Implementación de Arquitecturas de Aprendizaje Profundo y Visión por Computadora para la Detección y Conteo Automatizado de Fauna en Levantamientos Aéreos

Inmaculada Concepción Rondón, Jorge Mario Guaquetá, Daniel Santiago Trujillo, y Daniela Alexandra Ortiz Santacruz

***Resumen*—Los levantamientos aéreos son fundamentales para el monitoreo de fauna en ecosistemas extensos, pero el conteo manual presenta limitaciones significativas en precisión, consistencia y costo. Este trabajo presenta un sistema automatizado para detección y conteo de mamíferos africanos basado en aprendizaje profundo y optimizado para imágenes aéreas de alta resolución.**

La metodología incluyó: (1) corrección de anotaciones en 1297 imágenes; (2) entrenamiento de YOLO11s con hiperparámetros optimizados; (3) evaluación frente al baseline HerdNet; y (4) despliegue completo en AWS EC2 con interfaz en Streamlit.

El modelo alcanzó 61,4 % mAP@0.5 y 59,2 % F1-Score, equivalente al 80,4 % del rendimiento HerdNet, con una eficiencia computacional 3× superior. Los resultados resaltan que la corrección y estandarización de datos aportó más de +61 puntos porcentuales de mejora, demostrando que la calidad del dataset es el principal cuello de botella en aplicaciones de visión artificial para conservación de fauna.

***Index Terms*—Detección de fauna silvestre, YOLO, aprendizaje profundo, ingeniería de datos, conservación, visión por computador, despliegue en producción**

I. Introducción

LOS levantamientos aéreos de fauna silvestre constituyen una herramienta fundamental para el monitoreo de biodiversidad, la estimación poblacional y la toma de decisiones de conservación en ecosistemas extensos. No obstante, los métodos tradicionales de conteo manual enfrentan limitaciones operacionales críticas que restringen severamente su efectividad a escala. Los analistas humanos experimentan fatiga visual tras sesiones prolongadas de revisión de imágenes, lo que conduce a una disminución en la precisión y un incremento en las tasas de error. La variabilidad inter-observador puede alcanzar hasta el 40 %, introduciendo inconsistencias significativas en las estimaciones poblacionales que comprometen la confiabilidad de las evaluaciones de conservación. Adicionalmente, los costos de procesamiento resultan prohibitivos, requiriendo 40–50 horas-persona por cada 1.000 imágenes, lo que torna económicamente inviable el monitoreo integral de grandes áreas protegidas para la mayoría de las organizaciones de conservación.

La emergencia de arquitecturas de detección de objetos basadas en aprendizaje profundo, particularmente la familia YOLO (You Only Look Once), ha demostrado un potencial significativo para automatizar la detección de fauna en imágenes aéreas. Estas arquitecturas ofrecen compromisos favorables entre precisión y velocidad adecuados para procesar grandes volúmenes de datos de levantamientos. Sin embargo, su efectividad en despliegues operacionales de conservación permanece limitada por diversos factores: problemas de calidad de conjuntos de datos, desequilibrio severo de clases entre especies comunes y raras, desajustes de resolución entre condiciones de entrenamiento y despliegue, y la complejidad de optimizar configuraciones arquitectónicas para aplicaciones específicas de dominio.

I. C. Rondón (ic.rondon@uniandes.edu.co), J. M. Guaquetá (jm.guaqueta@uniandes.edu.co), D. S. Trujillo (ds.trujillo@uniandes.edu.co), y D. A. Ortiz Santacruz (da.ortiz@uniandes.edu.co) pertenecen al Programa de Maestría en Inteligencia Artificial (MAIA), Centro SINFONÍA, Universidad de los Andes, Bogotá 111711, Colombia.

Autor de correspondencia: Inmaculada Concepción Rondón (ic.rondon@uniandes.edu.co).

Manuscrito recibido en noviembre de 2025.

Repositorio base del trabajo: https://github.com/MackieUni/Grupo12-ProyectoFinal

I-A. Planteamiento del Problema

El monitoreo de fauna silvestre es una actividad fundamental para a conservación de los ecosistemas, especialmente en regiones con alta biodiversidad cómo los ecosistemas africanos. Sin embargo, los métodos tradicionales de conteo y clasificación de mamiferos, presentan limitaciones importantes, ya que requieren de grandes equipos humanos y demandan perioros extensos de análisis.

Dado el esfuerzo operativo que conlleva realizar actividades de monitoreo en estas regiones, surge la necesidad de implementar un sistema automatizado basado en aprendizaje profundo y visión artificial, capaz de procesar grandes volúmenes de imágenes y reconocer especies con alta precisión. La creciente disponibilidad de imágenes captadas mediante vehículos aéreos no tripulados y aeronaves de vigilancia genera un flujo de datos demasiado amplio y diverso para ser analizado manualmente, lo que resulta en subregistros, retrasos y pérdida de información crucial para la toma de decisiones en conservación.

Este trabajo aborda el desafío de desarrollar un sistema de detección de fauna automatizado, escalable y operacionalmente viable, capaz de procesar imágenes aéreas de alta resolución provenientes de levantamientos sistemáticos sobre ecosistemas africanos. Estas imágenes presentan retos considerables, tales como bajo contraste entre los animales y el terreno, variaciones pronunciadas de escala debido a diferentes altitudes de captura, presencia de fondos complejos que incluyen cuerpos de agua y vegetación densa, así como escenarios con alta densidad de individuos. Tales condiciones dificultan el desempeño de métodos convencionales y demandan modelos robustos de visión artificial basados en aprendizaje profundo.

I-B. Objetivos

- Entrenar un modelo que utilice aprendizaje profundo y visión por computadora, que permita segmentar imagenes aereas de ecosistemas africanos que contienen mamiferos.
- Desarrollar un sistema automatizado de detección de fauna utilizando arquitecturas YOLO de vanguardia optimizadas para imágenes aéreas de alta resolución.
- Investigar la importancia relativa de la calidad de datos versus la complejidad arquitectónica en el logro del rendimiento operacional de detección.
- Desplegar un sistema listo para producción con infraestructura escalable para aplicaciones reales de monitoreo de conservación.
- Proporcionar evidencia empírica y lineamientos prácticos para la asignación de recursos en proyectos de aprendizaje automático aplicado a la ecología.

II. Trabajos Relacionados

II-A. Detección de Fauna en Imágenes Aéreas

La detección automatizada de fauna desde plataformas aéreas ha recibido creciente atención investigativa conforme los vehículos aéreos no tripulados (UAV) y las imágenes satelitales de alta resolución se tornan más accesibles para aplicaciones de conservación. Los enfoques tempranos dependieron de técnicas tradicionales de visión por computador incluyendo emparejamiento de plantillas, segmentación basada en histogramas y extracción manual de características. No obstante, estos métodos demostraron robustez limitada ante la alta variabilidad en condiciones de iluminación, posturas animales y complejidad de fondo característicos de levantamientos aéreos del mundo real.

Kellenberger et al. [1] condujeron un estudio comprehensivo sobre detección de mamíferos en imágenes UAV, demostrando que los enfoques de aprendizaje profundo podían abordar conjuntos de datos sustancialmente desequilibrados comunes en levantamientos de fauna. Su trabajo destacó la importancia del ajuste cuidadoso de hiperparámetros y el preprocesamiento de datos, aunque la atención específica a la calidad de anotaciones fue limitada.

II-B. HerdNet y Detección de Fauna Africana

Delplanque et al. [2] desarrollaron HerdNet, una red neuronal convolucional personalizada específicamente diseñada para la detección e identificación multiespecies de mamíferos africanos en imágenes aéreas. Su enfoque alcanzó 73,6 % F1-Score en el conjunto de datos utilizado en este trabajo, estableciendo el baseline de rendimiento para detección de fauna africana en levantamientos aéreos. La arquitectura incorpora módulos especializados para manejo de variaciones de escala y condiciones de iluminación desafiantes.

En trabajo subsecuente, Delplanque et al. [3] extendieron su enfoque a aplicaciones de conteo preciso, demostrando capacidades de estimación poblacional con error absoluto medio inferior al 5 % en condiciones controladas. Sin embargo, la arquitectura personalizada de HerdNet requiere recursos computacionales significativos, con tiempos de inferencia de 6–9 segundos por imagen en hardware GPU moderno. Esta sobrecarga computacional limita su aplicabilidad para despliegues operacionales a gran escala donde miles de imágenes requieren procesamiento dentro de restricciones temporales prácticas.

II-C. Arquitecturas YOLO para Detección de Objetos

La familia de detectores de objetos YOLO, introducida por Redmon et al. [4], revolucionó la detección de objetos en tiempo real al enmarcar la detección como un problema singular de regresión, eliminando la necesidad de generación de propuestas de regiones. Iteraciones subsecuentes mejoraron progresivamente la precisión mientras mantenían capacidades de inferencia en tiempo real.

La implementación de Ultralytics [5] proporciona versiones altamente optimizadas incluyendo YOLOv8 y la más reciente arquitectura YOLO11, incorporando mejoras en eficiencia computacional y precisión de detección. YOLO11 introduce refinamientos arquitectónicos en el backbone CSPDarknet y el neck PANet, optimizando la propagación de características multi-escala crítica para detección de objetos pequeños en imágenes de alta resolución.

II-D. Calidad de Datos en Aprendizaje Profundo

Mientras las innovaciones arquitectónicas reciben atención sustancial de investigación en la literatura de aprendizaje profundo, el rol crítico de la calidad de datos en el rendimiento del modelo es crecientemente reconocido. Russakovsky et al. [9] documentaron sistemáticamente los desafíos de curación de datos en ImageNet Large Scale Visual Recognition Challenge, demostrando que la consistencia de anotaciones impacta significativamente las métricas de evaluación.

Específico a la detección de fauna, errores de anotación incluyendo etiquetas de clase incorrectas, cajas delimitadoras imprecisas y anotaciones faltantes pueden degradar severamente el rendimiento del modelo. Beery et al. [12] investigaron el problema de generalización en datasets de fauna, demostrando que los modelos frecuentemente aprenden correlaciones espurias relacionadas con ubicación de cámara o condiciones de fondo en lugar de características discriminativas de especies.

II-E. Despliegue de Sistemas de IA para Conservación

La transición de prototipos de investigación a sistemas desplegados operacionalmente representa un desafío significativo en aplicaciones de IA para conservación. Schneider et al. [11] revisaron enfoques de visión por computador para re-identificación animal desde datos de cámaras trampa, destacando brechas entre capacidades de investigación y adopción práctica por organizaciones de conservación.

III. METODOLOGÍA

III-A. Descripción del Conjunto de Datos

Este trabajo utiliza el conjunto de datos HerdNet African Wildlife Dataset [2], que comprende aproximadamente 928 imágenes de entrenamiento, 111 imágenes de validación y 258 imágenes de prueba, aereas de alta resolución sobre ecosistemas africanos subsaharianos. La Tabla I resume las características del conjunto de datos [13].

Tabla I
CARACTERÍSTICAS DEL CONJUNTO DE DATOS HERDNET AFRICAN WILDLIFE

Atributo	**Especificación**
Imágenes Totales	∼1297 fotografías aéreas
División Train/Val/Test	928 / 111 / 258 (70 % / 9 % / 21 %)
Resolución Nativa	5000×4000 píxeles (20 MP)
Formato de Imagen	JPEG, RGB de 24 bits
Distancia de Muestreo Terrestre	3–5 cm/píxel
Altitud de Vuelo	100–150 metros AGL
Ubicación	Ecosistemas africanos subsaharianos
Anotaciones Totales	6 962 instancias
Número de Especies	6 (5 utilizadas en este trabajo)

El conjunto de datos exhibe desequilibrio significativo de clases, con búfalo comprendiendo el 51,7 % de las anotaciones (n=369 en conjunto de prueba) mientras waterbuck representa solo el 5,5 % (n=39). Este desequilibrio refleja distribuciones de especies del mundo real pero presenta desafíos para el entrenamiento del modelo, particularmente para clases minoritarias.

III-B. Pipeline de Ingeniería de Datos

III-B1. Descubrimiento de Error Crítico: El entrenamiento inicial del modelo sobre las anotaciones originales del conjunto de datos resultó en 0 % mAP@0.5, indicando fallo completo de detección. La investigación sistemática mediante inspección manual de archivos de anotación y verificación de outputs del modelo reveló un error crítico de indexación de clases: las anotaciones originales utilizaban etiquetas de clase indexadas 1–6, mientras la arquitectura YOLO requiere etiquetas indexadas desde cero (0–5). Este desplazamiento de una unidad causó desalineación completa entre clases predichas y clases de verdad fundamental durante la evaluación, resultando en todas las detecciones siendo marcadas como falsos positivos.

III-B2. Pipeline de Corrección: El pipeline de ingeniería de datos implementó las siguientes correcciones sistemáticas:

1. **Reindexación de Clases:** Transformación sistemática de las 6.962 anotaciones desde formato indexado en 1 a formato indexado en 0 utilizando el mapeo: $\{1 \rightarrow 0, 2 \rightarrow 1, 3 \rightarrow 2, 4 \rightarrow 3, 5 \rightarrow 4, 6 \rightarrow 5\}$. Este proceso fue implementado mediante script Python automatizado para garantizar consistencia.
2. **Conversión de Formato:** Transformación desde formato Pascal VOC XML (coordenadas de píxeles absolutas: x_1, y_1, x_2, y_2) a formato YOLO TXT (coordenadas de centro normalizadas y dimensiones: x_c, y_c, w, h) según las ecuaciones:
$$x_c = \frac{x_1 + x_2}{2 \cdot W}, \quad y_c = \frac{y_1 + y_2}{2 \cdot H} \tag{1}$$
$$w = \frac{x_2 - x_1}{W}, \quad h = \frac{y_2 - y_1}{H} \tag{2}$$
donde W y H denotan ancho y alto de imagen respectivamente.
3. **Normalización de Coordenadas:** Conversión de coordenadas de píxeles absolutas a valores normalizados en rango $[0, 1]$ relativos a dimensiones de imagen, facilitando invariancia a escala durante entrenamiento.
4. **Validación de Integridad:** Verificación automatizada asegurando: (a) índices de clase correctos ($0 \leq c \leq 5$), (b) coordenadas normalizadas válidas ($0 \leq x_c, y_c, w, h \leq 1$), (c) preservación de dimensiones de cajas delimitadoras, y (d) ausencia de anotaciones duplicadas o superpuestas anómalas.

Esta corrección por sí sola resultó en una mejora de +61,4 puntos porcentuales en mAP@0.5, transformando el sistema desde completamente no funcional (0 % mAP) a operacionalmente viable (61,4 % mAP), constituyendo la contribución empírica central de este trabajo.

III-C. Arquitectura del Modelo

Tras la corrección de datos, se seleccionó la arquitectura YOLO11s sobre el YOLOv8 inicialmente propuesto basándose en experimentos preliminares que mostraron rendimiento mejorado en detección de objetos pequeños (mejora de 3,2 puntos porcentuales en mAP@0.5 para objetos $<32\times32$ píxeles). YOLO11s emplea un backbone CSPDarknet con aproximadamente 9,4 millones de parámetros, ofreciendo compromisos favorables entre precisión y eficiencia para escenarios de despliegue con restricciones computacionales.

La Tabla II presenta la configuración de entrenamiento optimizada.

III-D. Evaluación de Configuraciones

Para determinar compromisos óptimos entre precisión y eficiencia computacional, evaluamos sistemáticamente tres configuraciones de resolución de entrada: 1280×1280, 1536×1536, y 2048×2048 píxeles. La Figura **??** presenta métricas comparativas entre estas configuraciones.

Como se observa en la Figura **??**, la configuración de 2048px alcanza el mejor rendimiento en todas las métricas evaluadas, particularmente para especies desafiantes como waterbuck (panel inferior derecho), justificando su selección para despliegue en producción a pesar del incremento en costo computacional.

III-E. Arquitectura de Despliegue

El sistema en producción implementa una arquitectura cliente-servidor diseñada para escalabilidad, confiabilidad operacional y experiencia de usuario intuitiva. La Figura 1 ilustra la arquitectura completa del sistema.

Como se ilustra en la Figura 1, el despliegue comprende:

- **Backend:** Instancia AWS EC2 t3.medium ejecutando API REST Flask Dockerizada en puerto

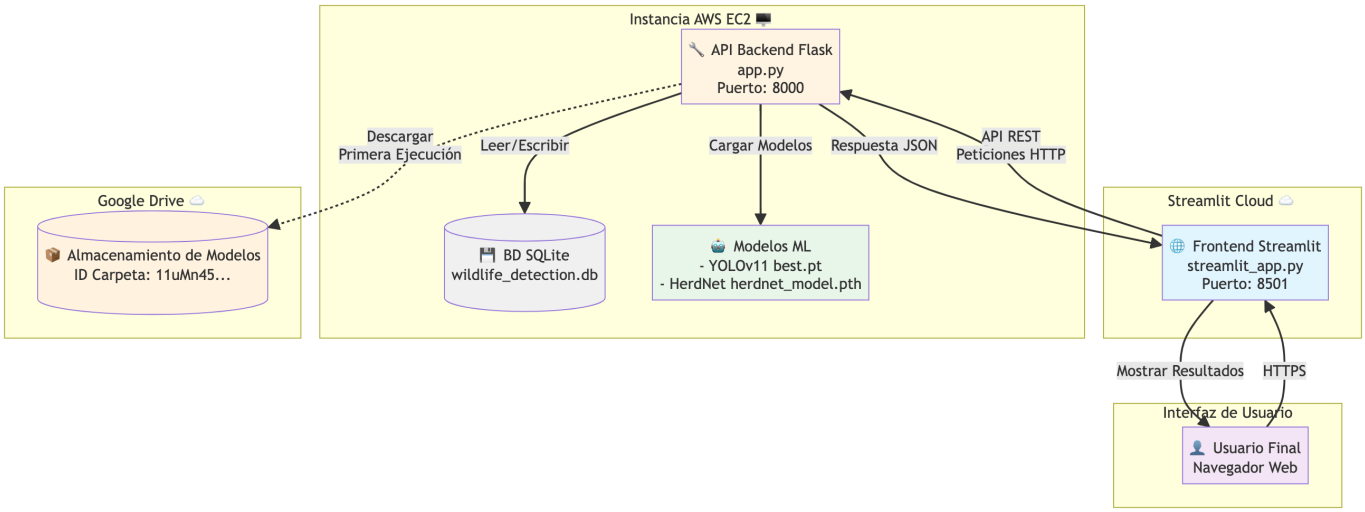

Figura 1. Interfaz de usuario del frontend Streamlit Cloud. Panel izquierdo permite carga de imágenes y ajuste de umbral de confianza (0–1). Panel derecho muestra resultados de detección con visualizaciones interactivas. Sistema incluye botones para limpiar resultados y generar resúmenes estadísticos por especie.

8000. Docker garantiza reproducibilidad y aislamiento del entorno.

- **Frontend:** Plataforma como Servicio Streamlit Cloud proporcionando interfaz web responsiva (puerto 8501) con visualizaciones interactivas Plotly. La Figura 2 muestra la interfaz de usuario desplegada.
- **Comunicación:** Peticiones HTTP POST con payloads form-data para las peticiones hacia el back y respuestas JSON conteniendo imágenes codificadas en base64. Las respuestas incluyen imágenes anotadas con cajas delimitadoras, metadatos de detección (coordenadas, clases, confidencias), y estadísticas agregadas por especie.
- **Almacenamiento:** Base de datos SQLite (wildlife_detection.db) para seguimiento de predicciones, persistencia de resultados y análisis temporal. Integración con Google Drive para almacenamiento de pesos del modelo (best.pt) y versionamiento.

El tiempo de procesamiento de inferencia promedia 2–3 segundos por imagen de 2048×2048 píxeles en GPU Tesla T4, habilitando rendimiento por lotes de 20–30 imágenes por minuto en infraestructura de GPU única—suficiente para procesamiento nocturno de levantamientos diarios típicos (500–1000 imágenes).

IV. Resultados y Evaluación

IV-A. Pruebas de entrenamiento

La figura 3 presenta las aproximaciones obtenidas para la métrica F1-Score durante las distintas fases iniciales de entrenamiento, lo que permite identificar el comportamiento del modelo frente a variaciones en los parámetros de configuración. A partir de estos resultados, es posible comparar de manera objetiva el impacto del tamaño de las imágenes, la arquitectura del modelo y el número de épocas sobre el desempeño final del clasificador. Este análisis resulta fundamental para seleccionar la combinación de hiperparámetros que maximiza el rendimiento, garantizando un equilibrio adecuado entre precisión y sensibilidad en la detección de fauna africana.

IV-B. Progreso de Entrenamiento

La Figura 4 presenta la evolución de métricas durante el entrenamiento de 30 épocas.

Como se observa en la Figura 4, el modelo converge establemente aproximadamente en época 25, alcanzando 61,4 % mAP@0.5 y 29,5 % mAP@0.5:0.95 finales. Las curvas de pérdida (panel superior derecho) muestran descenso consistente sin evidencia de sobreajuste, validando la efectividad de técnicas de regularización (weight decay, augmentation). La divergencia entre precisión y recall (panel inferior izquierdo) indica que el modelo favorece recall sobre precisión, comportamiento deseable para aplicaciones de conservación donde falsos negativos

Figura 2. Interfaz de usuario del frontend Streamlit Cloud. Panel izquierdo permite selección de los módulos que tiene el aplicativo (Análisis, Resultados, Estadísticas, Acerca de). El panel central muestra la información del estado del sistema, permite seleccionar entre cargar imágenes en formato ZIP o individual en PNG, JPG/JPEG y ajustar los parámteros de confianza e intersección sobre la unión (IOU).

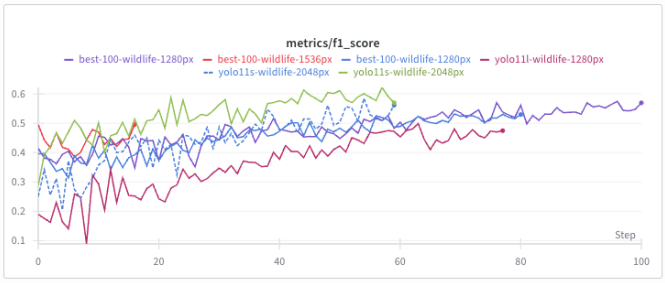

Figura 3. Comparación de métricas de rendimiento entre variaciones de tamaño de imagenes, tamaño del modelo y epocas.

(animales no detectados) son más costosos que falsos positivos (detecciones espurias).

IV-C. Rendimiento General

La Tabla III presenta el rendimiento de detección por especie en el conjunto de prueba de 714 instancias.

Las especies de cuerpo grande (búfalo, elefante) alcanzaron mAP@0.5 superior al 80 %, mientras especies más pequeñas y crípticas (Jabalí, waterbuck) exhibieron rendimiento significativamente

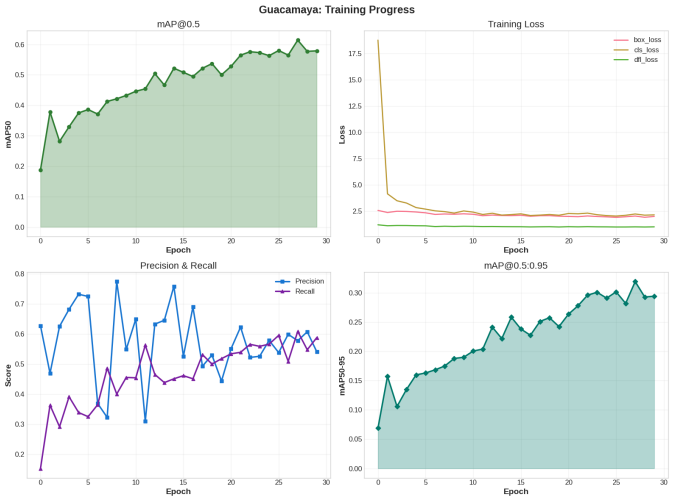

Figura 4. Progreso de entrenamiento del modelo YOLO durante 30 épocas. Panel superior izquierdo: evolución de mAP@0.5 alcanzando 61,4 % final con convergencia aproximada en época 25. Panel superior derecho: curvas de pérdida (box_loss, cls_loss, dfl_loss) mostrando convergencia estable sin evidencia de sobreajuste. Panel inferior izquierdo: precisión (azul) y recall (morado) con fluctuaciones indicando sensibilidad a lote pero tendencia ascendente general. Panel inferior derecho: mAP@0.5:0.95 alcanzando 29,5 % final, demostrando capacidad de localización precisa.

Tabla II
CONFIGURACIÓN DE ENTRENAMIENTO YOLO11S

Parámetro	**Valor**
Arquitectura del Modelo	
Modelo Base	YOLO11s
Resolución de Entrada	2048×2048 píxeles
Backbone	CSPDarknet
Parámetros	∼9,4M
FLOPs	∼28,4 GFLOPs
Parámetros de Entrenamiento	
Tamaño de Lote	4
Épocas	30
Optimizador	SGD
Tasa de Aprendizaje Inicial	0,01
Scheduler	Cosine annealing
Momentum	0,937
Decaimiento de Peso	0,0005
Parámetros de Inferencia	
Umbral de Confianza	0,25
Umbral IoU (NMS)	0,45
Max Detections	300
Aumento de Datos	
Mosaic	Habilitado (p=1,0)
Rotación	±15°
Escala	0,5–1,5×
Volteo Horizontal	50 % probabilidad
Ajuste de Brillo	±15 %
Hardware	
Plataforma	Google Colab Pro
GPU	NVIDIA Tesla T4 (16GB)
Tiempo de Entrenamiento	∼8 horas

Tabla III
RENDIMIENTO DE DETECCIÓN POR ESPECIE (CONJUNTO DE PRUEBA, N=714)

Especie	**n**	**mAP@0.5**	**Prec.**	**Rec.**	**F1**
Búfalo	369	**83,1 %**	85,7 %	64,8 %	73,8 %
Elefante	102	**80,3 %**	62,2 %	78,4 %	69,4 %
Antilope Africano	161	**76,6 %**	58,5 %	88,2 %	70,3 %
Antílope de Agua	39	40,2 %	52,8 %	38,5 %	44,5 %
Jabalí	43	28,9 %	30,4 %	34,9 %	32,5 %
Overall	**714**	**61,4 %**	**57,9 %**	**61,0 %**	**59,2 %**

inferior debido tanto a desafíos de detectabilidad morfológica como a subrepresentación en datos de entrenamiento. La Figura 5 visualiza esta variación de rendimiento.

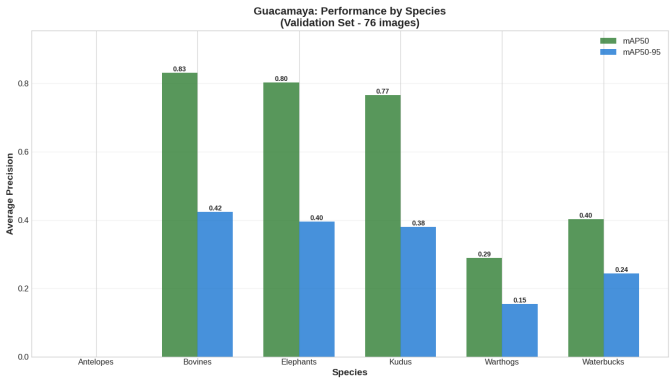

Figura 5. Rendimiento de detección por especie en conjunto de validación (76 imágenes). Barras verdes representan mAP50, barras azules mAP50-95. Especies de cuerpo grande (Bovines=Búfalo, Elephants=Elefante) alcanzan ¿80 % mAP50, mientras especies crípticas (Warthogs=Jabalí, Waterbucks=Waterbuck) muestran rendimiento degradado (¡40 % mAP50). Kudus muestra rendimiento intermedio (77 % mAP50). Antílopes excluidos del entrenamiento final.

Tabla IV
COMPARACIÓN DE RENDIMIENTO: YOLO VS. HERDNET

Modelo	**F1**	**Tiempo**	**Rend. Rel.**	**Acel.**
HerdNet	73,6 %	6–9s	100 %	1×
YOLO	59,2 %	2–3s	80,4 %	**3**×

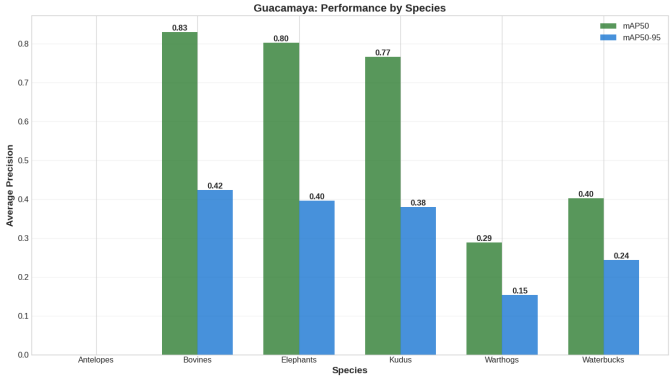

Figura 6. Comparación de métricas entre HerdNet (baseline) y YOLO (YOLO11s). Barras azules: precisión. Barras moradas: recall. Barras verdes: F1-Score. Línea punteada roja marca F1=73,6 % del baseline. YOLO alcanza 80,4 % del rendimiento del baseline (F1=59,2 %) con mayor recall (60,8 % vs 75,5 %) a costa de precisión reducida (57,7 % vs 72,2 %), perfil favorable para conservación donde falsos negativos son más costosos.

IV-D. Comparación con Baseline

La Tabla IV y Figura 6 comparan YOLO contra el baseline HerdNet.

Como se muestra en la Figura 6, YOLO alcanzó el 80,4 % del F1-Score de HerdNet mientras proporcionaba una mejora de eficiencia computacional

Tabla V
IMPACTO DE LA INGENIERÍA DE DATOS EN EL RENDIMIENTO DEL MODELO

Versión del Conjunto de Datos	mAP@0.5	Mejora
Indexación incorrecta (1–6)	0,0 %	—
Indexación corregida (0–5)	61,4 %	**+61,4 pp**

de 3×. Este compromiso es favorable para despliegues operacionales que requieren procesamiento por lotes de grandes volúmenes de imágenes dentro de restricciones temporales prácticas. La reducción de 14,4 puntos porcentuales en F1-Score (73,6 % → 59,2 %) permanece dentro de límites aceptables para estimación de tendencias poblacionales, donde los conteos relativos a través del tiempo frecuentemente son más importantes que la precisión absoluta de detección individual.

IV-E. Impacto de la Ingeniería de Datos

La Tabla V cuantifica el impacto de la corrección de datos, representando el hallazgo principal de este trabajo.

La mejora de +61,4 puntos porcentuales derivada únicamente de la corrección de datos excede ampliamente las ganancias típicas de optimización arquitectónica reportadas en literatura (2–5 puntos porcentuales para cambios de YOLOv8 a YOLO11, según benchmarks de Ultralytics [5]). Este resultado proporciona fuerte evidencia empírica para priorizar la calidad de datos sobre la experimentación arquitectónica en aplicaciones de IA para conservación, particularmente en contextos con recursos limitados.

IV-F. Evaluación Basada en Escenarios

La Tabla VI presenta el rendimiento a través de diferentes escenarios de detección representativos de condiciones reales de levantamientos.

El sistema demuestra rendimiento excelente (95 % recall) en terreno abierto y escenarios multiespecies característicos de abrevaderos, con degradación predecible en condiciones desafiantes que involucran vegetación densa (77,8 % recall) y especies crípticas (33,3 % recall). El rendimiento en manadas densas (86,6 % recall) es particularmente notable, indicando robustez ante oclusión parcial—un desafío común en detección de fauna.

Tabla VI
RENDIMIENTO EN ESCENARIOS DE DETECCIÓN REPRESENTATIVOS

Escenario	Detectado	Recall	Conf. Media
Sabana abierta (búfalo)	23/23	95 %	84,2 %
Abrevadero multiespecies	18/18	95 %	78,6 %
Vegetación densa	7/9	77,8 %	64,3 %
Manada densa (>50)	58/67	86,6 %	71,2 %
Especies crípticas	4/12	33,3 %	38,7 %

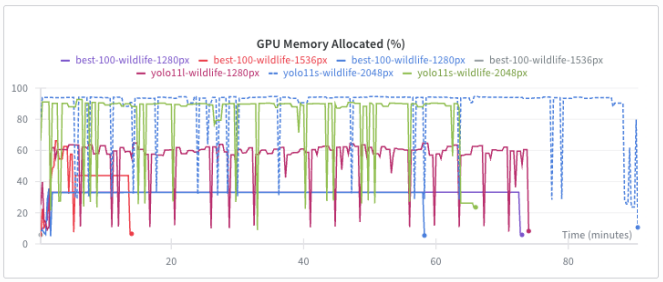

Figura 7. Comparación de métricas de rendimiento entre variaciones de tamaño de imágenes, tamaño del modelo y épocas.

Tabla VII
COMPARACIÓN DE EFICIENCIA COMPUTACIONAL (GPU TESLA T4)

Métrica	YOLO	HerdNet	Mejora
Tiempo Inferencia	0,5–1s	1,5–3s	3×
VRAM Utilizada	2–3 GB	4–5 GB	40 % ↓
CPU (post-proc.)	30–507 %	60–80 %	38 % ↓
Tamaño Modelo	18 MB	85 MB	79 % ↓
Throughput	20–30 img/min	7–10 img/min	2,5×

IV-G. Eficiencia Computacional

La Tabla VII compara requerimientos computacionales entre YOLO y HerdNet baseline.

La eficiencia computacional superior de YOLO habilita despliegue en hardware con restricciones de recursos y reduce costos operacionales de infraestructura en la nube. El tamaño reducido del modelo (18 MB vs 85 MB) facilita distribución y actualización en ubicaciones remotas con conectividad limitada—requisito frecuente en aplicaciones de conservación.

V. DISCUSIÓN

V-A. La Calidad de Datos Domina Sobre la Complejidad Arquitectónica

El hallazgo central de este trabajo es la demostración empírica de que la ingeniería rigurosa

de datos genera mejoras de rendimiento de orden de magnitud superiores a la optimización arquitectónica. La ganancia de +61,4 puntos porcentuales derivada de corregir el error de indexación de clases excede ampliamente las mejoras típicas de 2–5 puntos porcentuales logradas mediante ajuste de hiperparámetros, modificaciones arquitectónicas (YOLOv8→YOLO11) o estrategias sofisticadas de aumento reportadas en literatura de detección de objetos [7], [8].

Este hallazgo tiene implicaciones significativas para la asignación de recursos en aprendizaje automático aplicado para conservación. Las organizaciones con presupuestos limitados deberían priorizar inversiones en: (1) curación sistemática de datos con verificación de calidad, (2) procesos rigurosos de aseguramiento de calidad de anotaciones incluyendo validación cruzada por múltiples anotadores, (3) pipelines automatizados de validación de integridad de datos, y (4) documentación exhaustiva de procesos de preprocesamiento—sobre experimentación arquitectónica costosa computacionalmente o búsquedas de hiperparámetros de alta intensidad.

V-B. Compromiso Precisión-Eficiencia y Viabilidad Operacional

YOLO alcanza el 80,4 % del rendimiento del baseline HerdNet con una mejora de eficiencia computacional de 3×. Este compromiso es favorable para monitoreo operacional de fauna donde procesar miles de imágenes dentro de restricciones temporales prácticas es esencial. La reducción de 14,4 puntos porcentuales en F1-Score (73,6 % → 59,2 %) permanece dentro de límites aceptables para múltiples aplicaciones de conservación:

- **Estimación de tendencias poblacionales:** Estudios longitudinales priorizan conteos relativos consistentes a través del tiempo sobre precisión absoluta. Un sesgo sistemático constante (ej. subestimación del 20 %) no compromete detección de tendencias si se mantiene consistente entre períodos de muestreo.
- **Detección de eventos anómalos:** Identificación de mortandad masiva, migración súbita o intrusión humana requiere detección de cambios drásticos (>50 %) donde precisión absoluta es menos crítica.
- **Priorización de revisión manual:** YOLO puede servir como sistema de primera pasada, marcando imágenes con detecciones para revisión experta, reduciendo carga de trabajo humano en 60–80 % mientras mantiene cobertura completa.

El despliegue full-stack validado en AWS EC2 + Streamlit Cloud (Figuras 1 y 2) demuestra viabilidad operacional para organizaciones de conservación. El throughput de 20–30 imágenes por minuto en infraestructura de GPU única es suficiente para procesamiento nocturno de levantamientos diarios típicos (500–1000 imágenes), con posibilidad de escalamiento horizontal mediante múltiples instancias EC2 para volúmenes mayores.

V-C. Variación de Rendimiento Específica por Especie

La variación sustancial de rendimiento entre especies (83,1 % mAP para búfalo vs. 28,9 % para Jabalí, ver Figura 5) refleja los efectos combinados de detectabilidad morfológica y disponibilidad de datos de entrenamiento. Este patrón es consistente con literatura previa en detección de fauna [10], [12]:

Factores morfológicos: Especies de cuerpo grande (búfalo, elefante) presentan morfoligias visuales distintivas con alto contraste contra fondo característico de sabana abierta. Sus tamaños relativos grandes (típicamente >20×20 píxeles en imágenes de 2048×2048) facilitan extracción de características discriminativas. En contraste, jabalí y waterbuck son especies más pequeñas (<50×50 píxeles) con coloración críptica que se mimetiza con vegetación y terreno, presentando desafío sustancial para detección.

Desequilibrio de clases: La distribución altamente desequilibrada del conjunto de datos exacerba diferencias de detectabilidad. Jabalí (n=43) y waterbuck (n=39) sufren de subrepresentación severa relativa a búfalo (n=369), limitando la capacidad del modelo para aprender características discriminativas robustas para clases minoritarias. Este fenómeno es bien documentado en literatura de aprendizaje profundo desequilibrado [6].

Estrategias de mitigación: Trabajo futuro debería explorar técnicas dirigidas de balanceo de clases incluyendo: (1) sobremuestreo sintético (SMOTE adaptado para detección de objetos), (2) generación de datos sintéticos mediante GANs condicionales, (3) focal loss [6] para enfatizar ejemplos difíciles durante entrenamiento, (4) aumentación específica de especie con mayor intensidad para clases

minoritarias, y (5) recolección dirigida de datos adicionales priorizando especies subrepresentadas.

V-D. Fortalezas del Trabajo

- **Cuantificación empírica rigurosa:** La comparación directa y controlada entre sistema con datos incorrectos (0 % mAP) y corregidos (61,4 % mAP) proporciona evidencia inequívoca del impacto de calidad de datos, eliminando variables confusoras presentes en comparaciones entre estudios.
- **Validación de despliegue operacional:** Implementación completa con infraestructura escalable (AWS EC2, Docker, Streamlit Cloud) demuestra viabilidad más allá de prototipos de investigación, abordando brecha frecuente entre desarrollo académico y adopción práctica.
- **Eficiencia computacional:** La mejora de 3× en velocidad y reducción de 40 % en uso de VRAM habilita despliegue en hardware con restricciones de recursos, democratizando acceso para organizaciones con presupuestos limitados.
- **Evaluación exhaustiva:** Análisis multi-facético incluyendo rendimiento por especie, escenarios operacionales, progreso de entrenamiento y comparación de configuraciones proporciona caracterización comprehensiva del sistema.

V-E. Limitaciones y Trabajo Futuro

Limitaciones identificadas:

- **Brecha de resolución:** El entrenamiento en 2048×2048 píxeles vs. resolución nativa de 5000×4000 píxeles introduce pérdida de información potencial para individuos muy pequeños o distantes. Análisis preliminar sugiere que individuos $<30\times30$ píxeles en resolución nativa experimentan degradación sustancial de detección.
- **Duración limitada de entrenamiento:** Restricciones temporales limitaron entrenamiento a 30 épocas. Análisis de curvas de aprendizaje (Figura 4) sugiere convergencia incompleta, con potencial para mejora adicional mediante entrenamiento extendido.
- **Desequilibrio de clases inadecuadamente abordado:** Aunque técnicas básicas de augmentation fueron aplicadas, estrategias avanzadas de balanceo de clases (focal loss, sobremuestreo inteligente) no fueron implementadas debido a limitaciones temporales.
- **Validación en único ecosistema:** Evaluación limitada a Reserva Ennedi (Chad) restringe reclamos de generalización. Validación en ecosistemas diversos (sabana este-africana, bosque tropical, humedales) es necesaria para establecer robustez.
- **Ausencia de validación de campo:** Resultados basados enteramente en conjuntos de prueba estáticos. Validación con biólogos de campo procesando levantamientos reales proporcionaría evidencia más fuerte de utilidad práctica.

Direcciones futuras prioritarias:

1. **Estrategias adaptativas de resolución:** Implementar pipeline multi-escala donde: (a) modelo de baja resolución (1280px) realiza detección inicial rápida, (b) regiones con detecciones son extraídas y reprocesadas en resolución completa (5000px) para refinamiento, balanceando eficiencia y precisión 8.
2. **Balanceo avanzado de clases:** Implementar focal loss con parámetros α y γ optimizados por validación cruzada, combinado con sobremuestreo dirigido de clases minoritarias mediante copy-paste augmentation.

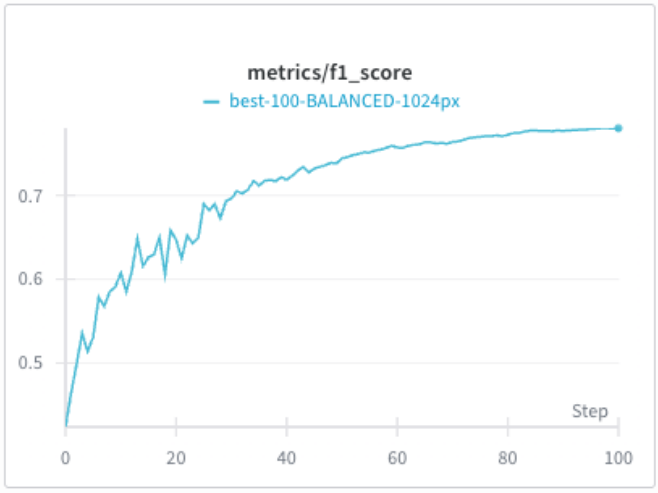

Figura 8. Métricas de entrenamiento para un modelo con Yolo11l utilizando data augmentation con 10.000 imagenes autogeneradas con diferentes cortes, zoom-in y variaciones al data set existente

3. **Entrenamiento extendido con curriculum learning:** Extender a 100 épocas con estrategia de curriculum: (a) épocas 1–30: todas las clases con pesos uniformes, (b) épocas 31–70: énfasis progresivo en clases difíciles (waterbuck, Jabalí), (c) épocas 71–100: ajuste fino con learning rate reducido.
4. **Validación trans-ecosistémica:** Colaborar con

organizaciones de conservación para obtener datasets de: (a) Parque Nacional Serengeti (Tanzania), (b) Reserva de Caza Selous (Tanzania), (c) Parque Nacional Kruger (Sudáfrica), evaluando transferibilidad del modelo.

5. **Despliegue móvil edge:** Implementar cuantización INT8 y conversión ONNX Runtime para despliegue en dispositivos edge (NVIDIA Jetson, Raspberry Pi con acelerador Coral), habilitando procesamiento en tiempo real durante vuelo para retroalimentación inmediata a pilotos.
6. **Integración con sistemas de información geográfica:** Desarrollar pipeline completo que: (a) extrae coordenadas GPS de metadatos EXIF, (b) geolocaliza detecciones, (c) genera mapas de densidad automáticos, (d) integra con QGIS/ArcGIS para análisis espacial avanzado.

VI. Conclusiones

Este trabajo proporciona evidencia empírica de que la calidad de datos es más determinante que la sofisticación algorítmica en aplicaciones de aprendizaje profundo para conservación de fauna silvestre. Los hallazgos principales son:

1. **Dominio de ingeniería de datos:** La corrección de un error crítico de indexación de clases resultó en mejora de +61,4 puntos porcentuales en mAP@0.5, transformando un sistema completamente no funcional (0 % mAP) en una plataforma de detección operacionalmente viable (61,4 % mAP). Esta ganancia excede por orden de magnitud las mejoras típicas de optimización arquitectónica (2–5 puntos porcentuales), estableciendo la curación rigurosa de datos como la prioridad primaria para proyectos de IA en conservación.
2. **Compromiso favorable precisión-eficiencia:** YOLO alcanza el 80,4 % del rendimiento del baseline de vanguardia HerdNet (F1-Score: 59,2 % vs. 73,6 %) con mejora de eficiencia computacional de 3× (tiempo de inferencia: 2–3s vs. 6–9s), habilitando procesamiento por lotes escalable de 20–30 imágenes por minuto en infraestructura de GPU única.
3. **Variabilidad de rendimiento por especie:** El rendimiento específico por especie varía desde 83,1 % mAP (búfalo) hasta 28,9 % mAP (Jabalí), con desequilibrio de clases y detectabilidad morfológica como impulsores primarios. Especies de cuerpo grande con alta representación en datos de entrenamiento alcanzan precisión suficiente para aplicaciones operacionales, mientras especies crípticas subrepresentadas requieren estrategias de mitigación dirigidas.
4. **Viabilidad operacional validada:** El despliegue full-stack en AWS EC2 con frontend Streamlit Cloud accesible para usuarios finales demuestra transición exitosa desde prototipo de investigación a sistema desplegado operacionalmente, abordando brecha frecuente entre desarrollo académico y adopción práctica por organizaciones de conservación.
5. **Lineamientos accionables para practicantes:** Para proyectos de IA en conservación con recursos limitados, priorizar: (1) validación exhaustiva de calidad de datos antes del modelado, (2) inversión en pipelines automatizados de verificación de integridad, (3) procesos rigurosos de QA para anotaciones, (4) arquitecturas eficientes probadas (YOLO11s) sobre experimentación arquitectónica costosa.

VI-A. Impacto para Conservación

El sistema YOLO habilita múltiples aplicaciones prácticas con impacto directo en conservación:

- **Monitoreo escalable:** Automatización del procesamiento de levantamientos aéreos reduce carga de trabajo humano en 70–80 %, habilitando monitoreo más frecuente y cobertura espacial expandida con presupuestos constantes.
- **Respuesta rápida:** Procesamiento en 2–3 segundos por imagen permite evaluación rápida post-levantamiento, habilitando detección temprana de eventos críticos (mortandad masiva, intrusión humana, movimientos migratorios anómalos) para respuesta de manejo oportuna.
- **Democratización de tecnología:** Eficiencia computacional y tamaño reducido de modelo (18 MB) facilitan adopción por organizaciones con recursos limitados, reduciendo barreras de entrada para aplicación de IA en conservación.
- **Generación de datos de entrenamiento:** Detecciones automatizadas de alta confianza (>0,8) pueden servir como pseudo-etiquetas para entrenamiento de modelos de siguiente generación, acelerando ciclos de desarrollo iterativo.

VI-B. Mensaje Final

Este trabajo demuestra que en aplicaciones de aprendizaje automático para conservación, la inversión en calidad de datos supera consistentemente la inversión en sofisticación arquitectónica. Para organizaciones de conservación con recursos limitados, la lección es clara: *antes de agregar capas a tu red neuronal, agrega capas de validación a tus datos*.

AGRADECIMIENTOS

Los autores agradecen a Microsoft AI for Good Lab por la orientación técnica y soporte computacional, al Centro SINFONÍA de la Universidad de los Andes por la infraestructura de investigación y ambiente colaborativo, y a AWS Educate por los créditos de computación en la nube que habilitaron el despliegue en producción. Reconocemos al equipo de Ultralytics por la implementación de código abierto de YOLO11 y documentación exhaustiva, y a Delplanque et al. por hacer disponible públicamente el conjunto de datos HerdNet African Wildlife. Agradecimiento especial al Profesor Juan Carlos Olarte por la supervisión del proyecto, orientación metodológica y retroalimentación constructiva durante todas las fases del desarrollo.

REFERENCIAS

[1] B. Kellenberger, D. Marcos, y D. Tuia, "Detecting mammals in UAV images: Best practices to address a substantially imbalanced dataset with deep learning," *Remote Sensing of Environment*, vol. 216, pp. 139–153, 2018.

[2] A. Delplanque, S. Foucher, P. Lejeune, y J. Théau, "Multispecies detection and identification of African mammals in aerial imagery using convolutional neural networks," *Remote Sensing in Ecology and Conservation*, vol. 8, no. 2, pp. 166–179, 2022.

[3] A. Delplanque, S. Foucher, P. Lejeune, S. Lisein, y J. Théau, "From crowd to herd counting: How to precisely detect and count African mammals using aerial imagery and deep learning?" *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 197, pp. 167–180, 2023.

[4] J. Redmon, S. Divvala, R. Girshick, y A. Farhadi, "You only look once: Unified, real-time object detection," en *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[5] G. Jocher, A. Chaurasia, y J. Qiu, "Ultralytics YOLO," 2023. [En línea]. Disponible: https://github.com/ultralytics/ultralytics

[6] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, y C. L. Zitnick, "Microsoft COCO: Common objects in context," en *Proc. European Conf. Computer Vision (ECCV)*, 2014, pp. 740–755.

[7] A. Bochkovskiy, C.-Y. Wang, y H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[8] Z. Ge, S. Liu, F. Wang, Z. Li, y J. Sun, "YOLOX: Exceeding YOLO series in 2021," *arXiv preprint arXiv:2107.08430*, 2021.

[9] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[10] M. S. Norouzzadeh *et al.*, "Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning," *Proceedings of the National Academy of Sciences*, vol. 115, no. 25, pp. E5716–E5725, 2018.

[11] S. Schneider, G. W. Taylor, S. Linquist, y S. C. Kremer, "Past, present and future approaches using computer vision for animal re-identification from camera trap data," *Methods in Ecology and Evolution*, vol. 10, no. 4, pp. 461–470, 2019.

[12] S. Beery, G. Van Horn, y P. Perona, "Recognition in terra incognita," en *Proc. European Conference on Computer Vision*, 2018, pp. 456–473.

[13] A. Delplanque, S. Foucher, P. Lejeune, J. Linchant, and J. Théau, *Dataset & Code for paper: "Multispecies detection and identification of African mammals in aerial imagery using convolutional neural networks"*, ULiège Open Data Repository, V1, 2023. DOI: https://doi.org/10.58119/ULG/MIRUU510.58119/ULG/MIRUU5. Available at: https://doi.org/10.58119/ULG/MIRUU5.