Detección y conteo automático de mamíferos africanos en imágenes aéreas mediante aprendizaje profundo: adaptación y evaluación del modelo HerdNet

Autores

Alejandro Aristizábal – a.aristizabals@uniandes.edu.co

Alexander Hernández – ja.hernandezp@uniandes.edu.co

Juan David Rico – jd.ricom1@uniandez.edu.co

Juan Felipe Jiménez – jf.jimnezc1@uniandes.edu.co

Resumen.

La expansión de las actividades humanas en África subsahariana ha intensificado los conflictos entre fauna silvestre y ganado, generando impactos ecológicos y sociales. Para apoyar estrategias de conservación, este estudio propone un modelo de aprendizaje profundo basado en HerdNet, orientado a detectar, contar y clasificar automáticamente animales en imágenes aéreas. Se utilizó un conjunto de datos públicos que incluye imágenes de seis especies, procesadas en parches de 512×512 píxeles. Las anotaciones originales en formato de bounding boxes fueron transformadas a coordenadas puntuales, optimizando así la detección en escenarios de alta densidad y oclusión.

Se exploraron cuatro configuraciones experimentales del modelo, aplicando estrategias de fine-tuning dirigidas sobre las capas profundas, uso de tasas de aprendizaje diferenciadas y el optimizador AdamW. El mejor desempeño se obtuvo en el Experimento 4, con un F1 Score de 72.62%, Recall de 84.8%, MAE de 0.97 y RMSE de 2.05, resultados que superan al modelo de referencia en precisión de conteo y estabilidad, aunque con menor precisión individual.

Los hallazgos sugieren que una adaptación cuidadosa de modelos preentrenados, combinada con estrategias de procesamiento de datos y evaluación automatizada (MLFlow), permite obtener resultados sólidos incluso con recursos limitados. Se concluye que este enfoque puede ser una herramienta valiosa para el monitoreo de biodiversidad, proponiendo líneas futuras como el entrenamiento multitarea, el uso de datos sintéticos y el despliegue en plataformas móviles de campo.

Palabras clave: *Visión por computadora*, *HerdNet*, *conservación de fauna*, *imágenes aéreas*, *detección de objetos/animales*

1. Introducción

La conservación de la biodiversidad en África subsahariana enfrenta desafíos significativos debido al crecimiento de las actividades humanas y la expansión territorial, lo que ha intensificado los conflictos entre la fauna silvestre y el ganado doméstico. Estos conflictos no sólo amenazan la supervivencia de especies emblemáticas, sino que también generan tensiones sociales y económicas en las comunidades locales que dependen de la ganadería para su sustento. Una gestión eficaz de estos conflictos requiere sistemas de monitoreo

precisos y eficientes que permitan identificar y cuantificar las poblaciones animales en tiempo real.

En este contexto, las imágenes aéreas ofrecen una herramienta valiosa para observar y analizar la distribución y migración de las manadas de animales en vastas áreas geográficas. Sin embargo, la interpretación manual de estas imágenes es laboriosa y propensa a errores, especialmente en entornos con alta densidad de animales, oclusiones y fondos complejos. La aplicación de técnicas de aprendizaje profundo, como las redes neuronales convolucionales, ha demostrado ser prometedora para automatizar la detección y clasificación de especies en imágenes aéreas, mejorando así la eficiencia y precisión del monitoreo de la biodiversidad.

El objetivo principal de este proyecto es desarrollar un modelo de aprendizaje profundo capaz de identificar y contar automáticamente animales en imágenes aéreas, clasificándolos por especie con una precisión comparable a la obtenida por modelos existentes como HerdNet. Para ello, se utilizará un conjunto de datos públicos que incluye imágenes aéreas de seis especies diferentes, distribuidas en conjuntos de entrenamiento, validación y prueba. El modelo se evaluará utilizando métricas estándar como Precisión, Recall, F1-Score, MAE y RMSE, y se implementará una aplicación para facilitar el cargue de las imágenes y la visualización de los resultados.

2. Estado del arte

La aplicación de inteligencia artificial (IA) en la conservación de la biodiversidad ha ganado relevancia en la última década, especialmente en el monitoreo de fauna silvestre mediante imágenes aéreas capturadas por drones. Este enfoque permite realizar censos de animales de manera más eficiente y menos invasiva que los métodos tradicionales. Inicialmente se identificaron los principales retos a los que se deben enfrentar este tipo de modelos:

- **Oclusión:** En la medida en que la densidad de la manada aumenta, los animales pueden cubrir parte de la superficie de otros individuos, lo cual puede limitar la capacidad de los modelos tradicionales de detección de objetos en la detección y conteo de animales.
- **Fondos complejos:** Los fondos de las imágenes aéreas suelen contener objetos confusos, que ocasionan que aumenten los números de falsos positivos y se pueden crear algunos sesgos indeseados.
- **Variación de la escala:** En las imágenes aéreas, el tamaño de los animales varía dependiendo de la distancia y la posición de la cámara, dificultando la detección y clasificación de animales de diferentes especies.
- **Distribuciones no balanceadas:** Las imágenes aéreas, usualmente están desbalanceadas en cuanto al número de animales que se identifican de cada especie, lo cual puede generar sesgos en los modelos, dado que se facilita la identificación de animales de la especie con mayor cantidad de individuos en los datos de entrenamiento.

Luego de identificar los principales retos, se revisan las contribuciones más significativas en este campo:

2.1 Modelos de detección de objetos en imágenes aéreas:

Otros estudios han explorado modelos de detección de objetos como Faster R-CNN, YOLO (You Only Look Once) y SSD para la identificación de animales en imágenes aéreas. Estos modelos han sido adaptados para abordar desafíos específicos del dominio, como la variación de escala y la complejidad del fondo. Sin embargo, enfrentan limitaciones en la detección de animales parcialmente ocluidos y en la clasificación precisa en entornos con alta densidad de objetos.

2.2 Avances en los modelos YOLO aplicados a la detección de fauna silvestre:

La serie de modelos YOLO ha sido ampliamente adoptada en tareas de detección de objetos en tiempo real debido a su eficiencia y precisión. Las versiones más recientes han incorporado mejoras significativas que las hacen especialmente adecuadas para la detección de fauna en imágenes aéreas.

YOLOv7 se destacó por establecer un nuevo estado del arte en detectores de objetos en tiempo real, superando a modelos anteriores en precisión y velocidad. Su arquitectura optimizada permite una detección más precisa de objetos en escenas complejas, como las encontradas en imágenes aéreas de fauna silvestre. Además han desarrollado variantes especializadas como WILD-YOLO, basada en YOLOv7, que mejora la detección de animales pequeños en imágenes aéreas mediante la adición de capas específicas y la optimización de la arquitectura para este propósito.

2.4 Modelos basados en Transformers:

Los modelos basados en Transformers han emergido como una alternativa prometedora a las arquitecturas tradicionales de redes convolucionales en tareas de visión por computadora. Su capacidad para capturar relaciones de largo alcance y contextos globales los hace especialmente útiles en escenarios complejos como la detección de fauna en imágenes aéreas.

En el contexto de imágenes aéreas, ViTDet (Visión Transformer Detector) ha sido evaluado en conjuntos como DOTA y RarePlanes, mostrando mejoras significativas en la detección de objetos con cajas delimitadoras horizontales, superando a sus contrapartes basadas en CNN en hasta un 17% en precisión promedio.

Otro modelo relevante es DFCformer, que introduce un enfoque de detección multi-escala utilizando Transformers. Este modelo aborda desafíos como la variación de escala y la confusión entre clases mediante la incorporación de mecanismos de atención adaptativos y módulos de fusión jerárquica, mejorando la detección de objetos pequeños y densamente agrupados en escenas aéreas.

Estos avances indican que los Transformers, al integrarse en arquitecturas de detección, pueden ofrecer mejoras sustanciales en la precisión y robustez de los modelos aplicados a la conservación de la biodiversidad mediante imágenes aéreas.

2.5 HerdNet: Detección y clasificación de animales en imágenes aéreas:

HerdNet es un modelo de red neuronal convolucional (CNN) diseñado específicamente para la detección y clasificación de animales en imágenes aéreas. Este modelo aborda problemas como oclusión y la variación de escala mediante el uso de puntos en lugar de cajas

delimitadoras para marcar los objetos de interés, lo que facilita el conteo de individuos en grupos de alta densidad. HerdNet ha demostrado un rendimiento notable en la identificación de especies y el conteo automático de animales, estableciendo un punto de referencia en este campo.

3. Metodología

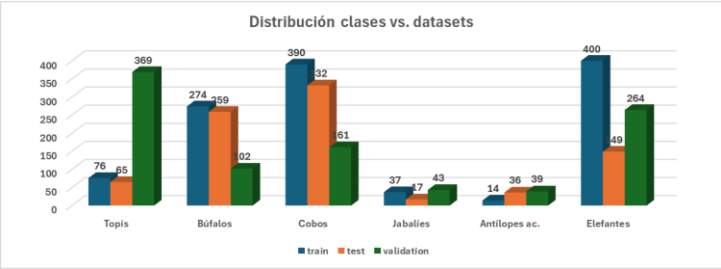
3.1 Abordaje del problema:

El objetivo principal del presente trabajo fue diseñar e implementar un sistema de detección de animales en imágenes aéreas mediante técnicas avanzadas de visión artificial, utilizando una arquitectura de red neuronal convolucional (CNN) especializada para detección y clasificación en contextos naturales. Para este propósito, se adoptó y se adaptó el modelo HerdNet, aplicando una serie de experimentaciones técnicas de ajuste con el fin de aumentar la precisión y robustez del sistema frente a las complejidades del entorno visual.

3.2 Datos utilizados:

El conjunto de datos empleado está compuesto por imágenes aéreas en alta resolución, segmentadas en tres subconjuntos: entrenamiento (157 imágenes), validación (111 imágenes) y prueba (121 imágenes). Los datos fueron tomados de la Universidad de Liege, disponibles en la siguiente fuente pública:
https://dataverse.uliege.be/file.xhtml?fileId=11098&version=1.0

En la siguiente gráfica podemos apreciar la distribución de las clases marcadas en las fotos tomadas de la fuente:



De acuerdo con esta distribución, las clases de elefantes, cobos y búfalos están ampliamente representadas en los tres conjuntos, mientras que jabalíes, antílopes acuáticos y topis representan una menor cantidad de ejemplos, particularmente en el conjunto de prueba. Esta distribución puede influir en el desempeño de los modelos, favoreciendo aquellas clases con mayor representación y dificultando la generalización en clases minoritarias. Cabe resaltar que las distribuciones presentadas, hacen referencia a la cantidad total de individuos de cada especie presentes en todas las imágenes.

3.3 Preprocesamiento:

El conjunto de imágenes fueron sometidas a un preprocesamiento previo al entrenamiento del modelo:

- *Generación y subdivisión de patches:*

Para mejorar la calidad de los datos imputados al modelo, dividimos las imágenes del entrenamiento, evaluación y validación en patches (subdivisiones de las imágenes). Este procedimiento se realizó para proporcionar a la máquina segmentaciones más detalladas, permitiendo un acercamiento a la imagen sin complicaciones en el procesamiento. Las imágenes se fraccionaron en *patches* de 512x512 píxeles, lo que resultó en un aumento en la cantidad de imágenes: 748 para entrenamiento, 600 para evaluación y 569 para validación.

- *Conversión de anotaciones:*

Los archivos de anotaciones (.csv) que contenían las posiciones de los animales en cada imagen fueron transformadas para representar cada objeto como un punto central en lugar de cajas delimitadoras (bounding boxes). Así, se calcularon las coordenadas x,y de cada objeto como el punto medio entre las máximas y mínimas que definían la ubicación de cada animal.

- *Aplicación de transformaciones de datos (Data Augmentation):*

Se aplicaron técnicas de aumento de datos al conjunto de entrenamiento utilizando la biblioteca albumentations. Esto incluyó voltear las imágenes vertical y horizontalmente con un 50% de probabilidad, realizar rotaciones aleatorias de 90 grados y ajustar brillo y contraste, además de aplicar desenfoque aleatorio y normalización de los valores de píxeles. Para los conjuntos de validación y prueba, solo se realizó la normalización de las imágenes.

- *Conversión de datos para el modelo:*

Se generaron anotaciones específicas para la detección de objetos mediante HerdNet, que incluyeron transformar las coordenadas de puntos en mapas de características, crear máscaras binarias a partir de estas coordenadas y reducir la resolución de las imágenes en validación y prueba para optimizar el uso de memoria y procesamiento.

- *Creación de datasets y dataloaders:*

Finalmente, se establecieron los tres conjuntos de datos y se definieron los Dataloaders correspondientes. Para los datos de entrenamiento se cargaron en batches de 2 imágenes con activación de shuffle, mientras que en los conjuntos de evaluación y validación se procesó una imagen por batch sin mezclar los datos. Con las imágenes segmentadas y procesadas, procedimos a realizar los entrenamientos correspondientes en nuestras experimentaciones.

3.4 Arquitectura del modelo:

La arquitectura principal utilizada es HerdNet con adiciones de fine-tuning en distintos experimentos que comprendieron unos ajustes con el propósito de mejorar la discriminación de especies.

Haciendo un recorrido en la arquitectura de HerdNet, este consiste en tres módulos principales:

- *Backbone:* Es un encoder basado en DLA-34 para extracción de características profundas.
- *Decoder:* Es un módulo que reconstruye las representaciones espaciales originales desde el espacio latente.

- *Heads:* Son cabezas separadas para detección y clasificación, implementadas con capas convolucionales 3x3 seguidas por activaciones ReLU y convoluciones 1x1.

En la fase de fine-tuning, se introdujo una capa personalizada adicional (Conv2D + BatchNorm + ReLU) posterior a la cabeza de clasificación, con el propósito de mejorar la discriminación de especies.

El primer experimento consistió en entrenar el modelo base con pesos pre entrenados. Se usó una combinación de *Focal Loss* y *CrossEntropy* Loss con pesos por clase para manejar el desbalance de clases.

El fine-tunning experimental que mejores resultados obtuvo consistió en un ajuste fino de las capas profundas del modelo - específicamente los bloques level4, level5 y la capa final fc en (del experimento 4 explicado más adelante) - mientras que el resto de los parámetros de la red, incluyendo las capas iniciales del backbone (level0 a level3) y el decoder DLA-34, fueron congelados para evitar su actualización durante el entrenamiento. A esto se sumóp la aplicación de tasas de aprendizaje diferenciadas por bloque de capas, el uso del optimizador AdamW, y un programador *scheduler* de tasa de aprendizaje tipo ReduceLROnPlateau, lo que permitió una mejor adaptación del modelo al dominio específico. El número de épocas varió entre 5 y 20 según el experimento, aunque se obtuvo como hallazgo el hecho que el modelo funciona lo más óptimamente posible con mínimo 10 épocas.

3.5 Experimentaciones

Commented [1]: Incluir ejemplo de una imagen de salida con cada experimento

- *Experimento 1:*

La primera experimentación constó en la inicialización del modelo con los pesos pre entrenados con dos configuraciones de pérdida: Uno de pérdida focal para manejo del desbalance en la detección de clases minoritarias, y un Cross Entropy con pesos ajustados y asignados a cada clase para mejorar el aprendizaje. Adicionalmente, se definió un optimizador Adam con una tasa de aprendizaje de 1e-4 y un *weight decay* de 1e-3 para prevenir el sobreajuste.

Para el procesamiento, se definieron 5 épocas de entrenamiento y se configuró el entrenador con su evaluador para cálculo de las métricas de rendimiento, siendo el f1 score la métrica principal.

Una vez entrenado el modelo, se cargaron los mejores pesos entrenados y se evaluó el modelo en el conjunto de prueba, generando el respectivo archivo .pkl del modelo entrenado y luego se configuró la integración con MLFlow para registro del modelo junto con sus métricas. Cada generación del modelo tuvo un tiempo estimado de procesamiento aproximado de una hora, y consumieron entre 10 a 20 unidades de cómputo.

- *Experimento 2: Ajuste fino superficial del modelo HerdNet:*

En este segundo modelo se buscó hacer un Fine-Tuning en base al modelo anterior manejando el mismo preprocesamiento de los datos por medio de la optimización únicamente de ciertas partes de la red.

Para este experimento se hicieron cambios clave en el Fine Tuning. Inicialmente, se congelaron todas las capas del modelo para evitar que el entrenamiento afecte los pesos de la parte más profunda de la red y solo se permitieron actualizaciones en las últimas capas del modelo, específicamente en las últimas capas convolucionales del backbone (level5) y en la capa encargada de la clasificación final fc.

Adicionalmente, se hizo una reducción de la Tasa de Aprendizaje a 1e-5, más bajo que el modelo anterior con el fin de evitar sobreajuste en las capas superiores, y se redujo la penalización weight decay de 5e-4 para mejorar la estabilidad del entrenamiento.

En cuanto al optimizador, se creó un optimizador Adam que afectaría únicamente las capas descongeladas, asegurando que el modelo no pierda el conocimiento previo y se mantuvo el mismo número de épocas.

- *Experimento 3: Fine-Tuning enfocado en la adaptación final:*

En esta tercera experimentación se congelaron todas las capas del modelo para preservar los pesos previamente entrenados y se descongelaron las capas superficiales, correspondientes a los niveles level 4 y 5, y la capa de clasificación fc con el fin de especializar las salidas del modelo para el nuevo dominio.

Con el objetivo de evitar el sobreajuste y permitir una adaptación controlada, se configuró una tasa de aprendizaje reducida (1e-5) y un parámetro de regularización *weight decay* de 5e-4. Además, se empleó una estrategia de aprendizaje diferencial con tasas distintas para cada bloque descongelado: 5e-6 para level4, 1e-5 para level5 y la capa fc. Esta segmentación permitió una refinación progresiva y controlada de los pesos.

Se utilizó el optimizador AdamW y se implementó un agendador de tipo ReduceLROnPlateau, que ajusta dinámicamente la tasa de aprendizaje si la métrica de validación F1 Score deja de mejorar. El modelo fue entrenado durante 10 épocas, evaluando el desempeño en cada iteración y seleccionando como checkpoint el modelo con mejor F1 Score sobre el set de validación.

- *Experimento 4: Entrenamiento enfocado en capas profundas: especialización controlada del modelo HerdNet.*

En esta cuarta experimentación, se tuvo como objetivo evaluar el desempeño del modelo HerdNet bajo condiciones controladas, replicando parcialmente el entorno propuesto en el artículo de referencia. Para ello, se aplicó un fine-tunning dirigido en las capas profundas del modelo, específicamente en level4, level5, y fc, mientras que el resto de los parámetros se mantuvieron congelados para preservar el conocimiento previamente adquirido.

Con el objetivo de optimizar la especialización de las capas descongeladas, se asignaron tasas de aprendizaje diferenciadas por grupo de capas, combinadas con el optimizador AdamW y un *scheduler* ReduceLROnPlateay, que ajustaba dinámicamente el aprendizaje en función del

rendimiento del F1 Score en validación. Las tasas utilizadas fueron: 5e-6 para el level4, 1e-5 para el level5, 5e-5 para la capa final fc y 2e-4 como *weight decay* global.

El entrenamiento se llevó a cabo durante 10 épocas, con validación al finalizar cada una y selección automática del mejor modelo con base en el F1 Score más alto obtenido, la cual, demostró una mejora significativa en la capacidad de generalización del modelo frente a otras versiones con fine-tuning, mostrando una mayor precisión en la detección de objetos en el conjunto de prueba.

3.6 Criterios de evaluación

Las métricas empleadas para validar el desempeño del modelo fueron las siguientes:

- *F1 Score:* Métrica principal que balancea precisión y exhaustividad.
- *Precisión y Recall:* Evaluadas individualmente para entender el tipo de error predominante.
- *MAE y RMSE:* Errores absolutos y cuadráticos de localización.
- *Accuracy:* Para clasificación por especie.

Los resultados fueron registrados por especie y de manera global (binaria) en cada experimento.

3.7 Entorno de implementación:

Todos los experimentos fueron realizados en Google Colab con acceso a GPU A100. Se emplearon las siguientes herramientas:

- *Pytorch:* Para el desarrollo del modelo.
- *Albumentations:* Para aumentos de datos.
- *MLFlow:* Como sistema de tracking y almacenamiento de los modelos.
- *down:* para carga de pesos pre entrenados.

Los modelos entrenados fueron almacenados en formato .pth y .pkl, y registrados en un servidor remoto de MLFlow para facilitar su despliegue posterior.

4. Resultados

Métricas	Experimento 1	Experimento 2	Experimento 3	Experimento 4
F1 Score	79,91%	62,91%	68,82%	72,62%
Recall	74,11%	84,31%	82,25%	84,80%
Precisión	86,69%	50,17%	59,16%	63,50%
Accuracy	93,91%	93,95%	94.87%	93,87%
MAE	0,5905	1,6098	1,3884	0.9736
RMSE	1,1280	4,683	2,6527	2,0580

5. Discusión

Los resultados que tomamos como referencia fueron obtenidos en una implementación previa de HerdNet:

F1-Score (%)	MAE	RMSE	AC (Confusión Promedio) (%)
83.5	1.9	3.6	7.8

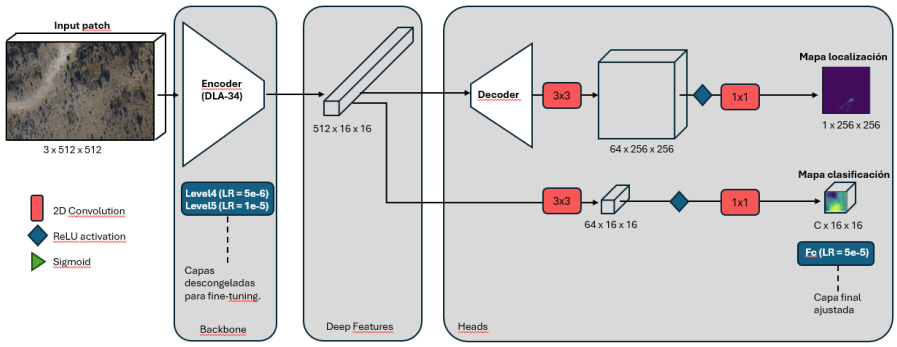
Los resultados obtenidos en las distintas experimentaciones permiten analizar en profundidad el comportamiento del modelo HerdNet y las implicaciones de aplicar técnicas de ajuste fino (fine-tuning) sobre diferentes niveles de su arquitectura. En general, el modelo demostró un desempeño prometedor en tareas de conteo y clasificación de animales en imágenes aéreas, particularmente en contextos de alta densidad y complejidad visual, sin embargo no se lograron superar las métricas del modelo con los hiperparámetros originales.

5.1 Rendimiento general

El *Experimento 1*, correspondiente al modelo base con pesos pre-entrenados, presentó el desempeño más equilibrado, alcanzando el F1 Score más alto (79.91%) y la mejor precisión (86.69%). Estos resultados reflejan la eficacia del modelo estándar al detectar y clasificar correctamente los animales, con un buen control de los falsos positivos y errores de conteo (MAE = 0.59, RMSE = 1.12).

Sin embargo, el *Experimento 4*, que aplicó un fine-tuning especializado únicamente en las capas profundas (level4, level5 y fc), logró también un alto F1 Score (72.62%) y una mejora significativa en el recall (84.80%), lo que sugiere una mayor capacidad para identificar verdaderos positivos. La precisión, aunque menor que en el experimento base, se mantuvo aceptable (63.50%). Esta estrategia permitió mejorar la generalización del modelo frente a variaciones en los datos, reduciendo además el error absoluto medio (MAE = 0.97), aunque con un ligero aumento del RMSE.

En la siguiente gráfica podemos ver la representación de los ajustes implementados en el *Experimento 4:*



5.2 Ventajas del enfoque

Una de las principales ventajas del modelo HerdNet es su capacidad de integrar detección y clasificación a partir de anotaciones puntuales, lo que facilita la adaptación a escenarios de alta densidad de objetos, como los rebaños en sabanas africanas. La estrategia de dividir las imágenes en parches, junto con el uso de aumentos de datos y mapas de calor como salidas del modelo, contribuyó significativamente a su capacidad de generalización.

Además, el uso de tasas de aprendizaje diferenciadas por bloque y schedulers dinámicos (como ReduceLROnPlateau) mostró ser una estrategia efectiva para mejorar la estabilidad del entrenamiento y permitir una mejor adaptación del modelo a las características del conjunto de datos.

5.3 Limitaciones observadas

Los retos iniciales se dieron principalmente en el procesamiento de las imágenes. Debido a su alta resolución y al peso de las mismas, no había sido posible procesarlas para ejecutar el pipeline de preprocesamiento, obligando al equipo a hacer metodologías de particionamiento. Esto no solo soluciona el problema de carga de data, sino que también se vio que las imágenes daban mejor detalle al modelo para facilitar su procesamiento y rendimiento, teniendo en cuenta que, ante el ojo humano, son imágenes donde no se logran ver o ubicar los animales de una manera fácil.

A pesar de los buenos resultados generales, se observaron algunas limitaciones. Por ejemplo, el Experimento 2, que solo ajustó las últimas capas del *modelo*, presentó una alta sensibilidad (recall = 83.62%) pero con una precisión muy baja (51.60%), lo que indica una tendencia del modelo a sobrepredecir objetos, generando un gran número de falsos positivos.

Así mismo, se mantiene como una limitación general de HerdNet la dificultad para estimar la ubicación precisa de los animales en escenarios con oclusión severa o cuando las especies comparten características visuales similares.

Otro reto significativo también consistió en el costo computacional para procesamiento del modelo en sus variaciones experimentales, en donde el grupo de trabajo ha tenido que hacer un gasto adicional para poder obtener acceso al consumo de unidades de cómputo de alto rendimiento para poder ejecutar estos modelos.

5.4 Comparación con trabajos previos

En nuestro caso, el Experimento 1 mostró el mejor desempeño general al aplicar el modelo HerdNet original sobre nuestros datos. Sin embargo, fue el Experimento 4 —con ajustes específicos en las capas profundas— el que obtuvo las mejores métricas y por tanto fue comparado con el modelo de referencia. En F1 Score, alcanzamos un 72.62%, frente al 83.5% de HerdNet, una diferencia explicable en parte por el uso de un subconjunto reducido de imágenes de entrenamiento debido a restricciones computacionales.

Al descomponer el F1 Score, nuestro modelo mostró un Recall superior (84.8% vs. 74.11%), pero una Precisión significativamente menor (63.5% vs. 86.69%). Esto indica que el modelo propuesto detecta más animales reales (menos falsos negativos), lo cual es deseable en

contextos de conservación, aunque a costa de un mayor número de falsos positivos. En contraste, el modelo de referencia es más conservador: menos detecciones, pero más precisas.

El MAE obtenido por nuestro modelo fue de 0.97, considerablemente menor al 1.9 del modelo de referencia. Esto sugiere una mayor exactitud promedio en el conteo, posiblemente gracias a técnicas como el fine-tuning localizado y el procesamiento por parches. Esta precisión es particularmente relevante cuando el conteo total impacta decisiones de manejo de fauna.

Asimismo, el RMSE de 2.05 (vs. 3.6 del modelo de referencia) refleja que nuestro modelo no solo comete menos errores promedio, sino que también evita errores extremos, lo que aporta estabilidad y confiabilidad en escenarios de monitoreo.

En resumen, aunque el F1 Score fue ligeramente inferior, las métricas de error muestran que el modelo propuesto cuenta con mayor precisión y consistencia en el conteo, posicionándolo como una herramienta efectiva para aplicaciones reales en conservación de biodiversidad.

6. Conclusiones y trabajo futuro

De acuerdo con el trabajo y experimentaciones hechas en este ejercicio, establecemos como uno de los hallazgos principales que el fine-tuning selectivo, aplicado sobre las capas más profundas del modelo HerdNet (específicamente level4, level5 y la capa clasificatoria fc), aporta mejoras notables en desempeño frente a otros enfoques más generales de ajuste. Esta estrategia, combinada con el uso de tasas de aprendizaje diferenciadas por grupo de capas y un *scheduler* adaptativo como ReduceLROnPlateay, permitió refinar el aprendizaje sin comprometer el conocimiento previo del modelo base. En particular, el *Experimento 4* logró los mejores resultados entre todas las variantes evaluadas, validando que una descongelación parcial bien dirigida puede mejorar la precisión y la capacidad de generalización del modelo.

Por otro lado, se observó que la estructura del dataset y el proceso de parcheo de imágenes grandes en secciones de 512x512 píxeles contribuyeron significativamente a la estabilidad computacional del cargue de los archivos para el entrenamiento del modelo, y a la capacidad del modelo para detectar puntos relevantes en imágenes complejas. Este flujo fue complementado con una evaluación mediante MLFlow, lo que permitió monitorear y comparar métricas clave como F1 Score, Accuracy, Precision, Recall, MAE y RMSE entre cada experimento.

En cuanto a las contribuciones del proyecto, se destaca el diseño y ejecución de cuatro variantes experimentales -incluyendo entrenamientos base y múltiples versiones de fine-tuning- , la implementación automatizada de todo el *pipeline* (desde el preprocesamiento hasta la evaluación final), así como la adaptación del código original de HerdNet para trabajar con nuevas imágenes y anotaciones específicas. También se desarrolló una estrategia reproducible de evaluación sobre validación controlada y se exportaron los modelos entrenados en formatos .pth y .pkl para uso posterior.

Como líneas de mejora o investigación futura, se propone ampliar la validación del modelo en imágenes provenientes de otras regiones o con especies animales distintas, para evaluar su robustez y capacidad de generalización. También se sugiere aplicar técnicas de aumento de datos específicas por clase o emplear métodos de generación sintética para contrarrestar el desbalance observado en algunas categorías. Otra dirección prometedora es el entrenamiento multitarea o el diseño de funciones de pérdida más completas que integren segmentación, detección y clasificación conjunta. Finalmente, se recomienda explorar backbones alternativos más livianos como MobileNet o EfficientNet para aplicaciones en tiempo real, y considerar el despliegue del modelo en escenarios de campo o mezclados con escenarios de ciudad, por ejemplo, mediante drones o estaciones remotas de monitoreo automatizado para la detección de fauna silvestre.

Referencias

- Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). *End-to-end object detection with transformers*. arXiv. https://doi.org/10.48550/arXiv.2005.12872
- Zhao, X., Shi, Y., Zhang, S., & Han, J. (2022). *DFCFormer: Detection transformer with feature and context fusion for remote sensing object detection. Drones*, 6(8), 188. https://doi.org/10.3390/drones6080188
- Jocher, G., Chaurasia, A., & Qiu, J. (2020). *YOLOv5 by Ultralytics*. GitHub repository. https://github.com/ultralytics/yolov5
- Meituan Applied Research. (2022). *YOLOv6: A single-stage object detection framework for industrial applications*. arXiv. https://arxiv.org/abs/2209.02976
- Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2023). *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*. arXiv. https://doi.org/10.48550/arXiv.2207.02696
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2019). *Focal loss for dense object detection. arXiv.*
- Kellenberger, B., Marcos, D., Lobry, S., & Tuia, D. (2019). *Half a percent of labels is enough: Efficient animal detection in UAV imagery using deep CNNs and active learning.arXiv.*

Anexos (opcional)

Material adicional relevante como fragmentos de código, configuraciones, detalles técnicos que no se incluyeron en el cuerpo principal del artículo.