Monitoreo de vida salvaje mediante detección de objetos en diferentes tipos de imágenes

Antezana, Matias | mantezana@udesa.edu.ar | Universidad de San Andrés
Giacometti, Mateo | mgiacometti@udesa.edu.ar | Universidad de San Andrés

Introducción

El siguiente trabajo a sido realizado con la finalidad de utilizar la conocida arquitectura de detección de objetos **YOLO** (en particular su versión más reciente, *YOLOv11*) para **detectar y clasificar** cada una de las **tres especies animales** (*vacas*, *ciervos* y *caballos*) presentes en el set de datos ***Aerial Wildlife Image Repository*** de la universidad de ***Mississippi***. Este mismo cuenta con una colección de *imágenes RGB*, *imágenes termales* y *coordenadas* de los animales en las imaǵenes etiquetadas por biólogos expertos.
Para alcanzar este propósito, se entrenaron **múltiples modelos basados en *YOLO***, utilizando la información proporcionada por el dataset. Se exploraron diversas combinaciones de datos, generando variaciones en los conjuntos de entrenamiento y evaluando estrategias de fusión de resultados de distintos modelos para incrementar la robustez y precisión del sistema de detección.

Metodología y Arquitecturas

Imágenes usadas para *train*, *validation* y *test* de los modelos

A continuación se presentan los diferentes *tipos de imágenes* empleadas para la construcción de los modelos probados en este trabajo:

- **Imágenes RGB:** Representan la información visual en los *tres canales* de *color primarios* (rojo, verde y azul).
- **Imágenes Termales:** Capturan la *radiación infrarroja* emitida por los objetos, proporcionando información de temperatura (tambien en tres canales).
- **Imágenes HST:** El espacio de color *RGB* se convierte al espacio *HSV* (Tono, Saturación, Valor), que se alinea mejor con la percepción humana. Luego, se genera el espacio *HST*, que combina los componentes de tono (*H*) y saturación (*S*) del espacio *HSV* con la información térmica (*T*) de las *imágenes termales* en *escala de grises*.
- **Imágenes GST:** La representación *GST* consta del canal *G*, que es la imagen *RGB* transformada a *escala de grises*; el canal S, que es la saturación de la representación *HSV* de la imagen *RGB*; y el canal *T*, que es la *imagen térmica* en *escala de grises*.

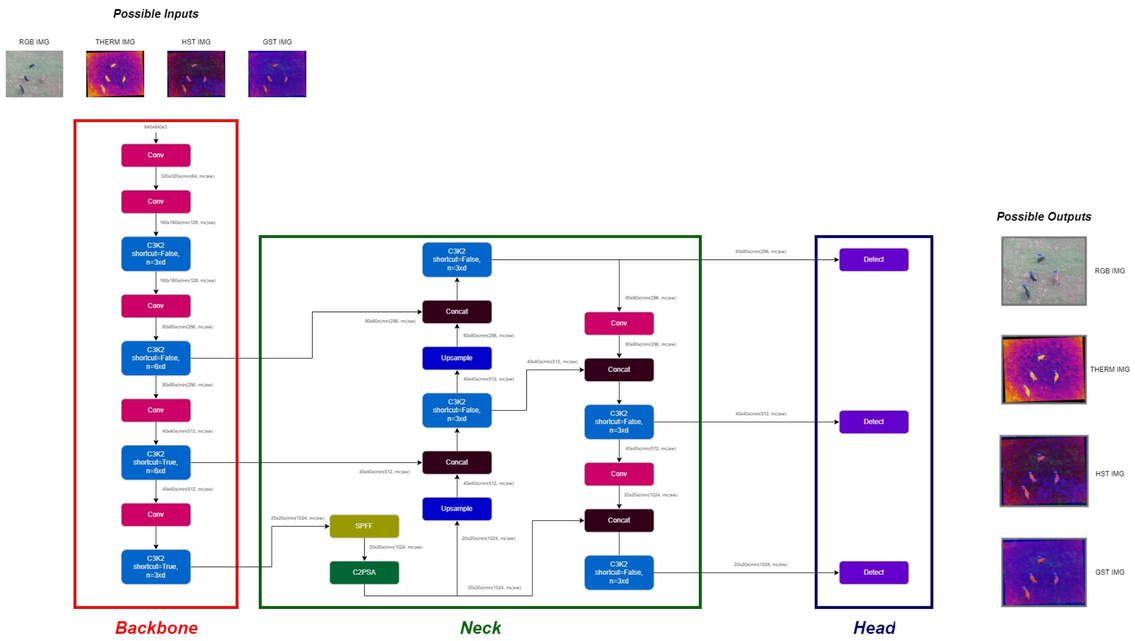

Figura 1: Arquitectura de *YOLOv11*, los diferentes tipos de imágenes empleadas en el trabajo y ejemplos de cómo podrían presentarse las detecciones en estas.

Búsqueda de hiperparámetros

Antes de realizar el entrenamiento de los modelos se llevó a cabo una **búsqueda de hiperparámetros** con el propósito de encontrar los que mejor maximizan el desempeño de cada modelo..

Combinación de resultados de modelos - *Late Fusion Approach*

La **Late Fusión** es un enfoque de *fusión de datos* en el que se procesan de manera *independiente* los datos de distintas fuentes, y la fusión ocurre en una etapa posterior, generalmente en el nivel de las salidas de los modelos. En este caso, se propone la combinación de los datos de dos modelos entrenados con diferentes tipos de imágenes: RGB y Termales.

Los resultados de detección obtenidos de ambas redes son procesados por un componente propuesto, no entrenable, que denominamos **sistema de emparejamiento de predicciones (PMS)**. La responsabilidad del PMS es proporcionar una única salida de detección basada en los resultados obtenidos de ambas ramas de procesamiento, fusionando así los resultados de ambas redes. Para esta implementación, el PMS toma como entradas el valor del parámetro de **intersección sobre unión (IoU)** para las bounding boxes de las salidas de ambas redes neuronales (debe ser mayor a 0.45 para ser válida) y la **probabilidad de que un objeto detectado pertenezca a una clase determinada**, calculada por cada red (debe ser mayor a 0.6 en ambas para ser válida).

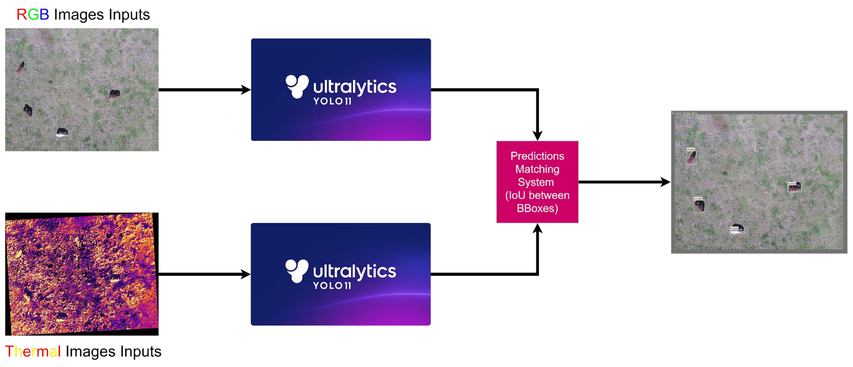

Figura 2: *Late Fusion* aplicado a *YOLOv11*. Se combinan las predicciones de un modelos entrenado con *imágenes RGB* y otro entrenado con *imágenes termales*.

Infraestructura utilizada para el entrenamiento

Para el entrenamiento de los módelo se utilizó la plataforma de **Google Cloud Platform** empleando **GPUs T4** lo cual permitió acelerar el proceso de aprendizaje y la búsqueda de hiperparámetros.

Resultados

Para este trabajo, se entrenaron *modelos* utilizando *cada tipo de imagen de manera individual* (realizando previamente la búsqueda de *hiperparámetros* correspondiente para cada caso), así como también un modelo que combina tanto i*mágenes RGB* como *imágenes termales*.

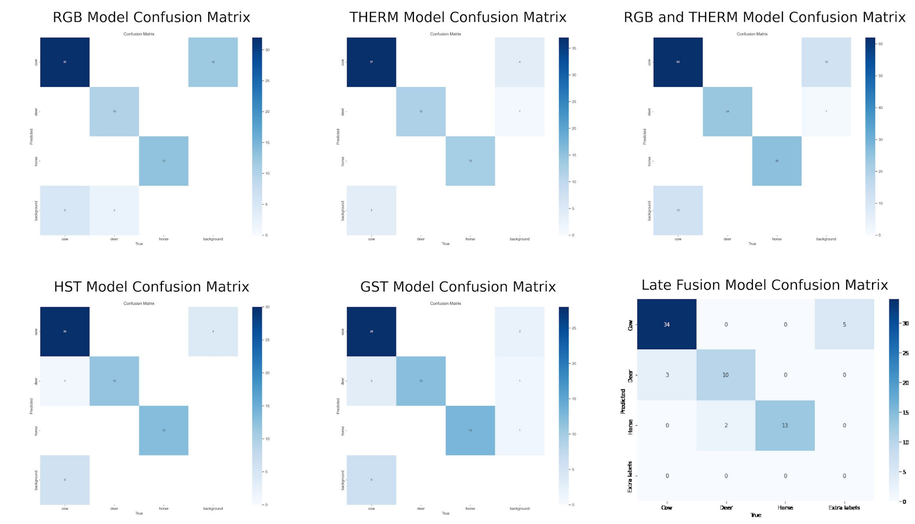

Figura 3: *Matrices de Confusión* de cada uno de los modelos realizados.

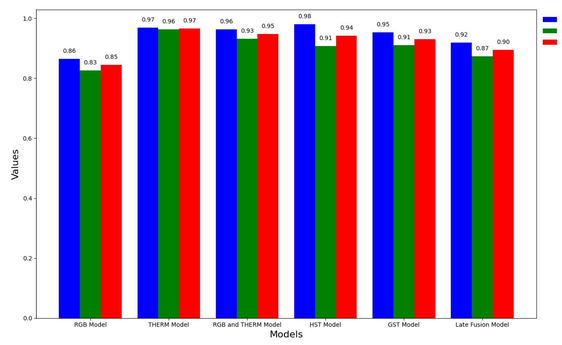

Figura 4: Métricas de *Precision*, *Recall* y *F1-Score* de cada modelo realizado.

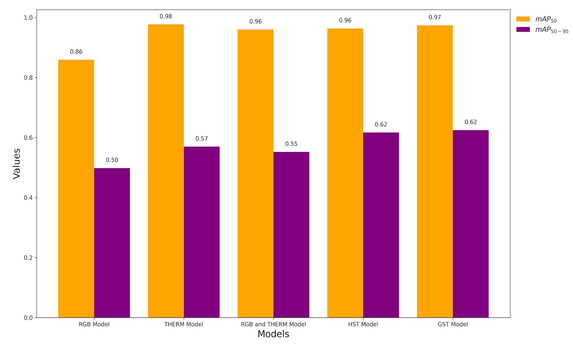

Figura 5: *Mean Average Precision* al 50% y desde 50% a 95% de cada modelo realizado.

Conclusiones

- El modelo basado exclusivamente en **imágenes termales** se destaca como el **más eficaz**, logrando la mayor *precisión* y *recall*. Esto sugiere que las diferencias de temperatura entre especies son características clave para la clasificación en este dataset.
- Las imágenes **RGB** parecen ofrecer información complementaria, pero no esencial, dado que su **combinación con las imágenes termales no mejora significativamente** los resultados. De hecho, en algunos casos, incluso podría introducir **ruido adicional.**
- Los modelos basados en transformaciones (***HST*** y ***GST***) también logran **buenos resultados**, pero no superan el rendimiento del modelo termal puro. Esto indica que, aunque útiles, estas transformaciones no capturan información crucial para este problema en comparación con la modalidad termal.
- La técnica de **Late Fusion** no ofrece beneficios significativos, lo cual podría estar relacionado con una subóptima integración de los modelos o con redundancia en la información.

Trabajo Futuro

- Implementación de la arquitectura ***Early Fusion*** (utilizando una versión más antigua de YOLO como la v5).
- Implementación de la arquitectura ***Middle Fusion*** (utilizando una versión más antigua de YOLO como la v5).
- Probar como *input* ***combinaciones de canales de los dos tipos (RGB y Termales) de imágenes diferentes*** a las presentadas.
- Utilizando la arquitectura Early Fusion, utilizar como input ***imagenes con mas de 4 canales*** (por ejemplo, ingresar en conjunto los 3 canales de las imágenes *RGB* y los 3 canales de las imágenes *termales*).
- Ampliar la **búsqueda de hiperparámetros** con muchas más iteraciones para cada modelo.

Referencias

[1] K. Roszyk, M. R. Nowicki, and P. Skrzypczyński. *Adopting the YOLOv4 Architecture for Low-Latency Multispectral Pedestrian Detection in Autonomous Driving*. Sensors, vol. 22, no. 3, pp. 1–19, 2022.
[2] S. Liang, H. Wu, L. Zhen, Q. Hua, S. Garg, G. Kaddoum, M. M. Hassan and K. Yu. *YOLO: Real-Time Intelligent Object Detection System Based on Edge-Cloud Cooperation in Autonomous Vehicles*. Journal of Advanced Transportation, vol. 2022, pp. 1–15, 2022.
[3] Roboflow Blog. *YOLOv11: How to Train Custom Data*.
[4] Ultralytics. *Ultralytics YOLO: Real-Time Object Detection and AI Models*.
[5] Ultralytics. *Ultralytics YOLO: Hyperparameter Tuning*.
[6] N. Rao. *YOLOv11 Explained: Next-Level Object Detection with Enhanced Speed and Accuracy*. Medium, 2024.