

Informe de Detección de Fauna — African Wildlife (YOLO11n)

Experimento en Google Colab

Objetivos:

- Entrenar YOLO11n sobre el dataset African Wildlife (Ultralytics).
- Evaluar métricas (P, R, mAP@0.5, mAP@0.5:0.95).
- Generar visualizaciones (curvas, ejemplos de inferencia) y consolidar un informe reproducible.

Configuración de entrenamiento

Parámetro	Valor
epochs	10
imgsz	640
batch	16
optimizer	auto
lr0	0.01
weight_decay	0.0005
device	None
data	african-wildlife.yaml
model	yolo11n.pt
name	train7

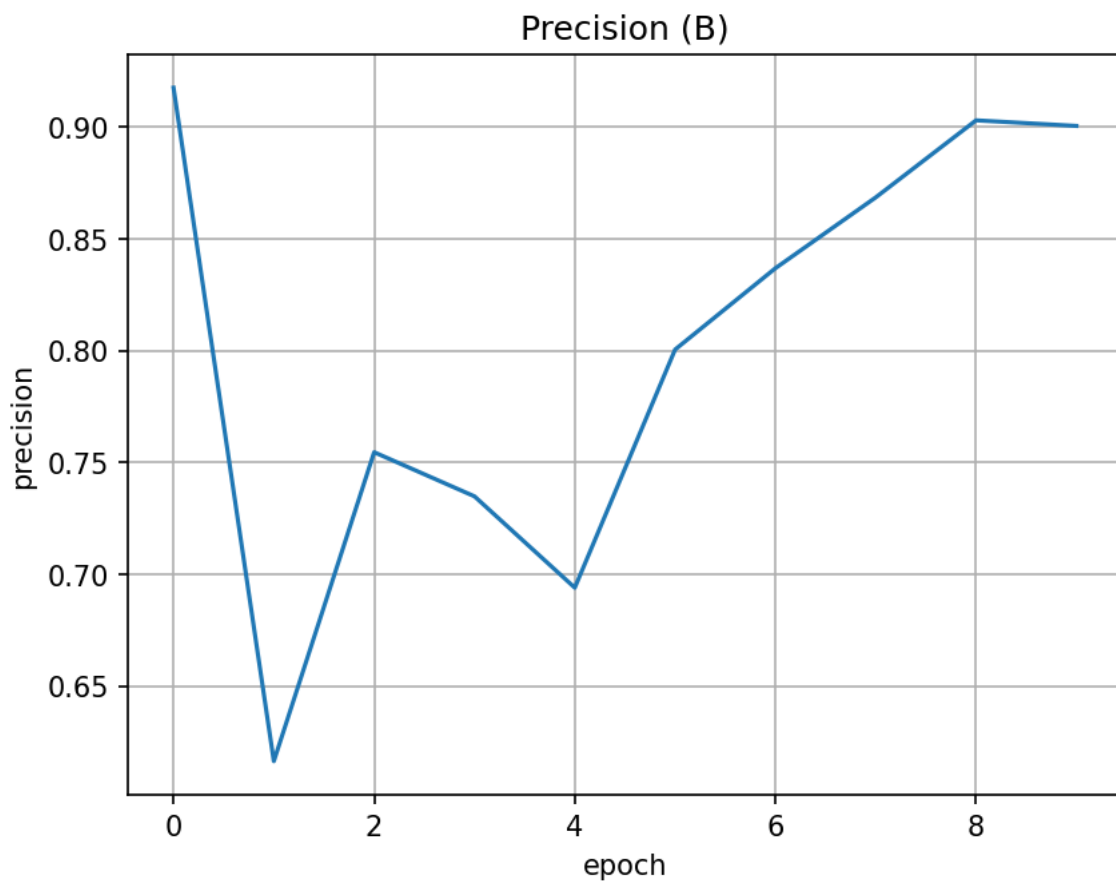
Métricas finales

Métrica	Valor
Precision	0.9004
Recall	0.9045

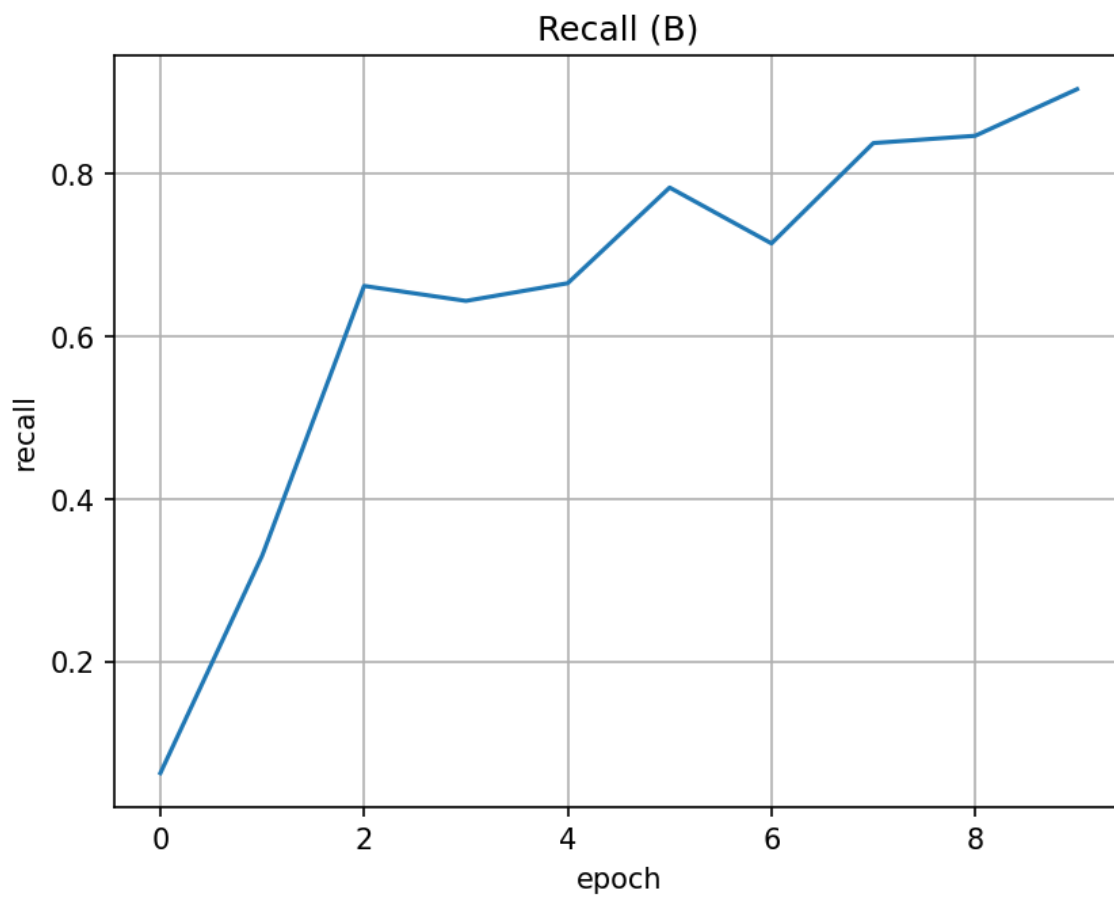
mAP50	0.9337
mAP50-95	0.7612
Box loss	0.6456
Cls loss	0.8156
DFL loss	1.0630

Curvas y visualizaciones

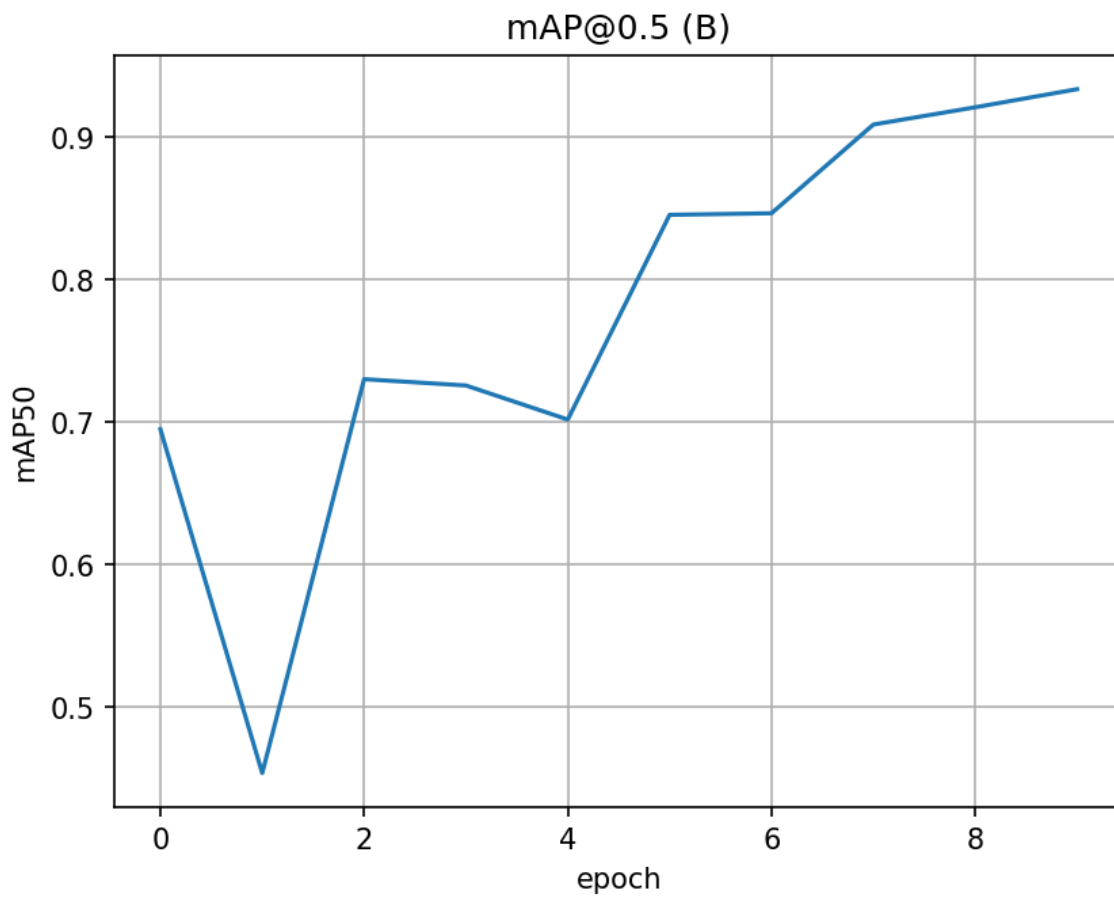
curve_precision.png



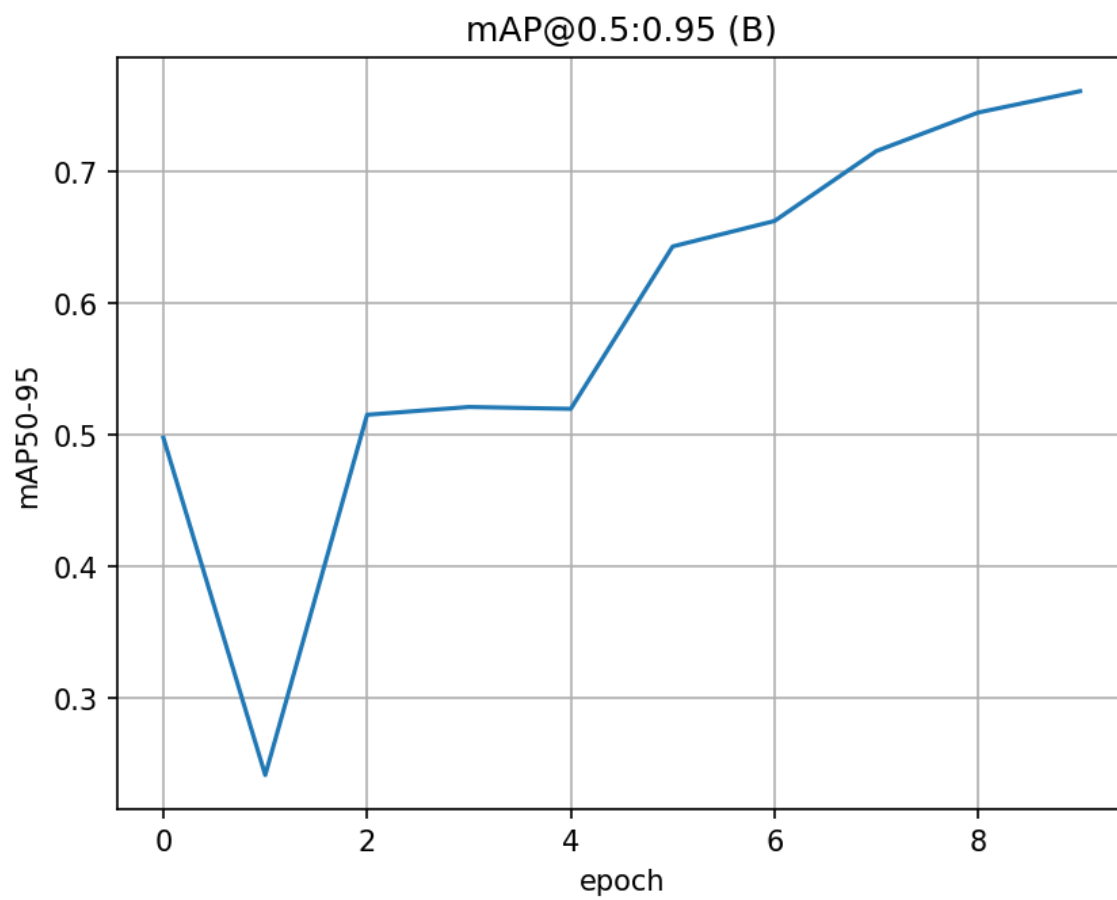
curve_recall.png



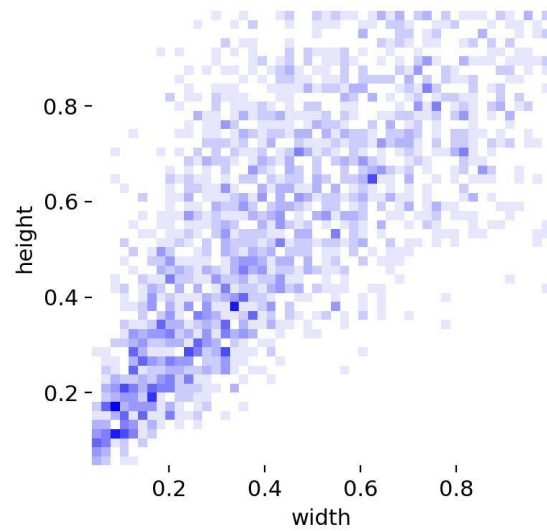
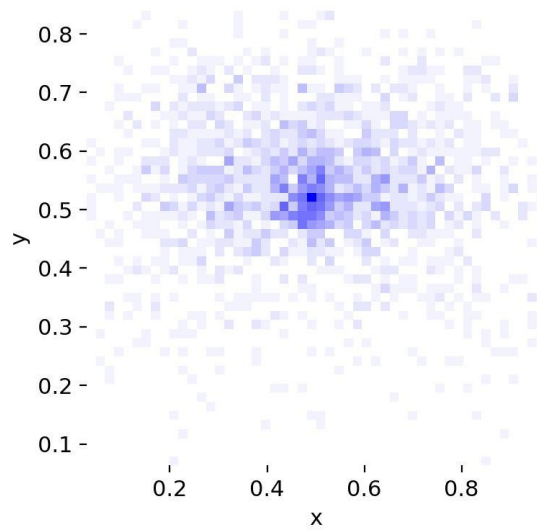
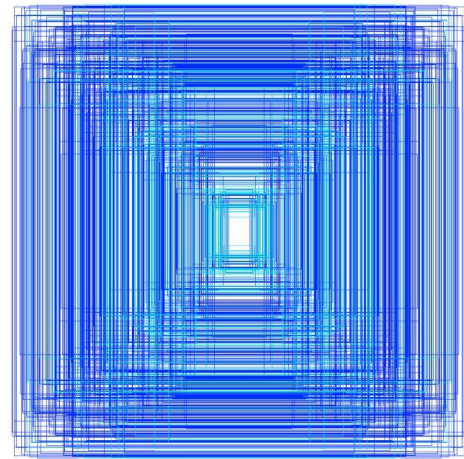
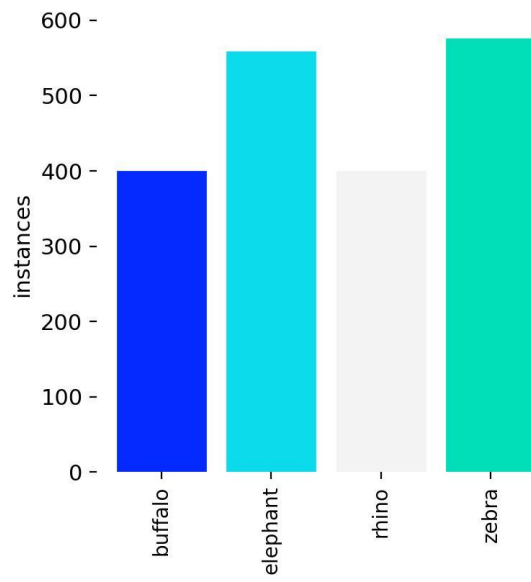
curve_map50.png



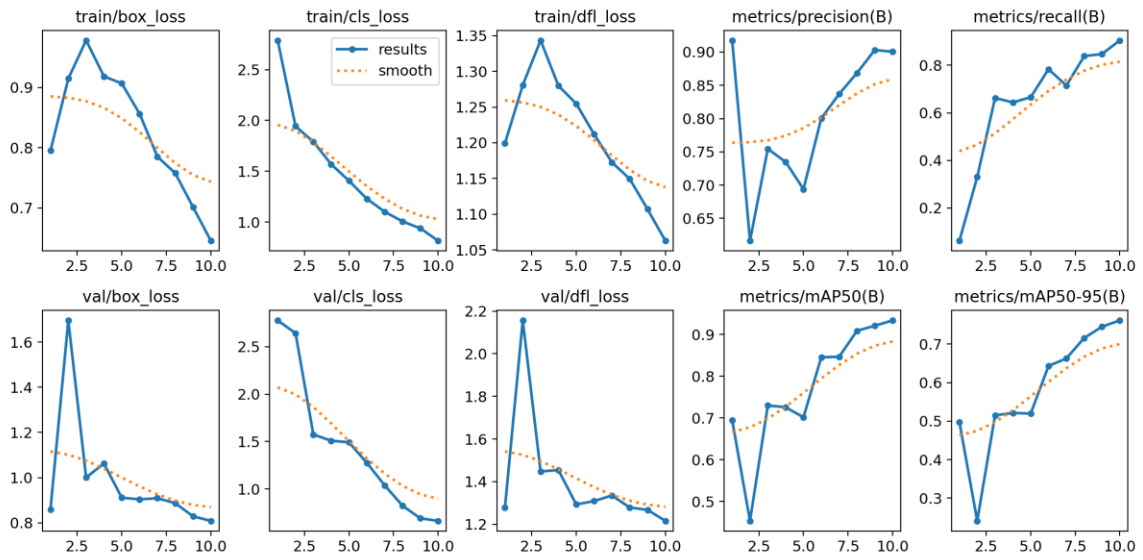
curve_map5095.png



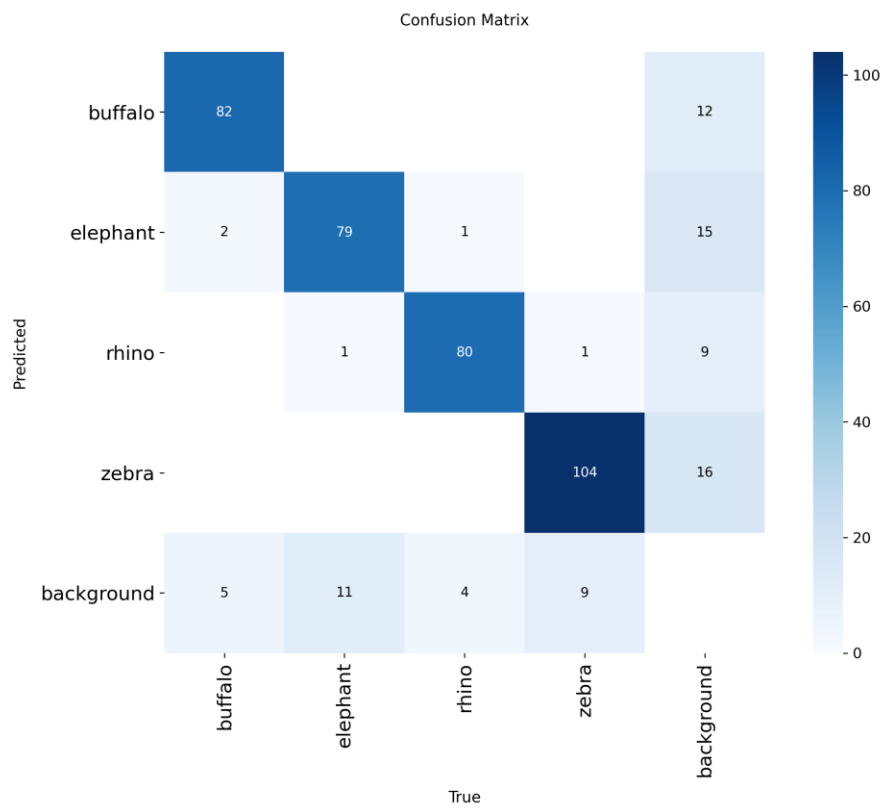
labels.jpg



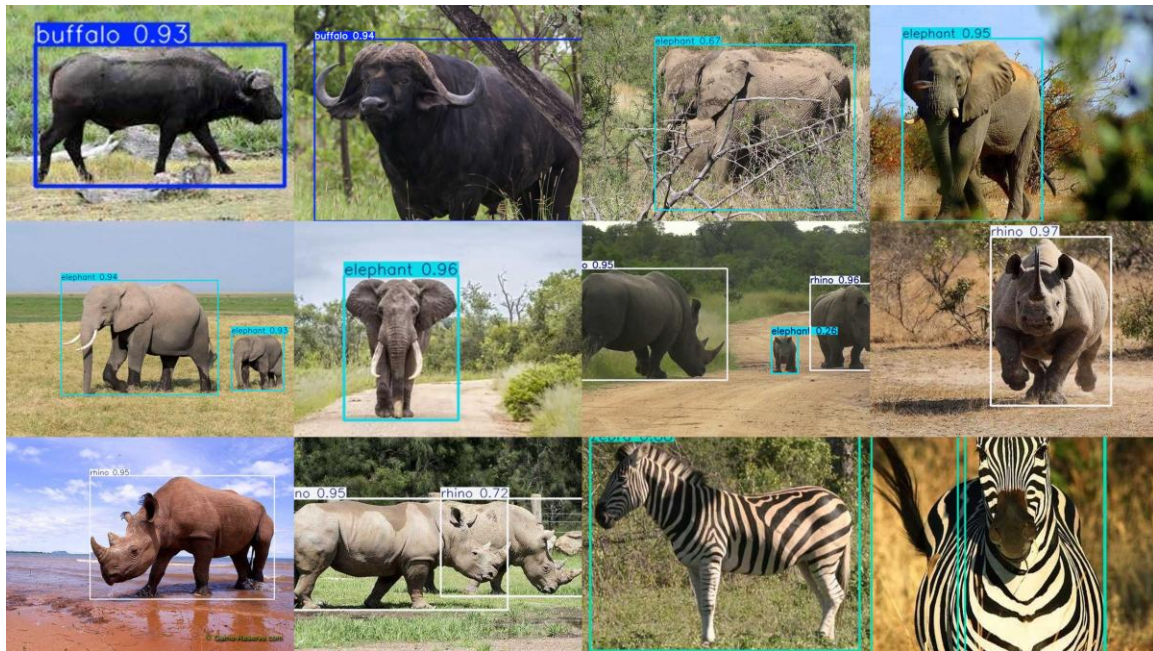
results.png



confusion_matrix.png



grid_predictions.png



Conclusiones (borrador)

El modelo YOLO11n alcanzó un desempeño final de:

- mAP@0.5 = 0.934
 - mAP@0.5:0.95 = 0.761
- con Precision=0.900 y Recall=0.905.

Observaciones:

- La curva mAP muestra mejora progresiva y estabilización al final del entrenamiento.
- Las clases con menor soporte de datos pueden beneficiarse de mayor augmentación o más épocas.
- Para uso en cámaras trampa, considerar umbrales de confianza específicos por clase.

Referencias

Ultralytics — African Wildlife Dataset: <https://docs.ultralytics.com/datasets/detect/african-wildlife/>

LearnOpenCV (Ankan Ghosh, 2025) — Fine-Tuning RetinaNet:

<https://learnopencv.com/finetuning-retinanet/>

Kaggle EDA Image Datasets — Fajri (2022): <https://www.kaggle.com/code/faldaoae/exploratory-data-analysis-eda-for-image-datasets>