

FINE-TUNING DE RETINANET Y ANÁLISIS DEL DATASET AFRICAN WILDLIFE

Edison Curi Aviles

Visión por Computadora aplicado a la biodiversidad

Abstract

La detección automática de fauna mediante visión computacional facilita el monitoreo de la biodiversidad. Este trabajo replica el fine-tuning del modelo **RetinaNet** usando **PyTorch** sobre el **African Wildlife Dataset** (~1279 imágenes). Se realizó un análisis exploratorio de datos para caracterizar tamaños y proporciones de los conjuntos de entrenamiento, validación y prueba. El modelo entrenado demostró capacidad para detectar varias especies con cajas delimitadoras precisas, evidenciando la importancia del preprocesamiento y el EDA. Los resultados muestran la aplicabilidad de técnicas de deep learning en entornos reales de conservación y proporcionan experiencia práctica en entrenamiento, evaluación y despliegue de modelos de visión por computadora.

Palabras clave: RetinaNet, African Wildlife Dataset, visión por computadora, detección de fauna, deep learning.

Introduction

La detección automática de fauna mediante visión computacional permite **optimizar el monitoreo de la biodiversidad** y apoyar en proyectos de conservación. Sin embargo, los datasets reales presentan desafíos como variación en dimensiones, iluminación y balance de clases. Para abordar estos retos, se emplean modelos de **detección de objetos** basados en *deep learning* como RetinaNet y YOLO, que han demostrado eficacia en entornos complejos. Este trabajo replica y adapta un tutorial de fine-tuning sobre un dataset de fauna africana, evaluando la aplicabilidad del modelo en un escenario real.

Objetivos:

- Reproducir el código propuesto por (Gosh, 2025) para la detección de vida silvestre
- utilizando el modelo RetinaNet y las bibliotecas Pytorch.
- Aplicar técnicas de Análisis Exploratorio de Datos a la colección de datos “African Wildlife dataset” publicado por Ultralytics.
- Sintetizar la información y los resultados obtenidos en un documento tipo poster de congreso.

Methodology

•Dataset: African Wildlife Dataset:

~1279 imágenes totales,

Tres sets: train, val, test.

Animales: Leones, elefantes, jirafas, cebras, etc.

Modelo: YOLOv8 (detector de objetos)

Epochs: 30, tamaño de imagen: 640x640, batch size:8

Preprocesamiento: Normalización de imágenes, división en train/val/test.

•**EDA:** Se analizó la distribución de tamaños de imágenes y la proporción de conjuntos, empleando histogramas y gráficos circulares.

•**Modelo:** Se aplicó fine-tuning de **RetinaNet**/YOLO con pesos preentrenados.

•Pre-Entrenamiento y Fine-Tuning:

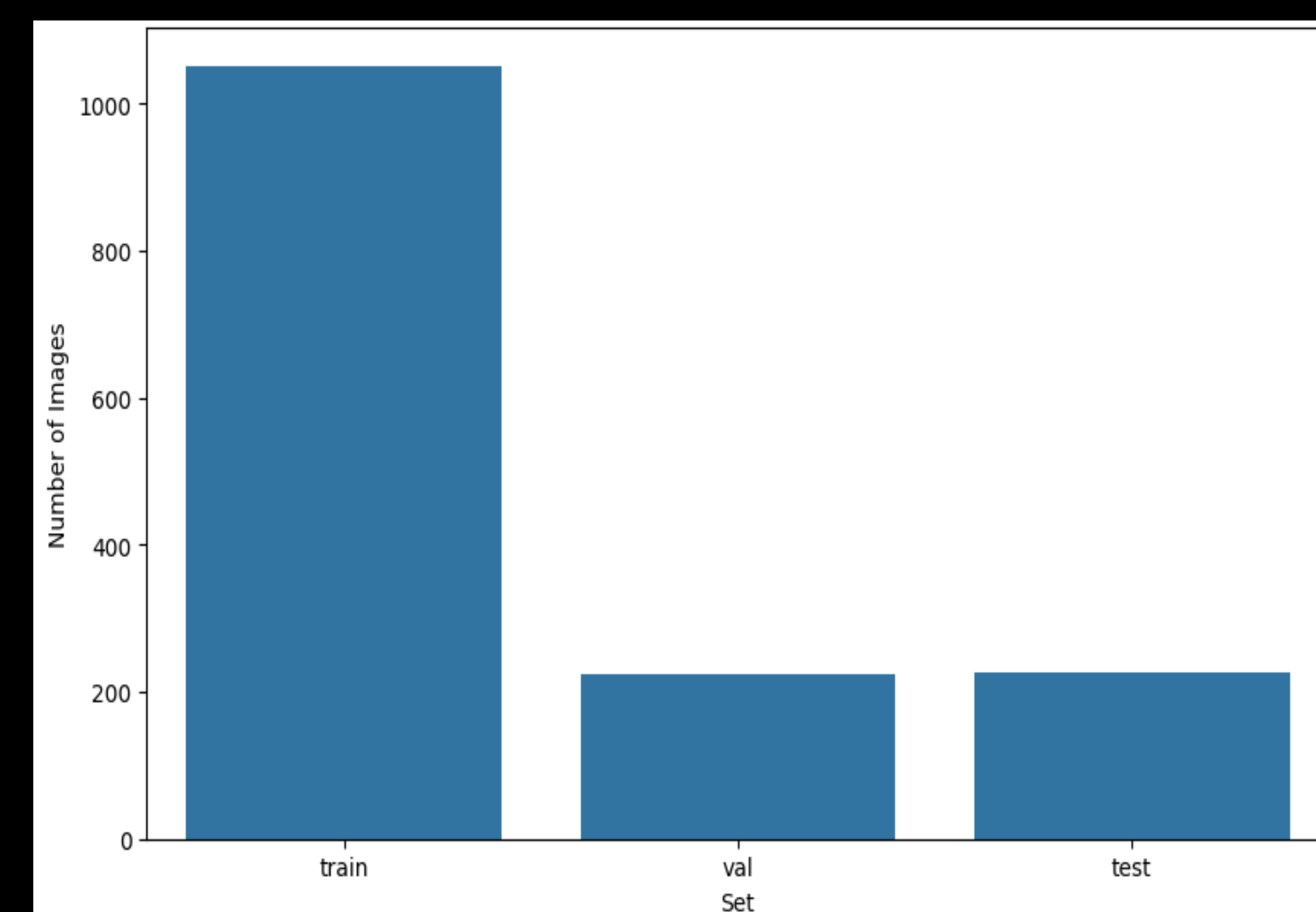
Usamos un modelo RetinaNet preentrenado en Pytorch.

Ajustamos el modelo con el **African Wildlife Dataset** (Replicando Gosh, 2025).

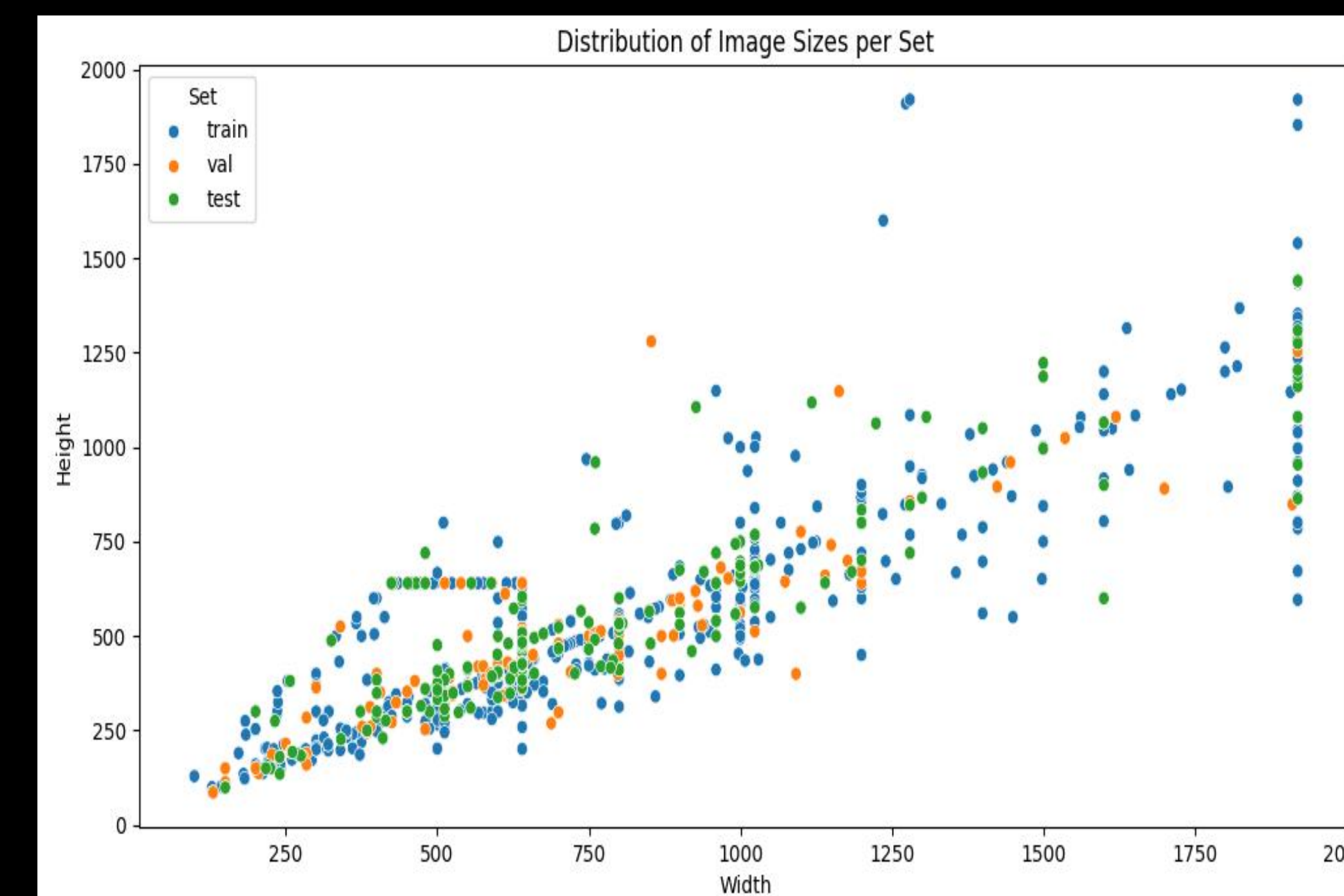
•**Evaluación:** Predicciones en imágenes de prueba y análisis de métricas (precisión, recall, mAP).

Results

Gráfica 01. cantidad de imágenes de conjunto **African Wildlife Dataset**.



Gráfica 02. Distribución de dimensiones de imágenes por set.



Gráfica 03. Proporción de imágenes por set.

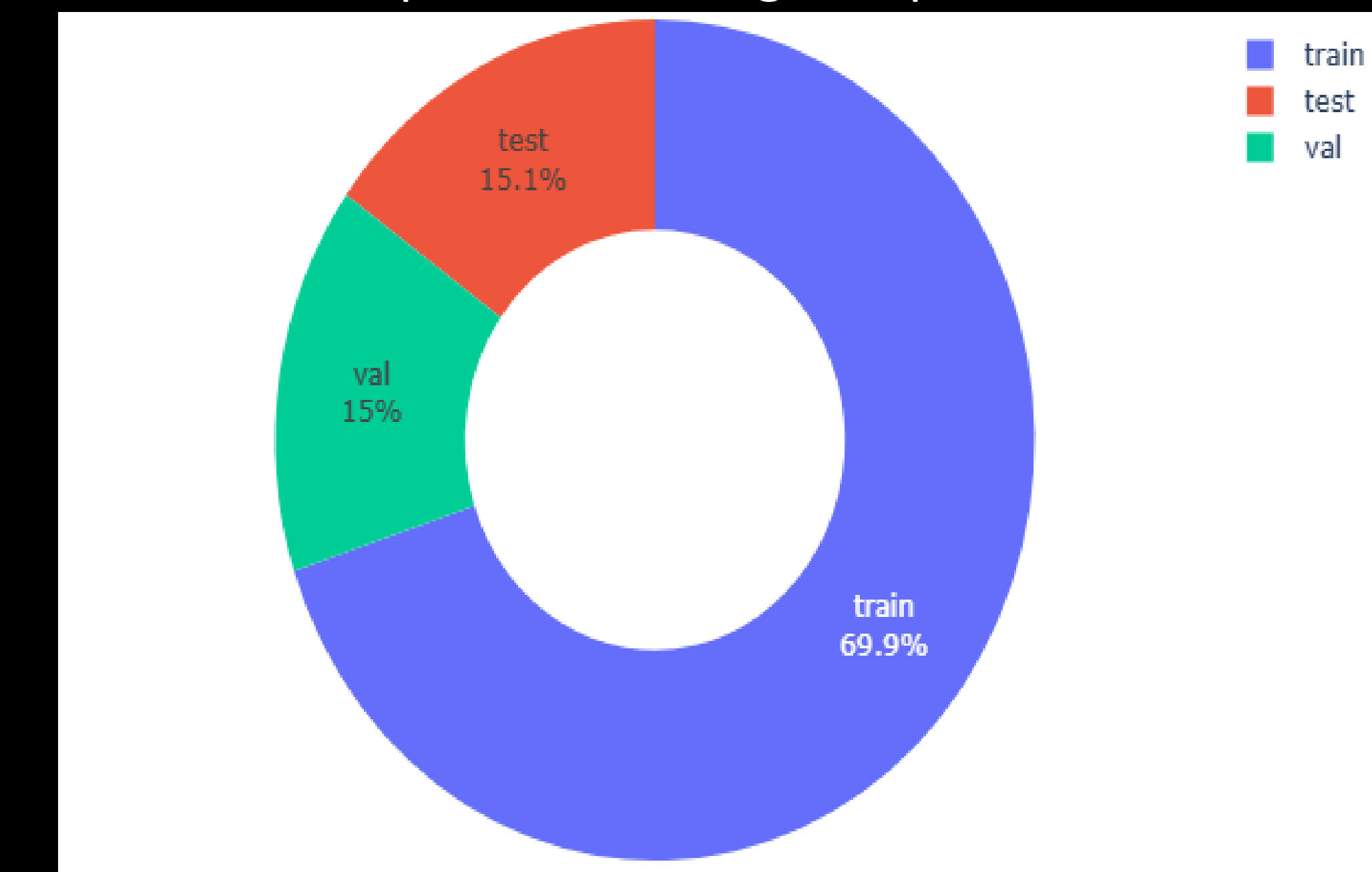
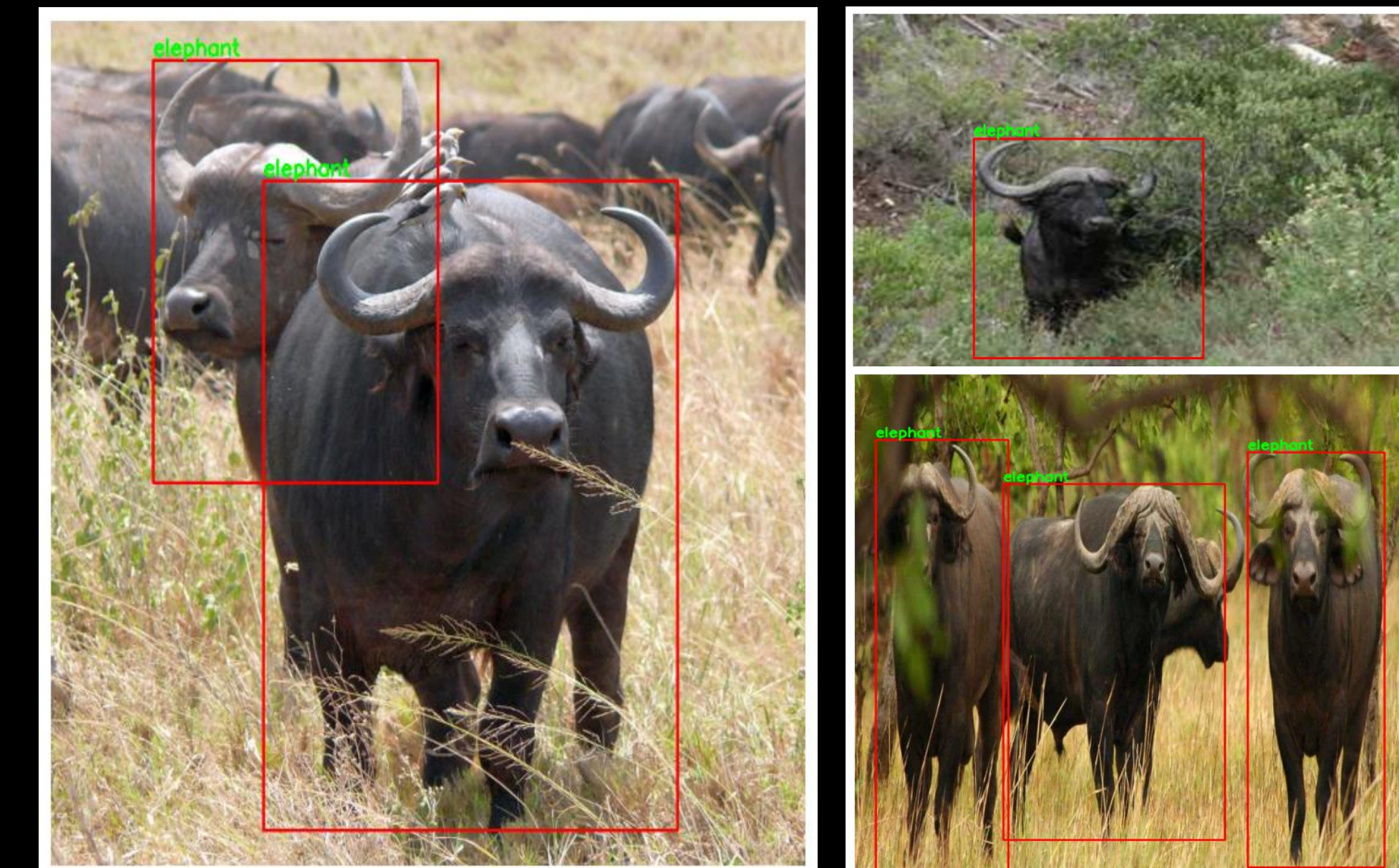
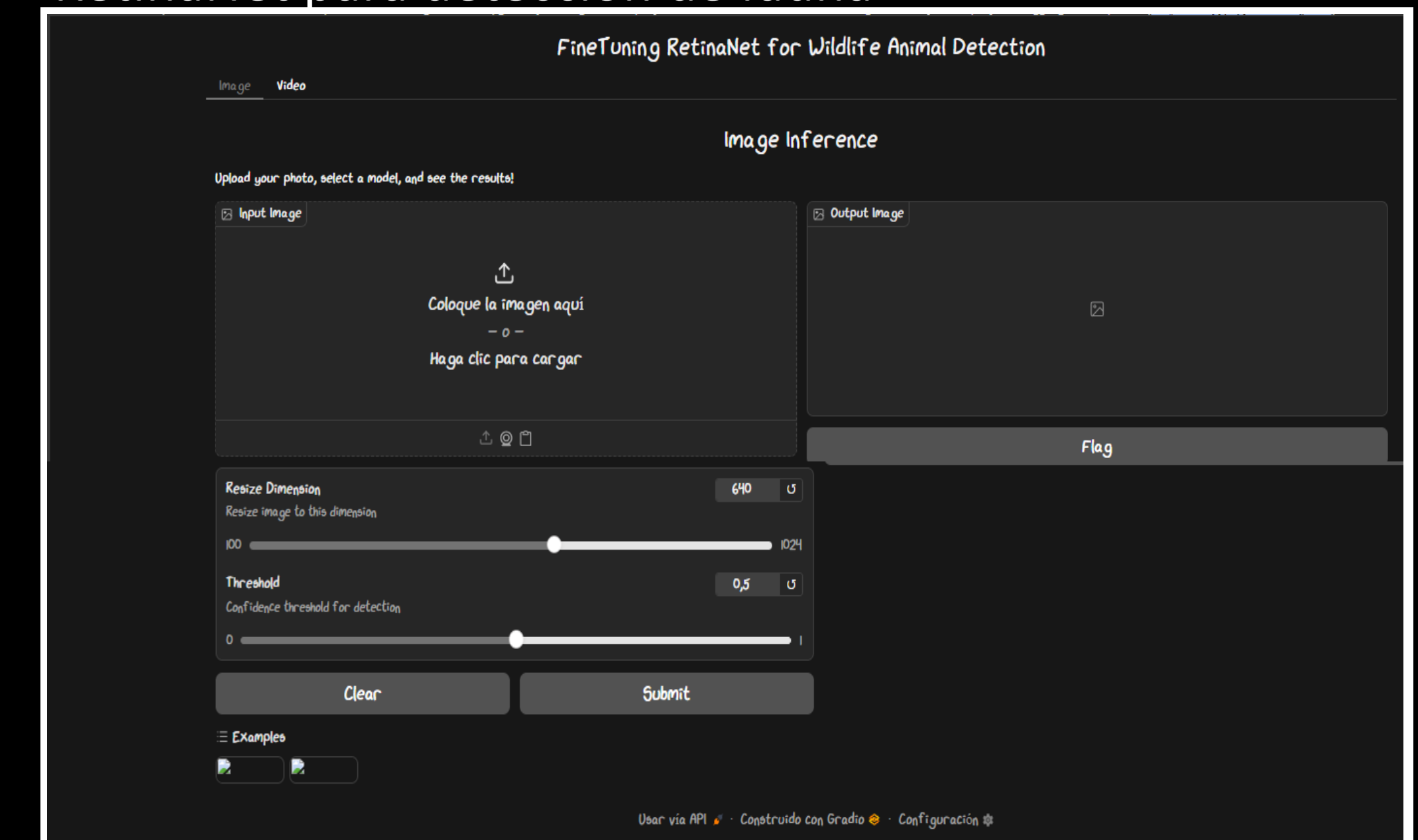


Imagen 01. Predicción de objeto en imágenes de vida Silvestre.



Se muestra las cajas delimitadoras generadas por el modelo RetinaNet preentrenado sobre African Wildlife Dataset.

Imagen 02. Interfaz (de Gradio) de inferencia de RetinaNet para detección de fauna



“Carga tu imagen, selecciona el modelo y observa los resultados de detección de animales”

Conclusion

- ❖ Se logró replicar el código de Gosh (2025), entrenando RetinaNet sobre el *African Wildlife Dataset* con PyTorch.
- ❖ El EDA permitió identificar la distribución de imágenes por conjunto y tamaño, evidenciando variaciones que afectan el entrenamiento del modelo.
- ❖ El modelo entrenado **detecta correctamente varias especies**, mostrando bounding boxes precisas en la mayoría de casos.

•Aprendizaje de la experiencia:

1. Replicar un código de un tutorial real ayuda a entender la integración de datasets, modelos y visualización de resultados.
2. Trabajar con datos reales evidencia la importancia del preprocesamiento, tamaño de batch, y elección de hiperparámetros.
3. Se obtiene una experiencia práctica de cómo entrenar, evaluar y desplegar modelos de visión computacional para detección de objetos.
4. **Próximos pasos / mejoras:**
5. Aumentar el dataset o aplicar data augmentation para mejorar detección de clases menos representadas.
6. Ajustar hiperparámetros y técnicas de post-procesamiento (como NMS) para mayor precisión.

Acknowledgements

Agradecimientos a instructora del curso Emilia Zeledón
A igual al Equipo del curso: Visión por computadora aplicado a la Biodiversidad-2025.