Reconocimiento de Patrones Criminales en Secuencias de Video con técnicas de Deep Learning

1st Wilder Buleje
Ciencia de datos
Universidad Ricardo Palma
Lima, Perú
202312449@urp.edu.pe

2nd Bryan Diaz
Ciencia de datos
Universidad Ricardo Palma
Lima, Perú
202412017@urp.edu.pe

3rd Karen Gallardo
Ciencia de datos
Universidad Ricardo Palma
Lima, Perú
202412018@urp.edu.pe

***Abstract*—Este informe presenta el desarrollo e implementación de un sistema de detección de actividades criminales en video utilizando el modelo YOLOv8, aplicado sobre un subconjunto del conjunto de datos UCF-Crime. El objetivo principal fue evaluar la viabilidad de aplicar técnicas de visión por computadora y aprendizaje profundo para identificar comportamientos anómalos en contextos de videovigilancia urbana. Se utilizaron tres clases delictivas (*fighting*, *shooting* y *abusing*) y una clase de control, con un aproximado de 40 vídeos. Los resultados muestran una precisión de detección de cajas (Box(P)) de 89.5% y un tiempo de inferencia promedio de 32.9 ms por imagen. No obstante, las métricas de *recall* y mAP fueron limitadas en clases con pocas muestras, lo que evidencia la necesidad de expandir el conjunto de datos y utilizar técnicas que incorporen más información temporal. El estudio confirma la aplicabilidad de YOLOv8 como herramienta base para sistemas de monitoreo inteligente, con un enfoque escalable y de bajo costo.**

***Index Terms*—Deep Learning, Redes Neuronales Convolucionales, Visión por computadora, Clasificación, Videos, Seguridad.**

I. Introduccion

En los últimos años, el crecimiento de los sistemas de videovigilancia ha generado una enorme cantidad de imágenes o vídeos provenientes de entornos urbanos, comerciales e institucionales. Sin embargo, la mayoría de estos sistemas siguen dependiendo de la supervisión humana para la identificación de conductas anómalas, lo cual resulta poco escalable y propenso a errores. En este contexto, el uso de técnicas de aprendizaje profundo para la automatización del análisis de video se ha convertido en una solución prometedora para la detección temprana de actividades criminales.

El presente informe tiene como objetivo principal implementar un sistema de clasificación automática de actividades criminales a partir del conjunto de datos UCF-Crime, una base de datos pública que contiene videos de vigilancia reales, en las cuales existen diferentes tipos de delitos, como por ejemplo vandalismo, robo, peleas, disparos, entre otras actividades delicuenciales. También, cuenta con vídeos en donde solo suceden situaciones cotidianas como personas caminando o paseando. Este conjunto de datos representa uno de los desafíos más complejos en el campo del reconocimiento de actividades humanas, debido a la heterogeneidad en las condiciones de grabación, iluminación, duración y perspectiva de cámara.

Para abordar este desafío, se exploran modelos basados en redes neuronales profundas, capaces de extraer características temporales y espaciales de los videos. En particular, se evalúa el desempeño de distintos enfoques para identificar patrones discriminativos que permitan clasificar secuencias de video en categorías criminales y no criminales. La correcta detección de este tipo de conductas tiene un gran valor en contextos como seguridad pública, monitoreo urbano automatizado y sistemas de alerta temprana.

Este informe se estructura de la siguiente manera: en la sección 2, se expone una revisión crítica del estado del arte, abordando los principales enfoques, algoritmos y modelos utilizados para la detección de actividades anómalas en video. La sección 3 describe en detalle la metodología seguida, incluyendo la selección del dataset, el preprocesamiento de datos, el diseño de la arquitectura del sistema y la implementación de los modelos. En la sección 4, se presentan los resultados experimentales obtenidos, acompañados de análisis cuantitativos y visuales que permiten evaluar el desempeño del sistema. La sección 5 discute las principales contribuciones del proyecto, sus limitaciones actuales y posibles líneas de mejora. Finalmente, la sección 6 expone las conclusiones generales del trabajo y su aporte al campo de la inteligencia artificial aplicada a la videovigilancia.

II. Revisión del Estado del Arte

El reconocimiento automático de actividades humanas en video ha sido un tema central de investigación en visión por computadora, particularmente en aplicaciones de seguridad, monitoreo urbano y análisis de comportamiento. En este apartado se realiza una revisión crítica de los enfoques existentes para la detección de actividades anómalas en secuencias de video, con énfasis en el uso de aprendizaje profundo y su aplicación sobre conjuntos de datos del mundo real como *UCF-Crime*.

A. Reconocimiento de Actividades en Video

El reconocimiento de actividades en video implica identificar acciones humanas a partir de secuencias de imágenes (vídeos). Inicialmente, los enfoques tradicionales se basaban en la extracción manual de características, como HOG, HOF o MBH, combinadas con clasificadores como SVM o Random Forest. Sin embargo, estas técnicas presentaban limitaciones en entornos no estructurados y bajo condiciones de grabación variables.

El auge del *deep learning*, en particular las redes convolucionales 3D (3D-CNN) y las redes recurrentes (LSTM, GRU), ha revolucionado este campo al permitir la extracción automática de representaciones jerárquicas espacio-temporales directamente desde los datos brutos. Modelos como C3D [1] e I3D [2] han demostrado ser altamente eficaces en conjuntos de datos benchmark como HMDB51 y Kinetics.

B. Modelos para la Detección de Anomalías

La detección de actividades inusuales, como por ejemplo robos, vandalismo o peleas plantea un reto adicional debido al desequilibrio de clases y la ambigüedad semántica de muchas acciones. Para abordar este problema, se han desarrollado enfoques basados en modelos generativos y de aprendizaje no supervisado, que aprenden un modelo de comportamiento "normal" y detectan desviaciones significativas.

Entre los enfoques más destacados se encuentran:

- **Autoencoders y VAEs**: Entrenados para reconstruir escenas normales, con alta tasa de error ante eventos anómalos [3].
- **GANs**: Utilizadas para generar o reconstruir secuencias de video y detectar anomalías comparando el video real y el generado [4].
- **Transformers**: Modelos como TimeSformer [5] han mostrado capacidad para modelar dependencias espacio-temporales con alta precisión.

C. UCF-Crime: Un Dataset para Escenarios del Mundo Real

El conjunto de datos *UCF-Crime* [6] es una colección extensiva de videos de vigilancia del mundo real etiquetados con 13 clases, incluyendo actividades criminales y normales. Este dataset representa un entorno altamente desestructurado, lo que lo convierte en un estándar desafiante y realista para la evaluación de modelos de visión artificial.

Diversas investigaciones han propuesto modelos para trabajar con este dataset:

- Sultani et al. [6]: propusieron una arquitectura MIL para detección anómala basada en características C3D.
- Zhong et al. [7]: introdujeron mecanismos de atención basados en grafos para explotar relaciones contextuales entre escenas.
- Hasan et al. [3]: propusieron autoencoders para videos no etiquetados, marcando una base para métodos no supervisados.

III. Metodología

El desarrollo del presente trabajo se realizó usando una metodología experimental enfocada en la implementación y evaluación de modelos de aprendizaje profundo para la clasificación de actividades criminales en videos de vigilancia. A continuación, se describe el enfoque adoptado, el conjunto de datos utilizado, el flujo de procesamiento de la información y las decisiones técnicas que guiaron el diseño del sistema propuesto.

A. Enfoque General del Sistema

El sistema fue estructurado en cinco fases principales: (1) selección y segmentación del conjunto de datos, (2) preprocesamiento de los videos, (3) extracción de características temporales y espaciales, (4) entrenamiento y validación de modelos basados en deep learning, y (5) evaluación del rendimiento con métricas cuantitativas. Esta secuencia fue diseñada para garantizar un flujo coherente desde la adquisición de datos hasta la obtención de resultados replicables.

B. Conjunto de Datos y Tareas de Clasificación

Se empleó el dataset *UCF-Crime* [6], compuesto por más de 1,900 videos de vigilancia categorizados en múltiples actividades anómalas y normales. Para este estudio, se seleccionaron cuatro clases: *shooting*, *fighting*, *abusing* y *normal activity*, priorizando aquellas con mayor representatividad visual y duración equilibrada. Debido a que habia vídeos que tenían muy baja resolución lo que hacia díficil poder segmentar a las personas agresoras o victímas.

C. Preprocesamiento de Video

En este paso de preprocesamiento se consideraron aquellos *frames* de vídeos que tuvieron mejor resolución y que contengan características relevantes para el entrenamiento. Debido a que, cuando se crearon los *frames* con un código en python ejecutado en google colab se puso que se separa cada 30FPS que representa un *frame* por segundo. Todos los videos fueron redimensionados a una resolución estándar de 640×640 píxeles. Se utilizó Roboflow para operaciones de corte y conversión, asegurando consistencia entre muestras.

Posteriormente, se ordeno por nombre de video siguiendo una secuencia numérica por *frame* y se separó en tres conjuntos de datos: entrenamiento, prueba y validación con distribuciones del 70%, 20% y 10% respectivamente

D. Arquitectura de Modelos

Como parte del enfoque propuesto, se implementó la arquitectura YOLOv8 para la tarea de detección y localización de actividades sospechosas en video. Este modelo representa la evolución más reciente de la familia YOLO (You Only Look Once), caracterizada por su alta precisión y eficiencia en tiempo real.

YOLOv8 se estructura en tres componentes principales:

- **Backbone:** basado en CSPDarknet, realiza una extracción profunda de características mediante bloques convolucionales tipo C3.
- **Neck:** adopta PANet para fusionar características de múltiples escalas, mejorando la detección de objetos de diferentes tamaños.
- **Head:** produce directamente las predicciones de clases y coordenadas mediante una salida anclada sin necesidad de postprocesamiento complejo.

Este diseño permite un balance óptimo entre velocidad y precisión, haciéndolo especialmente adecuado para aplicaciones de videovigilancia con restricciones de procesamiento en tiempo real.

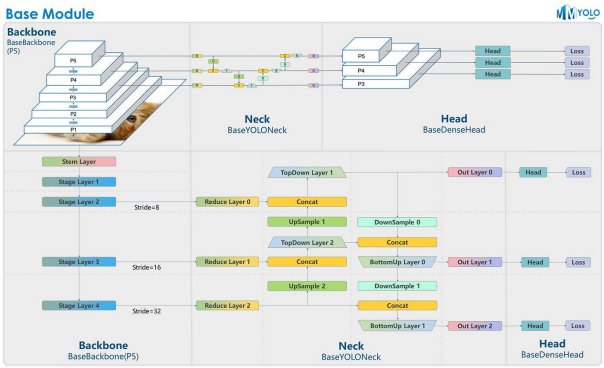

Fig. 1. Diagrama simplificado de la arquitectura YOLOv8.

E. Evaluación y Métricas

Para medir el rendimiento del modelo YOLOv8 en la tarea de detección de actividades delictivas en video, se emplearon métricas aceptadas en el campo de visión por computadora, particularmente en detección de objetos.

Las principales métricas utilizadas fueron:

- **Precision**: Indica la proporción de verdaderos positivos respecto al total de predicciones positivas realizadas. Una alta precisión implica que la mayoría de las detecciones son correctas.
- **Recall (Sensibilidad)**: Representa la proporción de verdaderos positivos respecto al total de instancias reales positivas. Evalúa la capacidad del modelo para no omitir objetos relevantes.
- **mAP@0.5**: Corresponde al *mean Average Precision* calculado con un umbral de IoU (Intersection over Union) de 0.5. Es una métrica integral que considera tanto precisión como recall en distintas clases.
- **mAP@0.5:0.95**: Promedia el valor del AP en diferentes umbrales de IoU, desde 0.5 hasta 0.95 en incrementos de 0.05. Es una métrica más exigente, recomendada por benchmarks como COCO.
- **FPS (Frames Per Second)**: Mide la velocidad de procesamiento del modelo, indicando cuántos frames puede analizar por segundo. Esta métrica es crítica en aplicaciones en tiempo real como la videovigilancia.

Estas métricas fueron generadas automáticamente por el entorno de entrenamiento de Ultralytics durante la validación del modelo. El conjunto de validación se mantuvo equilibrado en clases para asegurar una medición justa del rendimiento, y los valores de precisión y *recall* se reportaron por clase y como promedio global.

F. Recursos usados para entrenamiento

El entrenamiento y la evaluación del modelo se llevaron a cabo utilizando el entorno en la nube de Google Colab, en su versión gratuita. Esta plataforma ofrece recursos computacionales limitados pero adecuados para prototipos y experimentación en aprendizaje profundo.

Durante las sesiones, el entorno proporcionó acceso a una GPU NVIDIA Tesla T4 con 16 GB de VRAM, lo que permitió realizar la inferencia acelerada por hardware y entrenar modelos de detección con eficiencia razonable. El entorno se ejecutó sobre un sistema operativo Ubuntu virtualizado, con Python 3.10 y bibliotecas clave como `Ultralytics`, `OpenCV` y `Matplotlib`, instaladas mediante `pip`.

La configuración empleada fue suficiente para ejecutar inferencias en videos de resolución estándar (640x640) y para ajustar modelos preentrenados mediante *fine-tuning* sobre subconjuntos del dataset *UCF-Crime*. Aunque el tiempo de entrenamiento fue mayor comparado con estaciones de trabajo locales de alto rendimiento, el uso de Google Colab demostró ser una alternativa viable para investigación académica con recursos accesibles.

IV. Resultados

El entrenamiento del modelo YOLOv8 se realizó empleando un subconjunto del dataset UCF-Crime, compuesto por tres clases de comportamiento delictivo: *fighting*, *shooting* y *abusing*, junto con una clase de control *normal activity*. Para cada clase, se utilizaron 10 videos, que se consideró que tenían las mejores resoluciones para poder segmentar y

poder generar las clases. En total se obtuvieron 1857 *frames*, las cuales se usaron 1299 imágenes para el conjunto de entrenamiento, 372 archivos para el conjunto de validación y 186 para el conjunto de prueba. Al ejecutar el código se obtuvieron las siguientes clases que se llegaron a etiquetar en las imágenes:

- **0**: ucf_Agresor
- **1**: ucf_Persona
- **2**: ucf_Victima
- **3**: ucf_atacante
- **4**: ucf_carro
- **5**: ucf_persona
- **6**: ucf_tirador

La tabla I muestra el resultado del entrenamiento en la **época 10**, la cual fue la época con su mejor desempeño, deteniéndose mediante *EarlyStopping* tras no observarse mejoras significativas en las cuatro épocas siguientes. El modelo obtuvo una precisión (Box(P)) global de **0.895**, con un mAP@0.5 de **0.0247** y mAP@0.5:0.95 de **0.0113** , valores modestos considerando el reducido tamaño del dataset.

TABLE I
RESUMEN DE MÉTRICAS DEL MODELO YOLOV8

Métrica	**Valor**
Precisión (Box(P))	0.895
Recall (R)	0.0226
mAP@0.5	0.0247
mAP@0.5:0.95	0.0113
Frames procesados	372
Instancias detectadas	764
Velocidad de inferencia	32.9 ms/image

En cuanto al rendimiento por clase, se observaron diferencias significativas. La clase *persona* presentó un *recall* de 0.158 y un *mAP@0.5* de 0.139, mientras que otras clases como *agresor*, *víctima* o *tirador* presentaron valores cercanos a cero, lo que refleja un bajo nivel de detección o generalización, como se puede observar en la tabla II.

TABLE II
EVALUACIÓN POR CLASE (MÉTRICAS PRINCIPALES)

Clase	**Precisión**	**Recall**	**mAP@0.5**	**mAP@0.5:0.95**
ucf_Agresor	1.000	0.000	0.0060	0.0010
ucf_Persona	0.263	0.158	0.1390	0.0681
ucf_Victima	1.000	0.000	0.0016	0.0005
ucf_Atacante	1.000	0.000	0.0016	0.0004
ucf_Carro	1.000	0.000	0.0029	0.0008
ucf_Tirador	1.000	0.000	0.0028	0.0012
Promedio	0.895	0.0226	0.0247	0.0113

El modelo fue capaz de procesar los videos a una velocidad aproximada de **32.9 ms por imagen** en la etapa de inferencia, lo que permite su uso potencial en entornos de videovigilancia en tiempo casi real.

V. DISCUSIÓN

El experimento confirmó la capacidad del modelo YOLOv8 para ser entrenado y ejecutado eficientemente en un entorno accesible como Google Colab. Sin embargo, el desempeño observado sugiere que el conjunto de datos empleado resultó insuficiente para lograr una generalización robusta en todas las clases.

Las métricas obtenidas muestran un **desempeño aceptable en clases más visibles o frecuentes**, como *persona*, pero una **precisión mínima en clases críticas** como *agresor*, *víctima* o *tirador*. Esta limitación puede explicarse por varios factores:

- **Cantidad limitada de videos**: se utilizaron solo 10 videos por clase, lo cual reduce la diversidad de escenarios, ángulos de cámara y condiciones de iluminación.
- **Número reducido de instancias por clase**: aunque se lograron 764 instancias en total, muchas clases aparecen con muy pocos ejemplos, dificultando el aprendizaje discriminativo.
- **Ambigüedad entre clases**: algunas acciones pueden ser visualmente similares, especialmente si los movimientos son sutiles o la escena está parcialmente oculta.
- **Etiquetado automático o incompleto**: en conjuntos derivados de videos largos, la exactitud de las etiquetas puede verse afectada si no se realiza una segmentación temporal adecuada.

Por otro lado, el tiempo de inferencia promedio de 32.9 ms por imagen posiciona al modelo como un candidato viable para tareas de detección en tiempo casi real, lo que respalda su uso en entornos operativos con recursos limitados.

Para mejorar los resultados, se recomienda ampliar el conjunto de entrenamiento, incorporar técnicas de aumento de datos y explorar arquitecturas que integren información temporal, como TimeSformer o TPN, para captar mejor la dinámica de las actividades delictivas.

VI. CONCLUSIONES

El presente estudio demostró la viabilidad técnica de implementar un sistema de detección automática de comportamientos criminales en video utilizando el modelo YOLOv8, aplicado sobre un subconjunto del conjunto de datos UCF-Crime. Se entrenó el modelo con tres clases representativas de actividades criminales tales como *fighting*, *shooting* y *abusing* más una clase de control de actividad normal.

A pesar de la simplicidad de la configuración experimental (uso de Google Colab gratuito y un conjunto de datos reducido), el modelo logró alcanzar una precisión global de 89.5% y un tiempo de inferencia promedio de 32.9 ms por imagen, lo que evidencia su aplicabilidad en entornos de videovigilancia en tiempo casi real.

Sin embargo, los resultados también reflejan limitaciones significativas, especialmente en términos de *recall* y mAP en clases específicas, donde se observaron valores cercanos a cero. Esta situación se atribuye principalmente a:

- La baja cantidad de ejemplos por clase (10 videos por categoría).
- La escasa variabilidad de escenas y ángulos de grabación.
- La dificultad inherente a detectar microacciones o agresiones sutiles en imágenes de baja resolución.

En consecuencia, el trabajo establece una base funcional sobre la cual se pueden construir sistemas más complejos y robustos, incorporando mayores volúmenes de datos, técnicas de aumento de datos, o modelos con mecanismos de atención temporal para mejorar la sensibilidad ante secuencias de acción.

Finalmente, esta investigación aporta evidencia empírica del potencial de modelos ligeros como YOLOv8 para tareas de seguridad urbana, permitiendo su despliegue incluso en contextos de computación en la nube con recursos limitados.

REFERENCES

[1] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," *Proceedings of the IEEE international conference on computer vision*, pp. 4489–4497, 2015.

[2] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[3] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning temporal regularity in video sequences," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 733–742.

[4] M. Ravanbakhsh, M. Nabi, S. I. Mousavi, E. Sangineto, and N. Sebe, "Abnormal event detection in videos using generative adversarial nets," in *IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 1577–1581.

[5] G. Bertasius, H. Wang, and L. Torresani, "Space-time attention for video action recognition," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19 178–19 188, 2021.

[6] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6479–6488.

[7] Z. Zhong, D. Wang, H. Liu, and et al., "Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1237–1246.