

# Aproximación a la identificación de la minería Ilegal en videos de cámaras FLIR a partir de la detección y la diferenciación de carreteras no pavimentadas y cuerpos de agua

María Alejandra Ariza Rangel\*, Camilo Andrés Daza Ramírez†,  
María Paola Reyes Gómez\*, Juan Diego Yepes Parra‡

\*Maestría en Biología Computacional,

†Pregrado en Ingeniería de Sistemas y Computación,

‡Maestría en Ingeniería de Sistemas y Computación,

Facultad de Ingeniería, Universidad de los Andes, Bogotá, Colombia,

Facultad de Ciencias, Universidad de los Andes, Bogotá, Colombia

**Resumen**—La Fuerza Aeroespacial Colombiana (FAC), en colaboración con la Universidad de los Andes, desarrolló un sistema automatizado para detectar minería ilegal en la Amazonía mediante el análisis de imágenes térmicas FLIR capturadas por sensores aerotransportados. El sistema se centra en la detección de carreteras no pavimentadas y cuerpos de agua como indicadores indirectos de actividad minera.

Se propuso un esquema de preprocesamiento multicanal que mejora la calidad visual de las imágenes térmicas al combinar representaciones sintéticas optimizadas para texturas, estructuras y segmentación espacial. Posteriormente, se evaluaron modelos de aprendizaje profundo, destacándose YOLOv11 con un mAP@50 de 65.2 % y YOLOv11s con un mAP@50–95 de 32.8 %, superando incluso a modelos más robustos. En contraste, RetinaNet obtuvo resultados limitados (mAP@50–95 de 12.2 %) debido a su sensibilidad al desbalance de clases y a las distorsiones térmicas.

Los resultados validan la viabilidad técnica del enfoque propuesto y su potencial integración en sistemas reales de monitoreo ambiental y operaciones militares. El sistema ofrece beneficios clave como la reducción del tiempo de análisis, la mejora en la segmentación y la toma de decisiones más precisa. Además, puede escalarse a otras aplicaciones, como la detección de deforestación, la gestión de recursos hídricos y el soporte a políticas públicas de conservación ambiental.

**Index Terms**—Palabras clave; separadas; por punto y coma.

## I. INTRODUCCIÓN

La minería ilegal representa una de las amenazas ambientales más críticas para la Amazonía colombiana. Sus impactos abarcan desde la deforestación acelerada hasta la contaminación de cuerpos hídricos por metales pesados como el mercurio, afectando no solo la biodiversidad, sino también la salud humana y la seguridad territorial. En respuesta a esta problemática, la Fuerza Aeroespacial Colombiana (FAC), en colaboración con la Universidad de los Andes, ha impulsado el uso de tecnologías emergentes para fortalecer la vigilancia aérea y la identificación de actividades ilícitas.

La vigilancia mediante cámaras térmicas FLIR (Forward-Looking Infrared), instaladas en aeronaves, permite obtener

imágenes de alta cobertura geográfica incluso en condiciones de baja visibilidad. No obstante, el análisis manual de estos videos presenta limitaciones operativas, como altos costos de tiempo, fatiga visual y subjetividad en la interpretación. Ante este desafío, las técnicas de aprendizaje profundo (deep learning) y visión por computador se consolidan como herramientas prometedoras para automatizar la detección de elementos indicativos de minería ilegal, como carreteras no pavimentadas y cuerpos de agua, los cuales suelen estar asociados a campamentos mineros clandestinos.

Diversos estudios han demostrado la eficacia de estas técnicas en contextos similares. Por ejemplo, Pardini et al. [1] lograron reducir significativamente los falsos positivos en la detección de pistas aéreas ilegales en la Amazonía mediante redes neuronales convolucionales paralelizadas. Ferreira et al. [3], por su parte, aplicaron un enfoque de fusión de datos para la identificación de cultivos ilícitos, alcanzando una precisión del 92.16 %. En el ámbito de la gestión hídrica, Teixeira et al. 2[2] utilizaron redes neuronales profundas para segmentar embalses en zonas semiáridas de Brasil con un 95 % de precisión en la métrica de Intersección sobre Unión (IoU). Estos antecedentes respaldan la aplicabilidad de modelos basados en deep learning para abordar problemas complejos de análisis territorial y vigilancia ambiental.

En este contexto, el presente trabajo tiene como objetivo general diseñar y desarrollar un modelo basado en aprendizaje profundo para el procesamiento de imágenes extraídas de videos capturados por cámaras FLIR, con el fin de mejorar su calidad, eliminar ruido y automatizar la detección de cuerpos de agua y carreteras no pavimentadas. La finalidad es facilitar la identificación automática de estos elementos y contribuir al reconocimiento de posibles campamentos asociados a la minería ilegal. Para lograrlo, se plantearon los siguientes objetivos específicos: (1) Implementar técnicas de preprocesamiento de imágenes que permitan mejorar la calidad de

los fotogramas extraídos de los videos, eliminando ruido y mejorando el contraste, con una reducción mínima del 20 % en la distorsión visual. (2) Diseñar y entrenar un conjunto de datos etiquetado específicamente para la identificación de carreteras no pavimentadas y cuerpos de agua en imágenes térmicas. (3) Desarrollar un modelo de aprendizaje profundo basado en redes neuronales convolucionales (CNN), utilizando arquitecturas como YOLO y Faster R-CNN, para la detección y clasificación de estos elementos. (4) Evaluar el desempeño de los modelos mediante métricas de detección como mAP50 y mAP50-95 con el fin de validar su aplicabilidad en escenarios operacionales reales.

## II. ESTADO DEL ARTE

Recientemente, el aprendizaje automático y la visión por computadora han mejorado significativamente el monitoreo ambiental mediante imágenes satelitales, especialmente en áreas como la detección de pistas clandestinas en la Amazonía. En este contexto, Pardini et al. [1] desarrollaron un modelo basado en redes neuronales para reducir los falsos positivos en un 26.6 %, optimizando el procesamiento de 43 a 32 horas mediante paralelización.

En la gestión hídrica, Albuquerque Teixeira et al. [2] aplicaron redes neuronales para segmentar embalses en Brasil, alcanzando una precisión del 95 % en la Intersección sobre Unión (IoU), mejorando así la monitorización de recursos hídricos en zonas semiáridas.

Por otro lado, Ferreira et al. [3] implementaron una fusión de datos de teledetección para detectar cultivos ilícitos, alcanzando una precisión del 92.16 % y reduciendo los falsos positivos al 5.87 %. Este enfoque es particularmente útil para identificar cultivos ilegales, como la *Cannabis Sativa*, en Brasil.

Pinto Hidalgo et al. [4] crearon un modelo con redes neuronales y datos geoespaciales para detectar infraestructuras de producción de coca en la frontera Venezuela-Colombia. Su metodología mejoró la vigilancia en zonas de difícil acceso, usando imágenes satelitales y bases de datos geoespaciales.

En cuanto a la detección de carreteras ilegales, Sloan et al. [5] utilizaron redes U-Net y ResNet-34 para identificar carreteras en Asia Pacífico con una precisión entre el 72 % y el 81 %. Su metodología podría aplicarse en ecosistemas tropicales para monitorear el impacto ambiental de infraestructuras no autorizadas.

A pesar de estos avances, aún existen desafíos en la optimización de modelos para diferentes condiciones geográficas y climáticas, y en la reducción de costos computacionales sin afectar la precisión. Estos estudios ofrecen valiosas herramientas para mejorar la vigilancia ambiental y la gestión sostenible de los recursos naturales.

## III. MÉTODOS

### III-A. Obtención de datos

Se dispone de 25 minutos de video etiquetado, el cual ha sido dividido en alrededor de 4,000 fotogramas en formato PNG. Estos fotogramas pertenecen a las Fuerzas Aéreas

Colombianas (FAC) y provienen de cámaras FLIR (Forward-Looking Infrared), sensores térmicos avanzados que permiten la detección de objetos y actividades a partir de la radiación infrarroja emitida por los cuerpos. Los fotogramas incluyen diversas capturas de territorios con características variadas, tales como montañas, cuerpos de agua, carreteras, vehículos, entre otros. Además, se cuenta con las coordenadas geográficas de las imágenes para facilitar la identificación de objetos dentro de ellas. No obstante, los fotogramas presentan ciertos problemas, como exceso de ruido, distorsión, sombras y baja resolución; así que fue necesario hacer un preprocesamiento de éstas (ver sección III-B).

### III-B. Preprocesamiento de las imágenes

Los datos fueron preprocesados en Python utilizando la biblioteca scikit-image, la cual forma parte del ecosistema de scikit-learn y ofrece una amplia colección de algoritmos para el procesamiento de imágenes. Las imágenes empleadas contienen tres canales correspondientes al modelo RGB (rojo, verde y azul). Para el preprocesamiento, se trabajó inicialmente con cada banda por separado y, posteriormente, se integraron para formar la imagen compuesta.

Adicionalmente, se aplicaron técnicas de aumentación de datos con el objetivo de incrementar el número de muestras, aportar mayor variabilidad y reducir el riesgo de sobreajuste (*overfitting*). Esta fase se llevó a cabo manipulando las representaciones de cada imagen a través de sus respectivos canales.

En el **canal rojo**, el preprocesamiento se enfocó en la reducción de ruido y la mejora de la calidad de la imagen. Inicialmente, se aplicó un filtro gaussiano, que suaviza la imagen sin introducir artefactos indeseados mediante una distribución gaussiana que pondera los píxeles vecinos. Se utilizó un valor de  $\sigma = 1.5$ , correspondiente a un suavizado intermedio. La imagen resultante se convirtió al tipo de datos `uint8`. Luego, se aplicó un filtro mediano, útil para eliminar ruido del tipo sal y pimienta y preservar los bordes. Este filtro se aplicó con una ventana de 5x5 píxeles.

Finalmente, se aplicó un filtro guiado no lineal que emplea la imagen original como guía para suavizar la imagen previamente filtrada. Este tipo de filtro conserva los bordes con mayor precisión. Se utilizó un radio de 5 y un valor de  $\epsilon = 10^{-2}$ , lo suficientemente bajo para preservar detalles importantes. El resultado fue una imagen suavizada con bordes claramente definidos, lo cual es fundamental para identificar adecuadamente elementos como carreteras sin pavimentar y cuerpos de agua.

En el **canal azul**, se utilizó la segmentación basada en el modelo de *Chan-Vese*, que separa automáticamente la imagen en regiones homogéneas en función de similitudes en color, textura o intensidad. El objetivo fue distinguir entre la *región de interés* (carreteras y cuerpos de agua) y el fondo.

Los parámetros empleados fueron:

- $\mu = 0.25$ : regula el suavizado de las fronteras segmentadas. Un valor bajo favorece la preservación de bordes.
- $t_{\text{tol}} = 10^{-3}$ : umbral de tolerancia para la convergencia del algoritmo.

- Número máximo de iteraciones: 100, con un paso de actualización de 0.5, lo cual promueve una segmentación precisa.

Este proceso permitió aislar regiones homogéneas, facilitando la eliminación del fondo y resaltando los objetos de interés.

En la **banda verde** se aplicó un filtro lineal del tipo Gabor, ideal para capturar patrones espaciales, especialmente texturas, en diferentes orientaciones y escalas. Se construyó un banco de filtros Gabor con diversas orientaciones, cuyas respuestas se promediaron para combinar la información de textura. Posteriormente, la imagen combinada fue normalizada al rango [0, 255], preparando los valores de los píxeles para su incorporación en el canal verde.

Este procesamiento es clave para resaltar patrones y texturas útiles en tareas de segmentación y detección de objetos en escenas complejas.

Finalmente, las imágenes resultantes de cada canal fueron combinadas para formar una única imagen RGB en la que cada banda aporta información única y complementaria.

### *III-C. Modelos de Detección de Objetos*

Para llevar a cabo la detección de objetos en las imágenes, se implementaron diversos modelos de aprendizaje profundo, específicamente: **YOLO**, **Faster R-CNN** y **RetinaNet**.

*III-C1. YOLO (You Only Look Once):* YOLO es un algoritmo de detección de objetos en tiempo real ampliamente utilizado en visión por computador. Su enfoque consiste en dividir una imagen en una cuadrícula y, en una sola pasada, predecir simultáneamente las cajas delimitadoras (bounding boxes) y las clases de los objetos. Gracias a esta arquitectura unificada, YOLO ofrece un excelente equilibrio entre velocidad y precisión.

En este proyecto se trabajó con tres variantes de YOLO, todas entrenadas con imágenes propias del conjunto de datos.

- **YOLO Small:** Esta versión del modelo es más liviana y está optimizada para funcionar en dispositivos con recursos computacionales limitados. Se entrenó desde cero utilizando la implementación de la librería *Ultralytics*. El proceso de entrenamiento se realizó con un máximo de **50 épocas** y un tamaño de imagen de **640 píxeles**, buscando un equilibrio entre velocidad de entrenamiento y precisión. Se utilizó un **batch size de 16**, y se aplicó la técnica de *early stopping* con una paciencia de 8 épocas. Además, se habilitó la **aumentación de datos** para mejorar la capacidad de generalización del modelo, introduciendo transformaciones aleatorias como rotaciones, escalados y variaciones en la iluminación. Se registró el tiempo total de entrenamiento para evaluar la eficiencia del modelo.
- **YOLO Large:** Esta variante representa una versión más robusta y precisa del modelo YOLO. También fue entrenada desde cero con el dataset propio, siguiendo una configuración similar a la del modelo Small: **50 épocas**, tamaño de imagen de **640 píxeles**, **batch size de 16** y **early stopping** con paciencia de 8 épocas. La diferencia principal radica en su mayor capacidad de representación,

lo que se traduce en un mejor rendimiento en tareas complejas, aunque con un mayor consumo de memoria y tiempo de entrenamiento. Se activó igualmente la aumentación de datos para reforzar la generalización del modelo ante condiciones variadas.

- **YOLO Large con fine-tuning (congelando el backbone):** En esta tercera configuración se aplicó una estrategia de *transfer learning*, partiendo de un modelo YOLO Large preentrenado en un conjunto de datos extenso. Se congeló el **backbone** (capas encargadas de extraer características generales), entrenando únicamente las capas finales responsables de la detección (*head*). Para ello, se bloquearon explícitamente las primeras 14 capas del modelo (model.0 a model.13), lo cual se reforzó utilizando el parámetro *freeze=14*. Como resultado, solo alrededor del **3 %** de los parámetros del modelo fueron entrenables, reduciendo significativamente el tiempo y los recursos necesarios. Esta variante fue entrenada durante **20 épocas**, con un **batch size de 8**, utilizando un **learning rate reducido de 0.001** y el optimizador **Adam**, ideal para ajustes rápidos. Se habilitó el uso de GPU si estaba disponible, y el modelo resultante fue exportado en formato **ONNX**, permitiendo su integración en sistemas de inferencia en tiempo real. Esta estrategia permitió adaptar eficazmente el modelo a un nuevo dominio de aplicación, aprovechando los conocimientos previos del modelo base.

*III-C2. Faster R-CNN:* Faster R-CNN es un modelo de detección de objetos de dos etapas ampliamente reconocido por su alta precisión. Su arquitectura combina una red de propuestas de regiones (*Region Proposal Network*, RPN) con una segunda etapa que clasifica y ajusta las cajas generadas. En este trabajo, se entrena el modelo desde cero utilizando imágenes propias y anotaciones personalizadas. El pipeline de entrenamiento y evaluación fue desarrollado en PyTorch, partiendo de un conjunto de imágenes anotadas en formato YOLO, cuyas etiquetas son convertidas a coordenadas absolutas  $[x_1, y_1, x_2, y_2]$  compatibles con Faster R-CNN. El modelo base es una versión preentrenada de fasterrcnn\_resnet50\_fpn, al cual se le reemplaza la cabeza de clasificación para detectar dos clases específicas: “carretera” y “río”, además de la clase de fondo. Durante el entrenamiento, se emplean técnicas como *gradient clipping* para mejorar la estabilidad numérica, *learning rate scheduling* para una mejor convergencia, y *early stopping* para prevenir el sobreajuste. El mejor modelo se guarda automáticamente según su rendimiento en la validación. Aunque este enfoque conlleva mayores tiempos de inferencia comparado con modelos más livianos como YOLO, ofrece resultados sobresalientes en términos de precisión.

*III-C3. RetinaNet:* RetinaNet es un modelo de detección de objetos de una sola etapa que destaca por su uso de la función de pérdida *Focal Loss*, diseñada específicamente para mitigar el impacto del desbalance de clases durante el entrenamiento. El modelo fue entrenado desde cero utilizando nuestras propias imágenes, implementando un pipeline completo

en PyTorch y Torchvision. Se define una clase personalizada `RetinaNetDataset` que carga imágenes y etiquetas en formato YOLO, transformando las coordenadas normalizadas en cajas delimitadoras absolutas compatibles con RetinaNet. Se construyen `DataLoaders` para los conjuntos de entrenamiento y validación, garantizando un rendimiento eficiente. Durante el entrenamiento, se permite la actualización de todos los parámetros del modelo para maximizar el aprendizaje.

Para todos los modelos, se evalúa el rendimiento periódicamente utilizando métricas estándar como  $mAP@0.5$  y  $mAP@0.5:0.95$ , que miden la precisión en la detección y clasificación correcta de objetos. Este enfoque asegura una evaluación rigurosa y comparativa de la efectividad de los modelos entrenados.

### *III-D. Despliegue*

Para facilitar la interacción del usuario con los modelos de detección de ríos y carreteras, se diseñó un sistema modular compuesto por dos componentes principales: un **frontend** y un **backend**. Esta arquitectura se eligió con el objetivo de separar las responsabilidades entre la presentación de la interfaz y la lógica de procesamiento, permitiendo así una mayor flexibilidad en el desarrollo.

El **backend** fue planteado en Python, dado que todos los modelos fueron desarrollados y entrenados en este lenguaje utilizando bibliotecas como PyTorch y Ultralytics. Se propuso el uso de un framework ligero como FastAPI para exponer servicios web que reciban imágenes desde el cliente, procesen la solicitud con el modelo seleccionado por el usuario y retornen los resultados en un formato estructurado. La arquitectura considera la posibilidad de cargar dinámicamente distintos modelos según la petición, evitando mantenerlos todos en memoria de forma simultánea.

El **frontend** se desarrolló utilizando el framework Svelte, debido a su simplicidad, bajo peso y rapidez de desarrollo. La interfaz fue concebida como una herramienta sencilla, enfocada en la usabilidad: permite al usuario cargar una o varias imágenes, seleccionar el modelo de detección que desea emplear, y enviar la solicitud al backend. La comunicación entre frontend y backend se plantea mediante peticiones HTTP tipo POST con los archivos y parámetros necesarios.

En cuanto al despliegue, se contempló inicialmente un entorno local para facilitar la prueba y validación del sistema durante el desarrollo. No obstante, dado el requerimiento de que este despliegue debe ser consultable para el equipo docente, se utilizaron contenedores (`Docker`) con servicios de la nube de AWS, para poder tener la solución en un ambiente productivo.

Todo el código fuente del proyecto, el cual contiene esta misma documentación, el back-end, front-end, y los notebooks utilizados para varios propósitos, se encuentran en el siguiente repositorio público de Github: <https://github.com/juanyepesp/isis4825-proyecto-final>

## IV. RESULTADOS

### *IV-A. Desafíos de captura y fusión multicanal de imágenes térmicas*

Los videos capturados por las cámaras FLIR de la Fuerza Aérea Colombiana (FAC) enfrentaron una serie de desafíos técnicos derivados de las complejas condiciones de captura aérea en la región amazónica colombiana. Estas imágenes térmicas estuvieron afectadas por múltiples problemas de calidad que dificultaron su procesamiento automático. Entre los factores más críticos se identificaron: un elevado nivel de ruido térmico, generado por las fluctuaciones del sensor y las condiciones ambientales extremas; distorsiones producto del movimiento constante y las vibraciones de la aeronave; sombras pronunciadas que ocultaban parcialmente los objetos de interés; y una resolución limitada, afectada tanto por la densa atmósfera como por la distancia entre la cámara y el terreno.

Estas condiciones adversas se reflejaron directamente en los resultados obtenidos en una fase preliminar, donde los modelos de detección fueron aplicados sobre imágenes sin ningún tipo de preprocesamiento. En esta etapa inicial, el desempeño fue considerablemente deficiente, con dificultades notorias para detectar correctamente los elementos relevantes en las escenas. Estos resultados iniciales subrayaron la necesidad imperativa de implementar un conjunto de técnicas de preprocesamiento especializadas, destinadas a mejorar la calidad y la coherencia de las imágenes antes de su uso en algoritmos de detección de objetos. Dicha preparación previa se vuelve fundamental para maximizar la efectividad de los modelos y garantizar resultados fiables y robustos en este contexto tan desafiante.

En la Figura 1 se presenta una visualización del esquema de preprocesamiento diseñado, el cual se basa en la integración de información complementaria a través de una imagen RGB sintética. Cada uno de los tres canales fue generado mediante transformaciones dirigidas a resaltar aspectos específicos de la imagen, maximizando así la riqueza de la representación resultante.

- El **canal rojo (R)** contiene la imagen suavizada tras aplicar una secuencia de filtros especializados: Gaussiano (para atenuar ruido global), Mediano (para eliminar ruido impulsivo sin afectar bordes), y Guiado (para conservar contornos estructurales relevantes). Esta combinación mejora significativamente la nitidez de las formas, permitiendo una mejor identificación de objetos lineales como carreteras o cuerpos de agua.
- El **canal verde (G)** representa una codificación textural obtenida a partir de un banco de filtros de Gabor. Estas funciones permiten capturar características orientadas, tales como rugosidad del terreno o variaciones de vegetación, útiles para segmentar regiones según su morfología superficial.
- El **canal azul (B)** incorpora una segmentación binarizada obtenida mediante el algoritmo Chan-Vese. Este procedimiento reduce la complejidad de la escena, facilita la

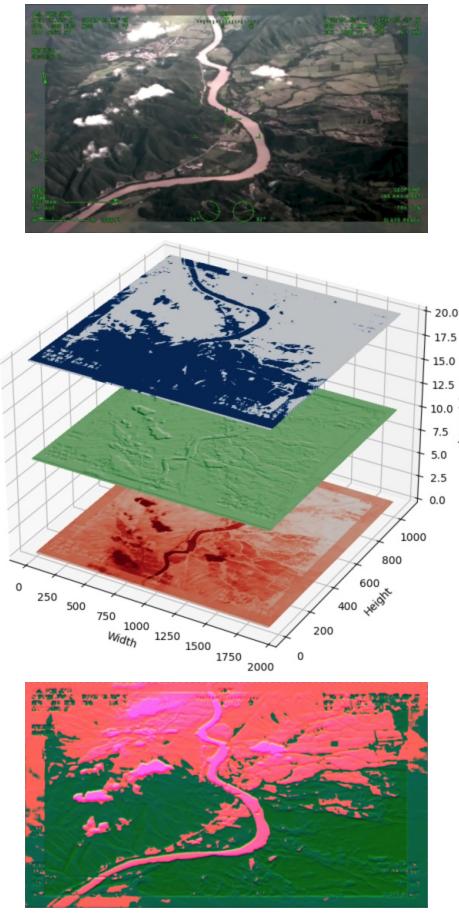


Figura 1. Fusión multicanal de preprocesamiento. Arriba: Imagen original. Centro: canal azul (segmentación chan-vese binarizada), canal rojo (imagen suavizada sin ruido), canal verde (respuestas texturales de filtros de Gabor) Abajo: imagen RGB resultante de la combinación de canales.

eliminación de fondo y permite enfocar la atención del modelo en las regiones más relevantes.

La fusión de estos tres canales da lugar a una imagen RGB enriquecida que combina suavizado estructural, textura direccional y segmentación espacial. Esta representación multicanal resulta particularmente adecuada para ser utilizada como entrada en sistemas de detección de objetos, al ofrecer una visión más robusta, informativa y discriminativa de las escenas térmicas complejas propias del entorno amazónico.

#### IV-B. Impacto del Preprocesamiento en la Detección

El esquema de preprocesamiento implementado demostró ser crucial para mejorar significativamente el rendimiento de los modelos de detección. Las transformaciones específicas aplicadas a cada canal (filtrado Gaussiano, Mediano y Guiado en el canal rojo; segmentación Chan-Vese en el canal azul; y codificación textural mediante filtros de Gabor en el canal verde) actuaron como un mecanismo de *aumentación implícita* de los datos. Esta estrategia permitió enriquecer la representación de las imágenes sin necesidad de recurrir a técnicas tradicionales de *data augmentation*.

Los resultados comparativos entre los modelos evaluados sobre imágenes originales (sin preprocesamiento) y preprocesadas muestran mejoras sustanciales en todos los casos, tanto en métricas estándar como en la capacidad de generalización de los modelos.

**YOLO11s** (9.4 millones de parámetros):

- **mAP@50:** de 33.7 % a 61.8 % (**+28.1 p.p.**)
- **mAP@50-95:** de 12.6 % a 32.8 % (**+20.2 p.p.**)

**YOLO11l** (25.3 millones de parámetros):

- **mAP@50:** de 44.0 % a 65.2 % (**+21.2 p.p.**)
- **mAP@50-95:** de 20.7 % a 32.1 % (**+11.4 p.p.**)

**YOLO11l Fine-tuned** (ajustado con pesos específicos del dominio):

- **mAP@50:** de 48.4 % a 51.7 % (**+3.3 p.p.**)
- **mAP@50-95:** de 19.5 % a 24.9 % (**+5.4 p.p.**)

Estos resultados confirman que el preprocesamiento no solo mejoró la calidad visual de las imágenes térmicas, sino que también optimizó de manera efectiva el desempeño de los modelos de detección automática. La incorporación de información estructural, textural y segmentada en los canales RGB permitió una interpretación más robusta de las escenas, lo cual facilitó la detección precisa de elementos de interés como carreteras sin pavimentar y cuerpos de agua en entornos selváticos complejos.

#### IV-C. Evaluación Comparativa de Arquitecturas de Detección

Tras los resultados positivos obtenidos con el esquema de preprocesamiento, se llevó a cabo una evaluación comparativa entre cinco arquitecturas modernas de detección de objetos. Para esta etapa, se emplearon exclusivamente imágenes preprocesadas, asegurando así condiciones óptimas de entrada para todos los modelos. Además de las variantes de YOLO11, se incluyeron dos arquitecturas ampliamente utilizadas en la literatura: Faster R-CNN y RetinaNet.

La Tabla I resume los resultados obtenidos, ordenados en función del rendimiento alcanzado en la métrica mAP@50. Se reportan también los valores de mAP@50-95 y el número de parámetros aproximado de cada modelo, lo cual proporciona una perspectiva adicional sobre su complejidad y eficiencia.

Cuadro I  
RENDIMIENTO DE ARQUITECTURAS EVALUADAS

Modelo	Param. (M)	mAP@50	mAP@50-95
YOLO11l	25.3	65.2 %	32.1 %
YOLO11s	9.4	61.8 %	32.8 %
Faster R-CNN	–	56.6 %	32.8 %
YOLO11l FT	25.3	51.7 %	24.9 %
RetinaNet	32.2	26.9 %	12.2 %

Los resultados obtenidos muestran que las variantes del modelo YOLO11, tanto YOLO11s (versión *small*) como YOLO11l (versión *large*), se destacaron como las arquitecturas más eficaces en el contexto de imágenes térmicas procesadas.

El modelo YOLO111, con 25.3 millones de parámetros, alcanzó la mayor precisión en la métrica mAP@50, con un 65.2 %, lo que indica una alta tasa de detección correcta de objetos cuando se permite un margen de error de hasta un 50 % en la superposición entre la predicción y la anotación real (*IoU*). Esta métrica, más permisiva, es útil para evaluar la capacidad general del modelo para localizar objetos, especialmente en escenarios donde el ruido o la resolución afectan la precisión de las cajas predichas.

No obstante, YOLO11s, con solo 9.4 millones de parámetros, mostró una eficiencia sobresaliente: obtuvo un mAP@50–95 de 32.8 %, superando ligeramente a su contraparte más grande. Esta métrica promedia la precisión en múltiples umbrales de *IoU* (desde 0.5 hasta 0.95, en incrementos de 0.05), y es más rigurosa, pues evalúa no solo la capacidad de detectar, sino también de localizar con precisión los objetos. El mejor resultado de YOLO11s en esta métrica sugiere una generalización más robusta y una mayor precisión espacial, a pesar de su menor complejidad computacional.

El modelo Faster R-CNN también presentó un rendimiento competitivo, con un mAP@50–95 igual al de YOLO11s (32.8 %) y un mAP@50 de 56.6 %. Aunque su precisión global fue inferior a la de los modelos YOLO11, su arquitectura de dos etapas demostró ser capaz de aprovechar el preprocesamiento aplicado, especialmente en términos de precisión detallada. En cambio, RetinaNet, con 32.2 millones de parámetros, obtuvo los valores más bajos tanto en mAP@50 (26.9 %) como en mAP@50–95 (12.2 %). Esto puede deberse a su mayor sensibilidad a condiciones adversas como el ruido térmico y el bajo contraste característico de las imágenes FLIR.

#### *IV-D. Desafíos Inherentes a las Imágenes FLIR*

A pesar de las mejoras sustanciales obtenidas mediante el preprocesamiento, los resultados alcanzados reflejan las limitaciones inherentes al trabajo con imágenes térmicas FLIR capturadas en condiciones operacionales reales. Este tipo de datos presenta desafíos particulares que afectan directamente el rendimiento de los algoritmos de detección:

- **Variabilidad térmica:** Las diferencias de temperatura entre los objetos de interés y su entorno pueden ser mínimas, especialmente durante ciertos períodos del día, lo que dificulta la diferenciación clara de las clases.
- **Resolución limitada:** La distancia entre la aeronave y la superficie terrestre impone restricciones importantes sobre la resolución espacial, reduciendo la capacidad de distinguir detalles finos.
- **Condiciones atmosféricas adversas:** La humedad, nubosidad y partículas en suspensión características de la región amazónica degradan la señal térmica captada por los sensores, introduciendo ruido e inconsistencias.
- **Movimiento de plataforma:** Las vibraciones, oscilaciones y desplazamientos de la aeronave durante el vuelo generan distorsiones que afectan la estabilidad espacial de las imágenes, introduciendo artefactos que comprometen la detección.

Estos factores explican por qué, incluso tras aplicar un pipeline de preprocesamiento avanzado, los valores absolutos de mAP se mantienen por debajo de los estándares comúnmente reportados en tareas de detección sobre imágenes RGB convencionales. En consecuencia, se reafirma la necesidad de desarrollar enfoques personalizados y robustos, específicamente diseñados para operar eficazmente sobre imágenes térmicas aéreas en contextos naturales y de difícil acceso como la selva amazónica.

#### *IV-E. Cumplimiento de Objetivos de Detección*

Los resultados obtenidos permiten afirmar que se cumplió satisfactoriamente el objetivo general del proyecto: desarrollar un modelo de detección automatizada basado en aprendizaje profundo para identificar carreteras y cuerpos de agua en imágenes térmicas FLIR capturadas por la Fuerza Aérea Colombiana. En particular, la arquitectura YOLO111 demostró ser la más efectiva, superando ampliamente las limitaciones de los enfoques tradicionales de análisis visual manual.

El modelo no solo logró detectar con precisión objetos de interés bajo condiciones operacionales reales, sino que también evidenció una mejora sustancial en su rendimiento gracias al esquema de preprocesamiento propuesto. En promedio, se observó un aumento de **17.5 puntos porcentuales en mAP@50**, lo que representa un avance significativo en la capacidad de interpretar escenas térmicas complejas.

Estos resultados confirman la viabilidad técnica de integrar soluciones de inteligencia artificial en los procesos de vigilancia aérea, y establecen una base robusta para futuras aplicaciones orientadas a la identificación temprana de actividades asociadas a minería ilegal u otras amenazas ambientales en la región amazónica. Asimismo, sientan las bases para la incorporación de este tipo de herramientas en sistemas operacionales de la FAC, mejorando la eficiencia y precisión en la toma de decisiones estratégicas.

#### *IV-F. Despliegue*

El despliegue del sistema se realizó en dos partes: front-end y back-end. La interfaz front-end, orientada al usuario final, fue desarrollada utilizando el framework Svelte, el cual se basa en JavaScript. Esta parte gestiona toda la interacción directa con el usuario, ofreciendo una experiencia ligera y responsive. El código fuente del front-end está alojado en un repositorio público de GitHub y fue desplegado utilizando Vercel, una plataforma de hosting gratuita que se integra fácilmente con GitHub para facilitar el despliegue.

Por otro lado, el back-end se encarga de la lógica principal del sistema. Este servicio toma una imagen como entrada, determina el modelo de detección a aplicar y ejecuta el procesamiento correspondiente. Fue desarrollado en Python y hace uso de diversos frameworks y librerías para tareas de visión por computador, incluyendo PyTorch, Ultralytics (YOLO), entre otros.

El back-end no es de acceso público y se encuentra desplegado en un clúster de procesamiento de AWS (Amazon Web Services), utilizando una instancia tipo EC2 y recursos

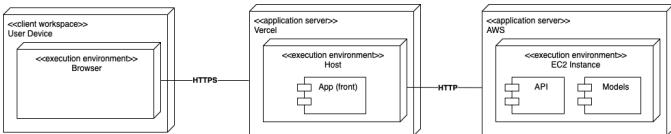


Figura 2. Diagrama de despliegue en UML

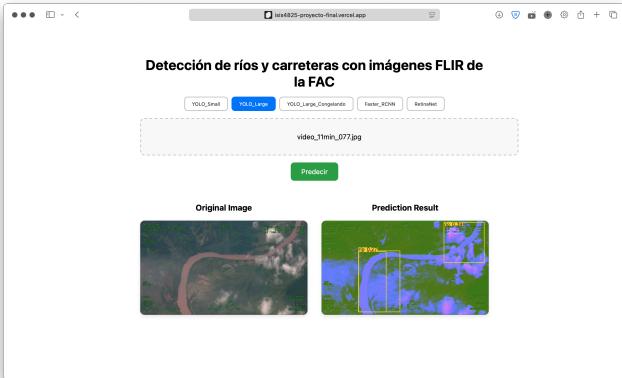


Figura 3. Pantallazo del despliegue en funcionamiento

computacionales suficientes para el análisis de imágenes. La comunicación entre el front-end y el back-end se realiza a través de peticiones HTTP a una IP específica. Este enfoque modular permite escalar cada componente de manera independiente y garantiza un entorno seguro y eficiente para el procesamiento y la entrega de resultados. El diagrama que muestra la arquitectura de despliegue se muestra en la figura 2.

Se puede apreciar un pantallazo del sistema en funcionamiento en la figura 3, el despliegue se puede consultar a continuación en este enlace: <https://isis4825-proyecto-final.vercel.app>

## V. DISCUSIÓN Y CONCLUSIÓN

El desarrollo de un sistema automatizado de detección de minería ilegal basado en imágenes térmicas representa un avance crucial en la transformación digital de las estrategias de vigilancia ambiental y defensa territorial. La combinación entre el preprocesamiento multicanal y el uso de modelos de aprendizaje profundo permitió superar limitaciones significativas en el análisis de imágenes FLIR, tales como el ruido térmico, la baja resolución espacial, y la interferencia atmosférica inherente a la región amazónica colombiana.

El esquema de preprocesamiento propuesto —basado en una representación RGB sintética donde cada canal fue optimizado para capturar distintas características visuales: suavizado estructural (R), texturas (G) y segmentación espacial (B)— se consolidó como un componente clave para enriquecer la representación de las imágenes y, por ende, potenciar el entrenamiento de los modelos. Esta estrategia fue coherente con enfoques previos en teledetección, como los descritos por Ferreira et al. (2019), quienes combinaron múltiples fuentes

de datos remotos para mejorar la identificación de cultivos ilícitos en Brasil, alcanzando una precisión del 92.16 %. Asimismo, Teixeira et al. (2024) lograron una segmentación precisa de embalses mediante redes neuronales profundas y preprocesamiento específico, alcanzando una precisión de 95 % en la métrica IoU .

Entre los modelos evaluados, YOLO111 demostró el mejor desempeño con un mAP@50 de 65.2 %, confirmando su capacidad para detectar con alta certeza elementos relevantes en imágenes complejas. Es destacable también el rendimiento del modelo YOLO11s, que con tan solo 9.4 millones de parámetros logró un mAP@50–95 de 32.8 %, superando incluso a su contraparte más robusta. Este hallazgo sugiere que modelos más livianos pueden ser más eficientes en contextos donde la localización precisa es más crítica que la clasificación general. Estas observaciones son coherentes con la literatura, en la cual modelos YOLO optimizados han demostrado ser eficientes y precisos en entornos con restricciones de hardware y condiciones de iluminación adversas (Sloan et al., 2024) .

En contraste, RetinaNet presentó un desempeño deficiente (mAP@50–95 de 12.2 %), posiblemente debido a su mayor sensibilidad al desbalance de clases y a las distorsiones térmicas propias de los sensores FLIR. Esta susceptibilidad también fue observada por Sloan et al. (2024), quienes destacaron la necesidad de adaptar arquitecturas específicas para condiciones atmosféricas y geomorfológicas extremas .

Un aspecto limitante fue la naturaleza indirecta de las clases detectadas. Aunque el modelo logró identificar carreteras no pavimentadas y cuerpos de agua —ambos elementos comúnmente asociados con la minería ilegal—, la ausencia de clases explícitas como dragas, campamentos o balsas restringe su aplicabilidad directa para pruebas legales o intervenciones operativas. Pinto Hidalgo et al. (2023) subrayan la importancia de combinar imágenes satelitales con datos geoespaciales para mejorar la precisión en la detección de infraestructuras ilegales de producción de coca, lo que sugiere una posible línea de mejora para este proyecto .

Desde una perspectiva operativa, los resultados alcanzados son altamente alentadores. La integración de estos modelos en sistemas de vigilancia aérea de la Fuerza Aeroespacial Colombiana (FAC) podría reducir significativamente los tiempos de análisis y aumentar la precisión en la identificación temprana de amenazas ambientales. Además, el sistema podría escalarse para apoyar tareas de monitoreo de deforestación, planificación territorial y gestión de recursos hídricos, en línea con aplicaciones discutidas por Teixeira et al. (2024) y Ferreira et al. (2019) .

En términos éticos y sociales, es imperativo asegurar que estas tecnologías se apliquen con protocolos de supervisión humana, salvaguardando los derechos de las comunidades locales. La inteligencia artificial no puede ni debe reemplazar el juicio humano, especialmente cuando se trata de intervenciones en territorios habitados o culturalmente sensibles.

Finalmente, este trabajo abre múltiples posibilidades de investigación. En particular, el uso de modelos multimodales que combinen imágenes térmicas con datos ópticos o espectra-

les, así como la implementación de arquitecturas basadas en transformers o aprendizaje auto-supervisado, podría mejorar la adaptabilidad del sistema a nuevos dominios con escasa disponibilidad de datos etiquetados.

VI. Conclusiones Este proyecto demuestra la viabilidad técnica y operativa de utilizar algoritmos de aprendizaje profundo, especialmente modelos YOLO y Faster R-CNN, para automatizar la detección de carreteras no pavimentadas y cuerpos de agua en imágenes térmicas FLIR capturadas por plataformas aéreas de vigilancia. La implementación de un esquema de preprocesamiento multicanal permitió mejorar notablemente la calidad de las imágenes y optimizar el desempeño de los modelos, alcanzando mejoras significativas en métricas como mAP@50 y mAP@50–95.

Los resultados obtenidos evidencian el potencial de estas herramientas para ser integradas en sistemas reales de monitoreo ambiental y vigilancia territorial, contribuyendo a la lucha contra la minería ilegal en la Amazonía colombiana. A su vez, el enfoque metodológico propuesto puede ser transferido a otros escenarios de análisis remoto, como la detección de deforestación, el monitoreo de cuerpos hídricos y la planificación de intervenciones sobre el territorio.

Se recomienda continuar con la expansión del conjunto de datos, incluyendo clases adicionales representativas de actividades mineras ilegales, así como explorar arquitecturas avanzadas que permitan una mayor precisión en entornos hostiles. La colaboración interdisciplinaria entre ingenieros, biólogos, militares y comunidades locales será clave para escalar estos desarrollos hacia aplicaciones de mayor impacto social, ambiental y geopolítico.

#### REFERENCIAS

- [1] G. R. Pardini, P. M. Tasinazzo, E. H. Shigemori, T. N. Kuck, M. R. Maximo, and W. R. Gyotoku, “Improved algorithm to detect clandestine airstrips in amazon rainforest,” *Algorithms*, vol. 18, no. 2, p. 102, 2025.
- [2] A. M. de Albuquerque Teixeira, L. V. Batista, R. M. da Silva, L. M. T. Freitas, and C. A. G. Santos, “Dynamic monitoring of surface area and water volume of reservoirs using satellite imagery, computer vision and deep learning,” *Remote Sens. Appl.: Soc. Environ.*, vol. 35, p. 101205, 2024.
- [3] A. Ferreira, S. C. Felipussi, R. Pires, S. Avila, G. Santos, J. Lambert, and A. Rocha, “Eyes in the skies: A data-driven fusion approach to identifying drug crops from remote sensing images,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 12, no. 12, pp. 4773–4786, 2019.
- [4] J. J. P. Hidalgo and J. A. S. Centeno, “Geospatial intelligence and artificial intelligence for detecting potential coca paste production infrastructure in the border region of venezuela and colombia,” *J. Appl. Secur. Res.*, vol. 18, no. 4, pp. 1000–1050, 2023.
- [5] S. Sloan, R. R. Talkhani, T. Huang, J. Engert, and W. F. Laurance, “Mapping remote roads using artificial intelligence and satellite imagery,” *Remote Sens.*, vol. 16, no. 5, p. 839, 2024.