtenido. Esto para su análisis e interpretación.

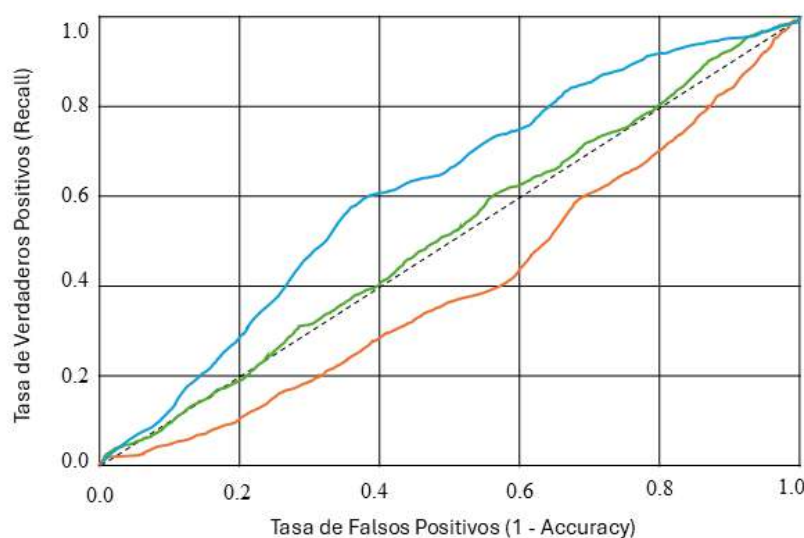


Figura 8. Curvas AUC-ROC.

5 MÉTODOS

En este apartado se detallan los pasos a seguir para resolver el problema. La finalidad de la presente sección es estructurar, guiar y fundamentar el proceso de investigación, asegurando la validez y la fiabilidad de los resultados.

En un primer momento se manifiestan el tipo y el diseño de la investigación, considerando el enfoque, el alcance, el diseño y la temporalidad. Así es como se define la estrategia que regirá la ejecución de esta investigación y garantiza el cumplimiento de los objetivos previamente establecidos.

Seguidamente se exponen los requerimientos o especificaciones del proyecto de investigación, estos se refieren a las herramientas y elementos por ocupar durante el desarrollo del proyecto. Los cuales abarcan tres conjuntos de datos (datasets) obtenidos a partir de plataformas públicas y gratuitas en línea que más adelante se presentan, y herramientas alojadas en la nube para el almacenamiento de los conjuntos de datos y el desarrollo de los algoritmos necesarios para la demostración de lo planteado previamente en la sección dedicada a la hipótesis.

Como tercera subsección, se tiene el diseño e implementación, en esta se explican los criterios de extracción de imágenes de video a partir de los conjuntos de datos que se poseen. Asimismo, se exhibe el comportamiento interno de cada uno de los modelos al procesar imágenes, pues es importante mencionarlo para comprender la lógica adoptada en la toma de decisiones.

Por último, se menciona la experimentación. Parte dedicada a la demostración de los resultados obtenidos mediante la manipulación y el procesamiento de los conjuntos de datos. Misma que se llevó a cabo en tres etapas, pues se desarrolló un algoritmo para cada objetivo específico, estas etapas se desglosan de la siguiente manera:

1. Configuración general y extracción de *frames*.
2. Entrenamiento de los modelos.
3. Testeo y métricas.

El primer algoritmo se especializó en la conexión con la herramienta alojada en la nube, donde se almacenan los conjuntos de datos, y en la extracción de imágenes de video

con una dimensión y una cantidad estandarizadas. Por otra parte, el segundo algoritmo se especializó únicamente en el entrenamiento puro de los modelos. Y, por último, el tercer algoritmo se especializó en evaluar los modelos frente a conjuntos de datos con mayor desbalance de clases, con el fin de medir la capacidad de generalización y la robustez de cada uno. Posteriormente, las métricas obtenidas se demuestran mediante representaciones visuales, como matrices de confusión y gráficas de barras.

5.1 Tipo y diseño de la investigación

Para garantizar el rigor científico y la reproducibilidad del presente estudio, se establece la clasificación metodológica de la investigación conforme a los criterios de (Hernández Sampieri, Fernández Collado, & Baptista Lucio, 2014) y de (Behar Rivero, 2008), considerando cuatro dimensiones fundamentales: el enfoque, el alcance, el diseño y la temporalidad.

De este modo, la presente investigación adopta un enfoque cuantitativo, dado que se fundamenta en la recolección de datos numéricos para el análisis estadístico de fenómenos medibles. Específicamente, se cuantifican variables de desempeño computacional (latencia de inferencia en milisegundos, tasa de cuadros procesados por segundo) y métricas de evaluación de modelos (sensibilidad, especificidad, *F1-Score*, exactitud global) mediante pruebas estandarizadas en conjuntos de datos predefinidos. El enfoque cuantitativo permite la validación objetiva de la hipótesis central a través de la comparación numérica entre arquitecturas, garantizando la generalización de los hallazgos dentro de las condiciones experimentales establecidas.

El alcance de la investigación se clasifica como correlacional-comparativo, con elementos explicativos, por las siguientes razones:

- Correlacional: se analiza la relación entre las variables independientes (arquitectura del modelo, complejidad paramétrica, estrategia de inferencia) y las variables dependientes (sensibilidad al desplazamiento de dominio, eficiencia computacional).
- Comparativo: se contrastan sistemáticamente tres arquitecturas distintas (EfficientNetB0, ResNet50 y YOLOv8) bajo idénticas condiciones experimentales, con el propósito de identificar diferencias significativas en su desempeño.

- Explicativo: más allá de describir las diferencias observadas, se busca explicar las causas arquitectónicas y operativas que explican por qué un modelo de una sola etapa (YOLOv8) presenta mayor resiliencia y eficiencia que los clasificadores densos de múltiples etapas.

En cuanto al diseño, el estudio se desarrolla bajo un diseño cuasiexperimental comparativo. Este enfoque contempla la manipulación de las variables independientes mediante el control de la arquitectura del modelo, considerando tres alternativas (EfficientNetB0, ResNet50 y YOLOv8), así como de los conjuntos de datos empleados para la evaluación (RLVS, SCVD y AIRTLAB). No obstante, al tratarse de modelos pre-entrenados evaluados sobre conjuntos de datos fijos, las unidades de análisis no corresponden a sujetos experimentales, por lo que la asignación aleatoria clásica a grupos no resulta aplicable. Con el propósito de garantizar que las diferencias observadas en el desempeño sean atribuibles exclusivamente a la arquitectura de los modelos, se mantienen constantes las condiciones experimentales, incluyendo el hardware utilizado (GPU Tesla 4), el entorno de desarrollo (Python 3.10), las librerías (Ultralytics 8.0), la resolución de entrada de las imágenes (640×640) y los hiperparámetros de inferencia, minimizando así la influencia de variables externas.

Finalmente, la investigación es de corte transversal, ya que la recolección de datos se realiza en un único momento para cada modelo evaluado. No se realiza seguimiento de la evolución de los modelos a lo largo del tiempo, sino que se comparan sus desempeños en un punto fijo del ciclo de desarrollo (post-entrenamiento con Transfer Learning).

5.2 Requerimientos y herramientas

Con la finalidad de asegurar la reproducibilidad de la experimentación y del análisis posterior de los resultados, en este apartado se proporciona una descripción de los datos de prueba, las plataformas de desarrollo, el lenguaje de programación y sus librerías.

5.2.1 Conjunto de datos de prueba

En esta investigación se utilizan tres conjuntos de datos de acceso libre obtenidos de plataformas públicas como GitHub y Kaggle. Estos tres conjuntos de datos se denominan RLVS (Real Life Violence Situations) (Mohamed, Mohamad, & Abd El Massih, 2019),

SCVD (Smart City CCTV Violence Detection) (Aremu, Li , & Alameeri, 2024) y AIRTLAB (airtlab, 2020). Estos contienen una diversidad de videos capturados por sistemas de videovigilancia empleados en entornos públicos reales, como calles, plazas o centros sociales, y en entornos privados controlados, como oficinas, cuartos o edificios, desde diferentes ángulos, con cambios de iluminación, con distintas calidades de imagen y con lejanía o cercanía de los actores involucrados en la incidencia violenta o no violenta. Igualmente, contienen escenas de películas o de deportes.

Cada uno de los conjuntos de datos cuenta con carpetas y subcarpetas que clasifican su contenido, principalmente bajo las etiquetas de “Violencia” y “No Violencia”, el resto de la información es irrelevante, al menos para el presente caso, razón por la cual se eliminan aquellas carpetas y subcarpetas innecesarias. El uso de conjuntos de datos preexistentes se fundamenta en el ahorro de tiempo y recursos para la grabación de una cantidad considerable de incidencias violentas y de entornos pacíficos desde diferentes ángulos, con cambios de iluminación y de calidad de imagen. Además, los conjuntos de datos públicos representan una fortaleza al estar respaldados por su uso intensivo en la comunidad de investigadores.

5.2.2 Herramientas de desarrollo

Los conjuntos de datos, los archivos generados durante el desarrollo del proyecto de investigación y los resultados obtenidos deben almacenarse para poder trabajar con ellos, analizarlos y visualizarlos posteriormente. Por tal motivo, se ha seleccionado la herramienta de almacenamiento en la nube, Google Drive. Se optó por almacenar los datos en línea, puesto que el recurso computacional propio presenta limitaciones significativas en cuanto a potencia de cómputo y capacidad de almacenamiento. Además, de esta manera, es posible agilizar el acceso del entorno de desarrollo de los algoritmos a los conjuntos de datos de prueba y a un espacio de almacenamiento para los archivos generados durante el proceso y los resultados finales. En última instancia, Google Drive ofrece de forma gratuita un total de 15 GB de almacenamiento en la nube, lo que resulta suficiente para el desarrollo de este trabajo (Google, 2018).

Google Drive, plataforma en línea desarrollada por Google, tiene como función principal permitir al usuario almacenar y compartir archivos en la nube, sustituyendo la ocupación local de dichos archivos (Google, 2012). La elección de Google Drive para el

almacenamiento de los conjuntos de datos también se basa en la flexibilidad de sus suscripciones y en las funcionalidades que ofrece la plataforma. Al utilizar el almacenamiento en la nube, el proyecto reduce costos y recursos computacionales, lo que agiliza su desarrollo.

Para el desarrollo de código y la documentación de los algoritmos mediante cuadernos de Jupyter, se utiliza Google Colab (*Collaboratory*), que es una plataforma en línea gratuita de Google que permite ejecutar cuadernos de Jupyter (combinación de código, herramientas de diseño textual y visual) en Python desde un navegador (Google, 2019). Además, Google Colab brinda la oportunidad de trabajar sin costo con GPU en la nube, con restricciones como la cantidad de iteraciones de código al día, el tiempo de habilitación de la GPU en el entorno o el uso de la RAM virtual. Es posible mejorar los beneficios de Google Colab mediante el pago de una cuota adicional. Sin embargo, los beneficios gratuitos de ambas plataformas y servicios satisfacen las necesidades del proyecto, y además reducen costos y recursos computacionales.

Adicionalmente, se destaca que existen otros entornos virtuales previamente configurados para el desarrollo de código y su almacenamiento, como Kaggle Notebooks. Sin embargo, la elección de Google Colab se fundamenta en los recursos gratuitos que ofrece, así como para no comprometer la compatibilidad entre herramientas. De esta manera, el desarrollo de los algoritmos y el almacenamiento de los resultados obtenidos se rigen por dos plataformas desarrolladas por la misma empresa tecnológica. Finalmente, al ser una de las plataformas más utilizadas por investigadores y centros de enseñanza, hace que su comunidad sea más grande respecto a otras plataformas y, de esta forma, sea sencillo consultar información sobre su uso o dar solución a problemáticas presentadas (Google, 2025).

5.2.3 Lenguaje de programación

Es importante destacar el uso del lenguaje de programación Python. La elección de este lenguaje de programación se fundamenta en la demanda que este ha adquirido para el desarrollo de tecnologías de punta en las que se aplican algoritmos genéticos, redes neuronales convolucionales (CNNs), Inteligencia Artificial, *Machine Learning*, *Deep Learning*, análisis y ciencia de datos. Además, el índice TIOBE, un indicador de la

popularidad de los lenguajes de programación a nivel mundial que se actualiza mes con mes demuestra que Python se coronó en primer lugar en octubre de 2021 (TIOBE, 2025).

Python es el lenguaje de programación pionero en el desarrollo de algoritmos de redes neuronales convolucionales por su alta capacidad de clasificación mediante técnicas de visión por computadora, ciencia de datos e inteligencia artificial. Al ser un lenguaje de programación multipropósito, cuenta con una amplia variedad de librerías que abordan cada una de las disciplinas descritas (sin mencionar aquellas que no son de importancia para la presente), acelerando los procesos de desarrollo. También es reconocido por ser un lenguaje de programación flexible ante las necesidades del desarrollador o el sistema. Además, por su alta demanda de uso involucra a una comunidad enorme, dando pauta a tener fácil acceso a documentación, solución a problemas y recursos. Por último, es un lenguaje de programación multiplataforma, ventaja que no se demuestra en el actual trabajo, pero es de indispensable valor para aquellos investigadores que no presentan problemas de recursos computacionales (Strategic Platform, 2019).

5.2.4 Librerías y paquetes de software

TensorFlow y Keras: funcionan como el motor tensorial y la interfaz de alto nivel para la carga de los conjuntos de datos y *frames*, el entrenamiento y validación de los modelos: EfficientNetB0 y ResNet50. Asimismo, proporcionan las herramientas necesarias para el mecanismo de *Early Stopping*.

Ultralytics: es la librería oficial y el ecosistema exclusivo del modelo YOLOv8. Indispensable para su ejecución, entrenamiento y validación.

OpenCV: su tarea consiste en capturar los flujos de video, decodificarlos y realizar la extracción iterativa de *frames* a una frecuencia estandarizada, preparándolos así para su uso en el entorno de las redes neuronales.

Pillow: actúa como un complemento útil para el procesamiento de imágenes estáticas, como el redimensionamiento, las conversiones de formato y la aplicación de técnicas de *Data Augmentation*. Funciones de utilidad ante el fenómeno de desplazamiento de dominio.

NumPy: constituye la base matemática subyacente de todo el flujo de trabajo. Dado que las redes neuronales procesan imágenes como matrices multidimensionales (llamadas

tensores en TensorFlow). Esta librería se encarga de gestionar las operaciones algebraicas a bajo nivel.

Pandas: permite estructurar los metadatos de los conjuntos de datos en tablas, facilitando así la gestión de las etiquetas binarias y el seguimiento de los dominios base para las pruebas ciegas.

Scikit-Learn: su uso consiste en generar métricas de eficacia estandarizadas (Accuracy, F1-Score) y representar los datos mediante matrices de confusión.

Matplotlib: permite la visualización de las curvas de aprendizaje durante el entrenamiento y la generación de representaciones visuales como las gráficas de barras.

Seaborn: su uso reside en la generación de gráfica de las matrices de confusión que aquí se presentan.

5.3 Diseño e implementación

Antes del entrenamiento de los modelos, es necesario recuperar imágenes (*frames*) correspondientes a todos los videos que componen cada clase de los conjuntos de datos, pues los modelos son entrenados a partir de esas imágenes. Para este caso en específico se ha definido la extracción de cinco *frames* por segundo (5 FPS) con la finalidad de brindar imágenes más concretas que aludan al tipo de clase al que corresponden y, de esta manera, asistir a la precisión de los modelos pre-entrenados.

Los modelos pre-entrenados seleccionados son EfficientNetB0, ResNet50 y YOLOv8. El primer modelo, EfficientNetB0 se enfoca en la eficiencia y *Transfer Learning*. Se define como una red neuronal convolucional entrenada con una gran variedad de imágenes de la base de datos de *ImageNet* (MathWorks, 2019). El segundo modelo, ResNet50 es igualmente una red neuronal convolucional que a diferencia del anterior enfatiza en su uso como línea base o una red neuronal clásica, además de utilizar conexiones residuales que le permiten saltar entre procesos de aprendizaje. Por último, “YOLOv8 es una arquitectura de modelos de visión por computadora basado en una red neuronal convolucional que resalta por su precisión y velocidad” (Ultralytics, 2023). La elección de dichos modelos se debe a su optimización para la eficiencia, lo que posibilita su despliegue en plataformas con GPU

limitada y, potencialmente, en aplicaciones en tiempo real. Además de basarse en el estudio del arte previamente demostrado. (ver *Tabla 2*).

Tabla 2. Elección de modelos pre-entrenados.

Modelo	Enfoque	Propósito de diseño	Dimensiones (píxeles)
EfficientNetB0	Eficiencia y <i>Transfer Learning</i>	Clasificación de imágenes	224x224 px
ResNet50	Baseline	Clasificación de imágenes	224x224 px
YOLOv8	Detección en tiempo real	Detección de objetos	640x640 px

Posteriormente se eligen las dimensiones de las imágenes (*frames*) para ser reconocidas después por los modelos pre-entrenados. La arquitectura de EfficientNetB0 y ResNet50 trabaja con dimensiones de al menos 224x224 píxeles para imágenes de entrada, mientras que YOLOv8 lo hace con dimensiones de 640x640 píxeles (MathWorks, 2019) (Ultralytics, 2025). Esto se debe a la naturaleza del propósito para el que se diseñó cada modelo, EfficientNetB0 y ResNet50 se orientan a la clasificación de imágenes, YOLOv8 por su parte se inclina a la detección de objetos en tiempo real, por lo que requiere de dimensiones más grandes de entrada para sus imágenes, evitando así la pérdida de elementos críticos.

Considerando los motivos anteriores, las dimensiones generales de extracción de los *frames* son de 640x640 píxeles. Sin embargo, es importante mencionar que, para llevar a cabo el entrenamiento de EfficientNetB0 y ResNet50, se redimensionaron las imágenes a 224x224 píxeles con el objetivo de trabajar con sus propias arquitecturas y ser más eficientes durante el entrenamiento. En otras palabras, los *frames* extraídos por cada video tendrán un tamaño de 640x640 píxeles. Donde se mantendrán esas dimensiones para el entrenamiento de YOLOv8 mientras que para el entrenamiento de EfficientNetB0 y ResNet50 se redimensionarán a un tamaño de 224x224 píxeles.

5.3.1 Algoritmos de entrenamiento y validación

La finalidad del presente proyecto consta en realizar una comparativa entre tres modelos pre-entrenados poniendo a prueba su precisión de detección ante incidencias de carácter violento, así como la velocidad de detección en tiempo real sin pérdida de datos.

Para llevar a cabo la comparativa, se obtuvieron tres conjuntos de datos donde el conjunto de datos con un mayor balance (RLVS), es decir, relación proporcional entre la

cantidad de videos de la primera clase respecto con los videos de la segunda clase se usa como base para el entrenamiento de los modelos, de esta manera será más específico y equilibrado el entrenamiento. El conjunto de datos con un menor balance con respecto al primero (SCVD) se utiliza para la validación de los modelos y el conjunto de datos con menor balance (AIRTLAB) de los tres se utiliza para la robustez de los modelos ante incidencias no previstas y su capacidad de adaptabilidad.

Para el acceso a la GPU que ofrece Google Colab, es necesario precargar los conjuntos de datos en alguna plataforma de almacenamiento en la nube para no consumir recursos locales del equipo de cómputo y agilizar procesos. Para este caso se ha mencionado Google Drive, y para acceder a los conjuntos de datos es necesario darle todos los permisos de acceso a diferentes tipos de archivos a Colab mediante código, incluso el de aquellos que no se estarán ocupando. El orden cronológico para el entrenamiento y la validación de los modelos consistió en 3 etapas:

1. Configuración general y extracción de *frames*.
2. Entrenamiento de los modelos.
3. Testeo y métricas.

Para llevar a cabo la primera etapa de la experimentación: configuración general y extracción de *frames*, se desarrolló un algoritmo automatizado encargado de la gestión de datos entre el almacenamiento en la nube (Google Drive) y el entorno de ejecución de código en línea (Google Colab). En este procedimiento se establece la comunicación entre las dos plataformas para compartir, manipular y almacenar archivos. También se instalan las librerías necesarias para el procesamiento, y tratamiento de los datos. Asimismo, asegura la extracción consistente de *frames* a una tasa de muestreo definida (5 FPS), optimizando el volumen de datos para los modelos posteriores.

Algoritmo 1 Primera etapa: configuración general y extracción de *frames*

Input: R_{origen} , $R_{destino}$, R_{temp} , FPS ; /* R_{origen} rutas de los datos crudos en Drive, $R_{destino}$ ruta de almacenamiento para los datos procesados, R_{temp} ruta temporal en el entorno local para el procesamiento de los datos, y FPS frecuencia de cuadros por segundo */

1: Conexión con Google Drive

```

2:  $R_{origen}$ ;
3: for each clase  $C$  en la lista de clases do
4: Crear directorio temporal local  $D_c$  dentro de  $R_{temp}$ 
5: Obtener lista de archivos de video  $V$  pertenecientes a la clase  $C$  en  $R_{origen}$ 
6: for each video  $v$  en  $V$  do
7: Cargar el archivo de video  $v$  en memoria
8: Obtener los  $FPS_{nativos}$  del video  $v$ 
9: Calcular el intervalo de salto de frames  $I = round(FPS_{nativo}/FPS_{objetivo})$ 
10: Inicializar contador de frames  $k = 0$ 
11: while el video  $v$  tenga frames por leer do
12: Leer el frame actual  $F_k$ 
13: if  $k(mod I) = 0$  then
14: Procesar  $F_k$ 
15: Guardar  $F_k$  como archivo de imagen en el directorio local  $D_c$ 
16: end if
17: Incrementar  $k$ 
18: end while
19: end for
20: Comprimir el contenido completo de  $D_c$  y exportar a  $R_{destino}$ 
21: Eliminar  $R_{temp}$  para liberar espacio
22: end for
23: Return mensaje de finalización

```

La segunda etapa consistió en el entrenamiento de los modelos. Dado el tamaño limitado del conjunto de datos y la necesidad de una convergencia eficiente, se optó por una estrategia de *Transfer Learning*. Esto implicó inicializar los modelos con pesos pre-entrenados en *ImageNet*, congelando las capas base para conservar la extracción de características visuales genéricas y entrenando únicamente una nueva cabecera de

clasificación adaptada a las clases del problema. Para garantizar la robustez del modelo y evitar el sobreajuste (*Overfitting*), se implementó un algoritmo de entrenamiento iterativo con mecanismos de *Data Augmentation* en tiempo real y *Early Stopping*, el cual detiene el proceso si la precisión de validación no mejora tras un número establecido de épocas (paciencia).

Algoritmo 2 Segunda etapa: entrenamiento de los modelos

Input: D_{train} , D_{val} , M_{lista} , H_{params} ; /* D_{train} dataset para el entrenamiento, D_{val} dataset para la validación, M_{lista} modelos, H_{params} hiperparámetros definidos ($\acute{E}pocas E_{max} = 15$, $Batch\ size\ B = 32$, $Paciencia\ P = 3$)* /

- 1: **Inicializar** entorno de ejecución GPU-T4
 - 2: **Cargar y descomprimir** D_{train} y D_{val} en el almacenamiento local
 - 3: Data Augmentation
 - 4: Redimensión de imágenes a 224x224 px
 - 5: **for each** arquitectura A en M_{lista} **do**
 - 6: **Instanciar** modelos base A_{base} con pesos pre-entrenados (ImageNet)
 - 7: **Congelar** los pesos A_{base} ($Trainable = False$)
 - 8: **Construir** cabecera de clasificación personalizada H
 - 9: GlobalAveragePooling2D
 - 10: Dense /* 256 neuronas, activación ReLU) */
 - 11: Dropout (0.5)
 - 12: Dense (2 neuronas, activación Softmax)
 - 13: **Compilar** modelo final $M = A_{base} + H$ /* usando optimizador Adam y pérdida Categorical Crossentropy*/
 - 14: **Inicializar** variable $Mejor_Score = 0$ y contador $Paciencia_Actual = 0$
 - 15: **for** época $e = 1$ hasta E_{max} **do**
 - 16: **Entrenar** M con un lote de datos de D_{train}
 - 17: **Evaluar** M usando D_{val} para obtener $Accuracy_{val}$
 - 18: **if** $Accuracy_{val} > Mejor_Score$ **then**
-

```

19: Actualizar  $Mejor\_Score = Accuracy_{val}$ 
20: Guardar pesos del modelo  $M$ 
21: Reiniciar  $Paciencia\_Actual = 0.23$  else
22: Incrementar  $Paciencia\_Actual$ 
23: end if
24:  $Paciencia\_Actual \geq P$  then
25: Romper ciclo de entrenamiento (Early Stopping)
26: end if
27: end for
28: end for
29: Return conjunto de archivos de modelos  $Pesos_{optimos}$ 

```

Finalmente, la tercera etapa evalúa el rendimiento de los modelos entrenados en dos vertientes de análisis: eficacia (capacidad de detección) y eficiencia (velocidad de procesamiento). Para evaluar la eficacia, se sometió a los modelos a una “prueba ciega” utilizando conjuntos de datos no vistos durante el entrenamiento (SCVD y AIRTLAB), permitiendo analizar el comportamiento de generalización y la resistencia al fenómeno de desplazamiento de dominio (*Domain Shift*). Se calcularon métricas estándar como la matriz de confusión, Exactitud (*Accuracy*), Sensibilidad (*Recall*) y *F1-Score*.

Paralelamente, para evaluar la viabilidad de implementación en sistemas de videovigilancia en tiempo real, se diseñó una prueba de latencia computacional. Este procedimiento midió el tiempo de inferencia promedio por cuadro (milisegundos) y la tasa de cuadros por segundo (FPS) en un entorno controlado, asegurando una comparativa justa bajo las mismas condiciones de hardware.

Algoritmo 3 Tercera etapa: testeo y métricas

Input: $M_{entrenados}$, D_{test} , Img_{dummy} , N_{iter} /* $M_{entrenados}$ Conjunto de modelos optimizados, D_{test} lista de conjuntos de datos externos, Img_{dummy} matriz de imagen sintética para pruebas de estrés de velocidad, N_{iter} número de iteraciones*/

1: **Inicializar** entorno de prueba en GPU

2: **Evaluación de eficacia**

3: **for each** Dataset D en D_{test} **do**

4: **Cargar** imágenes de prueba de D

5: **for each** Modelo M en $M_{entrenados}$ **do**

6: **Generar** predicciones para todas las imágenes en D

7: **Comparar** con las etiquetas reales

8: **Calcular** matriz de confusión $C_{M,D}$

9: **Calcular** métricas globales: $Accuracy, Recall_{violencia}, F1_Score$

10: **Almacenar** resultados en $Reportes_{métricas}$

11: **end for**

12: **end for**

13: **Evaluación de eficiencia**

14: **for each** Modelo M en $M_{entrenados}$ **do**

15: **Realizar** inferencia de calentamiento (Warm-up)

16: **Inicializar** cronómetro $T_{total} = 0$

17: **for** $i = 1$ hasta N_{iter} **do**

18: **Iniciar** tiempo t_{start}

19: **Ejecutar** inferencia: $M(Img_{dummy})$

20: **Detener** tiempo t_{end}

21: $T_{total} = T_{total} + (t_{end} - t_{start})$

22: **end for**

23: **Calcular** latencia promedio $L_{avg} = T_{total}/N_{iter}$

24: **Calcular** frames por segundo $FPS = 1/L_{avg}$

25: **Registrar** (M, L_{avg}, FPS) en $Reporte_{velocidad}$

26: **end for**

27: **Generar** gráficas

28: **Return** $Reportes_{métricas}, Reporte_{velocidad}$

5.4 Experimentación

En esta última sección del aparato metodológico se describen los procedimientos empleados tanto en la experimentación como en el análisis de los resultados.

5.4.1 Configuración inicial

La configuración inicial se orienta a la conexión entre Google Colab y Google Drive para el acceso y almacenamiento de los conjuntos de datos y los archivos que se vayan generando durante el proceso, como la extracción de los *frames* a partir de los videos de los conjuntos de datos.

Después de establecer una conexión entre plataformas, se instala la librería de OpenCV para el procesamiento y extracción de las imágenes de video. Luego se procede con la definición de las rutas donde se almacenan los conjuntos de datos para luego poder acceder a ellos. Para esto, debieron cargarse previamente en Google Drive. Una vez hecho, se crean por medio de código, las rutas donde se almacenarán los *frames* por extraerse, y finalmente, se extraen los *frames* de cada video de todos los conjuntos de datos (ver *Figura 9*).

5.4.2 Entrenamiento

Como se ha creado un algoritmo para cada etapa, es necesario establecer una conexión entre Google Colab y Google Drive para cada uno al inicio e instalar las nuevas librerías en cada caso. Seguidamente, se realiza el entrenamiento de los modelos a partir del acceso a los *frames* obtenidos del conjunto de datos RLVS, (ver *Figura 10*), si después de 3 épocas seguidas no existe una mejora en la precisión, se activa el mecanismo de *Early Stopping* y el modelo conserva su estado óptimo (la mejor época antes de empezar a memorizar).



Figura 9. Diagrama de flujo de la configuración inicial.

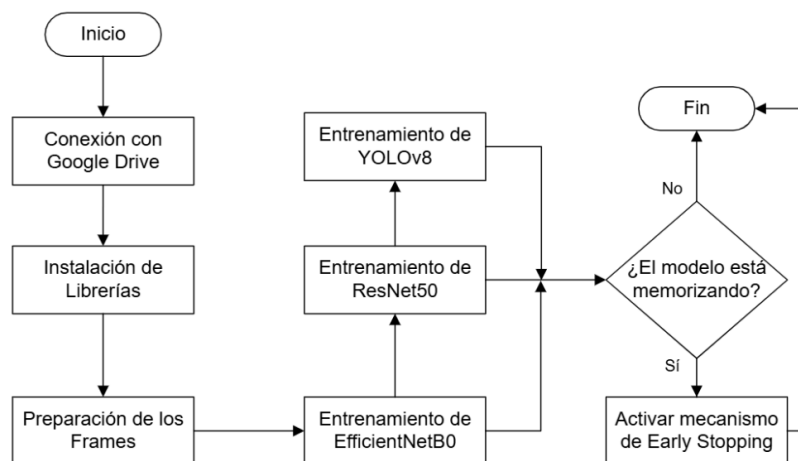


Figura 10. Diagrama de flujo del entrenamiento de los modelos.

5.4.3 Evaluación de las métricas

Se repite el código orientado a la conexión entre plataformas e instalación de librerías, si es necesario, se instalan nuevas librerías. Para la evaluación de los modelos pre-entrenados, se cargan los *frames* que se extrajeron del conjunto de datos SCVD y los modelos previos. Como resultado, se genera un archivo *.csv con las métricas obtenidas por cada modelo y, a partir de estas, se construye la matriz de confusión de la evaluación con SCVD (ver *Figura 11*). Lo mismo sucede con AIRTLAB. (ver *Figura 12*).

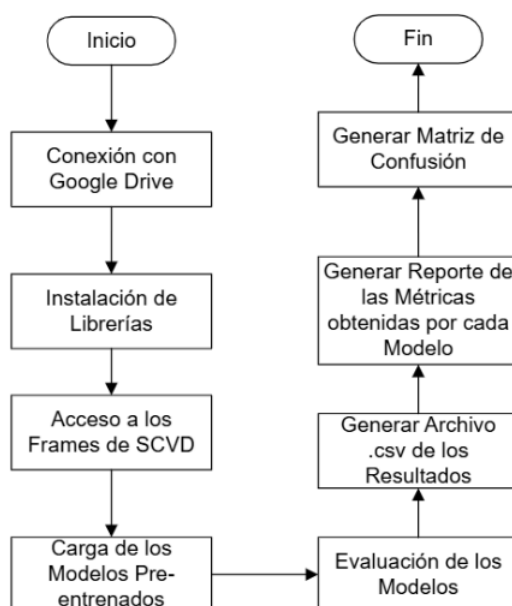


Figura 11. Diagrama de flujo de la evaluación con SCVD.

5.4.4 Prueba de latencia

En el mismo cuaderno del algoritmo orientado a la evaluación de los modelos, en un bloque de código separado se cargan los modelos y, a partir de una matriz de imagen sintética, se inicia una iteración de 100 pasadas para medir la velocidad de los modelos. Al final se genera una gráfica de barras que compara la latencia de cada modelo. (ver *Figura 13*).

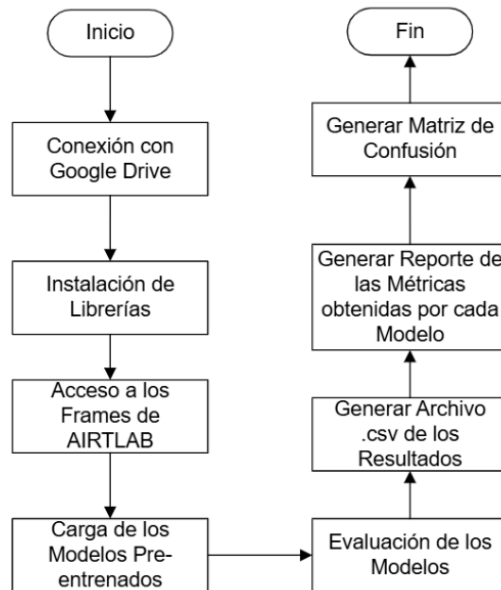


Figura 12. Diagrama de flujo de la evaluación con AIRTLAB.

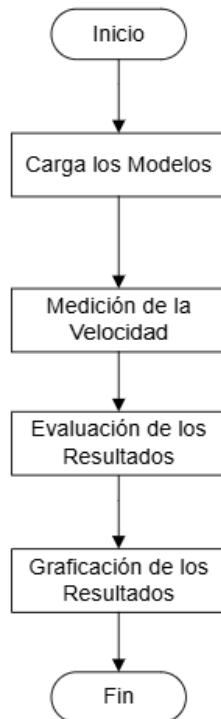


Figura 13. Diagrama de flujo de la prueba de latencia de los modelos.

6 RESULTADOS

En este apartado se detallan los resultados obtenidos a partir del entrenamiento (*training*) y evaluación (*testing*) de los tres modelos pre-entrenados, así como una comparativa entre los mismos con la finalidad de determinar el modelo más viable para su implementación en un sistema de videovigilancia.

Como primer punto, los tres modelos fueron entrenados con el conjunto de datos (RLVS), que mostró un mayor balance entre la cantidad de frames clasificados como “No Violencia” y aquellos clasificados como “Violencia”. De esa manera se omite obtener resultados afectados por el desplazamiento de dominio (*Domain Shift*).

Por consiguiente, se realizó la misma comparativa entre los resultados obtenidos en la evaluación de cada modelo a partir del conjunto de datos SCVD, mismo que se encuentra ligeramente desbalanceado hacia la clase “Violencia”. Finalmente, como buena práctica para determinar la robustez de cada modelo se realizó una segunda evaluación, en esta ocasión con el conjunto de datos con mayor desbalance (AIRTLAB) hacia la clase “Violento” (Violencia) y de esta forma demostrar las conclusiones de detección a las que llega cada uno de los modelos ante incidencias no contempladas.

Posteriormente, se realizó una valoración de la eficiencia computacional de cada modelo, poniendo a prueba su velocidad de detección, considerando el tiempo que le tomó analizar un *frame* (latencia) y la velocidad con la que realizó la detección.

6.1 Exploración de los conjuntos de datos

Para analizar los resultados de una manera correcta y justa, es conveniente analizar primero los conjuntos de datos utilizados, para conocer su estructura, tipos de escenas que contienen y complejidades particulares.

6.1.1 Real Life Violence Dataset (RLVS)

Este conjunto de datos fue el utilizado para el entrenamiento de los modelos al presentar un balance del 50% dedicado a la clase “Violencia” y el otro 50% a la clase “No Violencia”. En la *Figura 14* se presentan algunas imágenes representativas de las escenas

violentas, mientras que en la *Figura 15* se muestran ejemplos de escenas etiquetadas como no violentas.

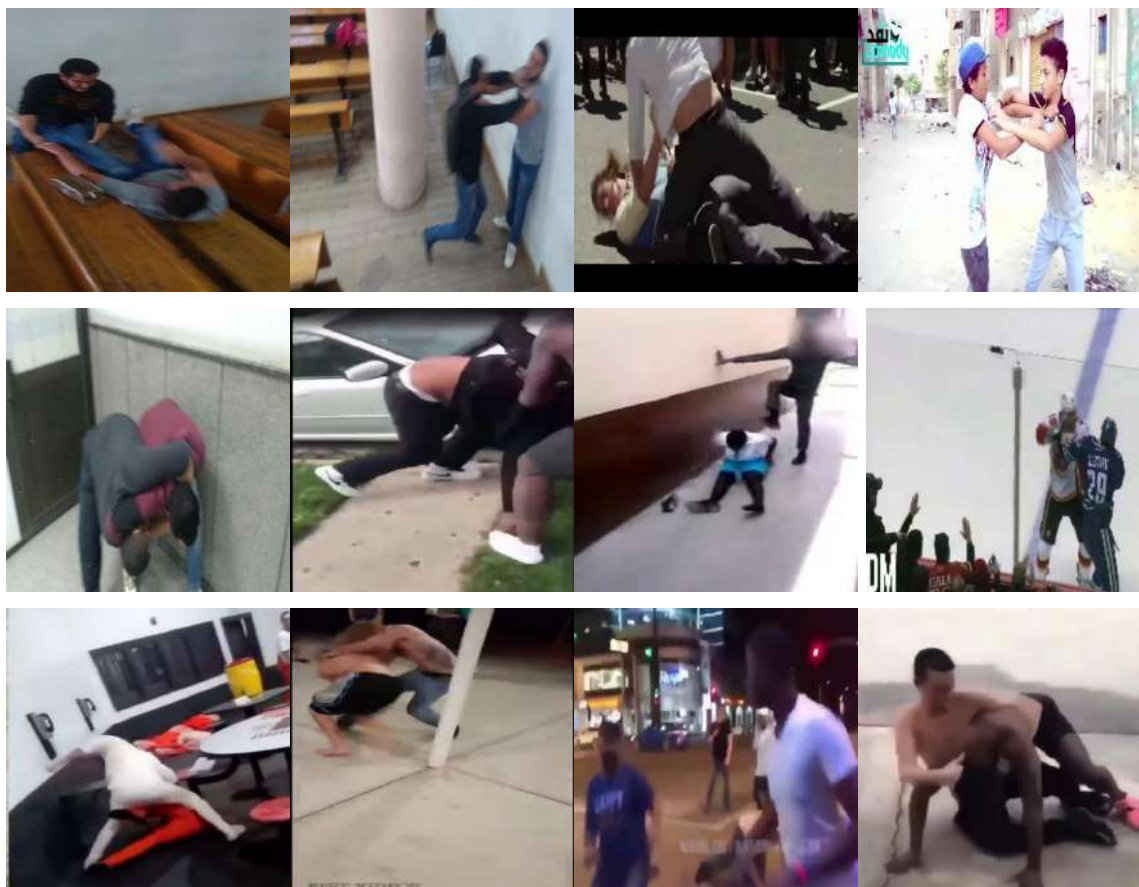


Figura 14. Escenas violentas del conjunto de datos RLVS.





Figura 15. Escenas pacíficas del conjunto de datos RLVS.

Entre las imágenes más difíciles de clasificar se encuentran aquellas que incluyen movimientos demasiado rápidos en los que es complejo interpretar a una persona y oclusiones parciales o totales. Algunos ejemplos se demuestran en las imágenes de la *Figura 16*.



Figura 16. Ejemplos de imágenes del conjunto de datos RLVS que plantean un desafío de clasificación.

La primera imagen muestra a dos mujeres intercambiando golpes. Sin embargo, por los movimientos bruscos y lo que logra capturar la cámara (ropa y cabello que oculta el rostro de las mujeres) presenta un desafío de interpretación. La segunda imagen es más concreta, pero no hay movimientos evidentes que puedan ser categorizados como violentos o pacíficos por completo, actúa como una imagen neutra que es discriminada a partir del *frame* anterior o siguiente. La tercera imagen representa la misma situación. Y la cuarta imagen es una

captura en negro que no exhibe ningún tipo de escena, y es discriminada igualmente que la segunda y tercera imagen, pues la cámara fue obstruida durante la escena.

6.1.2 Smart City CCTV Violence Detection (SCVD)

SCVD es el conjunto de datos utilizado para la evaluación de los modelos en una primera etapa orientada a la generalización de estos. La elección de este conjunto de datos para tal tarea se fundamenta en su balance de clases, presenta una ligera inclinación hacia la clase “Violencia”, pero es aceptable. Algunos ejemplos de las escenas violentas de este conjunto de datos se muestran en la *Figura 17*, mientras que las escenas no violentas se presentan en los ejemplos de la *Figura 18*.



Figura 17. Escenas violentas del conjunto de datos SCVD.



Figura 18. Escenas pacíficas del conjunto de datos SCVD.

Para este caso, las imágenes más complicadas para su clasificación fueron aquellas que demuestran una imagen borrosa o el fenómeno de detección se encuentra lejano, como se ilustra en los ejemplos de la *Figura 19*.



Figura 19. Ejemplos de imágenes del conjunto de datos SCVD que presentan un desafío de clasificación.

La primera imagen corresponde a un escenario pacífico en el que a una primera vista podría sugerir una escena violenta. En la segunda imagen se observa una incidencia de

violencia física a la lejanía del ángulo de la cámara. Sucede la misma situación con la tercera imagen, sin en cambio, la complejidad aquí es la falta de nitidez de la imagen. En última instancia, la cuarta imagen sugiere el mismo escenario que la primera.

6.1.3 AIRTLAB

En último lugar, AIRTLAB es el conjunto de datos utilizado para la evaluación de los modelos, inclinado hacia la robustez de estos, derivado de la falta de balance que posee. Pues presenta una tendencia hacia la clase “Violencia”. De manera similar a los casos anteriores, se presentan muestras de escenas violentas en la *Figura 20*, mientras que las no violentas se ilustran en los ejemplos de la *Figura 21*.

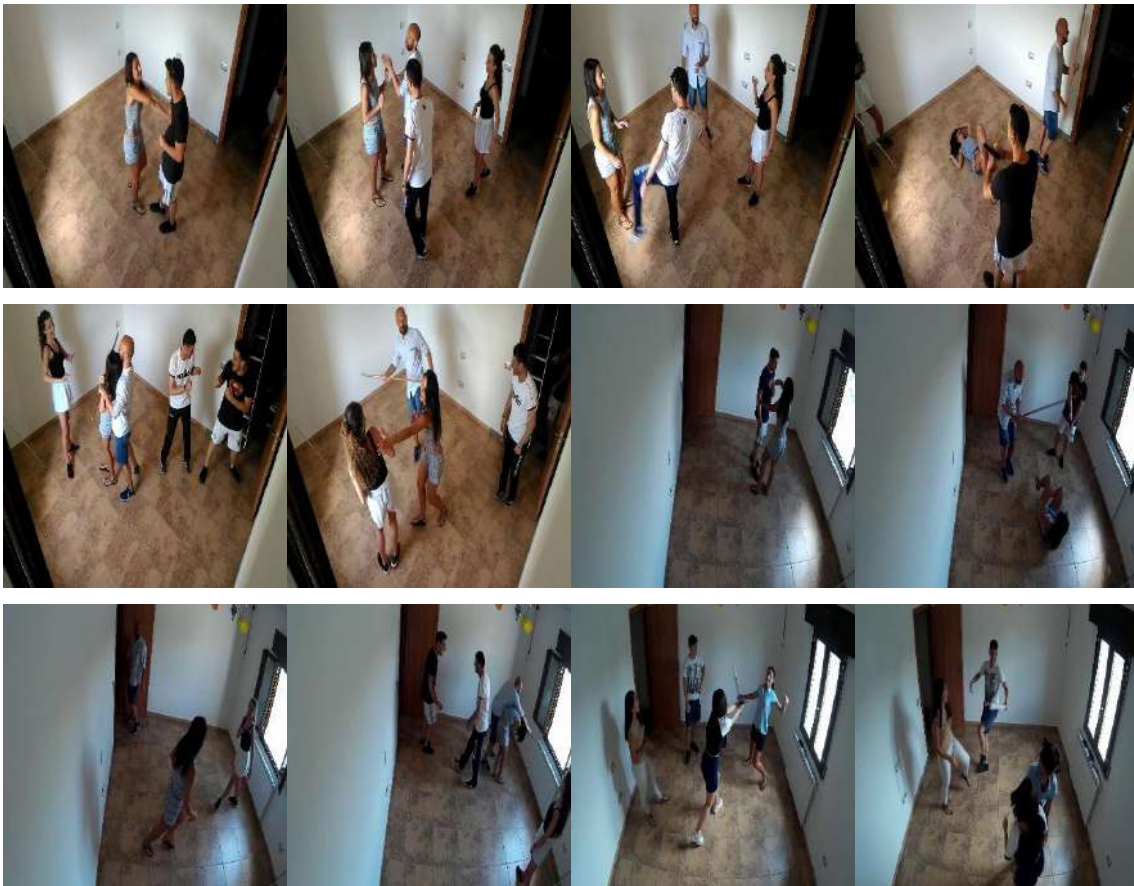


Figura 20. Escenas violentas del conjunto de datos AIRTLAB.



Figura 21. Escenas pacíficas del conjunto de datos AIRTLAB.

De igual manera, las imágenes que representan un reto de clasificación para este conjunto de datos se muestran en la *Figura 22*.

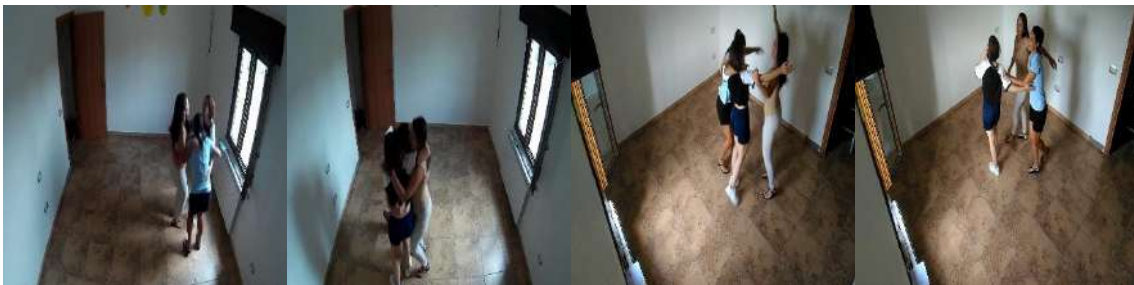


Figura 22. Ejemplos de imágenes del conjunto de datos AIRTLAB que presentan un desafío de clasificación.

Todas las imágenes presentadas en la *Figura 22* originalmente pertenecen a la clase de no violencia, pero suponen un reto en la detección al exponer actividades pacíficas enérgicas que pueden confundirse con escenarios violentos.

6.1.4 Extracción de frames

Cada uno de los conjuntos de datos se encuentran etiquetados bajo dos clases: “Violencia” y “No violencia” para el caso de RLVS y SCVD (Mohamed, Mohamad, & Abd El Massih, 2019) (Aremu, Li, & Alameeri, 2024), mientras que para AIRTLAB se hace uso de las etiquetas “Violento” y “No violento” (airtlab, 2020), aun así, los tres conjuntos de datos se clasifican de manera binaria entre incidentes violentos y escenarios pacíficos.

A través de la *Tabla 3* que se describe a continuación se dan a conocer las características generales de los conjuntos de datos, tales como la fuente de obtención, la cantidad de vídeos que dispone, el formato del contenido multimedia y bajo que licencia de distribución se mantienen.

Tabla 3. Características de los conjuntos de datos seleccionados.

Dataset	Volumen de datos	Formato	Licencia
AIRTLAB (GitHub)	350 videos etiquetados como “no violentos” y “violentos”.	.mp4	El dataset se publica libremente con fines educativos y de investigación. “The dataset is freely released for research and educational purposes.” (airtlab, 2020)
RLVS (Kaggle)	1,000 videos sobre violencia y 1,000 sobre escenarios pacíficos recopilados de YouTube.	.mp4	Se distribuye libremente por los autores siempre y cuando el uso del conjunto de datos se utilice con fines investigativos o de ingeniería (Mohamed, Mohamad, & Abd El Massih, 2019).
SCVD (Kaggle)	1,000 videos sobre violencia y 1,000 sobre escenarios pacíficos recopilados de YouTube.	.avi	CC BY-NC-SA 4.0

RLVS es el conjunto de datos utilizado para el entrenamiento de los tres modelos pre-entrenados, pues a diferencia de los otros conjuntos de datos es el que presenta un mayor balance entre la cantidad de videos clasificados como “Violentos” y aquellos clasificados como “No Violentos”, además no cuenta con subclasificaciones de acuerdo con el ángulo de la cámara de videovigilancia, parámetros de iluminación o capturas con diferentes tipos de cámara. En otras palabras, únicamente cuenta con dos carpetas denominadas como “Violencia” y “No violencia” donde cada una de ellas cuenta con 1,000 videos, dando un total de 2,000 videos. En breve, es la mejor opción para el entrenamiento de los modelos.

Por otra parte, SCVD se destina a la validación del entrenamiento previo de los modelos pre-entrenados poniendo a prueba las generalidades de cada modelo, y AIRTLAB es sometido para pruebas robustez ante escenarios no previstos y desbalanceados con la finalidad de observar la adaptabilidad de cada modelo ante estas situaciones. Esto por dos razones, donde la primera radica en la cantidad de videos correspondientes a las clases “Violencia” y “No violencia”, es decir, a diferencia de RLVS estos conjuntos de datos poseen una menor cantidad de videos válidos para el entrenamiento. La segunda radica en el balance a partir de la cantidad de *frames* obtenidos para cada clase correspondiente a cada conjunto de datos. Cabe mencionar que esta última razón deriva después de la extracción y obtención de *frames* para cada caso. En la *Tabla 4* se muestra el tipo de balance de los conjuntos de datos utilizados para evaluar los modelos.

Tabla 4. Balance de los conjuntos de datos en función del número de frames extraídos.

Dataset	Violencia	No violencia	Juicio
SCVD	3,690 frames (54%)	4,031 frames (46%)	Ligeramente desbalanceado hacia la clase Violencia, pero aceptable.
AIRTLAB	3,365 frames (67%)	1,648 frames (33%)	Desbalanceado hacia la clase Violencia.

6.2 Desempeño durante la fase de entrenamiento

La secuencia de entrenamiento de los modelos que aquí se demuestra fue aleatoria, por lo que no se rige bajo ningún tipo de preferencia sobre el rendimiento o arquitectura del modelo. Asimismo, se recalca el uso del conjunto de datos RLVS al presentar un balance adecuado entre clases. Donde la extracción de *frames* quedo de la siguiente manera: 29,337 *frames* de la clase “Violencia” y 27,511 *frames* de la clase “No Violencia”, dando un total de 56,848 *frames* para el entrenamiento de los modelos.

La única diferencia clave del entrenamiento consta en la dimensión de las imágenes de entrada para los modelos. Como se ha mencionado en líneas anteriores, para evitar propiciar una ventaja o desventaja en el rendimiento de los modelos con respecto a otro. Las dimensiones de extracción de los *frames* se estandarizaron a 640x640 píxeles, por lo que, para el entrenamiento de EfficientNetB0 y ResNet50 se redimensionaron a un tamaño de 224x224 píxeles. Mientras que se conservaron las dimensiones iniciales para el

entrenamiento de YOLOv8. Sin embargo, esta diferencia no perjudica o beneficia a algún modelo sobre otro.

Otro punto importante es la estandarización a 15 épocas en lotes de 32 imágenes para el entrenamiento de los modelos EfficientNetB0 y ResNet50. Por otro lado, el entrenamiento de YOLOv8 se manejó a un total de 10 épocas. Más adelante se dará a conocer el porqué de esta diferencia. Para todos los modelos se estableció el mecanismo de *Early Stopping* después de 3 épocas (o iteraciones) en las que no se presenta una mejora, es decir, el modelo ha comenzado a memorizar. Retornando y preservando al estado óptimo de aprendizaje.

Finalmente, cabe señalar que el entrenamiento se dividió en un 80% de entrenamiento y un 20% de evaluación a partir del conjunto de datos RLVS. Por lo que, se utilizaron 45,479 *frames* para el entrenamiento y 11,369 *frames* para la evaluación.

6.2.1 Hiperparámetros

Número de Épocas de los Modelos de Múltiples Etapas

EfficientNetB0 y ResNet50 son modelos de *Transfer Learning*, es decir, previamente fueron entrenados para la detección de otro tipo de fenómenos. Por lo que, ya cuentan con una inteligencia general que les permite identificar bordes, texturas y formas. Un entrenamiento después de ello está enfocado en la afinación (*fine-tuning*). En este caso, para aprender específicamente sobre la clasificación de incidencias de violencia física. Por lo tanto, no requiere de un elevado número de épocas para lograr aprender. Sumado a ello, 15 es un múltiplo de 3, brindándole tiempo suficiente de activación al mecanismo de *Early Stopping*.

Número de Épocas del Modelo de Una Etapa

Ultralytics posee algoritmos de optimización distintos y su documentación oficial sugiere que el modelo decida de manera autónoma sobre sus épocas. Para *fine tuning* se utilizan 10 épocas (Ultralytics, 2024).

Lotes (Batch size)

Las imágenes procesadas son arreglos multidimensionales por lo que ocupan altas cantidades de video. Google Colab es un entorno controlado electo por la falta de recurso y poder computacional propio. Al hacer uso de una membresía gratuita, la RAM y GPU que

este entorno proporciona, también se ven limitados, pero superan las características del equipo local. Así que para procesar lotes de 64 imágenes o más provocaría el colapso de la memoria de la tarjeta gráfica y lotes muy pequeños de alrededor de 16 imágenes o menos causarían un aprendizaje inestable. Motivos suficientes por los que se optó por un *Sweet Spot*, es decir, un punto de equilibrio entre ambos extremos. De esa manera, se definieron lotes de 32 imágenes.

Early Stopping

Por otra parte, para lograr optimizar el recurso computacional limitado es que se establece el mecanismo de *Early Stopping* (parada temprana). Un mecanismo capaz de regularizar el comportamiento del entrenamiento de un modelo deteniéndolo en automático en cuanto el rendimiento de aprendizaje se reduzca y, por ende, comience a memorizar (Pal, 2023).

Se estableció un parámetro de paciencia de 3 épocas para el *Early Stopping* con el objetivo de tolerar las fluctuaciones naturales de la función de pérdida, asegurando una tendencia real de *Overfitting* antes de abortar el entrenamiento prematuramente.

Regla 80/20

Es un estándar común utilizado en el entrenamiento de modelos de *Machine Learning* proveniente del principio de Pareto. Este señala que el 20% de las causas de un fenómeno repercute directamente en el 80% de los efectos. En el entrenamiento de un modelo este principio indica que el 20% del conocimiento posibilita el 80% de la comprensión de un tema en particular (Cabezudo, 2023).

6.2.2 Entrenamiento de EfficientNetB0

EfficientNetB0 utilizó 45,479 imágenes (80%) para su entrenamiento y 11,369 (20%) para una autoevaluación interna. En la función de pérdida se cuantifica la inexactitud de las predicciones del modelo durante el entrenamiento. Lo deseado en *Machine Learning* es que el valor de este indicador tienda a cero. En este sentido, de acuerdo con la *Figura 23* tanto el entrenamiento como la evaluación interna muestran una tendencia a la baja. Esto evidencia que el modelo logra comprender las características intrínsecas de la violencia física.

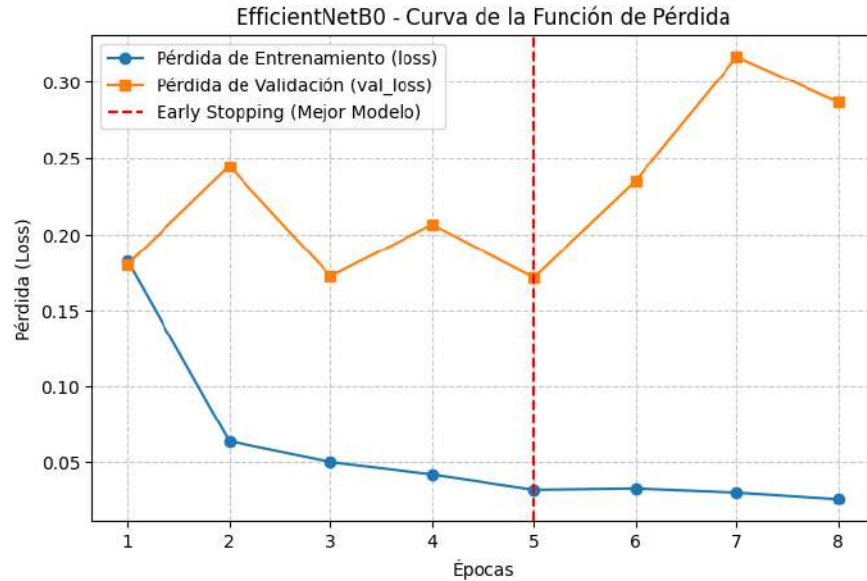


Figura 23. Curva de la función de pérdida del entrenamiento de EfficientNetB0.

Sin embargo, a partir de la sexta época la pérdida en la evaluación ha incrementado de forma errática. Y continua de esa manera en las épocas posteriores, por lo que el modelo ha dejado de aprender patrones generales y ha comenzado a memorizar (*Overfitting*) los datos de entrenamiento, degradando su capacidad de generalización ante datos nuevos. Por tal razón, después de tres épocas que no presentaron una mejora, se activó el mecanismo de *Early Stopping* y abortó el entrenamiento en la octava época. De esta manera, se descartaron los pesos con *Overfitting* y se restauró el estado de la quinta época del modelo.

6.2.3 Entrenamiento de ResNet50

Al igual que EfficientNetB0, ResNet50 aplicó la regla del 80/20 durante su entrenamiento. En la *Figura 24* ResNet50 demuestra una fluctuación instantánea en la función de pérdida de la validación que persiste en el resto de las épocas. Por lo tanto, el mecanismo de *Early Stopping* se activa después de 3 épocas de empeoramiento. Es decir, el modelo demostró una inclinación temprana hacia el *Overfitting*. De este modo, se restauró el estado de la primera época del modelo.

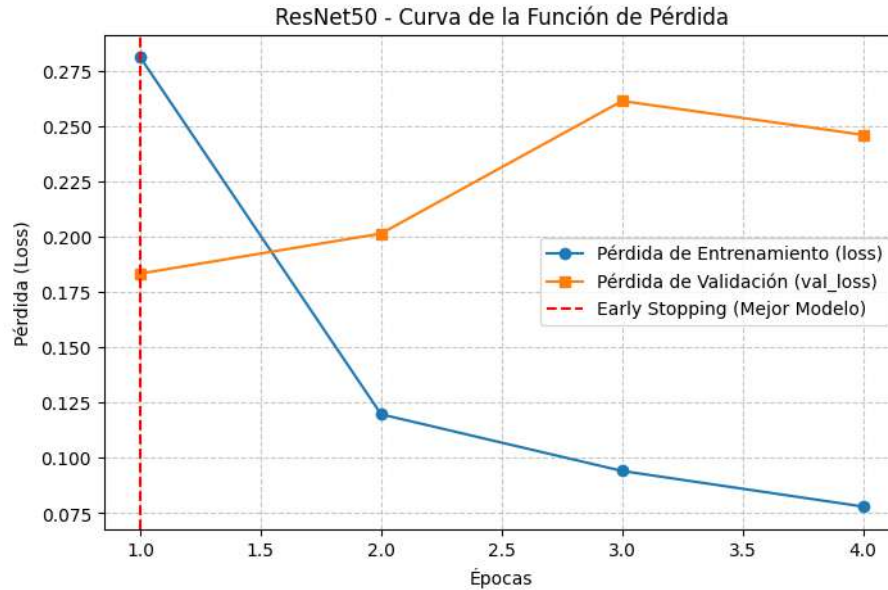


Figura 24. Curva de la función de pérdida del entrenamiento de ResNet50.

6.2.4 Entrenamiento de YOLOv8

De la misma manera que los modelos de múltiples etapas, YOLOv8 llevo a cabo su entrenamiento bajo la regla de 80/20. A diferencia de los dos modelos anteriores, YOLOv8 se configuró para un ciclo cerrado de 10 épocas, las necesarias y suficientes para entrenamientos de afinación. Además, este modelo cuenta con su propio mecanismo de *Early Stopping*, conocido como punto máximo de rendimiento (best.pt) derivado de sus algoritmos de optimización.

En el entrenamiento de YOLOv8 tanto la función de pérdida del entrenamiento como la función de pérdida de la evaluación (validación) descienden de forma armónica hasta la novena época. (ver *Figura 25*). En la décima y última época, a primera vista podría deducirse que las funciones de pérdida descendieron homogéneamente, pero no es el caso. La función de pérdida de la evaluación presenta una ligera fluctuación que la eleva. Por lo tanto, YOLOv8 detecta dicha fluctuación, marcándola como el inicio de *Overfitting*. Y restaura el estado de la novena época. A partir de un mecanismo interno que rastrea las métricas de entrenamiento con el objetivo de extraer y guardar los pesos correspondientes al punto de máximo rendimiento en la validación.

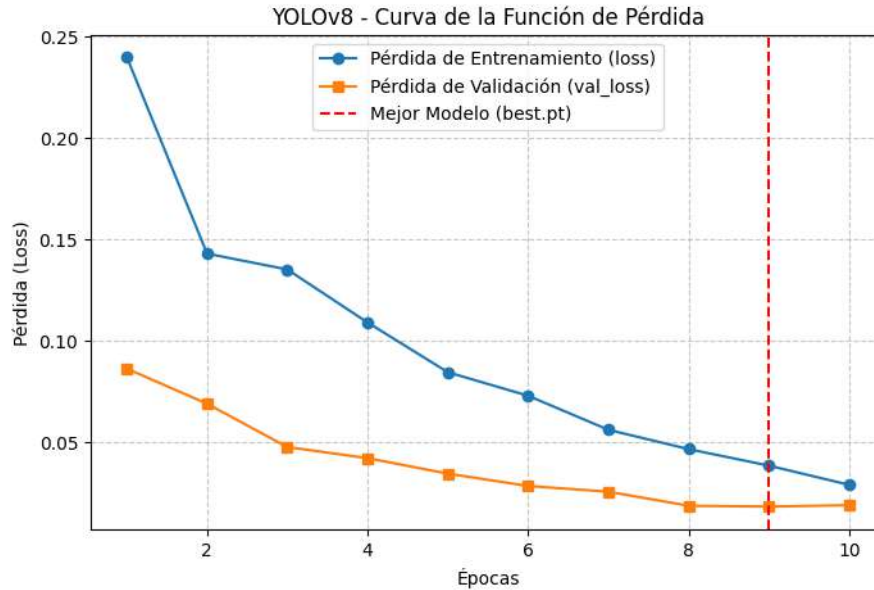


Figura 25. Curva de la función de pérdida del entrenamiento de YOLOv8.

6.2.5 Exactitud del entrenamiento de los modelos

Finalmente se demuestra una comparativa entre la exactitud (*Accuracy*) obtenida por cada modelo durante la etapa de entrenamiento. Donde de forma general, los modelos exhiben resultados alentadores al rebasar un porcentaje del 85%. (ver *Tabla 5*).

Tabla 5. Exactitud del entrenamiento de los modelos.

Modelo	Exactitud (Accuracy)	Épocas
EfficientNetB0	98.91%	8
ResNet50	88.87%	4
YOLOv8	99.40%	10

Tal resultado es sinónimo de una extracción exitosa sobre las características de violencia en situaciones controladas. El dimensionamiento de las imágenes para cada modelo y la cantidad de *frames* extraídos y su distribución según la regla del 80/20, desempeñó un papel significativo en la obtención de los resultados.

6.3 Desempeño durante la fase de evaluación

El desplazamiento de dominio (*Domain Shift*) consiste en la diferencia representativa entre la distribución de datos del conjunto de datos de origen utilizado para el entrenamiento y la del conjunto de datos de destino utilizado para la evaluación (ApX Machine Learning, 2026). En otras palabras, para los modelos las imágenes de SCVD difieren de las de RLVS. Esto se debe al cambio en los escenarios, tales como la iluminación, los ángulos, el tipo de cámara y diferentes variables aplicadas en las imágenes de RLVS. A partir de este fenómeno, los modelos fueron sometidos a pruebas de generalización con el conjunto de datos SCVD y a pruebas de robustez con el conjunto de datos de AIRTLAB.

6.3.1 Pruebas de Generalización

Para las pruebas de generalización se usó el conjunto de datos SCVD, dado que presentaba un desequilibrio de clases aceptable. Donde, en el proceso de extracción, se reportaron un total de 4,690 frames para la clase “Violencia” y 4,031 para la clase “No Violencia”. Así se extrajeron 8,721 *frames* del conjunto de datos SCVD para las pruebas de generalización en la primera fase de evaluación de los modelos.

Esta fase está orientada a demostrar la exactitud real de los modelos frente a escenarios desconocidos tomando como base el entrenamiento previo. En la *Tabla 6* se presenta la exactitud global que los modelos han alcanzado en la evaluación de las pruebas de generalización.

Tabla 6. Exactitud de la evaluación de los modelos frente a la prueba de generalización.

Modelo	Exactitud (Accuracy)
EfficientNetB0	57.93%
ResNet50	55.88%
YOLOv8	57.38%

La detección de los modelos frente a incidencias de violencia física se degradó en al menos un 38.66%, una cifra ineludible. Seguidamente se exhibe una comparativa entre la exactitud obtenida durante el entrenamiento y la que aquí se presenta. (ver *Tabla 7*).

Tabla 7. Comparativa entre la exactitud del entrenamiento contra la evaluación de generalización.

Modelo	Exactitud de Entrenamiento	Exactitud de Evaluación
EfficientNetB0	98.91%	57.93%
ResNet50	88.87%	55.88%
YOLOv8	99.40%	57.38%

A pesar de la implementación del mecanismo de *Early Stopping* los modelos se mostraron incapaces de generalizar. Esto probablemente sucedió debido a la identificación del escenario mostrado en las cámaras del conjunto de datos utilizado para el entrenamiento, en lugar de detectar la acción de violencia en sí misma. Al cambiar el ángulo, la iluminación o el escenario del conjunto de datos, el rendimiento del modelo se degradó ante el fenómeno de desplazamiento de dominio.

De forma numérica, EfficientNetB0 demostró un 15.41% de detección de incidentes no violentos de manera certera, un 30.81% de incidentes diagnosticados como falsos positivos, es decir, casos de “No Violencia” fueron detectados como casos de “Violencia”. Un 11.26% de detección de falsos negativos, en otras palabras, incidentes categorizados como “Violencia” pasaron desapercibidos por el modelo. Por último, un 42.51% de detección de incidentes puramente violentos. Para obtener la exactitud final del modelo, se suman únicamente los porcentajes correspondientes a la detección de incidencias violentas y no violentas, sin presentar falsos positivos. Esto se resume en la suma de los verdaderos positivos más los verdaderos negativos. Finalmente, este porcentaje es del 57.93%. (ver *Figura 26*).

En la *Tabla 8* se observa un marcado desbalance en la capacidad de recuperación (*Recall*) de EfficientNetB0. Esto se demuestra en la correcta identificación del 79% de los casos clasificados como violentos. Sin embargo, posteriormente registra una caída drástica al 33%. Lo cual indica que el modelo, ante la incertidumbre generada por el desplazamiento de dominio, opta por clasificar la mayoría de las muestras como casos violentos, lo que se traduce en una precisión global moderada, pero con una inclinación hacia falsos positivos.

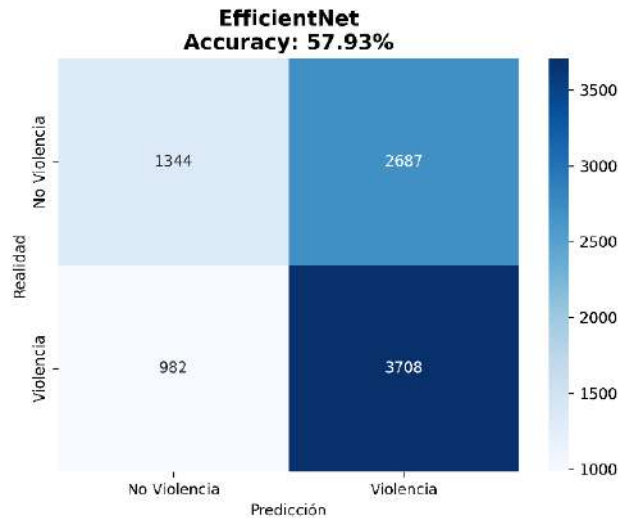


Figura 26. Matriz de confusión de la evaluación de EfficientNetB0 en las pruebas de generalización.

Tabla 8. Métricas de la evaluación de generalización para EfficientNetB0.

Clase	Accuracy	Recall	F1-Score
No violencia	0.58	0.33	0.42
Violencia	0.58	0.79	0.67

ResNet50 reportó un 55.88% de exactitud en la detección de incidencias violentas y no violentas, un 2.05% menos que EfficientNetB0. Desglosando cada uno de los casos expuestos en la matriz de confusión, el 35.22% corresponde a incidencias no violentas tanto en la predicción como en la realidad; un 10.99% de incidentes no violentos fueron categorizados erróneamente como incidentes violentos; un 33.12% de incidentes violentos en el entorno real pasaron ante el modelo como incidentes no violentos; y, finalmente, un 20.65% corresponde a la detección de incidentes violentos. (ver Figura 27).

Por otro lado, en la Tabla 9 se revela la incapacidad del modelo pre-entrenado para generalizar las características de violencia en el nuevo dominio de SCVD. El dato más crítico que presenta es el *Recall* de violencia de apenas 38%, lo cual significa que el modelo ignora el 62% de las amenazas reales, clasificándolas incorrectamente como “No Violencia”. Aun así, demuestra una métrica de “No Violencia” aceptable, siendo esta del 76%, este comportamiento conservador lo vuelve ineficaz para tareas de seguridad, donde la prioridad es no omitir situaciones de riesgo, es decir, minimizar falsos negativos.

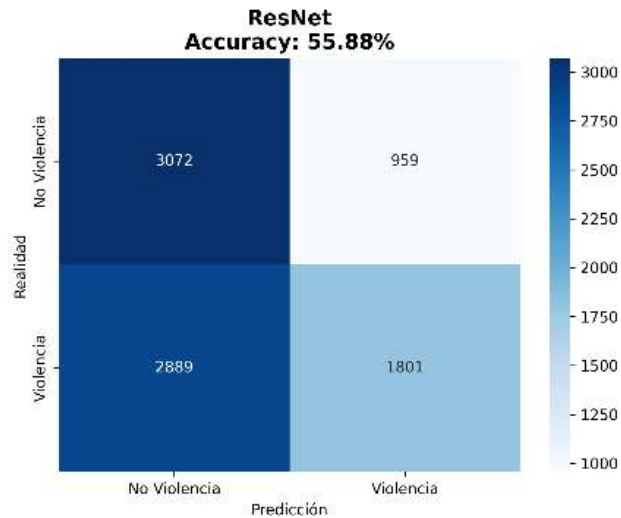


Figura 27. Matriz de confusión de la evaluación de ResNet50 en las pruebas de generalización.

Tabla 9. Métricas de la evaluación de generalización para ResNet50.

Clase	Accuracy	Recall	F1-Score
No violencia	0.52	0.76	0.61
Violencia	0.65	0.38	0.48

Ahora bien, YOLOv8 expone una exactitud del 57.38% teniendo una exactitud del 0.55% menor en relación con EfficientNetB0, mientras que al lado de ResNet50 obtuvo una exactitud con un incremento del 1.5%. Para cada uno de los cuadrantes de la matriz de confusión se consiguió un 7.85% en la detección de incidencias categorizadas como “No Violencia”, un 38.36% de incidentes no violentos fueron detectados como el caso contrario, un 4.25% de incidentes violentos fueron detectados bajo la etiqueta de “No Violencia”, y finalmente, un 49.52% de incidentes violentos fueron detectados como tales. (ver *Figura 28*).

Por otro lado, YOLOv8 en comparación con los dos modelos anteriores demuestra un perfil con mayor afin en la detección de violencia, esto se respalda con el resultado de 92% obtenido en su *Recall*. Evidenciando una robustez superior para capturar la clase de interés, a pesar de las diferencias de iluminación y de entorno del conjunto de datos SCVD. Sin embargo, esto conllevó un compromiso (*trade-off*) significativo: su especificidad colapsó al 17% de “No Violencia”. En otras palabras, YOLOv8 prioriza la seguridad, garantizando la

detección de casi todos los eventos violentos a costa de generar un volumen considerable de alertas falsas. (ver *Tabla 10*).

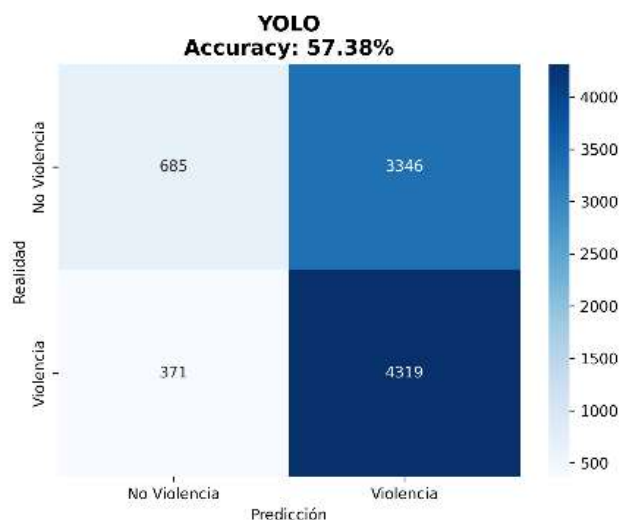


Figura 28. Matriz de confusión de la evaluación de YOLOv8 en las pruebas de generalización.

Tabla 10. Métricas de la evaluación de generalización para YOLOv8.

Clase	Accuracy	Recall	F1-Score
No violencia	0.65	0.17	0.27
Violencia	0.56	0.92	0.70

Considerando las matrices de confusión de cada modelo, el desplazamiento por dominio se enfatiza más en EfficientNetB0 y en YOLOv8. Esto quiere decir que ambos modelos presentaron más falsos positivos que ResNet50, convirtiéndolos en modelos con sesgo o de alta sensibilidad. Por otro lado, ResNet50 se demostró conservador al ignorar incidencias de violencia física. (ver *Tabla 11*).

Por ahora, pareciera que el modelo óptimo para la detección de incidencias violentas es EfficientNetB0, demostrando una exactitud mayor en comparación con los otros modelos, YOLOv8 pasaría a estar en segundo lugar y ResNet50 en el tercer lugar. Sin embargo, dicha conclusión es prematura y aún no constituye prueba de latencia.

Tabla 11. Comparativa entre los modelos a partir de sus matrices de confusión.

	EfficientNetB0	ResNet50	YOLOv8
Verdaderos Positivos	15.41 %	35.22 %	7.85 %
Falsos Positivos	30.81 %	10.99 %	38.36 %
Falsos Negativos	11.26 %	33.12 %	4.25 %
Verdaderos Negativos	42.51 %	20.65 %	49.52 %
Exactitud (Accuracy)	57.93 %	55.88 %	57.38 %
Debilidad	Falsos Positivos	Falsos Negativos	Falsos Positivos
Conclusión	Estabilidad aparente	Conservador, omite incidencias de violencia real	Hipersensible, confunde eventos pacíficos con incidentes violentos

6.3.2 Análisis del área bajo la curva

Para complementar la evaluación de aciertos y errores de las matrices de confusión, se analizó el Área Bajo la Curva ROC (AUC). Ya que esto permite evaluar el nivel de capacidad de los modelos en la separación matemática de dos clases, sin depender de un umbral de decisión fijo.

La línea intermedia punteada que se encuentra de esquina a esquina y divide la gráfica a la mitad, representa un *AUC* de 0.5, el cual equivale al peor escenario posible. Las líneas que se encuentran por encima de ella representan que el modelo ha aprendido, y la que se encuentra por debajo demuestra una confusión sistemática. (ver *Figura 29*).

Con esto entendido, EfficientNetB0 y YOLOv8 se encuentran marginalmente por encima. Esto indica que, a pesar de haber sufrido una caída crítica en su capacidad de generalización derivada del cambio de contexto, aún conservan una ligera capacidad predictiva real.

Por el contrario, ResNet50 cayó por debajo del límite. Esto recae en la detección conservadora que demostró, pues matemáticamente un *AUC* inferior a 0.5 señala que el modelo está invirtiendo las predicciones sistemáticamente.

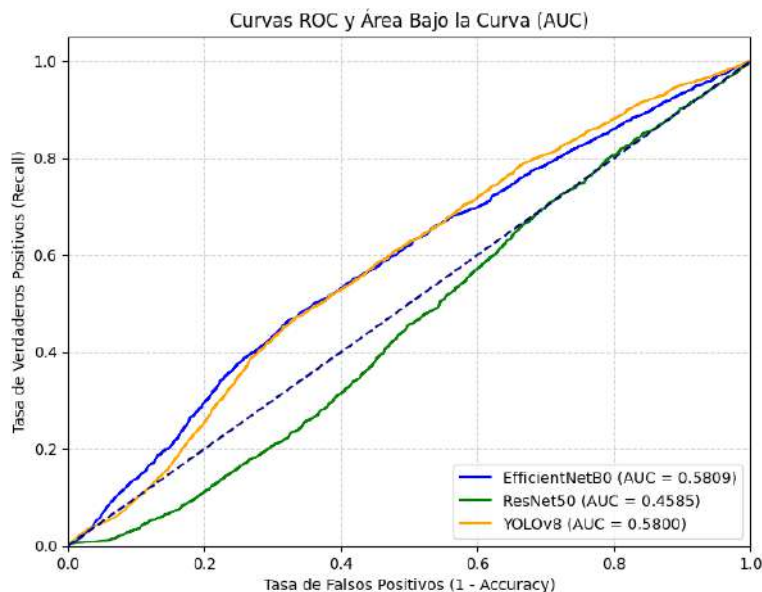


Figura 29. Área bajo la curva de la evaluación para pruebas de generalización.

6.4 Pruebas de Robustez

Para las pruebas de robustez se usó el conjunto de datos AIRTLAB al presentar el mayor desbalance de clases. Donde en el proceso de extracción se adquirieron 6,696 *frames* para la clase “Violencia” (3,359 de la primera cámara y 3,337 de la segunda cámara) y 3,287 *frames* para la clase de “No Violencia” (1,643 de la primera cámara y 1,644 de la segunda cámara). Así se extrajeron 9,983 *frames* del conjunto de datos AIRTLAB para las pruebas de robustez en la segunda fase de evaluación de los modelos.

Esta fase esta orienta a demostrar la exactitud real de los modelos frente a escenarios desconocidos tomando como base el entrenamiento previo. Finalmente, se contrastará con lo conseguido en la prueba de generalización. En la *Tabla 12* se indica la exactitud global que los modelos han conseguido durante la evaluación de las pruebas de robustez.

Tabla 12. Exactitud de la evaluación de los modelos frente a la prueba de robustez.

Modelo	Exactitud (Accuracy)
EfficientNetB0	67.29%
ResNet50	41.09%
YOLOv8	69.44%

En esta ocasión, la detección de los modelos frente a incidencias de violencia física se degradó en un 36.45%, una cifra menor con respecto a lo conseguido en las pruebas de generalización. Probablemente porque los escenarios son más claros y controlados dentro del conjunto de datos AIRTLAB. (ver *Tabla 13*).

Tabla 13. Comparativa sobre la exactitud en las diferentes etapas.

Modelo	Exactitud de Entrenamiento	Exactitud de Evaluación (Generalización)	Exactitud de Evaluación (Robustez)
EfficientNetB0	98.91%	57.93%	67.29%
ResNet50	88.87%	55.88%	41.09%
YOLOv8	99.40%	57.38%	69.44%

Los resultados de la prueba con AIRTLAB parecen más alentadores. EfficientNetB0 tuvo un incremento del 9.36%; ResNet50 exhibió una degradación continua a pesar del cambio en los datos, con una nueva caída del 14.79%; y YOLOv8 incrementó su capacidad de detección en un 12.06%. Todas estas cifras, en comparación con las obtenidas en la prueba de generalización.

Ahora, considerando únicamente el entrenamiento y la evaluación para las pruebas de robustez. EfficientNetB0 presentó una caída del 31.62%, ResNet50 del 47.78% y YOLOv8 del 29.96%.

Para EfficientNetB0 se obtuvo una exactitud del 67.29%, al sumar los porcentajes obtenidos en los cuadrantes de Verdaderos Positivos y de Verdaderos Negativos. El desglose para cada cuadrante es el siguiente: el modelo consiguió un 7.46 % de detección en escenarios pacíficos, un 25.41% de detección falsa en escenarios violentos, es decir, confundió un evento pacífico con uno violento. Obtuvo un 7.30% de falsos positivos, donde incidentes violentos se clasificaron como no violentos, y, por último, se detectó un 59.82% de violencia. (ver *Figura 30*).

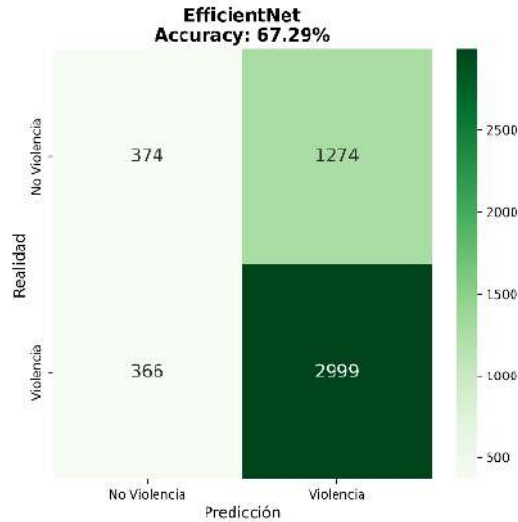


Figura 30. Matriz de confusión de la evaluación de *EfficientNetB0* en las pruebas de robustez.

EfficientNetB0 demostró una mejor capacidad de generalización, logrando recuperar el 89% de los eventos clasificados como violentos. Sin embargo, su incapacidad para distinguir correctamente la minoritaria clasificada como “No Violencia” (23%) sugiere que el modelo no ha aprendido características determinantes de la acción, sino que se apoya de un sesgo a la clase mayoritaria. (ver *Tabla 14*).

Tabla 14. Métricas de la evaluación de robustez para *EfficientNetB0*.

Clase	Accuracy	Recall	F1-Score
No violencia	0.51	0.23	0.31
Violencia	0.70	0.89	0.79

ResNet50, por su parte obtuvo una exactitud total de 41.09%. Reportando un 27.40% de detección de eventos pacíficos (verdaderos positivos), un 5.46% de confusión donde no existieron incidentes violentos, pero fueron clasificados como tal. Obtuvo un 53.44% de casos violentos detectados como eventos pacíficos, y obtuvo un 13.68% de detección en eventos violentos. (ver *Figura 31*).

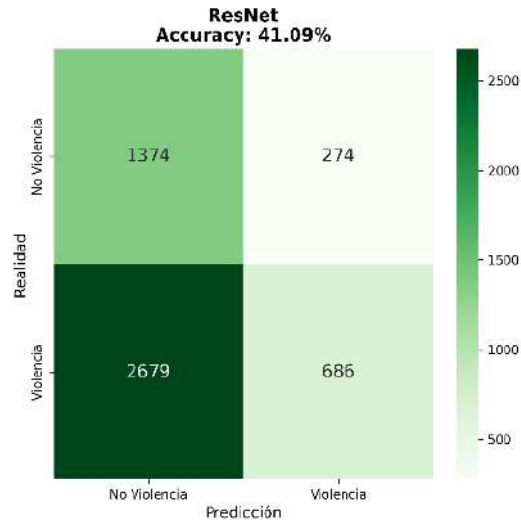


Figura 31. Matriz de confusión de la evaluación de ResNet50 en las pruebas de robustez.

Este modelo presentó una degradación severa de su capacidad predictiva ante AIRTLAB, al ser incapaz de detectar un porcentaje crítico de violencia, que fue del 80%. Este resultado confirma que ResNet50 fue altamente susceptible al sobreajuste de fondo (*Background Overfitting*) al ser entrenado con un solo dominio, convirtiéndose en un modelo ineficiente en entornos de seguridad dinámica. (ver *Tabla 15*).

Tabla 15. Métricas de la evaluación de robustez para ResNet50.

Clase	Accuracy	Recall	F1-Score
No violencia	0.34	0.83	0.48
Violencia	0.71	0.20	0.32

Finalmente, YOLOv8 alcanzó un 69.44% de exactitud en la detección de eventos pacíficos e incidencias violentas. Donde reporta un 3.35% de detección en escenarios no violentos, un 29.52% de detección en escenarios no violentos que se clasificaron como su parte contraria, un 1.03% de incidentes violentos confundidos como escenarios pacíficos, y finalmente, un 66.08% de detección de incidentes violentos. (ver *Figura 32*).

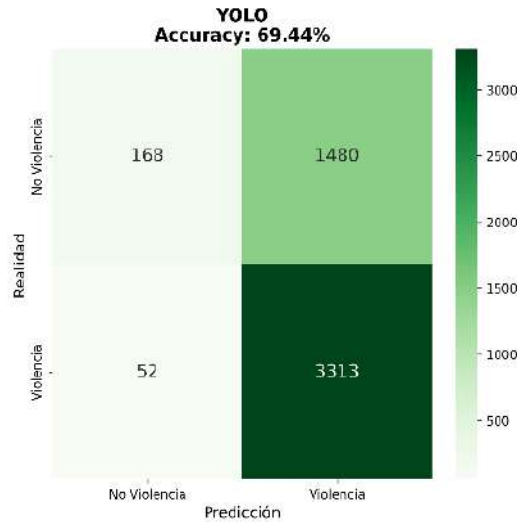


Figura 32. Matriz de confusión de la evaluación de YOLOv8 en las pruebas de robustez.

De manera global, ResNet50 colapso al considerar una significativa cantidad de falsos negativos, EfficientNetB0 no demostró haber aprendido características determinantes en la detección de violencia y, finalmente, YOLOv8 reveló un mejor rendimiento en comparación con los dos modelos anteriores teniendo como debilidad la sensibilidad ante la detección de agresiones. De acuerdo con el objetivo principal del proyecto, YOLOv8 es el modelo que lidera la detección de violencia. (ver *Tabla 16*).

Tabla 16. Métricas de la evaluación de robustez para YOLOv8.

Clase	Accuracy	Recall	F1-Score
No violencia	0.76	0.10	0.18
Violencia	0.69	0.98	0.81

Aun sin realizar el testeo de la velocidad de detección entre los modelos, podría sugerirse que ninguno de los modelos tiene un sólido fuerte. Donde, a primera vista, dos de ellos se inclinan por una alta cantidad de falsos positivos y uno de ellos por falsos positivos. Es decir, EfficientNetB0 y YOLOv8 sobreestiman los casos de incidentes de violencia, mientras ResNet50 pasa por alto incidentes violentos. Al menos en este caso. (ver *Tabla 17*).

Los tres modelos demuestran una debilidad significativa. Sin embargo, aún no es posible realizar conclusiones, derivado de la falta de la prueba de velocidad de detección. No

solo se desea un modelo que sea capaz de detectar incidentes violentos, sino también uno que sea capaz de realizarlo en tiempo real con la menor pérdida de imágenes posible.

Tabla 17. Comparativa entre los modelos a partir de sus matrices de confusión.

	EfficientNetB0	ResNet50	YOLOv8
Verdaderos Positivos	7.46%	27.40%	3.35%
Falsos Positivos	25.41%	5.46%	29.52%
Falsos Negativos	7.30%	53.44%	1.03%
Verdaderos Negativos	59.82%	13.68%	66.08%
<i>Accuracy</i>	67.29%	41.09%	69.44%
Debilidad	Falsos Positivos	Falsos Negativos	Falsos Positivos
Conclusión	Estabilidad aparente	Omite incidencias de violencia real	Confunde eventos pacíficos con incidentes violentos

Con la finalidad de respaldar lo analizado a partir de las métricas (*Accuracy*, *Recall*, *F1-Score*) conseguidas durante la evaluación en la prueba de robustez de los modelos, se presenta a continuación, la gráfica correspondiente a el Área Bajo la Curva ROC (AUC). Para evaluar, así, el nivel de capacidad de los modelos para la separación matemática de dos clases, sin depender de un umbral de decisión fijo.

En la *Figura 33* se demuestra que los tres modelos, a pesar de presentar una caída porcentual considerable, mantienen una capacidad predictiva real. Para este caso, ResNet50 no presenta una confusión sistemática y el puntaje de AUC de EfficientNetB0 y YOLOv8 incrementó respectivamente, posicionando a YOLOv8 como el modelo con un mejor aprendizaje, teniendo una puntuación de 0.7101, un valor cercano a 1.

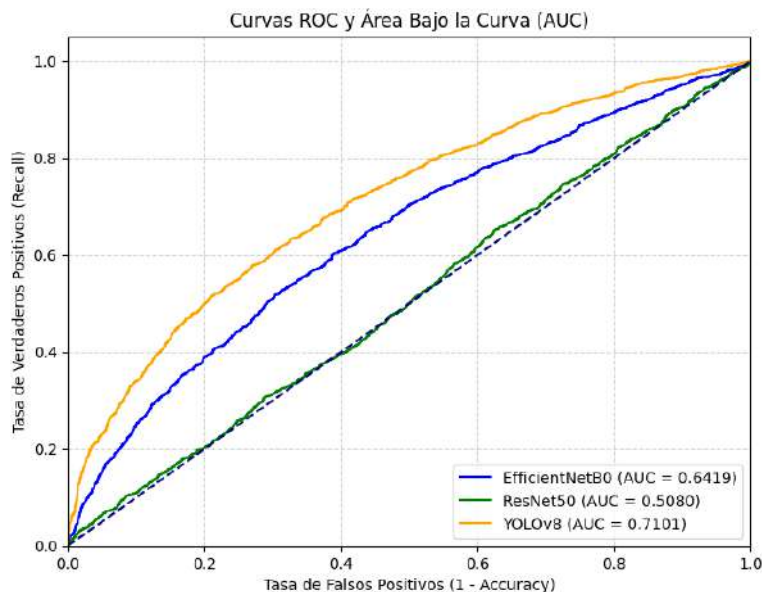


Figura 33. Área bajo la curva de la evaluación para pruebas de robustez.

6.5 Evaluación de eficiencia computacional

Un primer parámetro para considerar al implementar un sistema de videovigilancia orientado a la detección o al reconocimiento es la exactitud en la detección de determinada acción. Que, para el presente, consiste en la detección de incidencias categorizadas como violentas, es decir, capturar agravios físicos entre personas dentro de entornos controlados como oficinas, cuartos o similares. El segundo parámetro son los FPS (de sus siglas, Fotogramas Por Segundo), dicho de otra manera, la velocidad con la que se realiza la captura de imágenes de video, evitando una pérdida significativa de estas. En caso contrario, podría resultar perjudicial. Por ejemplo, perder las imágenes donde se comete la incidencia física o el acto violento sería crítico porque un incidente violento ya no podría evidenciarse como tal.

Ahora, para entenderlo mejor, un video se encuentra compuesto por una serie de imágenes que al moverse generan la ilusión de movimiento, así como una larga sucesión de puntos conforman una línea. A mayor tasa de fotogramas, más fluido se muestra el video. Por ende, a menor tasa de fotogramas, menor fluidez en el video (Argüello, 2023).

En la *Figura 34* cada uno de los círculos representa un fotograma. Donde hay más círculos, se forma algo parecido a una línea con relleno, un objeto que, gracias al conjunto

de círculos, pudo tener continuidad. Por otra parte, donde existe una menor cantidad de círculos, se pierde la continuidad y estos pasan a verse como simples círculos separados cada vez con mayor distancia.

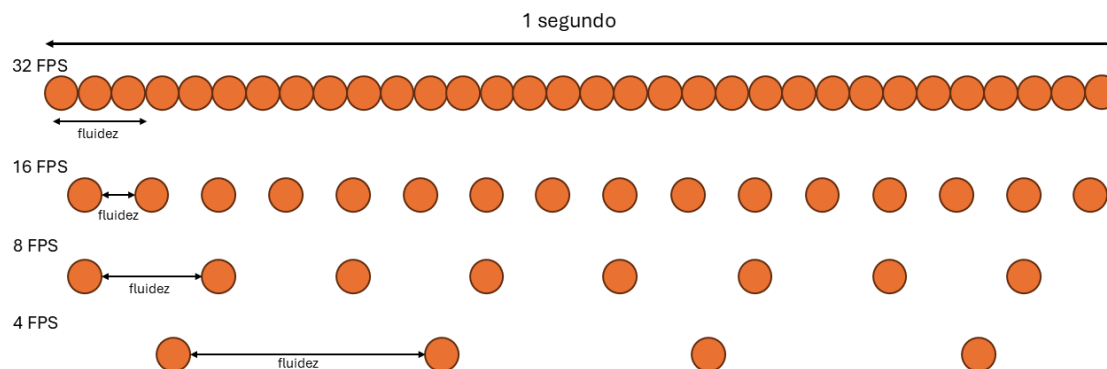


Figura 34. Representación de los FPS.

La figura anterior es una analogía a la fluidez de video con respecto a la cantidad de fotogramas que este contiene. A mayor número de imágenes (*frames*, fotogramas) se podrá visualizar un video más suave y con continuidad.

Una vez mencionado lo anterior, es indispensable que un sistema de videovigilancia mantenga fluidez. Los FPS en este rubro son un parámetro que fortalece la detección en tiempo real ante incidencias de violencia física y evita la pérdida de datos de clasificación importantes. Dicho de otra manera, el modelo pre-entrenado más eficiente para la detección de incidencias de violencia física deberá ser preciso y veloz (poseer la mayor tasa de FPS). Por lo que a continuación se demuestra gráficamente la velocidad demostrada por cada modelo en la prueba de eficiencia. (ver *Figura 35*).

Una vez que los modelos fueron entrenados y evaluados considerando generalidad y robustez, se les aplicó una prueba de velocidad para evaluar la detección de incidentes violentos. De acuerdo con la gráfica de barras, EfficientNetB0 y ResNet50 muestran una velocidad de detección considerablemente baja ante incidencias de violencia física, pues con mucha frecuencia estas incidencias se cometen con rapidez. Ambos modelos demostraron exactitud al ser clasificadores. Sin embargo, en la actualidad, en términos de velocidad, no resultan viables para la detección en tiempo real.

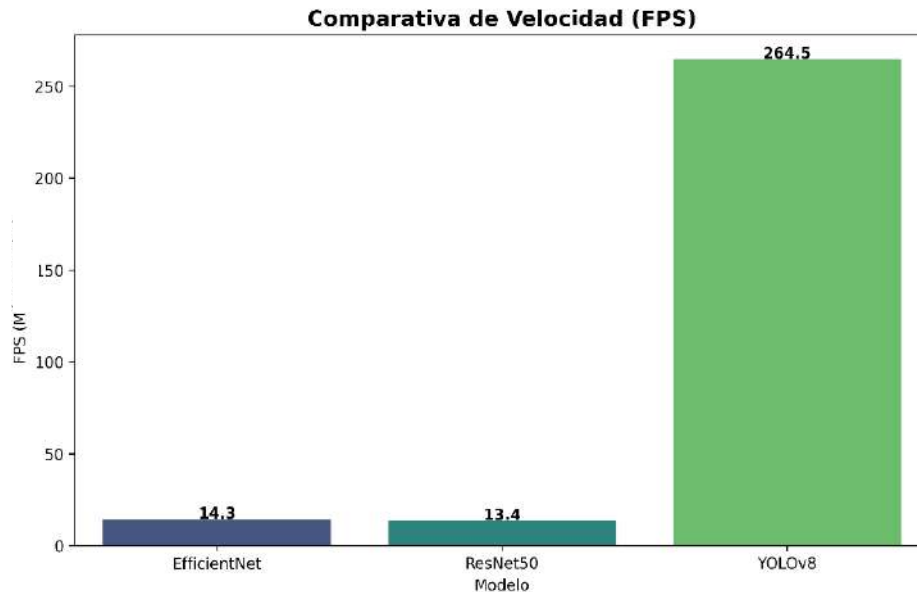


Figura 35. Comparativa de velocidad (FPS).

Por el contrario, YOLOv8 obtuvo una puntuación que supera incluso el promedio de fotogramas por segundo para la detección de incidencias en tiempo real (>30 FPS); superó por aproximadamente 19 veces a EfficientNetB0 y ResNet50. Es decir, su velocidad es extrema; incluso supera las necesidades del presente. Convirtiéndolo en el modelo más apto, considerando que para el proyecto se busca aquel modelo que mejor relación entre exactitud de detección y velocidad en tiempo real ofrezca.

En la *Tabla 18* que se muestra a continuación se presenta el tiempo en milisegundos que le tomó a cada modelo analizar por *frame* (*ms/frame*).

Tabla 18. Comparativa de latencia y velocidad de inferencia.

	EfficientNetB0	ResNet50	YOLOv8
Latencia	70.07 ms/frame	74.46 ms/frame	3.78 ms/frame
Velocidad	14.3 FPS	13.4 FPS	264.5 FPS

Contemplando los resultados demostrados en la gráfica de barras y en la *Tabla 18* sobre la velocidad de cada modelo pre-entrenado se revela que existe una diferencia significativa. Mientras que EfficientNetB0 y ResNet50 promedian entre 14.3 y 13.4 FPS, YOLOv8 supera los 260 FPS, siendo casi 19 veces más rápido. Esta brecha se justifica en

base a su diseño, ResNet50 y EfficientNetB0 fueron diseñados como clasificadores densos de múltiples etapas que procesan la imagen a través de bloques residuales complejos, lo que genera cuellos de botella computacionales (Sandaruwan, 2024). Por el contrario, YOLOv8 es un modelo de una sola etapa que realiza la predicción sobre toda la imagen en una única pasada de la red, minimizando drásticamente la latencia computacional (Silva Gúzman, 2020).

6.6 Síntesis y comparativa global

A pesar de que los modelos presentaron porcentajes significativos durante su entrenamiento, estos se deterioraron en la evaluación de las pruebas de robustez y, sobre todo, en las de generalización (Elgendy, 2025). Mostrando a EfficientNetB0 y YOLOv8 como modelos hipersensibles frente a incidencias de violencia física, mientras que a ResNet50 se le exhibió como un modelo conservador, es decir, ignora las incidencias de violencia física (Radečić, 2026). Además, de acuerdo con la prueba de eficiencia computacional, este modelo es capaz de procesar pobremente a 13.4 fotogramas por segundo. Siendo así un modelo que no presenta la mejor relación entre la eficacia y la eficiencia.

Por otro lado, EfficientNetB0 además de ser hipersensible, demuestra una tasa de 14.3 fotogramas por segundo. Siendo mejor que ResNet50, pero igualmente inviable en la detección en tiempo real. Por último, YOLOv8 mostró una baja capacidad de detección en la prueba de generalización, pero se recuperó en la prueba de robustez. Asimismo, obtuvo la mejor tasa de fotogramas por segundo entre los modelos de múltiples etapas.

De esta forma, se concluye que a pesar de la hipersensibilidad de YOLOv8 frente a incidencias de violencia física. Posee la mejor relación entre la eficacia y la eficiencia, al evidenciar una velocidad de 264.5 fotogramas por segundo.

La justificación de este análisis reside en los resultados obtenidos frente al entrenamiento en las pruebas de generalización y robustez, donde los modelos cayeron en su exactitud de entrenamiento, teniendo múltiples causas, como el sobreajuste al fondo (Background Overfitting); los modelos perdieron su capacidad discriminativa al memorizar los escenarios de los datos del dominio de origen y no el contexto del fenómeno por clasificar (Radečić, 2026). Por otra parte, la heterogeneidad entre los datos de los conjuntos de datos,

mientras RLVS exhibe escenas claras y deportivas en su mayoría en las que se muestran a una sola o pocas personas en los escenarios pacíficos, SCVD y AIRTLAB presentan escenarios pacíficos fuera del deporte, con imágenes menos claras, contextos diferentes e incluso multitudes de personas. Además de mencionar la iluminación y el cambio de ángulo. Asimismo, la cantidad de datos del conjunto RLVS no se compara de manera equitativa con la del resto de los conjuntos de datos.

Un aspecto más a considerar son las limitaciones computacionales. Al contar con una unidad de procesamiento gráfico (GPU) limitada, el modelo se ve forzado a trabajar bajo condiciones de configuración desfavorables, igualmente en la restricción de los procesos matemáticos complejos o su capacidad de realizar cálculos simultáneamente (Google, 2025).

CONCLUSIONES

La viabilidad de un modelo de seguridad no depende exclusivamente de su exactitud estadística en entornos controlados, sino también de su capacidad para operar en tiempo real y de su resiliencia ante escenarios visuales desconocidos. En este sentido, se evaluó el desempeño de tres modelos pre-entrenados de *Deep Learning* (EfficientNetB0, ResNet50 y YOLOv8) aplicados a la detección automatizada de violencia en secuencias de video, con el propósito de determinar el modelo que ofrezca el mejor equilibrio entre la resiliencia ante el desplazamiento de dominio (eficacia) y el procesamiento en tiempo real (eficiencia).

Para ello, en primera instancia, se acondicionaron los conjuntos de datos mediante la extracción de frames de cada uno a una frecuencia determinada, estableciendo así a RLVS como el conjunto de datos de dominio de origen para el entrenamiento de los modelos, a SCVD para la evaluación en las pruebas de generalización y a AIRTLAB para la evaluación en las pruebas de robustez, como los datos de dominio de destino. Seguidamente, los modelos se sometieron a un entrenamiento a partir del mecanismo de *Transfer Learning* rigiéndose bajo un total de 15 épocas para los modelos de múltiples etapas y 10 épocas para el modelo de una etapa. Adicionalmente, se estableció el mecanismo de *Early Stopping* con el objetivo de detener al modelo en caso de que presente memorización de los datos y restaurarlo a su estado óptimo.

Posterior al entrenamiento y la evaluación, los modelos fueron sometidos a una prueba de eficiencia computacional mediante pruebas de estrés, con las que se midieron la latencia de inferencia y la tasa de procesamiento en un entorno de hardware controlado. Como resultado, los modelos de múltiples etapas exhibieron una velocidad de entre 13 y 15 fotogramas por segundo, lo que los vuelve inviables para la detección en tiempo real. Mientras tanto, el modelo de una etapa exhibió una tasa de 264.5 fotogramas por segundo, rebasando por completo una cifra deseable en sistemas de videovigilancia en tiempo real. De esta manera, YOLOv8 se corona como el modelo con la mayor tasa de fotogramas en este caso.

Finalmente, a pesar de los destacables resultados en la exactitud de los modelos durante el entrenamiento, su capacidad de generalización se vio gravemente degradada ante

el desplazamiento de dominio. Causado principalmente por el cambio en el contexto de los datos y el desbalance de clases. Puesto que el conjunto de datos para el entrenamiento fue el único con simetría en la cantidad de videos para cada clase y a diferencia de los conjuntos de datos para la evaluación (pruebas de generalización y robustez), la clase de no violencia se encuentra conformada mayoritariamente por videos orientados al deporte mientras en los conjuntos de datos para la evaluación se componen de escenarios pacíficos en calles o centros de actividad e igualmente presentan multitudes o un mayor número de personas en cámara.

De este modo, ResNet50 fue el modelo que menor tasa de fotogramas y capacidad de detección presento al omitir incidencias de violencia física de forma discriminada. EfficientNetB0 demostró una capacidad de detección hipersensible frente a incidencias de violencia física, pero una tasa de fotogramas insuficiente para la detección en tiempo real. Por lo tanto, asumiendo el compromiso (*trade-off*) entre falsas alarmas y velocidad, YOLOv8 se posiciona como la única solución técnicamente viable para una implementación real de seguridad inmediata en el presente proyecto de investigación.

Como líneas de trabajo futuras para perfeccionar este sistema y mitigar su debilidad frente a las falsas alarmas, se propone la integración de algoritmos de análisis temporal, tales como redes neuronales recurrentes (LSTM) o mecanismos de seguimiento óptico (*Optical Flow*). De este modo, al añadir una dimensión de análisis secuencial, se permitiría que el modelo evaluara la presencia de acciones contundentes en un cuadro y comprendiera la intención sostenida del movimiento, refinando así su exactitud sin comprometer su velocidad de reacción.

Referencias

- 2M Technology. (19 de abril de 2024). *Evolution of Video Surveillance: From Closed-Circuit Analog to IP Cloud-Based Systems*. Recuperado el 17 de noviembre de 2025, de Advanced Security Systems for Complex & High-Risk Environments: <https://www.2mtechnology.net/evolution-of-video-surveillance-from-closed-circuit-analog-to-ip-cloud-based-systems/>
- Abundez, I. M., Alejo, R., Primero Primero, F., Granda Gutiérrez, E. E., Portillo Rodríguez, O., & Antonio Velázquez, J. A. (18 de julio de 2024). Threshold Active Learning Approach for Physical Violence Detection on Images Obtained from Video (Frame-Level) Using Pre-Trained Deep Learning Neural Network Models. *Multidisciplinary Digital Publishing Institute*, 17(7), 316. doi:<https://doi.org/10.3390/a17070316>
- Agramonte, D. A. (2024). *Implementación de un Sistema de Detección de Equipos de Protección*. Arequipa, Perú: Universidad Católica de Santa María. Recuperado el 24 de noviembre de 2025, de <https://repositorio.ucsm.edu.pe/server/api/core/bitstreams/94681511-ad7a-418b-b9fb-d99e634f082b/content>
- AI, L. (22 de julio de 2024). *Grad-CAM (Gradient-weighted Class Activation Mapping)*. Recuperado el 20 de julio de 2026, de Medium: <https://medium.com/@divakar1591/grad-cam-gradient-weighted-class-activation-mapping-8e1aeaf96d94>
- airtlab. (2020). *A Dataset for Automatic Violence Detection in Videos*. Recuperado el 19 de noviembre de 2025, de GitHub.com: <https://github.com/airtlab/A-Dataset-for-Automatic-Violence-Detection-in-Videos>
- Alzubaidi, L., Zhang, J., Humaidi, A., Al-Dujaili, A., Duan, Y., Al-Shamma, O., . . . Farhan, L. (31 de marzo de 2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1), 53. doi:10.1186/s40537-021-00444-8

- Amazon Web Services. (2026). *What is Data Augmentation?* Recuperado el 13 de mayo de 2026, de Amazon Web Services, Inc.: <https://aws.amazon.com/what-is/data-augmentation/>
- ApX Machine Learning. (2026). *Adapting Models to Different Data Distributions*. Recuperado el 08 de enero de 2026, de apxml.com: <https://goo.su/mRkUts>
- Aremu, T., Li, Z., & Alameeri, R. (2024). *Smart-City CCTV Violence Detection Dataset (SCVD)*. Recuperado el 19 de noviembre de 2025, de Kaggle.com: <https://www.kaggle.com/datasets/toluwaniaremu/smartcity-cctv-violence-detection-dataset-scvd>
- Argüello, F. (06 de octubre de 2023). *FPS en Videovigilancia, Fotogramas por segundo*. Recuperado el 15 de enero de 2026, de Infoteknico.com: <https://www.infoteknico.com/fps-en-videovigilancia/>
- Bakhshi, A., García Gómez, J., Gil Pita, R., & Chalup, S. (enero de 2023). Violence Detection in Real-Life Audio Signals Using Lightweight. *Procedia Computer Science*, 222, 244-251. doi:10.1016/j.procs.2023.08.162
- Banuprasad. (28 de junio de 2025). *VisDrone Dataset*. Recuperado el 20 de julio de 2026, de kaggledatasets: <https://www.kaggle.com/datasets/banuprasadb/visdrone-dataset>
- Behar Rivero, D. S. (2008). *Metodología de la Investigación*. Madrid: Shalom.
- Bermejo Nieves, E., Deniz Suarez, O., Bueno García, G., & Sukthankar, R. (2011). Violence Detection in Video Using Computer Vision Techniques. Berlin, Heidelberg: Springer Berlin Heidelberg. doi:https://doi.org/10.1007/978-3-642-23678-5_39
- Bilous, N., Malko, V., Frohme, M., & Nechyporenko, A. (11 de noviembre de 2024). Comparison of CNN-Based Architectures for Detection of Different Object Classes. *Multidisciplinary Digital Publishing Institute*, 5(4), 2300-2320. doi:<https://doi.org/10.3390/ai5040113>

- C3.ai. (2026). *F1 Score*. Recuperado el 03 de julio de 2026, de C3.ai:
<https://c3.ai/resources/glossary/data-science/f1-score>
- Cabezudo, R. (04 de agosto de 2023). *80/20 | Machine Learning*. Recuperado el 12 de marzo de 2026, de CloudAppMasters: <https://www.cloudappmasters.com/80-20-machine-learning/>
- Cásares, C. (07 de noviembre de 2022). *5 tendencias clave en tecnología de videovigilancia*. Recuperado el 17 de noviembre de 2025, de syscom.mx:
<https://www.syscom.mx/blog/las-5-principales-tendencias-en-tecnologia-de-cctv>
- Cloudflare. (2026). *What does buffering mean?* Recuperado el 05 de mayo de 2026, de Cloudflare.com: <https://www.cloudflare.com/learning/video/what-is-buffering/>
- Deep. (16 de marzo de 2019). *Introduction to SIFT(Scale Invariant Feature Transform)*. Recuperado el 20 de julio de 2026, de Medium.com:
<https://medium.com/@deepanshut041/introduction-to-sift-scale-invariant-feature-transform-65d7f3a72d40>
- Deniz, O., Serrano, I., Bueno, G., & Kim, T.-K. (12 de octubre de 2015). Fast violence detection in video. *2014 International Conference on Computer Vision Theory and Applications (VISAPP), 2*, 478-485. doi:10.5220/0004695104780485
- Durães, D., Veloso, B., & Novais, P. (29 de agosto de 2023). Violence Detection in Audio: Evaluating the Effectiveness of Deep Learning Models and Data Augmentation. *International Journal of Interactive Multimedia and Artificial Intelligence*, 8(3), 72-84. doi:<https://doi.org/10.9781/ijimai.2023.08.007>
- Elgendy, A. (13 de marzo de 2025). *What is overfitting in computer vision and how to prevent it?* Recuperado el 20 de julio de 2026, de Ultralytics:
<https://www.ultralytics.com/blog/what-is-overfitting-in-computer-vision-how-to-prevent-it>

Emergent Mind. (30 de diciembre de 2025). *Bidirectional GRU (BiGRU)*. Recuperado el 20 de julio de 2026, de emergentmind.com:

<https://www.emergentmind.com/topics/bidirectional-gated-recurrent-unit-bigru>

Figma. (10 de julio de 2026). *What is RGB? How RGB color works in design*. Recuperado el 20 de julio de 2026, de Figma: <https://www.figma.com/resource-library/what-is-rgb/>

GeeksforGeeks. (23 de julio de 2025). *Mel-frequency Cepstral Coefficients (MFCC) for Speech Recognition*. Recuperado el 20 de julio de 2026, de GeeksforGeeks: <https://www.geeksforgeeks.org/nlp/mel-frequency-cepstral-coefficients-mfcc-for-speech-recognition/>

GeeksforGeeks. (24 de julio de 2025). *What is TensorRT?* Recuperado el 20 de julio de 2026, de GeeksforGeeks: <https://www.geeksforgeeks.org/deep-learning/what-is-tensorrt/>

GeeksforGeeks. (18 de mayo de 2026). *Bidirectional LSTM in NLP*. Recuperado el 20 de julio de 2026, de geeksforgeeks.org: <https://www.geeksforgeeks.org/nlp/bidirectional-lstm-in-nlp/>

Google. (2012). *Almacena y comparte archivos en línea*. Recuperado el 17 de diciembre de 2025, de Google Workspace: <https://workspace.google.com/products/drive/>

Google. (2018). *Choose a plan that works for you*. Recuperado el 19 de diciembre de 2025, de Google.com: <https://goo.su/qFvzKL>

Google. (2019). *Google Colab*. Recuperado el 17 de diciembre de 2025, de Google.com: <https://colab.research.google.com/>

Google. (2025). *Datasets: Class-imbalanced datasets*. Recuperado el 20 de mayo de 2026, de Google for Developers: <https://developers.google.com/machine-learning/crash-course/overfitting/imbalanced-datasets>

- Google. (2025). *Google Colaboratory Frequently Asked Questions*. Recuperado el 19 de diciembre de 2025, de Google.com:
<https://research.google.com/colaboratory/faq.html>
- Google. (2025). *What is a GPU and its role in AI?* Recuperado el 20 de julio de 2026, de Google Cloud: <https://cloud.google.com/discover/gpu-for-ai>
- Google. (2026). *Classification: Accuracy, recall, precision, and related metrics*. Recuperado el 03 de julio de 2026, de Google for Developers:
<https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall>
- Google. (2026). *What is a convolutional neural network?* Recuperado el 20 de julio de 2026, de Google Cloud: <https://cloud.google.com/discover/what-are-convolutional-neural-networks>
- Gu, C., Wu, X., & Wang, S. (06 de mayo de 2020). Violent Video Detection Based on Semantic Correspondence. *IEEE Access*, 8, 85958-85967.
doi:10.1109/ACCESS.2020.2992617
- Gunarathne, A. (02 de noviembre de 2025). *Local Binary Patterns (LBP) — Explained*. Recuperado el 20 de julio de 2026, de Medium.com:
<https://medium.com/@ayeshagunarathne/local-binary-patterns-lbp-explained-4eb5364492f3>
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2014). *Metodología de la investigación*. España: McGraw Hill España.
- IBM. (9 de agosto de 2024). *What is AI?* Recuperado el 12 de junio de 2026, de Ibm.com:
<https://www.ibm.com/think/topics/artificial-intelligence>
- IONOS. (10 de febrero de 2025). *Long Short-Term Memory: redes con memoria a corto-largo plazo*. Recuperado el 20 de julio de 2026, de IONOS Digital Guide:
<https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/long-short-term-memory/>

- Kim, E. (07 de abril de 2023). *Understanding Single Shot Detector (SSD)*. Recuperado el 20 de julio de 2026, de Medium: <https://medium.com/@elvenkim1/understanding-single-shot-detector-ssd-782e4b93aec3>
- Lenovo. (2024). *Computer Vision AI: Transforming Technology Landscape*. Recuperado el 17 de noviembre de 2025, de Lenovo.com: <https://www.lenovo.com/in/en/glossary/computer-vision-ai/?orgRef=https%253A%252F%252Fwww.google.com%252F>
- Massiris Fernández, M. M. (2021). *Visión artificial en la gestión y apoyo a la seguridad de los trabajadores*. Bahía Blanca, Argentina: Universidad Nacional del Sur. Recuperado el 24 de noviembre de 2025, de <https://repositoriodigital.uns.edu.ar/handle/123456789/5753>
- Massobrio, A. (2026). *3D Convolutional Neural Network (3D CNN) — A Guide for Engineers*. Recuperado el 20 de Julio de 2026, de Neuralconcept.com: <https://www.neuralconcept.com/post/3d-convolutional-neural-network-a-guide-for-engineers>
- MathWorks. (2019). *Red neuronal convolucional EfficientNet-b0*. Recuperado el 26 de diciembre de 2025, de MathWorks.com: <https://la.mathworks.com/help/deeplearning/ref/efficientnetb0.html>
- Merit, K., & Beladgham, M. (31 de diciembre de 2025). AI-based Violent Incident Detection in Surveillance Videos to Enhance Public Safety. *The Journal of Telecommunications and Information Technology*, 102(4), 77-89. doi:<https://doi.org/10.26636/jtit.2025.4.2328>
- Mohamed, E., Mohamad, H., & Abd El Massih, M. (27 de abril de 2019). *Real Life Violence Situations Dataset*. Recuperado el 19 de noviembre de 2025, de Kaggle.com: <https://www.kaggle.com/datasets/mohamedmustafa/real-life-violence-situations-dataset/data>

- Negre, P., S. Alonso, R., González Briones, A., Prieto, J., & Rodríguez González, S. (20 de junio de 2024). Literature Review of Deep-Learning-Based Detection of Violence. *Multidisciplinary Digital Publishing Institute*, 24(12), 4016. doi:10.3390/s24124016
- Nithin Sai, A., & Siva Prasad, K. (septiembre de 2023). Machine Learning Software for the Detection of Violence from CCTV Live Footage. *Journal of Image Processing and Artificial Intelligence*, 9(3), 12-18. doi:10.46610/JOIPAI.2023.v09i03.002
- NVIDIA. (2017). *Convolutional Neural Network (CNN)*. Recuperado el 27 de junio de 2026, de NVIDIA Developer: <https://developer.nvidia.com/discover/convolutional-neural-network>
- NVIDIA. (2025). *NVIDIA T4 Tensor Core GPUs for Accelerating Inference*. Recuperado el 20 de julio de 2026, de NVIDIA: <https://www.nvidia.com/en-us/data-center/tesla-t4/>
- Pal, D. S. (03 de noviembre de 2023). *Achieving Optimal Performance: The Role of Early Stopping in Fine-Tuning Models*. Recuperado el 08 de enero de 2026, de InstaDataHelp | Unearth New Dimensions: <https://goo.su/gRIw3>
- Parlamentario, Instituto de Derecho. (2024). *Robustez; "robustness"*. Recuperado el 17 de noviembre de 2025, de Idpucm.com: <https://www.idpucm.com/robustez-robustness/>
- PC Componentes. (31 de agosto de 2022). *Qué es un cuello de botella de un PC y cómo solucionarlo*. Recuperado el 05 de mayo de 2026, de PC Componentes: <https://goo.su/eULUD4R>
- Primero Primero, F., Cervantes Ambriz, D., Alejo Eleuterio, R., Granda Gutiérrez, E., Sánchez Jaime, J., & ValdovinosRosas, R. (15 de septiembre de 2025). Strategic Sample Selection in Deep Learning: A Case Study on Violence Detection Using Confidence-Based Subsets. *Multidisciplinary Digital Publishing Institute*, 17(9), 1536. doi:<https://doi.org/10.3390/sym17091536>

- Radečić, D. (12 de junio de 2026). *Overfitting vs. Underfitting: A Practical Guide to Model Diagnostics*. Recuperado el 20 de julio de 2026, de DataCamp:
<https://www.datacamp.com/tutorial/overfitting-vs-underfitting>
- Raihan Talha, K., Bandapadya, K., & Monirujjaman Khan, M. (13 de julio de 2022). Violence Detection Using Computer Vision Approaches. *2022 IEEE World AI IoT Congress (AlloT)*, 544-550. doi:10.1109/AlloT54504.2022.9817374
- Rajini, Deekshitha, Kashish, Yoganarasimhan, & Sinchana. (2026). *Deep Learning-Enhanced Video Surveillance System for Detecting Violent Behavior Using Anomaly Detection Techniques*. Recuperado el 21 de enero de 2026, de
<https://www.scribd.com/document/848309831/Violence-Detection>
- Red Hat. (15 de abril de 2026). *What is deep learning?* Recuperado el 13 de mayo de 2026, de Redhat.com: <https://www.redhat.com/en/topics/ai/what-is-deep-learning>
- roboflow. (23 de junio de 2025). *What Is ResNet-18? How to Use the Lightweight CNN Model*. Recuperado el 24 de noviembre de 2025, de Roboflow Blog:
<https://blog.roboflow.com/resnet-18/>
- Rodríguez, P. N. (2025). *Inteligencia artificial fiable para la detección de violencia en vídeo*. Salamanca, España: Universidad de Salamanca. doi:10.14201/gredos.170077
- Roggio, F., Trovato, B., Sortino, M., & Musumeci, G. (15 de noviembre de 2024). A comprehensive analysis of the machine learning pose estimation models used in human movement and posture analyses: A narrative review. *Heliyon*, 10(21), e39977. doi:<https://doi.org/10.1016/j.heliyon.2024.e39977>
- Salman, M., Abbas, N., ur Rahman, S. I., Al Alshaikh, M., & Saudagar, A. K. (28 de marzo de 2026). An embedded deep learning framework for real-time violence detection and alert generation. *Scientific Reports*, 16(10805). doi:<https://doi.org/10.1038/s41598-026-44939-x>
- Sánchez, G. P. (2021). *Detección de Víctimas mediante Redes Neuronales e Imágenes Térmicas*. Madrid, España: Escuela Técnica Superior de Ingenieros Industriales

Universidad Politécnica de Madrid. Recuperado el 02 de diciembre de 2025, de https://oa.upm.es/67546/1/TFG_GUILLERMO_PRIETO_SANCHEZ.pdf

Sandaruwan, H. (13 de enero de 2024). *EfficientNetB0 architecture - Stem Layer*.

Recuperado el 04 de febrero de 2026, de Medium.com:

<https://medium.com/image-processing-with-python/efficientnetb0-architecture-stem-layer-496c7911a62d>

SharkYun. (31 de octubre de 2024). *Computer Vision — Object Detection, One-Stage vs Two-Stage detectors*. Recuperado el 17 de noviembre de 2025, de Medium:

<https://sharkyun.medium.com/computer-vision-object-detection-one-stage-vs-two-stage-b05dbff88195>

Silva Gúzman, E. R. (2020). *Entrenamiento de la Red Neuronal Convolucional YOLO para objetos*. Universidad Nacional Autónoma de México. Ciudad Universitaria, CDMX: Universidad Nacional Autónoma de México. Recuperado el 04 de febrero de 2026, de <https://goo.su/hD5s2zh>

Soni, S. (11 de octubre de 2023). *YOLO v2 Comprehensive Tutorial: Building on YOLO v1 Mistakes*. Recuperado el 24 de noviembre de 2025, de Medium.com:

<https://medium.com/@sachinsoni600517/yolo-v2-comprehensive-tutorial-building-on-yolo-v1-mistakes-aa7912292c1a>

Strategic Platform. (21 de septiembre de 2019). *Por qué python es el mejor lenguaje para IA y redes neuronales*. Recuperado el 19 de diciembre de 2025, de Strategicplatform.com: <https://strategicplatform.com/articulos/redes-neuronales-python>

Taboga, M. (2021). *Domain shift*. Recuperado el 17 de noviembre de 2025, de Statlect.com: <https://www.statlect.com/machine-learning/domain-shift>

TIOBE. (2025). *TIOBE Index for December 2025*. Recuperado el 19 de diciembre de 2025, de TIOBE Index: <https://www.tiobe.com/tiobe-index/>

Ultralytics. (12 de noviembre de 2023). *Explore Ultralytics YOLOv8*. Recuperado el 26 de diciembre de 2025, de Documentación de Ultralytics:
<https://docs.ultralytics.com/es/models/yolov8>

Ultralytics. (29 de junio de 2024). *Insights on Model Evaluation and Fine-Tuning*.
Recuperado el 10 de marzo de 2026, de Ultralytics Docs:
<https://docs.ultralytics.com/guides/model-evaluation-insights>

Ultralytics. (11 de mayo de 2025). *What is the ideal imgsz for Document classification in yolo11*. Recuperado el 26 de diciembre de 2025, de Ultralytics.com:
<https://community.ultralytics.com/t/what-is-the-ideal-imgsz-for-document-classification-in-yolo11/1065>

Ultralytics. (17 de abril de 2026). *What is ResNet-50 and what is its relevance in computer vision?* Recuperado el 01 de julio de 2026, de Ultralytics.com:
<https://www.ultralytics.com/blog/what-is-resnet-50-and-what-is-its-relevance-in-computer-vision#what-is-resnet-50>

Unity. (2025). *Computer Vision*. Recuperado el 27 de junio de 2026, de Unity.com:
<https://unity.com/glossary/computer-vision>

Ye, L., Liu, T., Han, T., Ferdinando, H., Seppänen, T., & Alasaarela, E. (09 de febrero de 2021). Campus Violence Detection Based on Artificial Intelligent Interpretation of Surveillance Video Sequences. *Multidisciplinary Digital Publishing Institute*, 13(4), 628. doi:<https://doi.org/10.3390/rs13040628>

Glosario de acrónimos

BiGRU: Bidirectional Gated Recurrent Unit. Es un modelo de redes neuronales que procesa secuencias de imágenes previas y posteriores a su estado actual para procesar el contexto temporal de un fenómeno determinado (Emergent Mind, 2025).

BiLSTM: Bidirectional Long Short-Term Memory. De la misma manera, es un modelo de red neuronal basado en LSTM que es capaz de retroceder a secuencias pasadas y adelantarse a secuencias futuras para así conocer el contexto temporal pasado y futuro (GeeksforGeeks, 2026).

C3D: Convolutional 3D. Es una arquitectura de *Deep Learning* que analiza la anchura, la altura y la profundidad de las imágenes (Massobrio, 2026).

CCTV: Closed-Circuit Television. Es el conjunto de cámaras de vídeo perteneciente a una red privada dedicadas a la videovigilancia y la visualización de sus grabaciones por medio de monitores.

CNNS: Convolutional Neural Networks. Una red neuronal convolucional es una red neuronal artificial orientada a la detección de patrones en imágenes o cualquier archivo de entrada con una estructura de matriz. Se utiliza en la visión por computadora, al estar inspirada en la visión humana (Google, 2026).

COCO: Common Objects in Context. Es un conjunto de datos que cuenta con una amplia variedad de datos para la evaluación de los modelos para tareas de visión por computadora desarrollados o afinados por investigadores (Ultralytics, 2023).

GPU T4: Graphics Processing Unit Tesla 4. Es la unidad gráfica de procesamiento versión Tesla 4 perteneciente a NVIDIA dedicada a trabajos de “procesamiento de alto rendimiento, la capacitación y la inferencia del *Deep Learning* y *Machine Learning*, basada en la nube” (NVIDIA, 2025).

GradCAM: Gradient-weighted Class Activation Mapping. Es una técnica utilizada para visualizar las decisiones que ha tomado una red neuronal convolucional con respecto a la clasificación de un fenómeno u objeto en especial. Permitiendo el análisis a través de

imágenes de calor a partir del cálculo de los gradientes. Se usa para explicar el proceso de toma de decisión de las CNN (AI, 2024).

LBPH: Local Binary Pattern Histograms. Consta de transformar los valores de una imagen en números enteros que pertenecen a un patrón (Gunarathne, 2025).

LSTM: Long Short-Term Memory. Es una técnica que le permite a una red neuronal conservar información pasada para cálculos en futuras predicciones (IONOS, 2025).

MFCC: Mel-Frequency Cepstral Coefficients. Es una técnica que representa el espectro de la voz por medio de matemáticas que capturan las características del habla, principalmente para tecnología emergente dedicada a las tareas de asistencia a través de comandos de voz (GeeksforGeeks, 2025).

MoSIFT: Motion Scale-Invariant Feature Transform. Es un algoritmo de extracción de características en puntos estratégicos de una imagen en movimiento sin importar el cambio de posición (Deep, 2019).

RGB: Red, Green, Blue. Es un modelo de color digital, el cual a partir de la combinación de rojo, verde y azul es posible crear nuevos colores (Figma, 2026).

SSD: Single Shot Detector. Es un algoritmo de *Deep Learning* basado en redes neuronales convolucionales que realiza la detección de objetos en una sola pasada hacia adelante (Kim, 2023).

STIP: Spatio-Temporal Interest Points. Es un algoritmo invariable en la detección de características de una imagen (Deep, 2019).

TensorRT: Tensor RunTime. Es un motor de alto rendimiento para el entrenamiento de modelos de *Deep Learning* y una biblioteca de herramientas desarrollado por NVIDIA (GeeksforGeeks, 2025).

VisDrone: Vision Meets Drone. Es un conjunto de datos capturados por drones para su uso en la evaluación de modelos de *Deep Learning* (Banuprasad, 2025).