UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

Clasificación y detección de seis mamíferos del ecuador utilizando la arquitectura de Deep Learning Yolov8

David Esteban Chamorro Enríquez

Ingeniería en Electrónica y Automatización

Trabajo de fin de carrera presentado como requisito
para la obtención del título de
Ingeniero en Electrónica

Quito, 15 de enero de 2024

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

HOJA DE CALIFICACIÓN
DE TRABAJO DE FIN DE CARRERA

Clasificación y detección de seis mamíferos del ecuador utilizando la arquitectura de Deep Learning Yolov8

David Esteban Chamorro Enríquez

Nombre del profesor, Título académico **Diego Benítez, Ingeniero en Electrónica**

Quito, 15 de enero de 2024

© DERECHOS DE AUTOR

Nombres y apellidos: David Esteban Chamorro Enríquez

Código: 00213784

Cédula de identidad: 1726588138

Lugar y fecha: Quito, 15 de enero de 2024

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en http://bit.ly/COPETheses.

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on http://bit.ly/COPETheses.

RESUMEN

El uso de modelos de Deep Learning para la automatización del monitoreo de animales cumple un rol fundamental para la preservación de las especies. El rendimiento de la red neuronal depende de la base de datos con la que se entrene y la configuración de los hiperparámetros. El proyecto se enfoca en el desarrollo de un clasificador que realice la detección de 6 mamíferos (Alouatta seniculus , Leopardus pardalis , Panthera onca , Puma concolor, Tayassu tajacu y Tapirus terrestre) utilizando la arquitectura de Yolov8. Se recopiló alrededor de 2000 imágenes por cada especie del portal digital iNaturalist y se generó 2 imágenes artificiales por cada original. Con el fin de lograr un modelo que compense velocidad de detección, precisión y carga computacional, se experimentó con las diferentes versiones de yolov8 y se varió el tamaño de batch. Los resultados con data augmentacion tiene una ventaja del 4.5% en precisión con relación a otros modelos. La prueba estadística k fold cross validation fue implementada para analizar el rendimiento general y particular de cada una de las especies.

Palabras clave: Yolov8, Deep Learning, preservación de especies, monitoreo de animales, clasificación de animales, detección de objetos.

ABSTRACT

The use of Deep Learning models for the automation of animal monitoring plays a fundamental role in preserving species. The neural network's performance depends on the database with which it is trained and the configuration of the hyperparameters. The project focuses on developing a classifier that detects six mammals (Alouatta seniculus , Leopardus pardalis , Panthera onca , Puma concolor, Tayassu tajacu and Tapirus terrestre), using the Yolov8 architecture. About 2000 images for each species were collected from the iNaturalist digital portal, and two artificial images were generated for each original. To achieve a model that maximizes trade-offs between detection speed, accuracy, and computational burden, we experimented with different versions of yolov8 and varied the batch size. The results with data augmentation have a 4.5% advantage in accuracy over other models. The k-fold cross-validation statistical test was implemented to analyze the overall and individual performance of each of the species.

Key words: Yolov8, Deep Learning, species preservation, animal monitoring, animal classification, object detection.

TABLA DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

La preservación de la biodiversidad del planeta es un problema que la sociedad ha enfrentado en las últimas décadas. Según la Unión Internacional para la Conservación de la Naturaleza (IUCN), desde inicios del siglo hasta el 2022, el porcentaje de especies en peligro ha crecido en un 400 % debido a actividades como el desarrollo residencial y comercial, la agricultura y acuacultura, la producción de energía y minería, entre otros [1]. Es decir, las acciones que han supuesto un avance para mejorar la calidad de vida de los humanos han puesto en riegos a la flora y fauna del planeta. Al ser la humanidad uno de los principales actores en el deterioro de la diversidad de especies, una fracción de la sociedad ha desarrollado un sentido de responsabilidad con la naturaleza, por lo que varios proyectos apoyados por el sector privado, gobiernos y/o ONGS han surgido para mejorar la supervivencia de la biodiversidad. Una de las iniciativas más conocidas son los objetivos de desarrollo sostenible 14 y 15 propuestos por la ONU en el 2015, donde se plantea detener la pérdida de la biodiversidad tanto terrestre como marítima al trabajar con el medio ambiente utilizando una base científica sólida [2]. Las diferentes propuestas suelen iniciar evaluando el estado y riesgo de las especies, luego se realiza un plan de acción y recuperación el cual es constantemente estudiado. A partir de estos resultados, se realizan recomendaciones sobre la mejor forma de reducir el riesgo, finalmente se implementa y se evalúa el plan ejecutado [3]. El monitoreo efectivo de especies es una de las soluciones más propicias que ayudan a reducir la amenaza de especies como consecuencia de la mejora en la automatización del proceso [4].

Uno de los países con mayor número de especies amenazados es Ecuador, alrededor de 2500 especies, entre plantas y animales, se encuentran en riesgo [5]. Ecuador, gracias a la disposición geológica, elevaciones naturales, cuerpos de agua e islas, cuenta con una gran

variedad de ecosistemas que facilitan el hábitat para muchas especies de animales y vegetales, es por esta razón que se lo considera como un país megadiverso [6]. La Constitución ecuatoriana en sus artículos 14 y 57 establecen que los ciudadanos y el Estado son responsables de la preservación del medio ambiente y conservación de ecosistemas y biodiversidad [7]. La política más importante implementada en Ecuador es la declaración de áreas, parques o reservas protegidas, como por ejemplo las Islas Galápagos y el Parque Nacional Yasuní, estos espacios son usados para monitorear y proteger a las especies ante las diferentes amenazas [8]. A pesar de los esfuerzos realizados, los resultados de los planes implantados no son alentadores, según el Plataforma Intergubernamental sobre Biodiversidad y servicios de los Ecosistemas (IPBES), el 65 % de los servicios de biodiversidad y ecosistemas (BES) se encuentran en declive, por lo que para mejorar el sistema de protección de especies es necesario optimizar el proceso de monitoreo con el trabajo en conjunto del Estado y la comunidad científica [9].

En la actualidad, la importancia de los datos para la toma de decisiones es trascendental. La gran mayoría de investigaciones no dejan nada al azar, sino que a partir de datos y mecanismos para tratar esta información se obtienen recomendaciones y posibles soluciones de cómo enfrentar un problema en específico. Por lo que las acciones para la conservación de especies deben contar con suficientes datos e información para lograr ser eficiente y mitigar los riesgos, sin conocer el contexto a partir de datos no se puede desarrollar políticas adecuadas [10]. Una de las alternativas más usadas para la recolección de datos de especies es el monitoreo de animales y plantas en su hábitat natural por medio de cámaras trampas. La preminencia de este sistema se debe a que no es invasivo con el ecosistema por lo que no supone un riesgo para la naturaleza [10-11]. Para que el monitoreo de especies no represente un gran costo económico, ni la demanda de tiempo para el personal humano sea excesiva es necesario automatizar el proceso de identificación, detección y

localización de los diferentes animales. La iniciativa que más ha ganado importancia son los algoritmos de visión computacional basados en modelos de Deep Learning, porque pueden realizar con mayor precisión, consistencia y repetitividad el análisis de imágenes, reduciendo así el tiempo de desarrollar estudios ecológicos y políticas de preservación [12].

Como hemos visto, el monitoreo de especies utilizando modelos de Deep Learning es una de las alternativas más viables para mitigar la amenaza de las especies. Al ser un tema tan emergente, se han desarrollado diverso algoritmos y arquitecturas de redes neuronales, cada una con sus ventajas y desventajas para diferentes aplicaciones. En nuestro caso, el objetivo es que el modelo realice la detección de varias clases de objetos, por lo que algoritmos como YOLO y sus diversas versiones, Redes Neuronales Convolucionales (CNN) y de Región (R-CNN) suelen ser las opciones más usadas. Una alternativa para el desarrollo de un modelo de Deep Learning que sea capaz de clasificar animales, es construir la red neuronal desde cero, es decir definir el tamaño, número de filtros o kernel, función de activación y el tipo de cada capa de la red, además se escoge un optimizador que favorezca el proceso de aprendizaje y ajuste de los pesos en cada iteración. Un estudio interesante del 2022 propone una red neuronal convolucional mejorada para clasificar 10 animales, donde la arquitectura del modelo implementa 4 etapas convolucionales de dos dimensiones, cada una con diferente tamaño y número de filtros y 2 capas densas de una dimensión. Se menciona que es preferible usar capas convolucionales de mayor tamaño en lugar de usar muchas capas de menor complejidad, especialmente en las primeras dos etapas de la red; porque al inicio, la red extrae las características locales y únicas de las imágenes. Para mejorar la eficiencia durante el entrenamiento del modelo se usó el optimizador Adam. A pesar de aumentar la precisión del modelo en comparación a una CNN clásica, los resultados del proceso de entrenamiento y validación no superan el 85 % y 70% respectivamente [13].

Para lograr una mayor precisión de las redes neuronales, se requiere tiempo, experiencia e implementar arquitecturas más complejas durante el proceso de diseño [14]; sin embargo, existen soluciones más sencillas como son el uso de los algoritmos “You only look onces” (YOLO). Con esta técnica se consigue aumentar la exactitud y velocidad en la detección de objetos en tiempo real [15], sin necesidad de elaborar la arquitectura del modelo, ya que el usuario solo debe enfocarse en armar la base de datos, realizar las anotaciones de los objetos a detectar y experimentar la mejor configuración de los hiper-parámetros de este. Para la creación de un modelo de detección y clasificación de 8 aves y mamíferos se analizó el rendimiento de tres diferentes algoritmos “Single-shot detection” (SSD), Faster R-CNN de dos etapas y YOLOv5, específicamente el modelo YOLOv5s. Al comparar las métricas de precisión, recall y media del promedio de precisión (mAP), el modelo que usó YOLOv5 mejora en un 3 y 15 % el parámetro de precisión y en un 1 y 4 % los valores de mAP en comparación a SDD y Faster R-CNN, no obstante, es superado en recall por la alternativa de Faster R-CNN. Estos resultados sugieren que YOLO favorece la detección de las especies, ya que consigue excelentes predicciones en situaciones complejas como problemas de iluminación, imágenes borrosas o abundante vegetación, pero los casos de falsos positivos aumentan ligeramente [16].

La primera versión de YOLO fue publicada en el 2015 por Ultralytics y durante los siguientes 8 años varias versiones de YOLO han sido desarrolladas con el fin de mejorar el rendimiento, tamaño, precisión y velocidad de los modelos. Para realizar una comparación cuantitativa de las últimas versiones de YOLO en un caso real, un grupo de investigación utilizó YOLOv5, YOLOv7 y YOLOv8 para el entrenamiento de un modelo que clasifique señales de tránsito y determinar en qué porcentaje mejora o empora las versiones del algoritmo. Para el experimento se utilizaron 4650 imágenes y la configuración de los hiper-parámetros fue de 50 épocas, 8 en tamaño de batch y el optimizador Adam. Los resultados

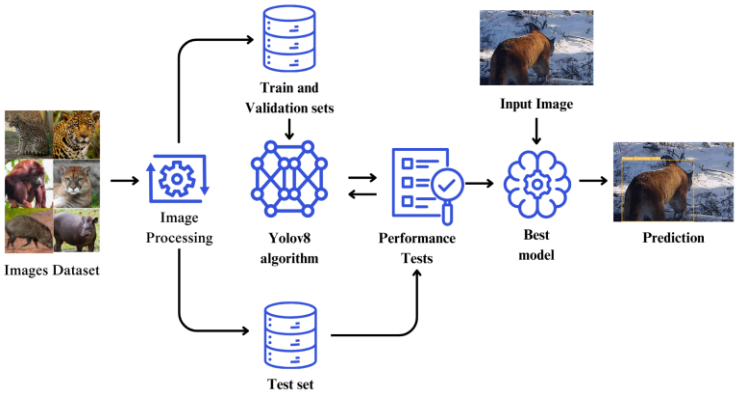
que obtuvieron indican que a pesar de que YOLOv7 consigue una precisión de 0.95, lo que es un 3% mayor a v8 y un 15% a v5, la versión más actualizada de YOLO supera significativamente a los otros modelos en términos de recall, mAP50 y mAP50-95, alcanzando un 0.941, 0.954 y 0.896 respectivamente, lo que es un 11%, 7% y 14% a su rival más próximo [17]. La mejora en estos parámetros indica que YOLOv8 cuenta con una mayor habilidad para detectar todas las instancias de las clases en la imagen y tiene un mejor rendimiento en modelos multiclase con un porcentaje de precisión alto en comparación a versiones anteriores [18]. YOLO-NAS tiene una gran capacidad para reducir la detección de falsos negativos pero su un índice de falsos positivos es elevado [19]. En este documento se va a explicar el desarrollo de un modelo de Deep Learning mediante el cual el algoritmo de YOLOv8 realice la detección de 6 especies endémicas del Ecuador.

DESARROLLO DEL TEMA

Materiales y metodología.

El objetivo de este proyecto es obtener un clasificador de 6 mamíferos del Amazonas ecuatoriano a partir del modelo de visión computación Yolov8; el cual, al recibir una imagen determine la probabilidad de a qué animal corresponde con relación a las 6 clases definidas y en caso de superar un umbral, dibuje una caja delimitando al objeto detectado junto con el nombre del mamífero y un valor de predicción. Los animales a seleccionar para el estudio son aquellos que han sido categorizados por la lista roja del UICN [1] o que el Ministerio del Ambiente del Ecuador los ha asignado con algún tipo de riesgos, que van desde vulnerable hasta en peligro de extinción, o aquellos que es indispensable preservarlos debido a que es único en la región. A partir de estos criterios, los mamíferos seleccionados son Alouatta seniculus (Mono Aullador Rojo), Leopardus pardalis (Ocelote), Panthera onca (Jaguar), Puma concolor (Puma), Tayassu tajacu (Pecari de Collar) y Tapirus terrestre (Tapir amazónico). En la figura 1 se muestra la estructura de la red neuronal.

Figura 1. Estructura de la red neuronal



A. Base de Datos

Para el proceso de entrenamiento, validación y pruebas del modelo es necesario recopilar un set de datos lo suficientemente robusto y extenso para que el algoritmo puede extraer las

características principales de los 6 mamíferos y compruebe si el aprendizaje es satisfactorio. La base de datos cuenta con un total de 11708 imágenes, se procuró que cada especie tenga un aproximado de 2000 imágenes. Inicialmente se pensaba utilizar 1200 imágenes por cada mamífero, pero los resultados de estos modelos no fueron lo esperado por lo que se optó por incrementar el número de datos. La gran mayoría de las imágenes se recopilaron del repositorio de datos científicos iNaturalist, el cual cuenta con millones de observaciones de diversas especies de animales y plantas [20]. Las imágenes que se encuentran pueden ser de cámaras trampas, fotografías profesionales o realizadas con teléfonos celulares, ya que cualquier persona, puede subir una observación a este portal web comunitario; por esta razón, es necesario filtrar la información e implementar protocolos de rigurosidad de la calidad de los datos [21]. La propia plataforma iNaturalist, al notar este conflicto en su base de datos, agregó la etiqueta de "Grado de Investigación" a aquellas imágenes que cuentan con el consenso de la comunidad sobre su identificación precisa, de esta forma la asociación de las observaciones con una especie debe ser calificadas por la comunidad científica. En la base de datos para el desarrollo del modelo, únicamente se usaron imágenes con la categoría de grado de investigación de iNaturalist.

Una vez construida la base de datos, se realizó un proceso de depuración, se buscó si existen datos repetidos y si la resolución de la imagen es superior o igual a 640x640 pixeles, ya que es el tamaño al que se va a escalar las imágenes, esto garantiza imágenes únicas y con buena definición. El algoritmo de YOLO requiere de la imagen y de su correspondiente anotación; es decir, un archivo de texto que incluya la información de los cuadros delimitadores que encierre a cada una de las especies a clasificar. Para familiarizarse con el set de datos, conocer el alcance del modelo y certificar la valía de las imágenes se realizó las anotaciones manualmente utilizando la herramienta CVAT; en donde, se buscaba encerrar con la mayor precisión posible a todos los animales que se encontraban a la vista y que

correspondan a una de las seis clases, sin importar si el cuerpo no estaba completo o se veían borrosos. Con el fin de mejorar el rendimiento del modelo se implementó la técnica de data augmentation, la cual incrementa el número de datos al generar artificialmente imágenes a partir de las originales [22]. Para automatizar el proceso de anotación del nuevo set de datos solo se realizaron traslaciones en el eje x y y, inversión horizontal y cambio de brillo aleatorios, por lo tanto, únicamente se produjo 2 imágenes por cada original, para evitar que las fotos conseguidas sean muy similares entre sí y el modelo se sobre ajuste al set de datos del entrenamiento. En los casos, en que los animales ya no eran visibles debido a las modificaciones realizadas en las imágenes, fueron eliminadas de la base de datos. Al final del procedimiento el número de imágenes creció a 34914 y cada especie contaba con alrededor de 5800 imágenes.

B. Modelo de Deep Learning Yolov8

En la primera mitad del 2023, YOLOv8 fue considerado como el estado del arte para la detección y seguimiento de objetos, segmentación de instancias, y clasificación de imágenes en tiempo real gracias a sus mejoras en precisión y velocidad [18]. La arquitectura del modelo se divide en tres: columna, cuello y encabezado. El encabezado implementa un modelo de “anchor-free split” en lugar de “anchor-boxes” para lograr resultados con mayor exactitud y eficiencia, ya que el sistema se encarga de predecir el centro del objeto en lugar de realizar un mapeo de cajas delimitadoras predefinidas para localizarlo, mejorando también el tiempo de respuesta [18-19]. El uso innovador de los módulos C2f, los cuales contienen bloques convolucionales, de división de canal y etapa cruzada parcial CSP, en la estructura de la columna y cuello representan una mejora en la detección y extracción de características de las imágenes [23]. Otro aspecto revolucionario de Yolov8 es el uso de mosaicos para el entrenamiento del modelo, técnica similar a data augmentation, pero en lugar de modificar las

características de la imagen, dispone de 4 imágenes en una sola en diferentes posiciones para generar nuevos casos de aprendizaje para el modelo. El algoritmo YOLO tiene 6 versiones pre entrenados con el set de datos COCO, cada uno cuenta con diferentes compensaciones entre número de parámetros, velocidad de respuesta, precisión y tiempo de entrenamiento.

C. Configuración del experimento

1) Set de datos para entrenamiento, validación y pruebas.

El primer paso en el desarrollo del modelo es dividir la base de datos recopilada estableciendo un porcentaje para la epata de entrenamiento, otro para la validación y el resto para pruebas. Cuando se tiene un gran número de imágenes por clase, se suele usar la proporción 70, 20 y 10 por ciento respectivamente, ya que con estos valores se tiene suficientes imágenes para el aprendizaje y ajuste del modelo y se puede determinar las métricas de precisión final [24]. Como en este caso no se tenía tanta diversidad de imágenes, se optó por priorizar la etapa de entrenamiento por encima de las pruebas. Los porcentajes que se usó fueron 75 % para entrenamiento, 20 % para validación y 5% para pruebas. Para garantizar que existan suficientes imágenes de cada mamífero, la asignación se realizó en relación con el total de datos para cada especie y no para el total de imágenes. En el experimento en donde se implementó data augmentation, se respetó los porcentajes señalados para que la forma en cómo se dividió el set de datos no influya en el rendimiento de los modelos y se pueda comparar únicamente el factor de aumentar imágenes artificialmente. En la siguiente tabla se muestran los resultados:

Tabla 1. División del conjunto de datos para cada animal.

Clases	Entrenamiento	Validación	Pruebas
Alouattta seniculus	1473	393	98
Leopardus pardalis	1641	438	109
Panthera onca	1496	398	100
Puma concolor	1488	397	99

Tayassu tajacu	1202	320	80
Tapirus terrestre	1482	395	99

2) Configuración del modelo

La configuración de los hiper parámetros es esencial para conseguir la mejor versión del modelo en términos de velocidad, precisión y carga computacional. Siguiendo con las recomendaciones del fabricador, en todos los casos, se escogió que las imágenes sean redimensionadas a 640x640 pixeles. Además, se seleccionó 100 épocas para el proceso de entrenamiento, con el fin que los diferentes parámetros tengan el suficiente número de iteraciones para converger a un valor. El optimizador fue configurado en modo automático para que el algoritmo sea el encargado de decidir que optimizador es el más adecuado en relación con el set de datos. En todos los entrenamientos se usó el Descenso de Gradiente Estocástico SGD con una tasa de aprendizaje de 0.01 y momentum de 0.9. Para conseguir el modelo con el mejor rendimiento posible se exploró las diferentes versiones de Yolov8 (nano, médium y extralarge), se varió el tamaño de batch y se comparó entre usar y no usar la técnica de data augmentation.

3) Experimentar con los modelos yolov8 preentrenados.

Al contar con seis alternativas de YOLOv8 para detección de objetos, la decisión de escoger uno de ellos dependerá del tipo de aplicación a realizar y que aspectos se quiere priorizar [22]. En el caso del clasificador de animales, es de interés contar con una precisión sobresaliente, reducir en lo posible el número de falsos positivos y que al ser multiclase el rendimiento del modelo sea consistente y acertado para todas sus clases, sin dejar de lado la importancia de la velocidad de predicción del sistema, ya que se espera que el algoritmo procese las imágenes en tiempo real. Para determinar que versión es la más adecuada para nuestro set de datos, se va a comparar los valores de precisión, recall, mAP50, mAP50-95,

tamaño del modelo y tiempo de detección. El tiempo de detección se refiere a la suma de los tiempos de preprocesamiento, inferencia, pérdidas y post procesamiento.

4) Ajuste del tamaño de batch.

El coste computacional que representa el desarrollo de modelos de Deep Learning es elevado debido al gigantesco número de operaciones que se realizan durante el entrenamiento de la red neuronal. Por lo tanto, es de interés optimizar la carga computacional de estos sistemas para facilitar la repetibilidad del proceso y reducir los tiempos de aprendizaje lo que va a permitir que se realicen un mayor número de experimentos para conseguir un modelo eficaz. Uno de los hiper parámetros que favorece la optimización de recursos computacionales es el tamaño de batch, el cual también puede llegar a influenciar en el rendimiento del entrenamiento especialmente en la función de pérdidas [25]. Para todos los procedimientos se va a usar la tarjeta gráfica RTX 3060ti, la cual cuenta con una VRAM de 8192 GB. Debido a esta limitación de memoria es esencial administrar lo mejor posible los recursos disponibles, por ello se va a probar con un batch de 4, 8, 12 y 16 para apreciar como impacta en las métricas del modelo, tiempo de entrenamiento y consumo de la memoria de la tarjeta gráfica. La versión de YOLOv8 que se va a usar, es el que sea escogido en el paso anterior.

5) Comparación del modelo con Data-Augmentatiom.

Por regla general, un mayor número de datos suele resultar en una mejora en el rendimiento de los modelos de Deep Learning, ya que la red neuronal cuenta con una gran diversidad de escenarios y casos de aprendizaje. Tener una base de datos tan extensa no suele ser viable en términos de almacenamiento ni es económicamente asequible; por lo tanto, la técnica de data augmentation surge como una alternativa cautivadora para extender artificialmente los sets de imágenes. Si se abusa de esta herramienta; es decir, se genera demasiadas nuevas fotos a partir de una original, el modelo va a sobre ajustarse al set de

datos de entrenamiento, por lo que las predicciones que realice no van a ser del todo precisas. Al generar únicamente 2 imágenes por cada una, se evita el problema de sobreajuste. Las curvas del modelo con data augmentation de precisión, recall, mAP50 y mAP50-95 en función de las épocas, van a ser comparadas con el modelo que tenga los mejores resultados utilizando metodología 1 y 2. Finalmente, el sistema que presente los valores más sobresalientes en precisión de detección global y por clases, recall y mAP, va a ser considerado como la mejor versión del clasificador de mamíferos.

6) K Fold Cross validation

Una vez se haya determinado el modelo con las mejores propiedades, es recomendable evaluar la efectividad del clasificador, para ello se usan técnicas estadísticas que generalicen las propiedades del sistema [26]. K fold cross validation es un proceso que divide en k veces diferentes el set de datos, los grupos de imágenes de entrenamiento y validación cambian para cada nuevo set de datos. Las imágenes que eran del grupo de validación pasan a ser de entrenamiento y cierta sección de entrenamiento pasa a ser de validación. Al generar estas nuevas organizaciones de los datos y entrenar el modelo con cada una de ellas, se consiguen los valores necesarios para estimar de forma global el rendimiento del modelo. En este caso se va a realizar 5 iteraciones del método estadístico y se va a realizar el promedio de los parámetros resultantes

Resultados y Discusión

En la tabla 2 se presentan los resultados de los modelos con yolov8n, yolov8m y yolov8x.Si se compara el modelo mediano con el nano, se puede ver una mejora del 4 % en el parámetro mAP50-95, lo que significa que esta versión realiza predicciones más precisas en casos difíciles, como puede ser, abundante vegetación o poca visibilidad, pero el tamaño de la red es 8 veces más grande y el tiempo de detección se cuadriplica. Con el modelo

yolov8x no se consigue mejoras que superen el 1% en ninguna de las métricas, además de que el tamaño y velocidad del modelo aumentan significativamente. Después de analizar los resultados, el modelo seleccionado es yolov8m ya que cuenta con un balance entre tiempo de respuesta y exactitud de predicción. La versión nano es muy veloz pero la detección no es lo suficientemente aceptable y el algoritmo extragrande es demasiado lento para la precisión que tiene.

Tabla 2. Métricas de los modelos preentrenados YOLOV8.

Modelo	**Yolov8n**	**Yolov8m**	**Yolov8x**
Precision	0.947	0.956	0.953
Recall	0.920	0.938	0.943
mAP@50	0.960	0.971	0.973
mAP@50-95	0.795	0.835	0.84
Tamaño [KB]	6.084	50.769	133.504
Tiempo de detección [ms]	2.8461	8.9954	25.9277

Los valores presentados en tabla 3 determinan que existe una relación inversa entre el tamaño de batch y el tiempo de entrenamiento es decir mientras más pequeño el valor de batch mayor va a ser el tiempo que le tome al modelo en entrenarse. Por otro lado, la relación entre batch y memoria de Vram es directamente proporcional, mientras más grande el uno mayor el otro. Considerando que la tarjeta gráfica utilizada es de 8192 Gb, la opción de 16 no es viable ya que, si se aumenta el número de imágenes, no se va a contar con la suficiente memoria. La opción de un batch de 4, a pesar de necesitar menos espacio, demanda de un mayor tiempo de entrenamiento, lo que va a prolongar el proceso de experimentación dificultando la optimización del modelo. Las últimas alternativas disponibles son batch de 8 y 12, la primera opción tiene una ventaja en términos de recursos computacionales ya que realiza el entrenamiento en prácticamente el mismo tiempo utilizando menos memoria de la tarjeta gráfica. Los resultados en términos de precisión, recall y mAP eran muy cercanos por lo que variar este hiper-parámetro no impacta en el rendimiento del modelo.

Tabla 3. Tiempo de entrenamiento y VRAM utilizados para cada batch.

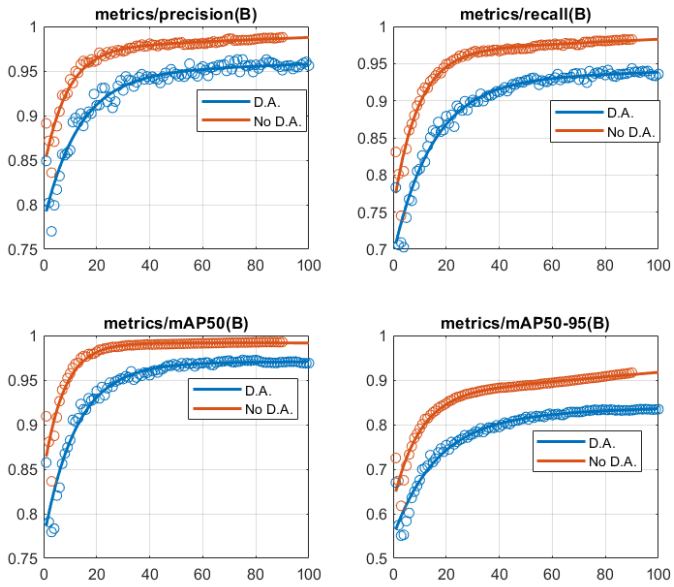
Tamaño de Batch	Tiempo de entrenamiento [h]	Memoria VRAM usada [GB]
4	6.75	2.56
8	5.98	4.1
12	5.97	5.9
16	5.42	7.5

La Figura 2 muestra la comparación de la variación de los parámetros de precisión, recall, mAP50 y mAP50-95 en función de la época en la que se encuentra el proceso de entrenamiento, entre el modelo con y sin técnica de data augmentation. Además, estas cuatro gráficas indican que el experimento que usa imágenes generadas artificialmente alcanza valores más altos en todas las métricas y las curvas llegan al estado estacionario en un menor número de épocas. Estas mejoras representan que el modelo con data augmentation tiene mayor precisión a la hora de la detección, sobre todo en los casos complejos, mitiga el problema de falsos positivos y negativos, requiere de un menor número de épocas permitiendo reducir los tiempos de entrenamiento y carga computacional. El coste de mejorar los valores de precisión, recall, mAP50 y mAP50-95 en un 3.1, 4.5, 2.2 y 8.2 (Table 4) por ciento respectivamente, es el tamaño de la red neuronal del nuevo sistema, el cual incrementa en un 400% y ahora se necesitan de 16 horas para el proceso de aprendizaje y validación. Al tener la misma configuración de hiper parámetros en ambos casos, la velocidad de detección no se ve afectada.

Tabla 4. Comparación de métricas entre caso con y sin data-augmentation.

Caso	Precision	Recall	mAP @ 50	mAP 50-95	Tamaño [KB]	Tiempo de detección [ms]	Tiempo de entrenamiento [h]
Sin Data-Aug	0.956	0.938	0.971	0.835	50 769	8.9954	6
Con Data-Aug	0.987	0.983	0.993	0.917	202 627	8.4814	16.2

Figura 2. Curvas de métricas entre modelo que usa y no usa data-augmentation.



El gráfico de precisión versus confianza y la Matriz de confusión o Tabla de vedad permiten realizar un análisis con mayor profundidad de la mejor versión del clasificador de los 6 mamíferos. La figura 3 señala cómo va a ser la precisión del modelo para diferentes umbrales de confianza tanto de cada uno de los 6 animales y de forma global. Estas curvas indican que la precisión del modelo es mayor al 80% para cualquier umbral mayor a 0.02 y que en el nivel de confianza de 0.953 todas las clases tiene una precisión del 100 %. El mono aullador y el pecarí de collar son las especies que tienen menor precisión para niveles de confianza bajos; en el caso del mono, se debe a que este mamífero suele estar en los árboles rodeado por abundante vegetación, en consecuencia, el modelo suele confundirse con las ramas (falso positivo) o por temas de visibilidad no realiza ninguna detección (falso negativo). En el caso del pecarí, la menor precisión puede estar relacionado con el número de imágenes recopilados para estas especies, ya que cuenta con menos datos en comparación al resto de animales. La figura 4, muestra un resumen del número de falsos positivos, falsos negativos y detecciones correctas para las 6 clases. En este gráfico se observa que el modelo

prácticamente no tiene problemas de clasificación entre las 6 especies, los inconvenientes se presentan cuando el modelo detecta como animal a la vegetación asignándole una clase y otros en que estando el animal no lo identifica.

Figura 3. Curva de precisión vs confianza del mejor modelo.

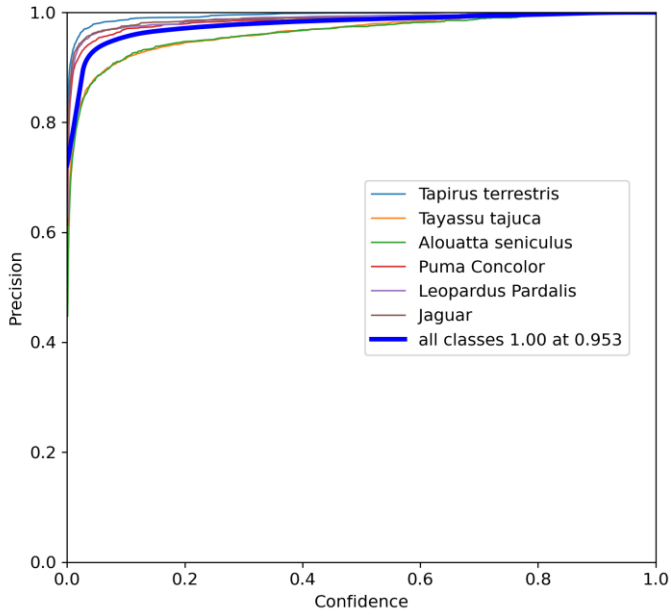
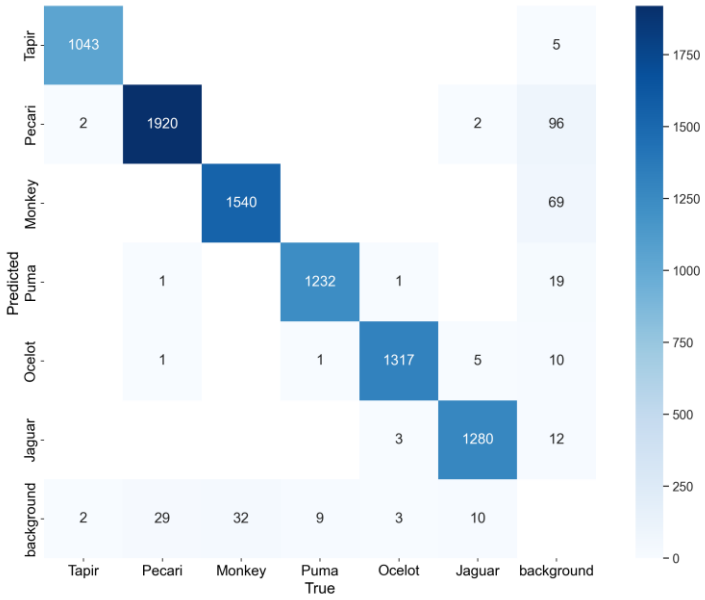


Figura 4. Matriz de confusión del mejor modelo.



En la tabla 5 se presentan los resultados del proceso de K flod cross validadtion. Cada parámetro corresponde al promedio de los valores resultante de los 5 entrenamientos con los

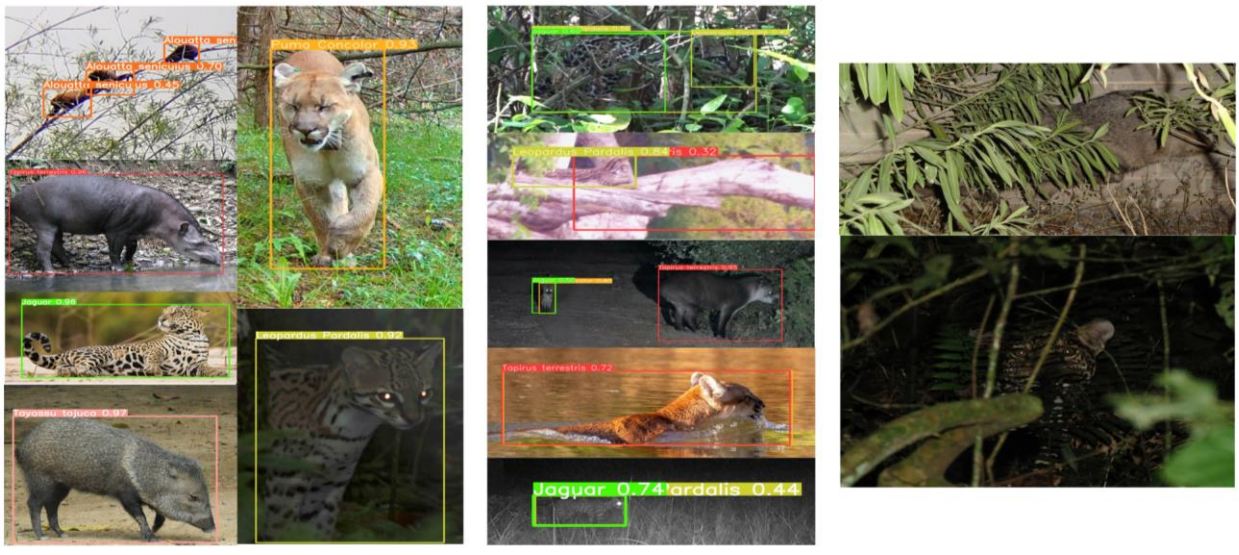
5 diferentes sets de datos. El valor de mAP@50-95 del mono aullador es de 0.8838 lo que confirma el problema de precisión del modelo para detectar correctamente a esta especie sobre todo en casos de mucha vegetación y poca visibilidad. Los demás parámetros de las diferentes especies indican la gran eficiencia que tiene el modelo para clasificar y detectar a estos animales.

Tabla 5. K-fold cross validation: generalización de las metricas del modelo.

Clases	Precision	Recall	mAP@50	mAP@50-95
Global	0.9888	0.9860	0.9936	0.9252
Tapirus terrestre	0.9966	0.9810	0.9950	0.9586
Tayassu tajacu	0.9780	0.9774	0.9924	0.9076
Alouattta seniculus	0.9772	0.9728	0.9916	0.8838
Puma concolor	0.9914	0.9866	0.9944	0.9428
Leopardus pardalis	0.9958	0.9948	0.9950	0.9218
Panthera onca	0.9950	0.9896	0.9950	0.9374

La figura 5 muestra los diferentes casos de predicciones: verdadera positivos, falsos positivos y falsos negativos; del modelo con las mejores prestaciones en términos de precisión, velocidad y tamaño.

Figura 5. Diferentes casos de predicción de modelos: (a) Verdaderos positivos, (b) Falsos positivos, (c) Falsos negativos



CONCLUSIONES

En este proyecto se determinó que el clasificador con mejor rendimiento en términos de precisión, velocidad y eficiencia computacional es el modelo que implementa data augmentation, usa el algoritmo yolov8m, tiene un batch size de 8, emplea el optimizador SGD con taza de aprendizaje de 0.01 y momentum de 0.9, redimensiona las imágenes de 640 pixeles y realiza el proceso de entrenamiento en 100 épocas. Para disminuir los casos falsos positivos se recomienda agregar al set de datos, imágenes de fondo, para que el modelo mejore la detección de las 6 clases. Si se quiere reducir el tiempo de respuesta del sistema y la carga computacional del clasificador, se puede explorar la alternativa de entrenar varios modelos de una sola clase y ensamblarlos, en lugar de desarrollar un modelo multiclase. Al ser un problema de solo una clase para cada modelo, surge la posibilidad de usar el algoritmo YOLO- NAS.

.

REFERENCIAS BIBLIOGRÁFICAS

[1] IUCN Red List, "The IUCN Red List of Threatened Species," IUCN Red List of Threatened Species, 2019. https://www.iucnredlist.org/resources/summary-statistics.

[2] M. Moran, "Bosques, desertificacion y diver- ´ sidad biologica," Desarrollo Sostenible, 2020. ´ https://www.un.org/sustainabledevelopment/es/biodiversity.

[3] M. F. Braby, "Threatened species conservation of invertebrates in Australia: an overview," Austral Entomology, vol. 57, no. 2, pp. 173–181, Jan. 2018, doi: https://doi.org/10.1111/aen.12324.

[4] N. M. Robinson et al., "How to ensure threatened species monitoring leads to threatened species conservation," Ecological Management & Restoration, vol. 19, no. 3, pp. 222–229, Aug. 2018, doi: https://doi.org/10.1111/emr.12335.

[5] H. M. Ortega-Andrade et al., "Red List assessment of amphibian species of Ecuador: A multidimensional approach for their conservation," PLOS ONE, vol. 16, no. 5, p. e0251027, May 2021, doi: https://doi.org/10.1371/journal.pone.0251027.

[6] R. Yang et al., "Cost-effective priorities for the expansion of global terrestrial protected areas: Setting post-2020 global and national targets," Science Advances, vol. 6, no. 37, p. eabc3436, Sep. 2020, doi: https://doi.org/10.1126/sciadv.abc3436.

[7] "CONSTITUCION DE LA REP ´ UBLICA DEL ECUADOR," ´ 2008. Available: https : //www.defensa.gob.ec/wp − content/uploads/downloads/2021/02/Constitucion − de − la − Republica − del − Ecuadoractene − 2021.pdf

[8] F. Cuesta et al., "Priority areas for biodiversity conservation in mainland Ecuador," Neotropical Biodiversity, vol. 3, no. 1, pp. 93–106, Jan. 2017, doi: https://doi.org/10.1080/23766808.2017.1295705.

[9] J. Kleemann et al., "Priorities of action and research for the protection of biodiversity and ecosystem services in continental Ecuador," Biological Conservation, vol. 265, p. 109404, Jan. 2022, doi: https://doi.org/10.1016/j.biocon.2021.109404.

[10] R. T. Buxton, S. Avery-Gomm, H.-Y. Lin, P. A. Smith, S. J. Cooke, and J. R. Bennett, "Half of resources in threatened species conservation plans are allocated to research and monitoring," Nature Communications, vol. 11, no. 1, Sep. 2020, doi: https://doi.org/10.1038/s41467-020-18486-6.

[11] [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, Jun. 2018, doi: https://doi.org/10.1073/pnas.1719367115

[12] B. G. Weinstein, “A computer vision for animal ecology,” Journal of Animal Ecology, vol. 87, no. 3, pp. 533–545, Nov. 2017, doi: https://doi.org/10.1111/1365-2656.12780.

[13] Z. Song, W. Gong, C. Li and T. T. Toe, ”Animals Image Classification Method Based on Improved Convolutional Neural Network,” 2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 2022, pp. 1997- 2001, doi: 10.1109/ITAIC54216.2022.9836676.

[14] P. K. Priya, T. Vaishnavi, N. Selvakumar, G. R. Kalyan and A. Reethika, ”An Enhanced Animal Species Classification and Prediction Engine using CNN,” 2023 2nd International Conference on Edge Computing and Applications (ICECAA), Namakkal, India, 2023, pp. 730-735, doi: 10.1109/ICECAA58104.2023.10212299.

[15] R. Thangaraj, C. J. J, S. M, R. S. K, S. Sasikumar and V. S, ”Automatic Detection and Classification of Wild Animal Species Using YOLO Models,” 2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN), Salem, India, 2023, pp. 315-322, doi: 10.1109/ICPCSN58827.2023.00058.

[16] D. Ma and J. Yang, ”YOLO-Animal: An efficient wildlife detection network based on improved YOLOv5,” 2022 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML), Xi’an, China, 2022, pp. 464-468, doi: 10.1109/ICICML57342.2022.10009855.

[17] F. N. Ortatas¸ and M. Kaya, ”Performance Evaluation of YOLOv5, YOLOv7, and YOLOv8 Models in Traffic Sign Detection,” 2023 8th International Conference on Computer Science and Engineering (UBMK), Burdur, Turkiye, 2023, pp. 151-156, doi: 10.1109/UBMK59864.2023.10286611.

[18] Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLO (Version 8.0.0) [Computer software]. https://github.com/ultralytics/ultralytics

[19] E. Casas, L. Ramos, E. Bendek and F. Rivas-Echeverr´ıa, ”Assessing the Effectiveness of YOLO Architectures for Smoke and Wildfire Detection,” in IEEE Access, vol. 11, pp. 96554-96583, 2023, doi: 10.1109/ACCESS.2023.3312217.

[20] iNaturalist. Observation Dataset. Available from https://www.inaturalist.org. Accessed [11/02/2023]

[21] P. Soroye et al., “The risks and rewards of community science for threatened species monitoring,” Conservation Science and Practice, vol. 4, no. 9, Aug. 2022, doi: https://doi.org/10.1111/csp2.12788.

[22] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” Journal of Big Data, vol. 6, no. 1, Jul. 2019, doi: https://doi.org/10.1186/s40537-019-0197-0.

[23] B. Carrillo-Perez, A. B. Rodriguez, S. Barnes and M. Stephan, ”Improving YOLOv8 with Scattering Transform and Attention for Maritime Awareness,” 2023

International Symposium on Image and Signal Processing and Analysis (ISPA), Rome, Italy, 2023, pp. 1-6, doi: 10.1109/ISPA58351.2023.10279352.

[24] M. -J. Zurita et al., ”Towards Automatic Animal Classification in Wildlife Environments for Native Species Monitoring in the Amazon,” 2023 IEEE Colombian Conference on Applications of Computational Intelligence (ColCACI), Bogota D.C., Colombia, 2023, pp. 1-6, doi: ´ 10.1109/ColCACI59285.2023.10226093.

[25] T. Takase, “Dynamic batch size tuning based on stopping criterion for neural network training,” Neurocomputing, vol. 429, pp. 1–11, Mar. 2021, doi: https://doi.org/10.1016/j.neucom.2020.11.054.

[26] T. -T. Wong and P. -Y. Yeh, ”Reliable Accuracy Estimates from k-Fold Cross Validation,” in IEEE Transactions on Knowledge and Data Engineering, vol. 32, no. 8, pp. 1586-1594, 1 Aug. 2020, doi: 10.1109/TKDE.2019.2912815.