

Redioma

Curso: Visión por Computadora aplicado a la Biodiversidad

Proyecto: Aprendizaje profundo aplicado a datos de Biodiversidad

Alberto Méndez Rodríguez, Costa Rica

Conjunto de datos utilizado:

El conjunto de datos para detectar y entrenar por medio de algoritmos de vision artificial se obtuvo de **Ultralytics. African Wildlife Dataset. Last Update: 17/03/2025** URL: <https://docs.ultralytics.com/datasets/detect/african-wildlife/>

Se tiene unas imágenes de animales de vida silvestre africana, como búfalos, elefantes, rinocerontes y cebra. Estos datos fueron bajados del sitio antes mencionados, básicamente se usaron 1052 imágenes para entrenar un modelo, 225 para validarlo y se usaron 227 de prueba.

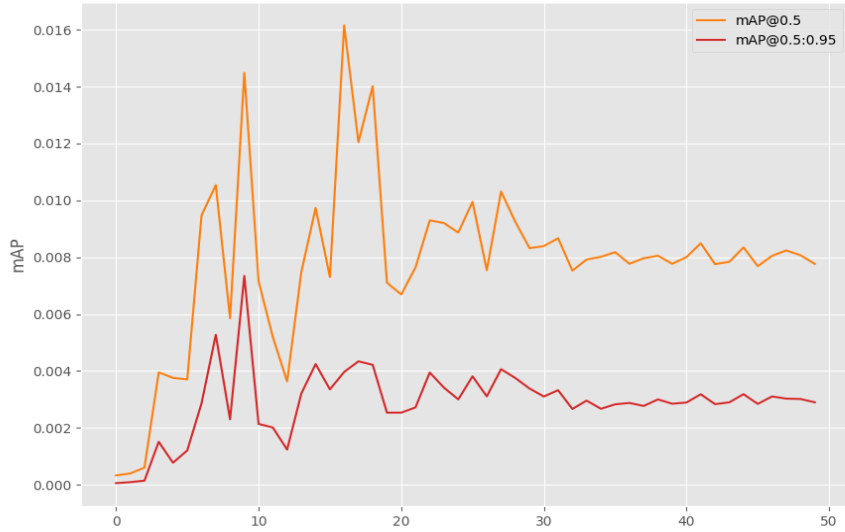
En cuanto al modelo usado para entrenar se uso YOLO11, obtenido de la pagina: <https://docs.ultralytics.com/es/models/yolo11/>.

Este modelo es la última interacción de la serie YOLO de **Ultralytis**, que detecta en tiempo real objetos, lo cual mejora su rendimiento y eficiencia.

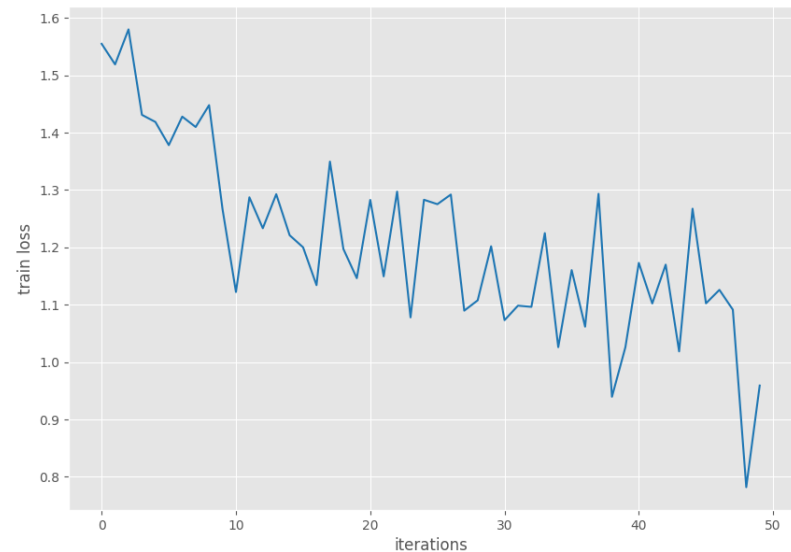
Resumen de resultados

Luego del corrido con 50 epoch, pues mayor número no fue posible por limitaciones de maquina de procesamiento, se obtuvo que el modelo se comporta bien aun para objetos traslapados ([mAP@0.5](#)). En el caso de la curva [mAP@0.5:0.95](#) los resultados son un poco inferiores en general.

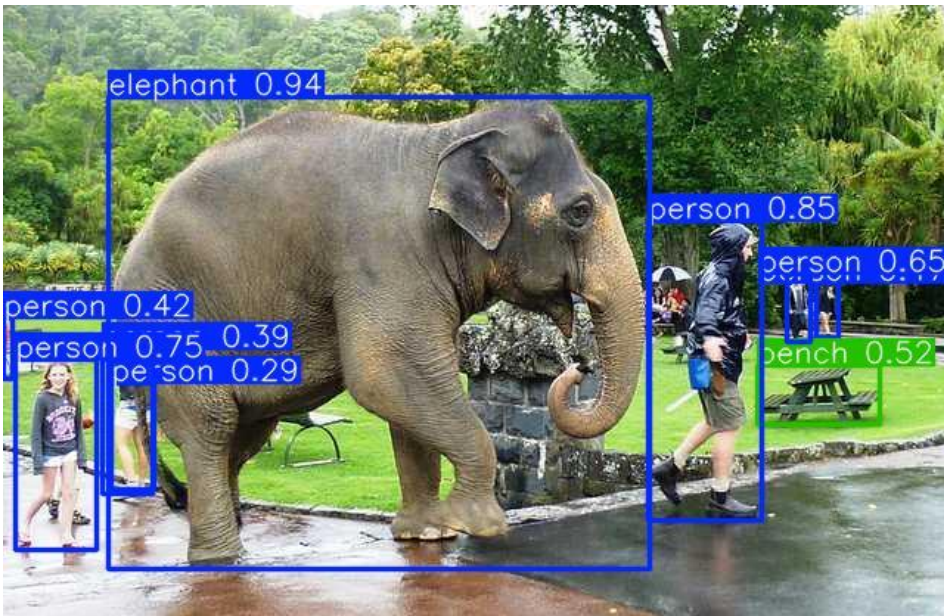
Sin embargo en ambos casos las corridas a EPOCH de menos de 20 interacciones es muy fluctuante, posiblemente por presencia de datos faltantes, después de 20 se estabilizan ambas curvas, lo que puede indicar que el modelo entro en un estado de no aprender nuevas características.



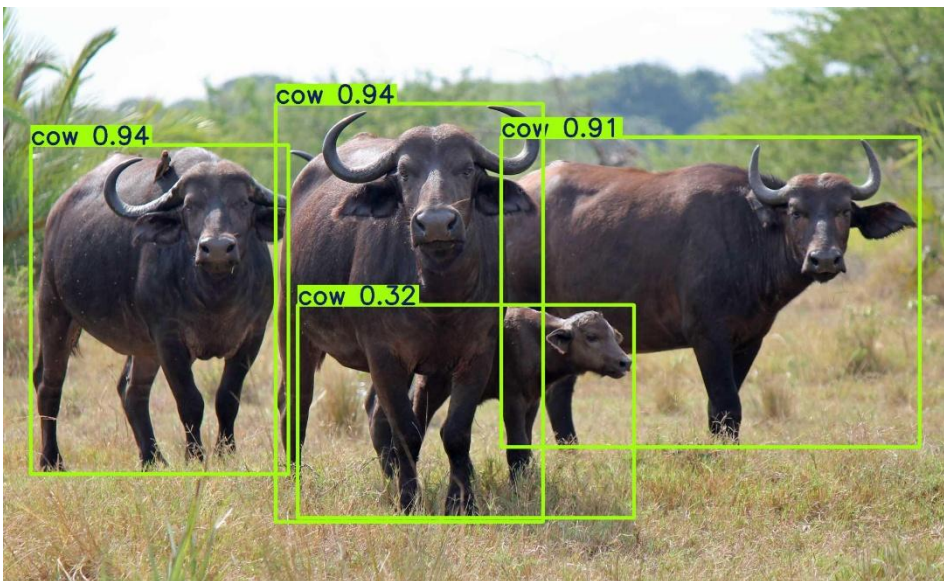
En el caso del gráfico de perdida de entrenamiento, vemos que el comportamiento del modelo es a la baja entre más corridas de interacciones se hagan, lo que indica que entre mas interacciones obtendrá mejores predicciones, eso si se nota una mayor divergencia en las interacciones, lo cual se nota por la cantidad de picos hacia arriba y hacia abajo.



En el caso de FineTuning RetinaNet (Gosh, 2025) Gosh, Ankan. FineTuning RetinaNet for Wildlife Detection with PyTorch: A Step-by-Step Tutorial. LearnOpenCV website URL: <https://learnopencv.com/finetuning-retinanet/>) el corrido de los script y los modelos propuestos, indica que los resultados fueron bastante buenos, si bien se corrió para dos imágenes, una de los cuales el modelo identificó bien los objetos encontrados



Sin embargo para esta otra imagen el resultado en el numero de objetos fue buena, pero no clasificándolo, posiblemente por las características introducidas del objeto Bufalo que lo confundió con Vaca, esto habría que tener más claro cuales fueron las características a buscar por objeto y corregirlas mas.



Reflexión sobre la experiencia

El ejercicio como mecanismo de observar las potencialidades de estos algoritmos y sus modelos, fue bastante interesante, sin embargo, el problema detectado es que son cajas negras, donde no sabemos cómo llegar a desarrollar modelos parecidos y ni siquiera analizar esos modelos.

Considero que hubiera sido más interesante ejecutar los modelos en clase y analizar cómo se hicieron. Sin embargo, como experiencia de lo que podríamos hacer es increíble.

El otro problema a tomar en cuenta es muchos de nosotros no tenemos disponibles supercomputadoras y debemos ejecutarlo en Colab, pero usar maquinas de mas núcleos aquí se hace complicado