

# Detección de Vida Silvestre en África usando RetinaNet y PyTorch

## Visión por computadora aplicada a la biodiversidad

Alexander Barrantes Herrera  
Redbioma  
2025

### A. Resumen Descripción del Dataset y de Análisis Exploratorio

**Dataset:** African Wildlife Dataset (Ultralytics, 2025).

**Clases:** buffalo, elephant, rhino, zebra

- Conteo de imágenes en train/val/test
- Número y distribución de bounding boxes
- Distribución de tamaños de imagen
- Visualización de ejemplos con GT

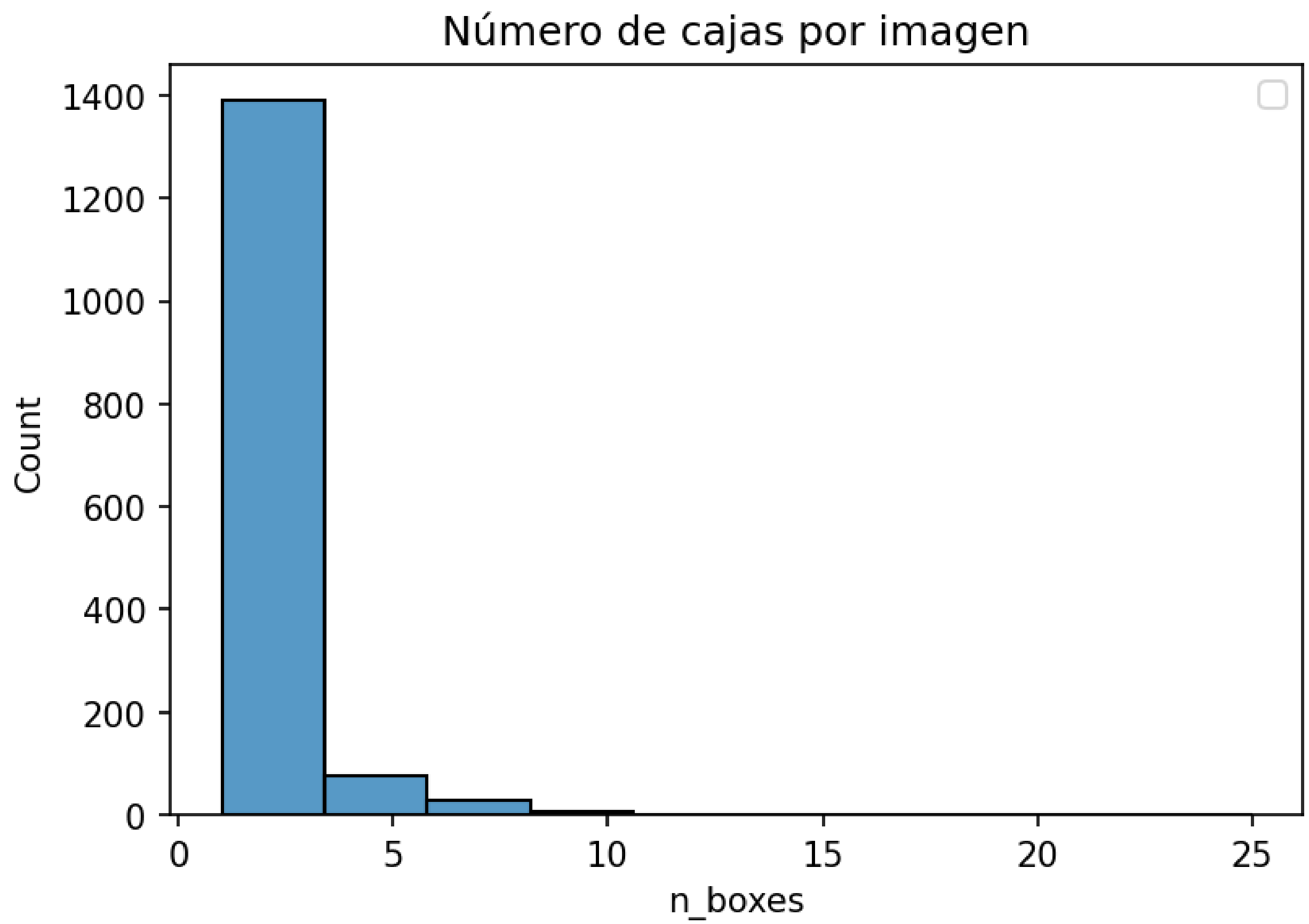


Figure: Número de Cajas por Imagen

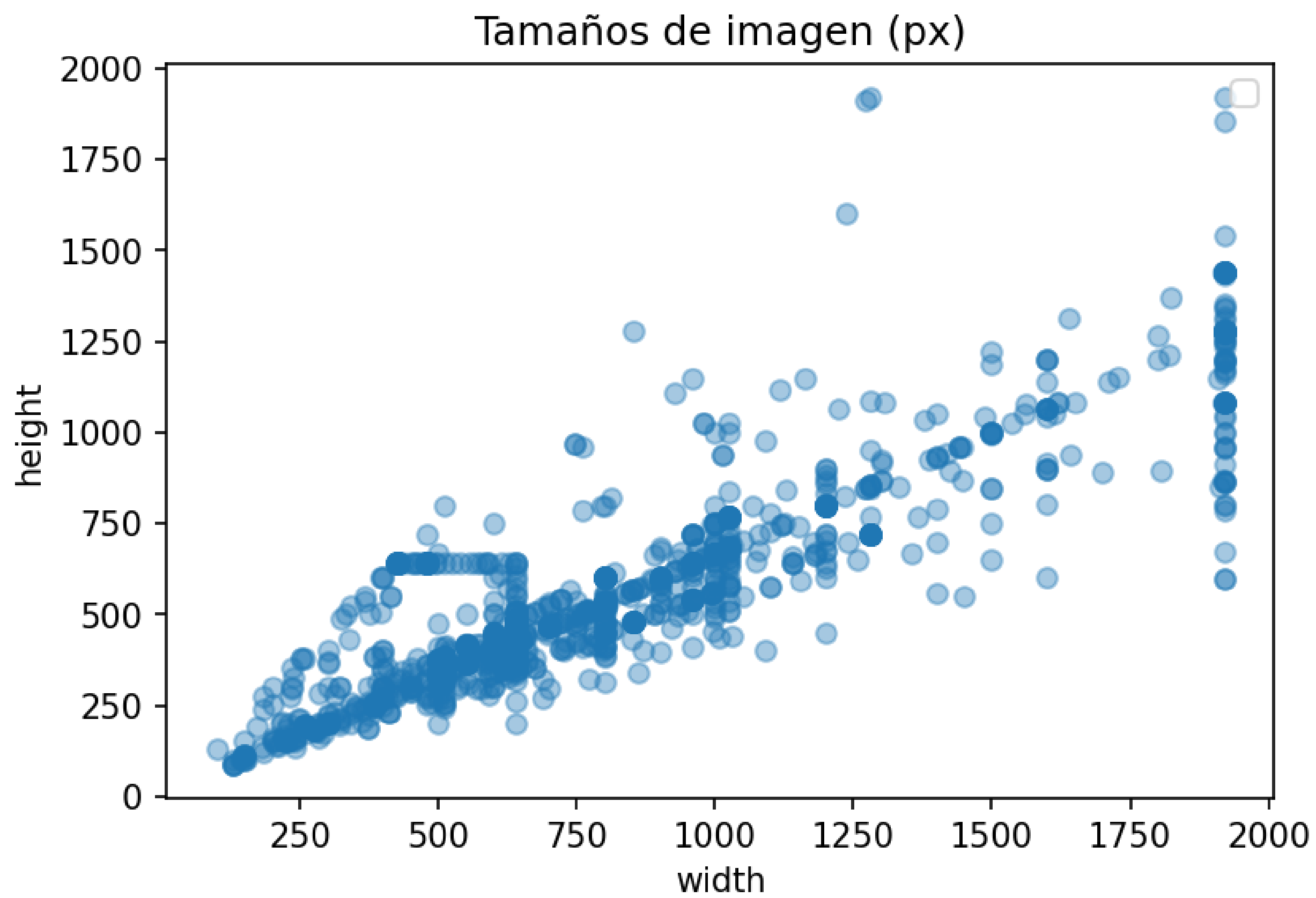


Figure: Tamaño de Imagen según PX

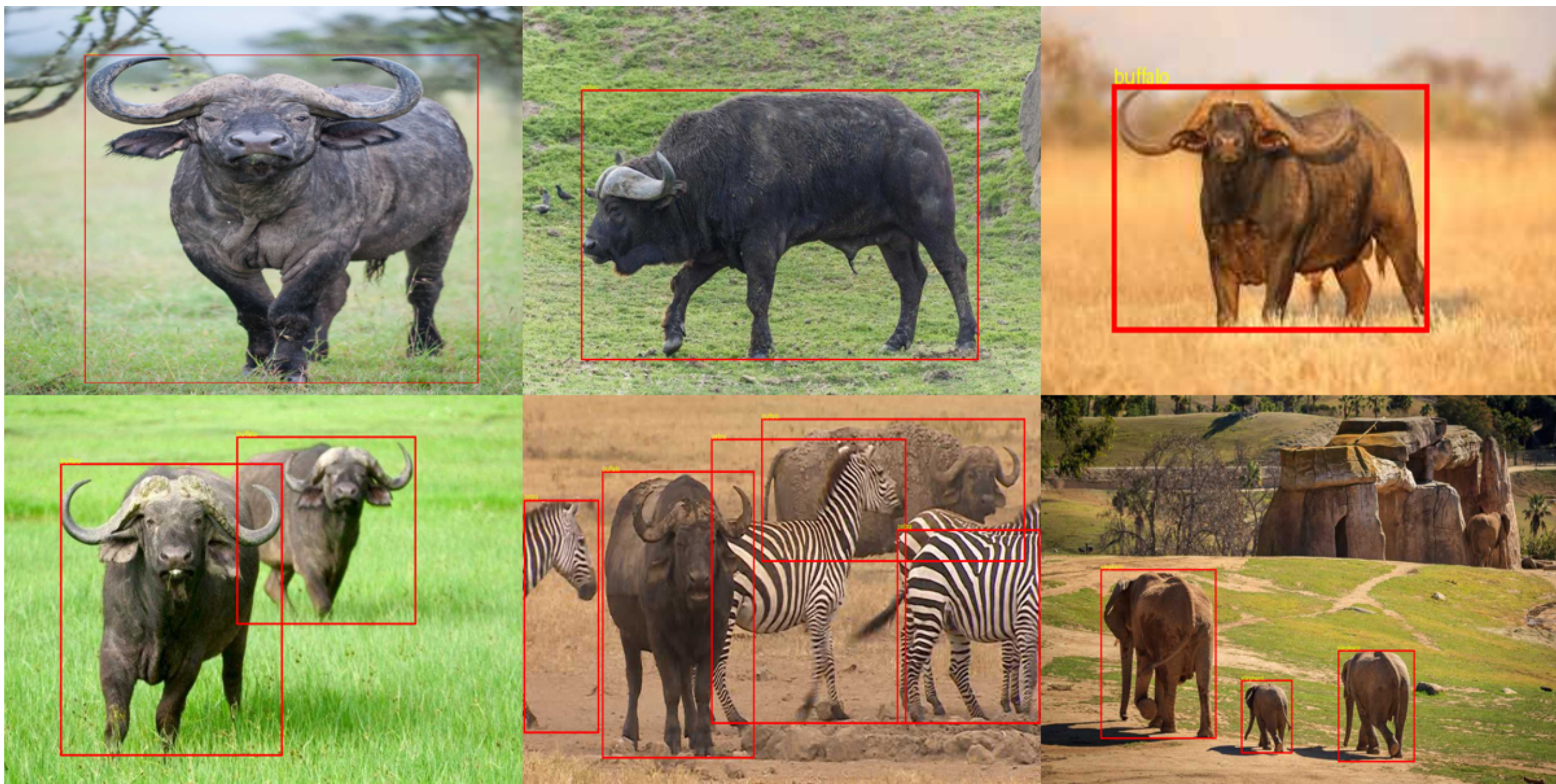


Figure: Visualización de Imágenes - Inicial

### B. Resumen Descriptivo del Modelo RetinaNet

**Arquitectura:** RetinaNet con ResNet50 preentrenado

**Configuración:**

- Clases: 4 + fondo
- Optimizador: AdamW, lr=1e-4
- Batch: 2, Epochs: 5
- Augmentations: Resize, Flip, ToTensor

### C. Resumen de Resultados del Modelo

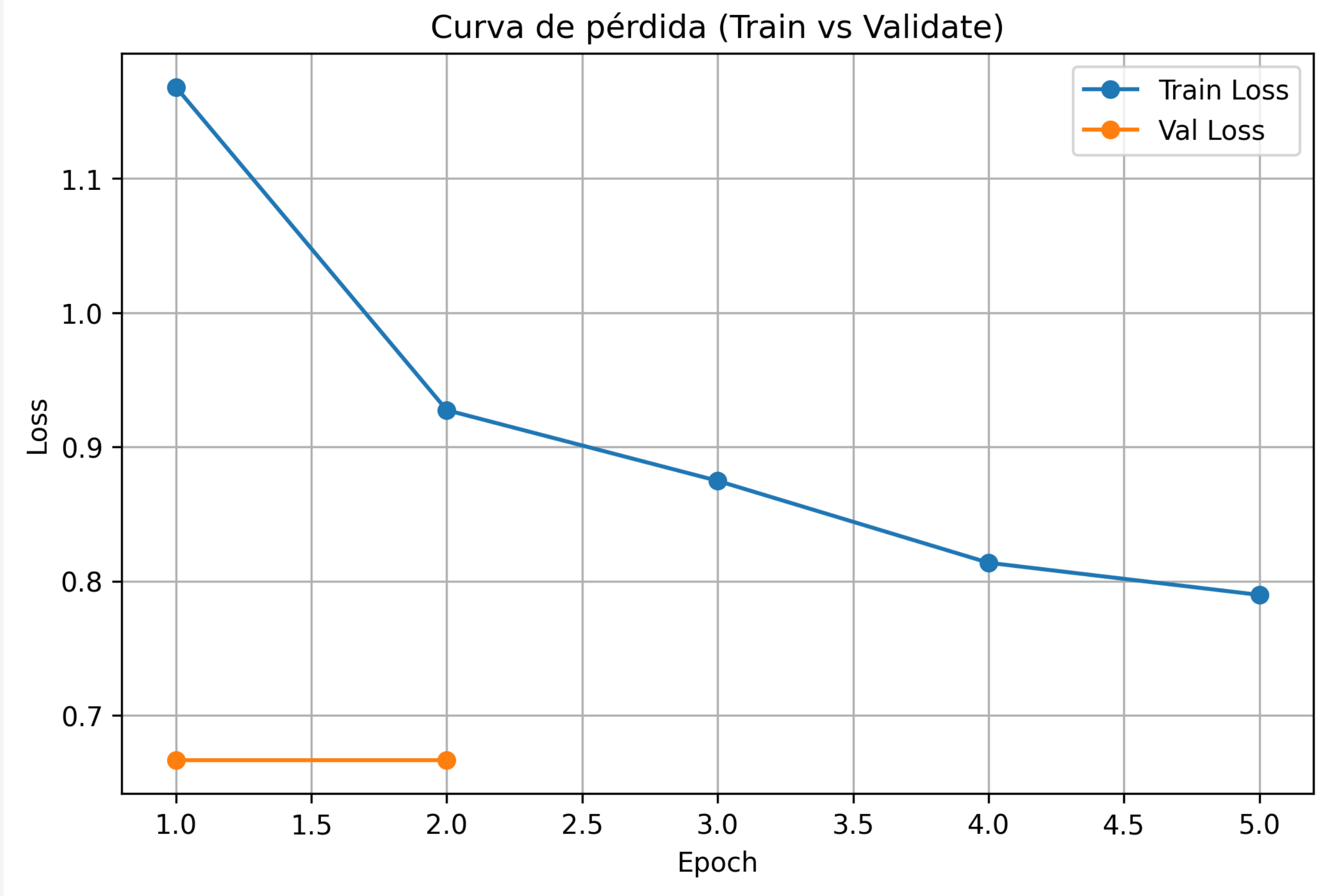


Figure: Curva de pérdida Train vs Val

**Resumen de métricas (época final):**

Métrica	Valor
mAP@0.5	0.351
mAP@0.5:0.95	0.176
Precision	0.412
Recall	0.536
F1 Score	0.463
Accuracy	0.602

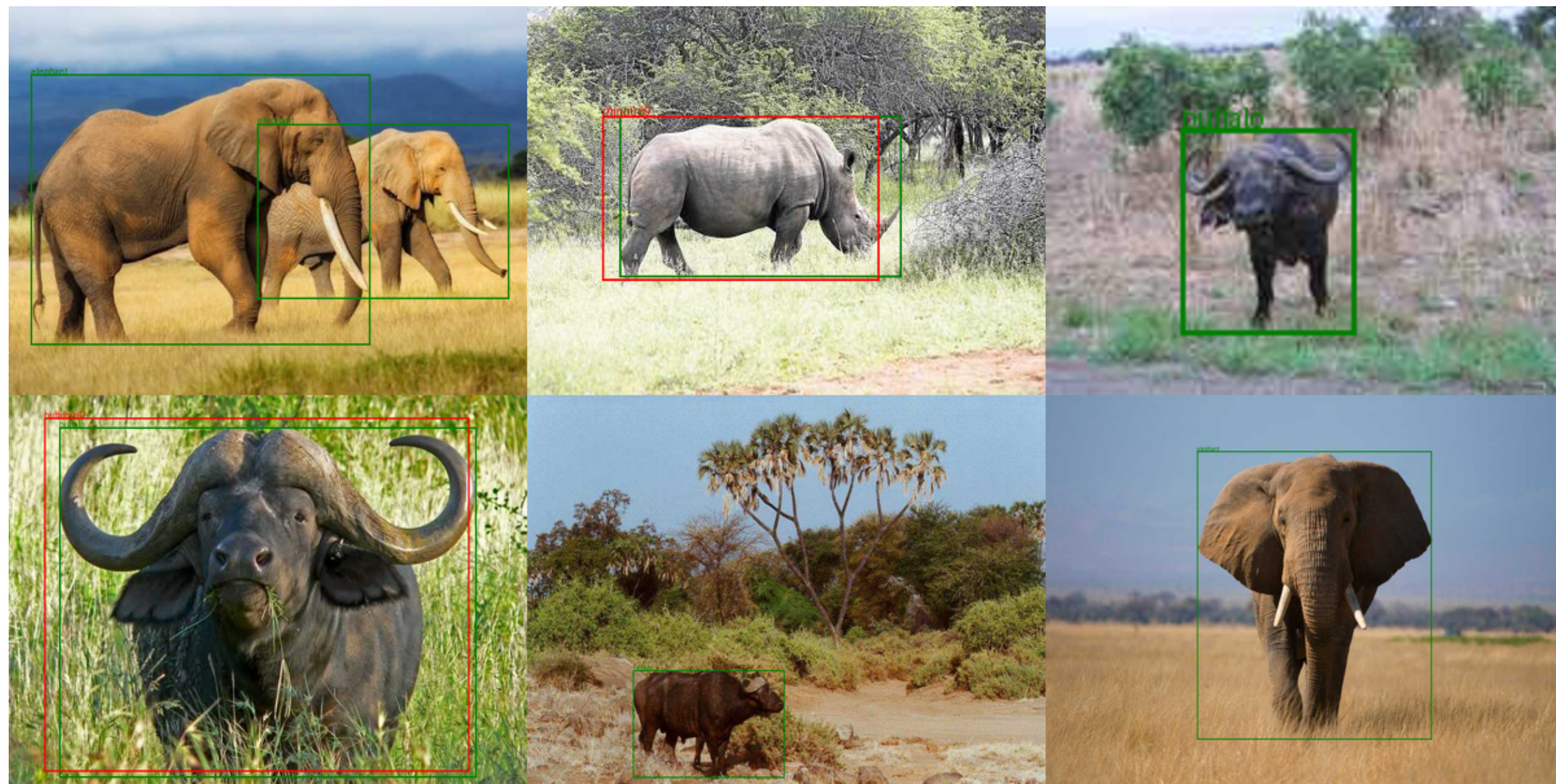


Figure: Ejemplos de validación: Green=GT, Red=Predicción

### D. Reflexión sobre la experiencia obtenida

- Replicar el tutorial (Gosh, 2025) permitió entender datasets en PyTorch.
- Dataset real: balance de clases y variabilidad de cajas.
- Visualización de predicciones vs GT fue crucial antes de cuantificar métricas.
- El pipeline entrenar-validar-visualizar es esencial en detección de objetos.