UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

Aplicación de algoritmos de aprendizaje automático y computer vision para la prevensión y detección de crimenes

.

Andrés Mateo Herrera Gordón

Ingeniería en Ciencias de la Computación

Trabajo de fin de carrera presentado como requisito
para la obtención del título de
Ingeniero en Ciencias de la Computación

Quito, 12 de Mayo de 2025

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Politécnico

HOJA DE CALIFICACIÓN
DE TRABAJO DE FIN DE CARRERA

Aplicación de algoritmos de aprendizaje automático y computer vision para la prevensión y detección de crimenes

Andrés Mateo Herrera Gordón

Nombre del profesor, Título académico **Felipe Grijalva, Ph. D**

Quito, 12 de Mayo de 2025

© DERECHOS DE AUTOR

Nombres y apellidos: Andrés Mateo Herrera Gordón

Código: 216962

Cédula de identidad: 1724061039

Lugar y fecha: Quito, 12 de Mayo de 2025

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en http://bit.ly/COPETheses.

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on http://bit.ly/COPETheses.

Aplicación de Algoritmos de Aprendizaje Automático y Visión por Computadora para la Prevención y Detección de Crímenes

Andrés Herrera
Ing. en Ciencias de la Computación
Universidad San Francisco de Quito
Quito, Ecuador
aherrerag@estud.usfq.edu.ec

***Resumen*—Este trabajo presenta el entrenamiento e implementación de algoritmos de aprendizaje automático y visión por computador, orientados a la prevención y detección de crímenes en tiempo real. El tema surge como respuesta al incremento de actos ilícitos en Ecuador, con el objetivo de reducir los tiempos de respuesta policial mediante la automatización del análisis de comportamientos sospechosos en videovigilancia. Se utilizaron modelos con redes neuronales profundas, específicamente *I3D* y una arquitectura híbrida *I3D+ConvLSTM*, capaces de identificar patrones en secuencias temporales. Los resultados obtenidos demuestran la efectividad de estas técnicas para anticipar y detectar eventos delictivos, constituyendo una herramienta valiosa para mejorar la seguridad ciudadana. Esta investigación representa un avance en la aplicación de visión por computadora a contextos urbanos, proponiendo una solución práctica, escalable y con alto potencial de implementación.**

***Palabras clave*—Visión por computador, aprendizaje automático, videovigilancia inteligente, redes neuronales profundas, *I3D*, ConvLSTM2D, detección de crímenes, análisis de video, seguridad ciudadana, comportamiento sospechoso.**

***Abstract*—This paper presents the training and implementation of machine learning and computer vision algorithms aimed at the real-time prevention and detection of crimes. The proposal emerges in response to the rising crime rates in Ecuador, with the objective of reducing police response times through the automated analysis of suspicious behaviors in video footage. Deep neural network models were employed, specifically *I3D* and a hybrid I3D-ConvLSTM2D architecture, capable of identifying anomalous patterns in temporal video sequences. The results demonstrate the effectiveness of these techniques in both anticipating and detecting criminal events, providing a valuable tool for enhancing public safety. This research represents a significant step forward in the application of computer vision to urban contexts, offering a practical, scalable solution with strong implementation potential.**

***Keywords*—Computer vision, machine learning, intelligent video surveillance, deep neural networks, *I3D*, ConvLSTM2D, crime detection, video analysis, public safety, suspicious behavior.**

I. Introducción

Ecuador ha experimentado un aumento constante en los índices de criminalidad en los últimos años. Esto ha generado preocupación social, temor, y ha evidenciado la necesidad de soluciones tecnológicas que fortalezcan las capacidades de prevención y respuesta del sistema de seguridad del país. Una comparación de los datos mensuales de 2023, 2024 y 2025 revela un incremento en los robos reportados durante el presente año. Hasta marzo de 2025, se han reportado un total de 3.750 casos, superando las cifras del mismo período en 2023, con un recuento de 3.301 casos, y en 2024, con 3.259. Esta tendencia al alza resalta la naturaleza alarmante de la situación actual [1]. En este contexto, el presente trabajo se centra en el desarrollo e implementación de un sistema inteligente basado en algoritmos de aprendizaje automático y visión por computadora para la detección y prevención de delitos en tiempo real.

Si bien existen diversas iniciativas basadas en inteligencia artificial orientadas a la seguridad urbana, estas presentan limitaciones al momento de capturar la complejidad espacio-temporal inherente a los eventos delictivos. Este estudio introduce un enfoque que integra redes neuronales profundas con mecanismos de análisis temporal, con el objetivo de identificar patrones de comportamiento anómalos y reducir los tiempos de respuesta operativa por parte de las autoridades.

Para la validación se utilizó el conjunto de datos UCF-Crime, compuesto por grabaciones de cámaras de seguridad que reflejan situaciones reales en entornos urbanos. A nivel metodológico, se implementaron y compararon dos modelos: una arquitectura basada en Two-Stream Inception 3D y una variante que incorpora una red ConvLSTM para capturar de forma más eficaz las dinámicas temporales de las secuencias de video. Además, se aplicó la técnica One Cycle Policy en combinación con LRFinder para optimizar la tasa de aprendizaje durante el entrenamiento. Este enfoque busca mejorar la captura de características temporales como una herramienta efectiva para la prevención del crimen.

II. Estado del Arte

El estudio del uso de visión por computadora y algoritmos de aprendizaje automático en la detección de actividades ilícitas ha tenido una evolución significativa a lo largo de los años. Esto se evidencia en varios trabajos de investigación, los cuales abarcan distintos enfoques sobre este mismo tema.

Uno de los enfoques más utilizados es el análisis de comportamiento previo al acto delictivo. El Martínez-Mascorro et

al. [2] proponen el uso de redes neuronales convolucionales tridimensionales (3DCNN) para la detección de comportamientos sospechosos en escenarios de hurto en tiendas. Se enfoca en identificar el PCB (Previous Crime Behaviour), el comportamiento anterior al crimen, con el fin de prevenir el acto delictivo. Este enfoque evidencia la eficacia de las 3DCNN como herramientas preventivas, y subraya el valor de analizar el comportamiento humano antes de que se materialice la actividad ilícita. De forma similar, en otro enfoque del ingeniero en ciencias de la computación Martínez-Mascorro et al. [3] se comparan tres configuraciones de entrenamiento: un modelo binario, un modelo multi-clase con salida binaria y un modelo completamente multi-clase. Se subraya la importancia de mantener una configuración de datos en modelo de clasificación binaria y destacando la efectividad del enfoque PCB para la predicción temprana de actividades anómalas.

Por otra parte, con la finalidad de aumentar la precisión en el reconocimiento de violencia en tiempo real, los matemáticos Ghosh y Chakrabarty [4] proponen una arquitectura Two-stream Convolutional Network basada en el procesamiento paralelo de dos tipos de entrada, frames RGB y flujo óptico. Estos permiten capturar de manera más robusta tanto las características espaciales como las temporales de los videos. Asimismo, los expertos Carreira y Zisserman [5] introducen el modelo Two-Stream Inflated 3D ConvNet (*I3D*). Este infla las redes convolucionales 2D entrenadas en ImageNet para poder operar en 3D. Al ser una Two-Stream, trabaja con frames RGB y flujos ópticos paralelamente. Al probarse en el ámbito de reconocimiento de acciones en comparación a modelos anteriores como el C3D, redes 3DCNN o redes recurrentes ConvLSTM, el modelo *I3D* alcanza una precisión del 0.98 en UCF-101 y 0.809 en HMDB-51, superándolos ampliamente. Se demuestra que el uso de arquitecturas infladas en 3D permite una comprensión más robusta de las acciones humanas en video, estableciendo un nuevo modelo para tareas de reconocimiento de comportamiento.

En el rubro de reconocimiento de crímenes, los investigadores Sultani et al. [6] introducen un enfoque para la detección de anomalías en videos de vigilancia utilizando un marco de aprendizaje profundo basado en la clasificación de instancias múltiples. Al igual que en otros casos, emplea etiquetas binarias y segmenta los videos en instancias que se procesan dentro del modelo como bolsas positivas o negativas, para luego entrenar con el modelo C3D. Este trabajo destaca la importancia de la detección preliminar de anomalías como un paso hacia una interpretación de eventos en entornos de vigilancia reales. Siendo uno de los primeros en utilizar el modelo UCF-Crime dataset.

Además, la profesora Mahareek et al. [7] proponen un enfoque de detección de anomalías en videos de vigilancia basado en la combinación de una red convolucional 3D (3DCNN) y una red convolucional de largo plazo (ConvLSTM). Su arquitectura se basa en la habilidad de las redes 3DCNN para obtener características espacio-temporales y de la habilidad de las redes ConvLSTM para obtener relaciones temporales. El modelo incluye una secuencia de cuatro capas 3DCNN con activaciones ReLU, normalización por lotes y técnicas de regularización como max-pooling y dropout, seguida de una capa ConvLSTM y una capa densa con activación softmax para la clasificación. Los resultados sugieren que la integración de 3DCNN con ConvLSTM constituye una solución robusta y eficaz para el análisis de videos de vigilancia.

Por lo tanto, la mayoría de los enfoques existentes se centran en redes 3DCNN estándar o en arquitecturas recurrentes de forma aislada. En este contexto, se propone una arquitectura novedosa que combina el modelo *I3D* con una capa ConvLSTM de forma secuencial, lo cual permite capturar de manera más precisa patrones espacio-temporales complejos asociados al comportamiento previo al crimen. A diferencia de trabajos previos que usan 3DCNN genéricas junto a ConvLSTM, la incorporación del modelo *I3D*, considerado más avanzado por su estructura inflada y uso de dos flujos de entrada, permite potenciar aún más la detección de patrones tempranos. Esta propuesta busca enfocarse especialmente en la prevención, es decir, en reconocer el momento previo al acto delictivo; por lo que se requiere un modelado temporal más profundo. Además, se plantea como una solución adaptable a contextos de vigilancia en tiempo real y se evalúa sobre el dataset UCF-Crime, utilizado en investigaciones similares.

III. Metodología

Esta investigación se realizó utilizando la versión 2.5.0 de TensorFlow y 2.5.0 de Keras. De igual forma, se utilizó el servidor A100 de 80 GB con el fin de mejorar el rendimiento de los entrenamientos.

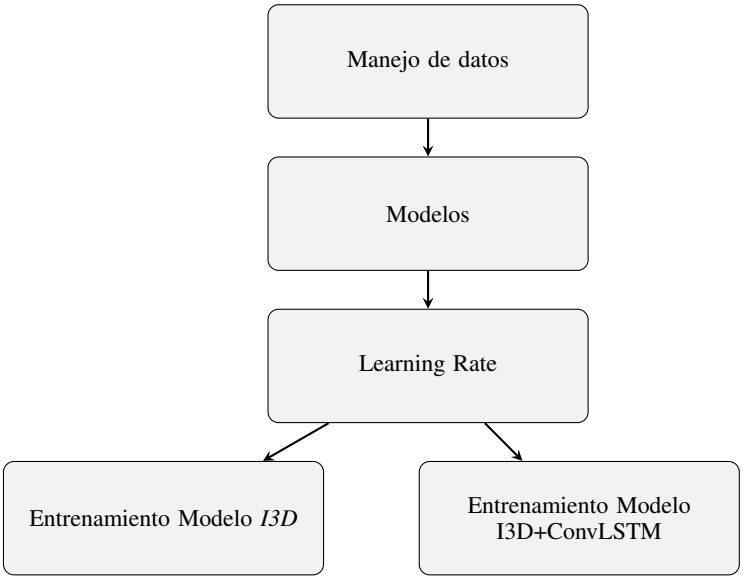

Figura 1. Procesos seguidos.

En la Figura 1 se puede observar el flujo en el que se realiza el proyecto. Empezando por el manejo de datos, seguido de las técnicas utilizadas para escoger el mejor learning rate. Finalizando con el entrenamiento de los modelos para posteriormente compararlos.

A. Manejo de datos

Como se menciona anteriormente, se utiliza un conjunto de datos de UCF-Crime. Este conjunto cuenta con 13 clases de distintos crímenes y un total de 1900 videos. Cabe recalcar que cuenta con videos repetidos en cada clase.

Primero, se decidió qué clases son las que iban a ser utilizadas para el entrenamiento. Se escogieron las siguientes clases: Abuse, Arson, Assault, Burglary, Fighting, Robbery, Shooting, Shoplifting, Stealing, Vandalism. La elección de estas categorías se basa en varios aspectos como: relevancia social, estas clases representan tipos de crímenes comunes en contextos urbanos y con alto impacto en la seguridad ciudadana; detectabilidad visual, las clases seleccionadas presentan características visuales distinguibles que pueden ser identificadas mediante técnicas de visión por computador, tales como movimientos agresivos e interacción con objetos específicos; impacto preventivo, la selección de estas clases responde a un balance entre importancia social, capacidad de ser detectadas visualmente y factibilidad técnica, permitiendo un enfoque estratégico para abordar el problema de la criminalidad en Ecuador.

Como segundo paso se escogió los videos basados en ciertas características importantes. Calidad del video, evitar videos extremadamente pixelados ya que pueden afectar la extracción de características visuales relevantes; iluminación y visibilidad, si el video es demasiado oscuro puede ser difícil detectar detalles, asimismo con los videos con la iluminación demasiado intensa; ángulos fijos de cámara, se utilizaron videos con ángulos de cámara fijos y mayormente con planos abiertos.

Continuando, a la elección de PCB, start frame y end frame para cada video de crimen. De esta forma, se toma en cuenta el momento previo al crimen para poder concentrar el modelo en obtener relaciones temporales. Al finalizar, se prosiguió a tratar los datos. Mediante scripts de Python, se revisó si algún video estaba repetido o si estaba tan oscuro que el modelo no podría obtener características. Después de tratar los videos, se procedió a balancear las clases para obtener la mitad de videos con un crimen y la otra mitad de videos normales.

Luego, se separó el dataset en entrenamiento (516 videos), validación (65 videos) y prueba (65 videos). De esta forma, los datos estaban listos para poder procesarse en el módulo generador de datos para Tensorflow. Para ello, se propone un módulo eficiente y robusto denominado VideoDataModule, desarrollado en TensorFlow. Este módulo es útil para la preparación, procesamiento y carga de datos de video en tareas de clasificación binaria, crimen y no crimen. Este componente es crucial para garantizar una entrada consistente y representativa al modelo de aprendizaje profundo, permitiendo una generalización efectiva del mismo en escenarios del mundo real.

El módulo presenta una implementación que permite extraer clips de una longitud fija desde los videos obtenidos, con la capacidad de manejar variaciones en la duración de los mismos y adaptarse dinámicamente durante el entrenamiento. Una de las principales fortalezas del módulo radica en su estrategia de muestreo temporal aleatorio durante la fase de entrenamiento, lo cual incrementa significativamente la diversidad en el entrenamiento y reduce el sobreajuste, evitando que el modelo aprenda el orden de los videos como una característica.

Además, este módulo incorpora técnicas de aumento de datos como: recorte aleatorio, cambios de brillo, contraste, tono, adición de ruido y desenfoque gaussiano. Estos son fundamentales para simular condiciones reales con variaciones de iluminación, movimiento y calidad de grabación. Estas transformaciones fortalecen la robustez del modelo frente a entornos complejos y nuevos ambientes.

VideoDataModule opera sobre un conjunto de datos estructurado en formato tabular, donde cada fila representa una muestra de video etiquetada. Las columnas incluyen la ruta al archivo de video, el cuadro de inicio PCB, el cuadro final, y la etiqueta binaria que indica si el fragmento de video corresponde a una actividad criminal (1) o no criminal (0). A partir de esta información, el módulo extrae una secuencia de números de frames de imágenes RGB, cada una re-dimensionada a 224×224 píxeles, que en conjunto conforman un tensor de forma (num frames, height, width, 3). Esta secuencia representa la entrada del modelo. La salida correspondiente es un vector codificado en one-hot de longitud 2, que indica la clase a la que pertenece la muestra. Este diseño de entrada y salida permite una integración directa con las arquitecturas de redes neuronales convolucionales utilizadas para clasificación de videos en esta investigación, garantizando consistencia y compatibilidad en todo el pipeline de entrenamiento y evaluación.

B. Modelos

Como se detalla anteriormente, se trabajará con dos modelos. Un modelo puro del *I3D* y un combinado de *I3D+ConvLSTM*. De esta forma, se puede evaluar el rendimiento del modelo base (*I3D*), que extrae eficientemente características espaciales y temporales mediante convoluciones 3D, frente a un modelo extendido que incorpora una capa ConvLSTM. Esta última capa permite capturar dependencias temporales de alto nivel, integrando la memoria interna de los estados recurrentes con el procesamiento convolucional. La combinación busca aprovechar la robustez del *I3D* en la extracción de patrones locales de movimiento, junto con la capacidad de modelado secuencial de la ConvLSTM, útil en escenarios donde los patrones de crimen se desarrollan gradualmente a lo largo del tiempo. El objetivo es determinar si la adición de memoria y sensibilidad a la dinámica temporal mejora la detección y clasificación de eventos delictivos en comparación con un enfoque puramente convolucional.

El modelo *I3D* originalmente trabaja con la versión 1.x de TensorFlow y sus respectivas dependencias. Con la finalidad de utilizar un modelo actualizado con dependencias actualizadas, se obtuvo el modelo *I3D* de un pull request al original; este actualiza el modelo para poder trabajar con versiones 2.x de TensorFlow.

Con el fin de facilitar la integración del modelo *I3D* dentro del flujo de trabajo de entrenamiento, evaluación y ajuste de hiperparámetros, envuelto en un modelo funcional de Keras. Esta decisión responde a varios motivos prácticos y técnicos. Al utilizar un modelo Keras se facilita la incorporación de nuevas capas, la modificación de salidas y la integración con otras arquitecturas, como en el caso del modelo combinado *I3D+ConvLSTM*. De igual forma, se ganan todas las funcionalidades como Callbacks y summaries, los cuales forman parte intrínseca de Keras.

Asimismo, se utiliza como métrica de evaluación la pérdida en validación. Añadiendo, además, como métrica de evaluación el AUC (Área Bajo la Curva) y exactitud para tener una referencia adicional del rendimiento del modelo.

Por otra parte, para elegir los hiperparámetros del modelo combinado. El modelo fue envuelto dentro de una clase Keras; esto permite que el tuner lo trate como una unidad optimizable. Dentro de esta clase, se expusieron los siguientes hiperparámetros que afectan el aprendizaje y la generalización del modelo:

Cuadro I
HIPERPARÁMETROS EXPLORADOS Y VALORES OPTIMIZADOS

Parámetro	***Hiperparámetros***	***Optimización***
Tamaño del kernel	3x3, 5x5	3x3
Número de filtros	32, 64, 128	128
Dropout	0.2, 0.3, 0.5	0.3
Dropout recurrente	0.2, 0.3, 0.5	0.5
Unidades densas	64, 128, 256	128
Dropout después de capa densa	0.2, 0.3, 0.5	0.5

En el Cuadro I se observan los hiperparámetros explorados. La razón para incluir estos hiperparámetros específicos es que cada uno de ellos controla distintos aspectos del aprendizaje: el tamaño del kernel influye en el contexto espacio-temporal que el modelo puede capturar; los filtros determinan la capacidad expresiva de la red; las tasas de dropout controlan el sobreajuste, especialmente importantes en modelos con un número de parámetros alto, como lo es el modelo híbrido *I3D+ConvLSTM*; el dropout recurrente ayuda a reducir el sobreajuste en operaciones convolucionales ConvLSTM al apagarlas aleatoriamente. Del mismo modo, el número de unidades en la capa Dense final actúa como un cuello de botella antes de la salida, impactando directamente la capacidad de abstracción del modelo; mientras que el último dropout es el proceso final previo a la operación softmax que fue usada en el modelo que obtiene las predicciones.

La métrica objetivo seleccionada para la optimización fue la exactitud en validación, ya que el objetivo principal del modelo es clasificar correctamente la clase del video.

Se definió un total de 10 combinaciones, con una ejecución por combinación de 1, lo que permite tener una exploración inicial razonable sin incurrir en altos costos de cómputo. Para evitar el sobreentrenamiento, se utilizó la técnica de early stopping, deteniendo el entrenamiento si la pérdida en validación no mejora durante tres épocas consecutivas.

Tras completar el proceso de optimización, los hiperparámetros encontrados por el tuner correspondieron a la combinación que dio paso a un mejor desempeño en el conjunto de validación, según la métrica de precisión de validación. Esta combinación fue la que produjo el mayor rendimiento en el conjunto de validación, según la métrica de exactitud de validación. El uso de un kernel size pequeño (3x3) permitió capturar patrones espacio-temporales más locales, mientras que un número elevado de filtros como 128 incrementó la capacidad representativa de la capa ConvLSTM. Los valores relativamente altos de dropout y recurrent dropout, 0.3 y 0.5, respectivamente, actuaron como mecanismos efectivos de regularización, reduciendo el sobreajuste. Por último, se seleccionaron 128 unidades en la capa densa, con un dense dropout de 0.5, lo que permitió una abstracción adecuada de las características extraídas antes de la capa de salida.

De esta forma, ambos modelos necesitan optimizar su learning rate para poder iniciar con el entrenamiento.

C. Learning Rate

Para entender cómo se comporta el learning rate durante el entrenamiento de los modelos, se utiliza la técnica de One Cycle Policy. Por lo que, se necesita encontrar el learning rate mínimo y el máximo. Para esto, se utiliza la técnica del learning rate finder.

Se realizan varios experimentos de los cuales se pudieron obtener gráficas de los learning rates óptimos para cada uno de los modelos. Después de analizar las gráficas, se elige el mínimo y el máximo.

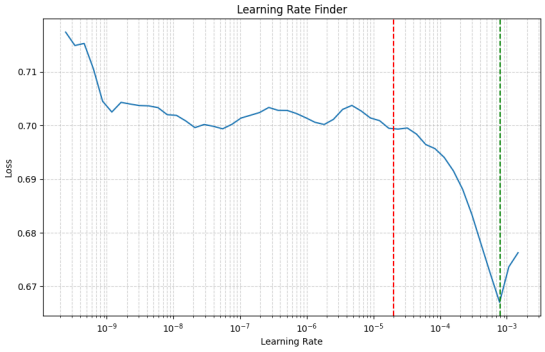

Figura 2. LRFinder modelo I3D.

En la Figura 2 LRFinder modelo *I3D*. Se puede apreciar cómo se comporta la tasa de aprendizaje para el modelo *I3D* y los valores escogidos para el máximo y el mínimo. De color rojo, el mínimo es 2×10^{-5} y el máximo es 8×10^{-4}. Estos son los valores que se utilizarán para One Cycle policy.

De forma similar, en la Figura 3 se puede apreciar el comportamiento de la tasa de aprendizaje durante el entrenamiento. Se observa un aumento de forma lineal del mínimo hasta el máximo, seguido por un descenso cosenoidal, para terminar en el mínimo nuevamente.

En la Figura 4, se ilustra el flujo de procesamiento del modelo RGB-I3D. Una secuencia de 32 fotogramas de video,

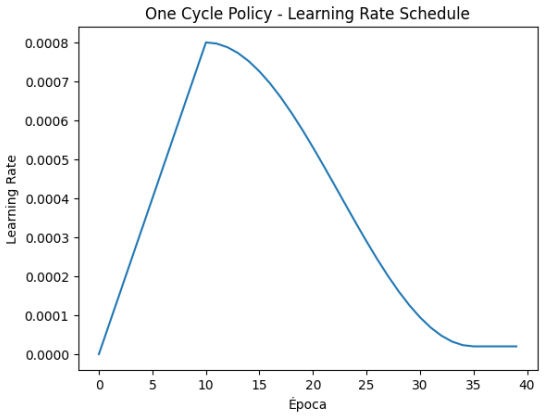

Figura 3. One Cycle modelo I3D.

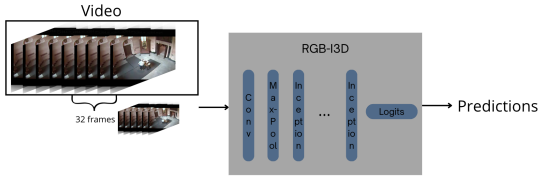

Figura 4. I3D flow diagram.

obtenidos de los videos, se introduce en el modelo, el cual luego propaga los datos a través de sus capas internas. Este proceso culmina con la generación de las predicciones finales.

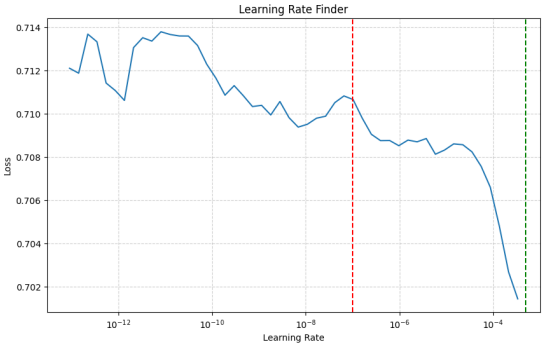

Figura 5. LRFinder modelo I3D+ConvLSTM.

En la Figura 5 LRFinder modelo *I3D+ConvLSTM*. Se puede apreciar cómo se comporta la tasa de aprendizaje para el modelo combinado con los valores escogidos para el máximo y el mínimo. De color rojo, el mínimo es 1×10^{-7} y el máximo es 5×10^{-4}. Estos son los valores que se utilizarán para One Cycle policy.

Por otro lado, en la Figura 6 se observa la forma en la que la tasa de aprendizaje se comporta durante el entrenamiento. Se puede notar que aumenta de forma lineal del mínimo hasta el máximo para luego tener un descenso cosenoidal con lo que termina en el mínimo nuevamente.

Además, la Figura 7 ilustra un flujo de procesamiento en dos etapas para los 32 fotogramas de video. La primera etapa

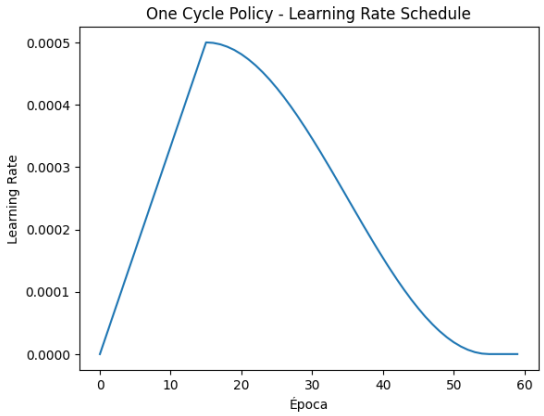

Figura 6. One Cycle modelo I3D+ConvLSTM.

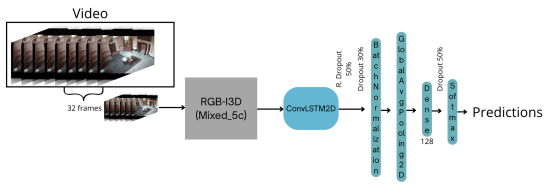

Figura 7. I3D flow diagram.

involucra el modelo RGB-I3D, que se utiliza para extraer características en el punto final *mixed-5c*; a diferencia de la Figura 4, donde se utiliza el modelo completo para la clasificación. Estas características extraídas se pasan luego por una capa ConvLSTM, seguida de normalización y promediado. Finalmente, las características procesadas se introducen en una capa densa y una activación *softmax* para generar las predicciones finales.

Con las arquitecturas de los modelos definidas y el comportamiento de la tasa de aprendizaje controlado y optimizado para cada configuración, se puede iniciar la fase de entrenamiento. Tras realizar varios experimentos, el modelo I3D mostró signos de sobreajuste alrededor de la época 40, por lo que su entrenamiento se limitó a 40 épocas. En contraste, el modelo I3D+ConvLSTM comenzó a sobreajustarse en la época 60, por lo que se entrenó durante 60 épocas. La metodología completa empleada en este proyecto se detalla en el repositorio de GitHub adjunto [8].

IV. Resultados y Discusión

A continuación, se presentan los resultados obtenidos, gráficos, métricas de desempeño y se discuten los hallazgos, resaltando las fortalezas y áreas de mejora para la investigación.

A. Entrenamiento y validación

Comenzando con el modelo *I3D*. Las gráficas de entrenamiento y validación muestran un buen comportamiento del modelo durante el entrenamiento.

Primero, en la Figura 8 se observa que la pérdida de entrenamiento (línea azul) disminuye progresivamente hasta

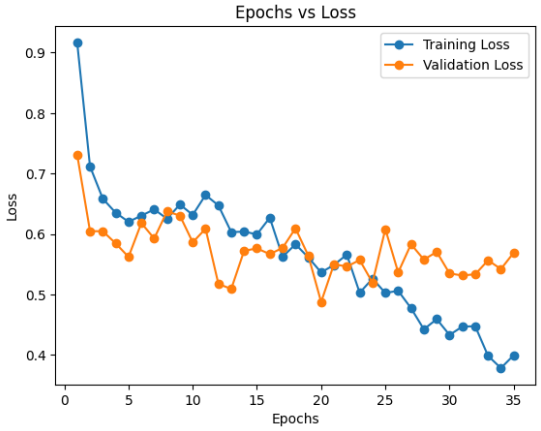

Figura 8. Épocas vs pérdida.

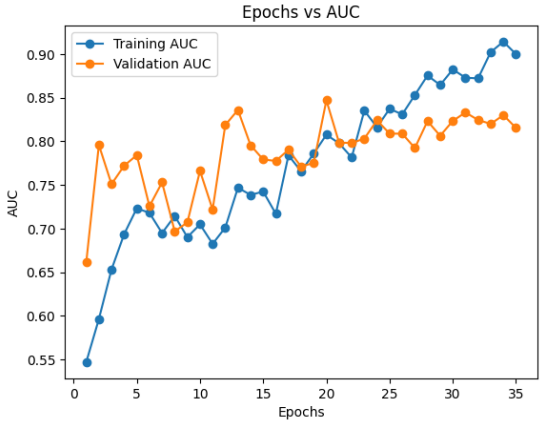

Figura 10. Épocas vs AUC.

alcanzar un valor cercano a 0.4, mientras que la pérdida de validación (línea naranja) se estabiliza alrededor de 0.55. Esta diferencia indica que el modelo logra ajustarse a los datos de entrenamiento, evitando llegar al sobreajuste, ya que no termina de entrenarse en las 40 épocas.

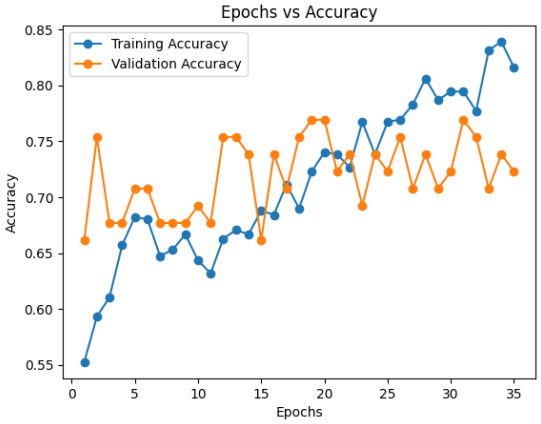

Figura 9. Épocas vs exactitud.

Sin embargo, la Figura 9 presenta que la exactitud de entrenamiento (línea azul) aumenta progresivamente hasta alcanzar un valor de 0.8, mientras que la exactitud de validación (línea naranja) se estabiliza alrededor de 0.70. Esta tendencia sugiere que el modelo mejora su capacidad de generalización, aunque existe una brecha entre el desempeño en los datos de entrenamiento y los de validación.

Además, en la Figura 10 se observa que el AUC de entrenamiento (línea azul) aumenta progresivamente hasta alcanzar un valor de 0.90 aproximadamente, mientras que el AUC de validación (línea naranja) se estabiliza alrededor de 0.80. Esta tendencia sugiere que el modelo mejora su capacidad de generalización, aunque existe una brecha entre el desempeño en los datos de entrenamiento y los de validación.

Gracias a los gráficos se puede concluir que el modelo ha logrado aprender características representativas de los datos de entrenamiento de manera efectiva. Comenzando con la Figura 8, donde se evidencia una disminución sostenida de la pérdida de entrenamiento junto con la estabilización de la pérdida de validación, lo cual sugiere que el modelo no ha alcanzado un punto crítico de sobreajuste, ya que aún existe margen de mejora si se continuara el entrenamiento. Después, la Figura 9 muestra un incremento progresivo en la exactitud de entrenamiento y una estabilización en la exactitud de validación, lo que indica una adecuada capacidad de generalización, aunque se mantiene una brecha entre ambos conjuntos. De manera similar, en la Figura 10, el comportamiento del AUC refleja un aumento consistente en el conjunto de entrenamiento y una estabilización en el conjunto de validación, evidenciando que el modelo es capaz de discriminar correctamente entre clases.

En corolario, los resultados indican que el modelo ha aprendido características relevantes sin sobreajustarse, logrando un equilibrio razonable entre desempeño en entrenamiento y en validación.

Por otra parte, los resultados de las gráficas de entrenamiento y validación para el modelo híbrido, *I3D* sumado con una capa ConvLSTM, son los siguientes:

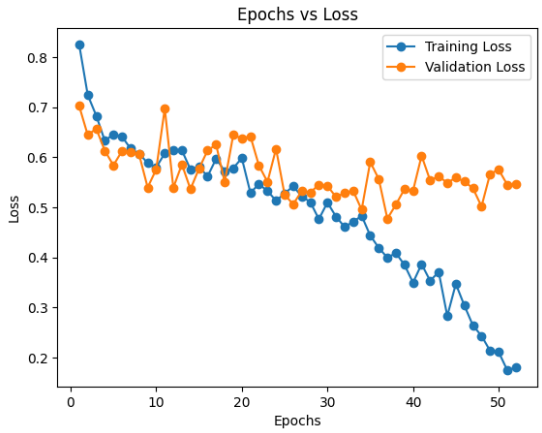

Figura 11. Épocas vs pérdida.

En la Figura 11 se observa que la pérdida de entrenamiento (línea azul) disminuye progresivamente hasta alcanzar un valor cercano a 0.2, mientras que la pérdida de validación (línea naranja) se estabiliza alrededor de 0.55. Esta diferencia indica que el modelo logra ajustarse a los datos de entrenamiento, llegando al sobreajuste en la época 40.

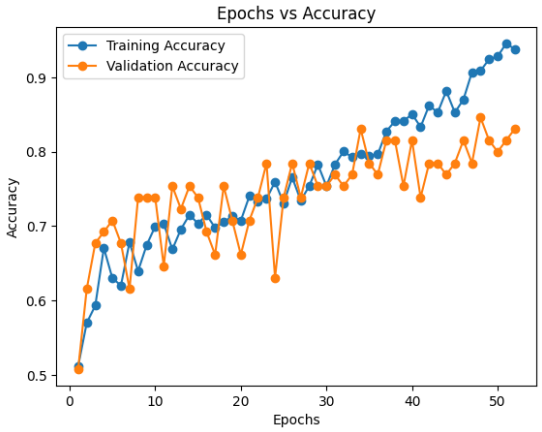

Figura 12. Épocas vs exactitud.

De igual manera, la Figura 12 presenta que la exactitud de entrenamiento (línea azul) aumenta progresivamente hasta alcanzar un valor de 0.95, mientras que la exactitud de validación (línea naranja) se estabiliza alrededor de 0.80. Aunque esta diferencia puede indicar indicios de sobreajuste, no necesariamente es crítica, ya que la exactitud de validación sigue siendo aceptablemente alta, lo que sugiere que el modelo conserva una capacidad razonable de generalización.

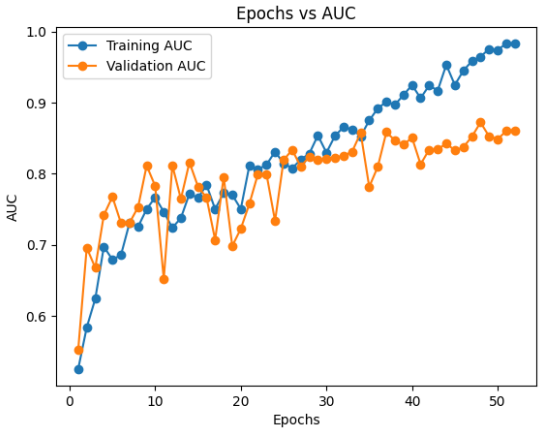

Figura 13. Épocas vs AUC.

Finalmente, en la Figura 13 se observa que el AUC de entrenamiento (línea azul) aumenta progresivamente hasta alcanzar un valor de 0.90 aproximadamente, mientras que el AUC de validación (línea naranja) se estabiliza alrededor de 0.85. Esta tendencia indica que el modelo mejora su capacidad de discriminación en el conjunto de entrenamiento, mientras mantiene un desempeño sólido en el conjunto de validación. La diferencia entre ambas curvas sugiere que el modelo generaliza de manera adecuada, con un riesgo de sobreajuste poco significativo.

Se puede apreciar que el modelo ha logrado un ajuste efectivo a los datos de entrenamiento, mostrando un comportamiento controlado en términos de generalización. En la Figura 11, la disminución de la pérdida de entrenamiento junto con la estabilización de la pérdida de validación evidencia que el modelo alcanza un buen ajuste, aunque presenta señales de sobreajuste alrededor de la época 40. Por su parte, en la Figura 12, la evolución de la exactitud refleja un desempeño elevado en el entrenamiento y una exactitud de validación aceptable, indicando que, si bien existen indicios de sobreajuste, el modelo mantiene una capacidad razonable para generalizar. Finalmente, la Figura 12 muestra un comportamiento del AUC que respalda la robustez del modelo, ya que la diferencia entre entrenamiento y validación es mínima, sugiriendo una adecuada capacidad de discriminación con bajo riesgo de sobreajuste.

En conjunto, los resultados sugieren que el modelo aprende características relevantes de los datos, alcanzando un buen compromiso entre ajuste y generalización. De igual forma, al entrenar el modelo con Keras se puede guardar el mejor modelo antes de que se note el sobreajuste.

Cuadro II
RESULTADOS DE ENTRENAMIENTO Y VALIDACIÓN.

Resultado	**I3D**		**I3D+ConvLSTM**	
	Entr.	***Val.***	***Entr.***	***Val.***
Pérdida	0.57	0.49	0.34	0.47
Exactitud	0.71	0.77	0.87	0.81
AUC	0.78	0.85	0.93	0.86

Como se muestra en el Cuadro II, el modelo *I3D+ConvLSTM* supera al modelo *I3D* en todas las métricas consideradas tanto en el conjunto de entrenamiento como en el de validación. Una mejora significativa se puede observar en la exactitud y el valor AUC durante el entrenamiento, manteniendo un desempeño competitivo en validación. Estos resultados sugieren que la incorporación de capas ConvLSTM permite capturar de manera más efectiva las dependencias temporales en los datos.

B. Evaluación en el conjunto de prueba

Una vez completado el proceso de entrenamiento, los modelos se evaluaron en el conjunto de prueba con el fin de analizar su capacidad de generalización. Esta etapa permite verificar el desempeño final de cada modelo en datos no vistos, proporcionando una medida objetiva de su efectividad en condiciones reales. Las métricas que se utilizan para la evaluación son exactitud, precisión, F1 score, AUC, y una matriz de confusión.

El Cuadro III presenta los resultados obtenidos por ambos modelos en el conjunto de prueba. Se observa que el modelo *I3D+ConvLSTM* supera consistentemente al modelo *I3D* en todas las métricas evaluadas. En particular, se destaca una

Cuadro III
DESEMPEÑO DE LOS MODELOS EN EL CONJUNTO DE PRUEBA.

Métrica	I3D	I3D+ConvLSTM
Exactitud	0.63	0.68
Precisión (avg)	0.63	0.68
F1-score (avg)	0.63	0.68
AUC	0.64	0.73

mejora en la exactitud del 0.05 (de 0.63 a 0.68), lo que sugiere una mayor capacidad del modelo para clasificar correctamente las muestras.

Además, la precisión y el F1-score también muestran un incremento, indicando que el modelo con capa ConvLSTM no solo comete menos errores en sus predicciones positivas, sino que también logra un mejor equilibrio entre precisión y sensibilidad. Finalmente, el valor del AUC (Área Bajo la Curva ROC) pasa de 0.64 a 0.73, lo cual implica que el modelo *I3D+ConvLSTM* tiene una mejor capacidad de discriminación entre clases. Estos resultados confirman que la incorporación de la capa recurrente mejora el desempeño general de la tarea.

Para analizar en detalle el comportamiento de los modelos en el conjunto de prueba, se utiliza la matriz de confusión. Gracias a esta, es posible identificar no solo la tasa de aciertos globales, sino también los patrones de confusión entre clases, lo que proporciona información valiosa para evaluar la robustez y la precisión del modelo en situaciones reales.

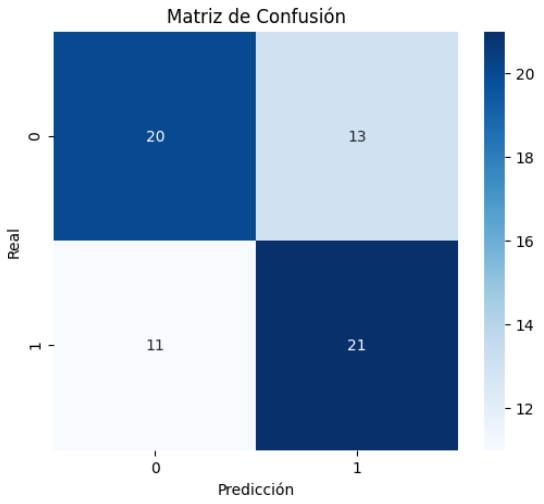

Figura 14. Matriz de confusión I3D.

Como se puede apreciar en la Figura 14, la matriz de confusión del modelo I3D permite observar el comportamiento del modelo sobre el conjunto de prueba. El modelo logró clasificar correctamente 20 muestras de la clase 0 y 21 de la clase 1, mientras que cometió 13 y 11 errores, respectivamente. Estos resultados reflejan un desempeño moderado, con una tendencia a confundir ambas clases de manera equilibrada. La precisión del modelo se ve limitada por la ambigüedad en la separación de clases, lo que sugiere que la arquitectura I3D por sí sola no es suficiente para capturar de manera óptima las características discriminativas de los datos.

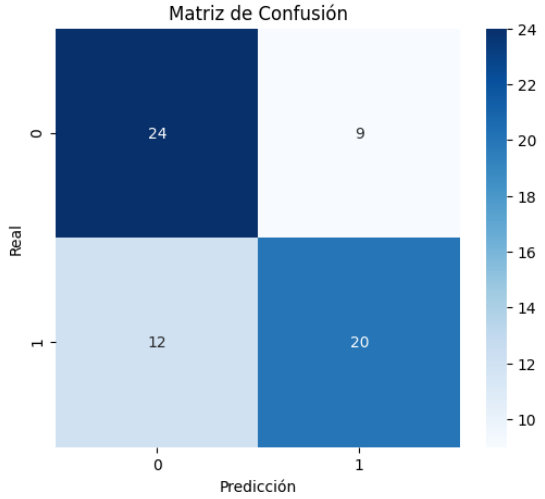

Figura 15. Matriz de confusión I3D+ConvLSTM.

Por otro lado, en la Figura 15 se presenta la matriz de confusión correspondiente al modelo *I3D+ConvLSTM*, donde se observa una mejora en la capacidad de clasificación con respecto al modelo *I3D*. En este caso, el modelo clasificó correctamente 24 muestras de la clase 0 y 20 de la clase 1, mientras que cometió 9 y 12 errores, respectivamente. Estos resultados reflejan un desempeño más equilibrado y una mejor diferenciación entre las clases, alcanzando una exactitud global del 0.68. La reducción en los errores de clasificación y la mejora general en las métricas indican que el uso de la capa ConvLSTM permitió al modelo capturar de mejor manera las dependencias temporales presentes en los datos.

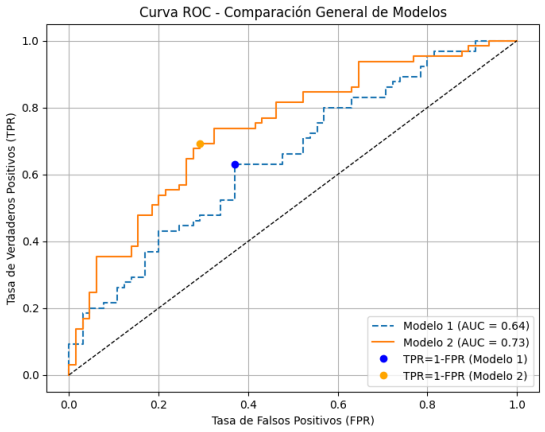

Figura 16. Curvas ROC

Asimismo, en la Figura 16 se presenta la comparación de las curvas ROC de los modelos. El modelo *I3D* obtuvo un AUC de 0.64, lo que indica una capacidad moderada para distinguir entre las clases. Por su parte, el modelo *I3D+ConvLSTM* alcanza un AUC de 0.73, lo que evidencia una mejora en su

rendimiento discriminativo. Además, se observa que el punto de equilibrio entre la tasa de verdaderos positivos (TPR) y la tasa de falsos positivos (FPR) está mejor posicionado en el modelo *I3D+ConvLSTM*, lo cual refuerza su capacidad para generalizar en la detección de eventos de crimen en el conjunto de prueba. Demostrando así que el uso de una capa ConvLSTM ayudó a capturar las características discriminativas en los datos, mejorando las predicciones en el conjunto de prueba.

Cuadro IV
RESUMEN DE PARÁMETROS ENTRENADOS EN LOS MODELOS.

Parámetro	I3D	I3D+ConvLSTM
Total de parámetros	12,318,550	17,642,710
Parámetros entrenables	12,282,034	17,605,938
Parámetros no entrenables	36,516	36,772

Se observa que los modelos propuestos cumplen con el objetivo general de apoyar la predicción y prevención de crímenes. Utilizando un conjunto de datos compuesto por 516 muestras de entrenamiento, 65 de validación y 65 de prueba, se obtienen resultados notables para los modelos. Los resultados demuestran que la capa ConvLSTM dentro de la arquitectura mejora la capacidad del modelo para capturar patrones temporales en los datos. Esta mejora es relevante para detecciones de crímenes, ya que presentan dinámicas espacio-temporales complejas. Comparado con la arquitectura sin este componente, el modelo híbrido *I3D+ConvLSTM* tiene un mejor desempeño en las métricas, lo cual respalda su utilidad en tareas de análisis predictivo con dependencias temporales.

V. CONCLUSIONES

Este trabajo presentó un sistema de detección y prevención de crímenes mediante redes neuronales profundas, específicamente utilizando las arquitecturas I3D e I3D+ConvLSTM. Los resultados experimentales revelaron que el modelo híbrido, al integrar capacidades de memoria temporal propias de redes recurrentes, logró un mejor rendimiento en métricas clave, alcanzando un AUC de 0.73 frente al 0.64 del modelo I3D. Este comportamiento resalta la efectividad de dar un mayor peso a los patrones temporales para el análisis de secuencias de video en contextos de reconocimiento de acciones.

No obstante, se identificaron desafíos como la falta de un conjunto de datos mayor, el posible sobreajuste y la discrepancia entre el rendimiento en entrenamiento y validación, lo que apunta a la necesidad de estrategias más eficaces de regularización y una mayor diversidad en los conjuntos de datos. Como trabajo futuro, se plantea como línea de investigación la incorporación de técnicas de aprendizaje autosupervisado, mecanismos de atención para mejorar la interpretabilidad y la evaluación del sistema en escenarios reales con condiciones variables y ruido ambiental. En conjunto, este estudio representa un avance hacia sistemas de vigilancia inteligente más precisos, adaptables y escalables.

Finalmente, el enfoque híbrido tiene potencial de aplicación en dominios que también dependen de patrones temporales para la detección de acciones o eventos; ejemplos de esto son la vigilancia en tiempo real en hospitales para detectar caídas o situaciones de emergencia, o el monitoreo en fábricas para prever fallos en maquinaria, o predicciones de comportamiento animal. Lo más importante a recalcar es que la implementación de este proyecto es viable gracias a la arquitectura I3D+ConvLSTM. La implementación de un sistema de supervisión y evaluación mejoraría su rendimiento, dando paso a un sistema de detección y prevención de crímenes altamente preciso. Este exhaustivo análisis demostró que el modelo podría implementarse para ayudar a las fuerzas de seguridad de un país como Ecuador a reducir el tiempo de respuesta e incluso anticipar incidentes delictivos.

REFERENCIAS

[1] Fiscalía General del Estado, "Analítica de cifras de robo," 2025, accedido el 10 de abril de 2025. [Online]. Available: https://www.fiscalia.gob.ec/analitica-cifras-de-robo/

[2] G. A. Martínez-Mascorro, J. R. Abreu-Pederzini, J. C. Ortiz-Bayliss, and H. Terashima-Marín, "Suspicious behavior detection on shoplifting cases for crime prevention by using 3d convolutional neural networks," *CoRR*, vol. abs/2005.02142, 2020. [Online]. Available: https://arxiv.org/abs/2005.02142

[3] G. A. Martínez-Mascorro, J. C. Ortiz-Bayliss, and H. Terashima-Marín, "Detecting suspicious behavior: How to deal with visual similarity through neural networks," *CoRR*, vol. abs/2007.15235, 2020. [Online]. Available: https://arxiv.org/abs/2007.15235

[4] D. K. Ghosh and A. Chakrabarty, "Two-stream multi-dimensional convolutional network for real-time violence detection," 2022. [Online]. Available: https://arxiv.org/abs/2211.04255

[5] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," 2018. [Online]. Available: https://arxiv.org/abs/1705.07750

[6] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6479–6488.

[7] E. A. Mahareek, E. K. ElSayed, N. M. ElDesouky, and K. A. ElDahshan, "Detecting anomalies in security cameras with 3d-convolutional neural network and convolutional long short-term memory," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 1, p. 993, Feb 2024.

[8] A. Herrera, "Crime-detection," https://github.com/AndresH1234/Crime_Detection, 2025, accessed: 2025-05.

[9] L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay," 2018. [Online]. Available: https://arxiv.org/abs/1803.09820

[10] A. Diba, M. Fayyaz, V. Sharma, A. H. Karami, M. M. Arzani, R. Yousefzadeh, and L. V. Gool, "Temporal 3d convnets: New architecture and transfer learning for video classification," 2017. [Online]. Available: https://arxiv.org/abs/1711.08200

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014. [Online]. Available: https://arxiv.org/abs/1409.4842

[12] E. Cesario, P. Lindia, and A. Vinci, "Multi-density crime predictor: An approach to forecast criminal activities in multi-density crime hotspots," *Journal of Big Data*, vol. 11, no. 1, May 2024.

[13] V. Sivamani, "One cycle policy-a deep understanding," May 2024. [Online]. Available: https://medium.com/@varunsivamani/one-cycle-policy-a-deep-understanding-6d4d352ec7b1

[14] A. Salimath, "How to use the learning rate finder in tensorflow," Apr 2019. [Online]. Available: https://medium.com/octavian-ai/how-to-use-the-learning-rate-finder-in-tensorflow-126210de9489

[15] L. N. Smith, "Cyclical learning rates for training neural networks," 2017. [Online]. Available: https://arxiv.org/abs/1506.01186