



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

**TITULO DEL TRABAJO DE TITULACIÓN
SISTEMA DE IDENTIFICACIÓN DE AVES DE LA COMUNA DOS
MANGAS USANDO INTELIGENCIA ARTIFICIAL**

AUTOR

Panchana Malavé, Jonathan Alejandro

EXAMEN COMPLEXIVO

**Previo a la obtención del grado académico en
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN**

TUTOR

Ing. Rosero Vásquez, Shendry Balmore, Ms. CC

Santa Elena, Ecuador


Año 2025




UPSE

**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

TRIBUNAL DE SUSTENTACIÓN



Ing. José Sánchez A. Msc.
DIRECTOR DE LA CARRERA



Ing. Shendry Rosero V. Msc.
TUTOR



Ing. Jaime Orozco I. Mgt
DOCENTE ESPECIALISTA



Ing. Majorie Coronel S. Mgti.
DOCENTE GUÍA UIC



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por Panchana Malavé Jonathan Alejandro, como requerimiento para la obtención del título de Ingeniero en Tecnologías de la Información.

La Libertad, a los 18 días del mes de junio del año 2025

TUTOR

**Shendry
Rosero**

Firmado digitalmente por Shendry Rosero
DN: cn=Shendry Rosero, gn=Shendry
Rosero, c=ES, Spain, e=ES, Spain,
e=shendry.rosero.v@gmail.com
Motivo: Soy el autor de este documento
Ubicación:
Fecha: 2025-06-20 14:29:05:00

Ing. Shendry Balmore Rosero

Vásquez, Ms. CC



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

DECLARACIÓN DE RESPONSABILIDAD

Yo, **JONATHAN ALEJANDRO PANCHANA MALAVÉ**

DECLARO QUE:

El trabajo de Titulación, Sistema de identificación de aves de la Comuna Dos Mangas usando Inteligencia Artificial previo a la obtención del título en Ingeniero en Tecnologías de la Información, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

La Libertad, a los 18 días del mes de junio del año 2025

EL AUTOR

A handwritten signature in blue ink, which appears to read 'Jonathan Malavé', is written over a horizontal line.

Jonathan Alejandro Panchana Malavé

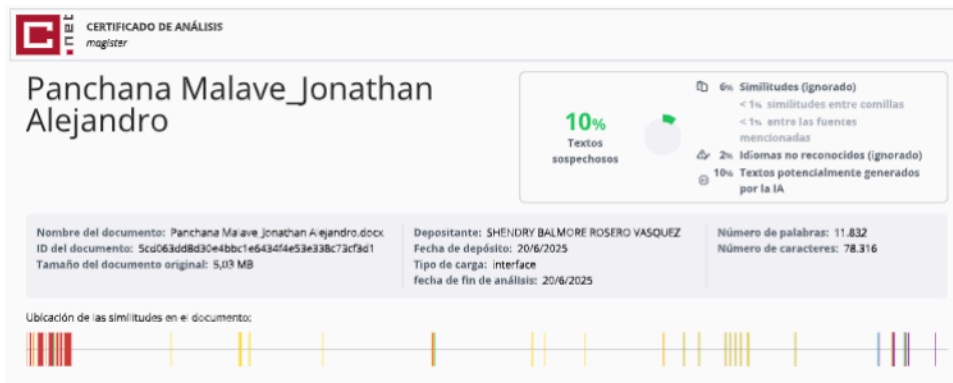


**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA**

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado Sistema de identificación de aves de la Comuna Dos Mangas usando Inteligencia Artificial, presentado por el estudiante, Jonathan Alejandro Panchana Malavé fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al 10%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.



TUTOR

Shendry Rosero

Firmado digitalmente por Shendry Rosero
DN: cn=Shendry Rosero, gn=Shendry Rosero, c=ES
Spain, e=ES_Spain_cn=shendry.rosero.v@gmail.com
Motivo: Soy el autor de este documento
Ubicación:
Fecha: 2025-06-20 14:29:05:00

Ing. Shendry Balmore Rosero Vásquez, Ms. CC



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

AUTORIZACIÓN

Yo, **JONATHAN ALEJANDRO PANCHANA MALAVÉ**

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales del presente trabajo de titulación con fines de difusión pública, además apruebo la reproducción dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

Santa Elena, a los 18 días del mes de junio del año 2025

EL AUTOR

A handwritten signature in blue ink, which appears to read "Jonathan Malavé", is written over a horizontal line.

Jonathan Alejandro Panchana

Malavé

AGRADECIMIENTO

Agradezco principalmente a mis padres, quienes me han acompañado a lo largo de mi vida y siempre me motivaron a que siga adelante académicamente, me han enseñado que con esfuerzo y dedicación puedo lograr lo que me proponga.

Total, agradecimiento a cada uno de los docentes de la Universidad Estatal Península de Santa Elena que impartieron sus conocimientos conmigo a lo largo de toda mi carrera universitaria.

Finalmente, también quiero agradecer a mis amigos, las personas con quienes he vivido un sinnúmero de anécdotas, en cada viaje, en cada fogata, en cada campamento, siempre es un gusto compartir a su lado.

Jonathan Alejandro, Panchana Malavé

DEDICATORIA

Este trabajo va dedicado a mis padres, Santa Dionicia Malavé Holguín y Moisés Alejandro Panchana Ortega, quienes lo han dado todo para educarme siempre de la mejor manera, quienes nunca han dudado en brindarme su apoyo incondicional en cada una de mis decisiones y que desde pequeño me enseñaron a nunca darme por vencido.

A mis hermanos, Geovanny Moisés Panchana Malavé quien siempre ha sido un ejemplo de hermano mayor y superación profesional, Karen Stephanny Panchana Malavé quien siendo la menor de la familia siempre ha demostrado ser una persona capaz de cumplir sus metas.

Jonathan Alejandro, Panchana Malavé

ÍNDICE GENERAL

TITULO DEL TRABAJO DE TITULACIÓN	I
TRIBUNAL DE SUSTENTACIÓN.....	II
CERTIFICACIÓN	III
DECLARACIÓN DE RESPONSABILIDAD	IV
DECLARO QUE:	IV
CERTIFICACIÓN DE ANTIPLAGIO	V
AUTORIZACIÓN.....	VI
AGRADECIMIENTO	VII
DEDICATORIA	VIII
ÍNDICE GENERAL	IX
ÍNDICE DE TABLAS	XII
ÍNDICE DE FIGURAS	XIII
RESUMEN	XV
ABSTRACT	XVI
INTRODUCCIÓN	1
CAPÍTULO I.....	2
1. FUNDAMENTACIÓN	2
1.1. Antecedentes	2
1.2. Descripción del Proyecto	4
1.3. Objetivos de Proyecto	6
1.3.1. Objetivo General.....	6
1.3.2. Objetivos Específicos	6
1.4. Justificación del Proyecto	7

1.5.	Alcance del Proyecto	8
CAPÍTULO II.....		9
2.	MARCO CONCEPTUAL, MARCO TEÓRICO Y METODOLOGÍA DEL PROYECTO	9
2.1.	Marco Conceptual	9
2.1.1	Machine Learning	9
2.1.1.1.	Tipos de Machine Learning	9
2.1.2.	Dataset	10
2.1.2.1.	El Dataset en Machine Learning.....	10
2.1.2.2.	Datos Sintéticos (Data Augmentation)	11
2.1.2.3.	Aumento de Imagen.....	11
2.1.3.	Python	13
2.1.4.	Streamlit.....	13
2.1.5.	Inteligencia Artificial (IA).....	13
2.1.6.	Redes Neuronales Convolucionales (CNN)	14
2.1.7.	ResNet 50.....	14
2.1.8.	¿Qué es LabelImg?	15
2.1.9.	Producto Mínimo Viable (MVP)	15
2.2.	Marco Teórico.....	15
2.3.	Metodología del Proyecto.....	17
2.3.1.	Metodología de la Investigación.....	17
2.3.2.	Técnicas e Instrumentos de Recolección de Datos.....	19
2.3.3.	Metodología de Desarrollo	20
CAPÍTULO III.....		22
3.	PROPUESTA.....	22
3.1.	COMPONENTES DE LA PROPUESTA	22

3.1.1.	Fase 1: Recopilación y Preparación de Datos.....	22
3.1.2.	Fase 2: Desarrollo del Algoritmo de Clasificación.....	25
3.1.3.	Fase 3: Desarrollo del Producto Mínimo Viable (MVP).....	28
3.1.4.	Fase 4: Pruebas y Validación del Sistema	30
	CONCLUSIONES.....	42
	RECOMENDACIONES.....	43
	ANEXOS.....	44
	Bibliografía	47

ÍNDICE DE TABLAS

Tabla 1: Prueba 1 imagen con buena calidad: Elaborado por el autor	31
Tabla 2: Prueba 2 imagen con mala calidad: Elaborado por el autor	32
Tabla 3: Prueba 3 varias aves de la misma especie en una foto: Elaborado por el autor	33
Tabla 4: Prueba 4 imagen con poco brillo: Elaborado por el autor	34
Tabla 5: Prueba 5 especie que no consta en el dataset: Elaborado por el autor.....	35
Tabla 6: Prueba 6 error en el formato del archivo: Elaborado por el autor	36
Tabla 7: Prueba 7 imagen desenfocada: Elaborado por el autor.....	37
Tabla 8: Prueba 8 imagen recortada: Elaborada por el autor.....	38
Tabla 9: Prueba 9 imagen de Tirano Tropical: Elaborado por el autor	39
Tabla 10: Prueba 10 imagen de Tangara Azuleja: Elaborado por el autor	40
Tabla 11: Resultados del entrenamiento del modelo: Elaborado por el autor	41

ÍNDICE DE FIGURAS

Ilustración 1: Transformación Geométrica y Fotométrica: Elaborado por [25]	12
Ilustración 2: Transformación Fotométrica: Elaborado por [25].....	12
Ilustración 3: Arquitectura del modelo ResNet50: Elaborado por [27].....	14
Ilustración 4: Organización del dataset inicial: Elaborado por el autor.....	22
Ilustración 5: Etiquetado de imagen en Labelimg: Elaborado por el autor	23
Ilustración 6: txt con la ubicación del objeto: Elaborado por el autor	23
Ilustración 7: Etiquetado guardado en formato txt: Elaborado por el autor	24
Ilustración 8: Instalación de la librería Albumentations: Elaborado por el autor	25
Ilustración 9: Función para leer y guardar nuevas etiquetas YOLO: Elaborado por el autor	25
Ilustración 10: Aumento de 5 imágenes por cada imagen original: Elaborado por el autor	25
Ilustración 11: Carga de dataset y aumento de datos: Elaborado por el autor.....	26
Ilustración 12: Entrenamiento y validación: Elaborado por el autor	26
Ilustración 13: Transfer learning: Elaborado por el autor.....	27
Ilustración 14: Callbacks y Entrenamiento: Elaborado por el autor	27
Ilustración 15: Número de épocas realizadas: Elaborado por el autor	28
Ilustración 16: Importación de librerías: Elaborado por el autor	28
Ilustración 17: Carga del modelo: Elaborado por el autor	29
Ilustración 18: Carga de diccionario de clases: Elaborado por el autor.....	29
Ilustración 19: Función para mejorar el nombre del resultado: Elaborado por el autor .	29
Ilustración 20: Subir la imagen del ave: Elaborado por el autor	30

Ilustración 21: Interfaz del sistema en streamlit: Elaborado por el autor	30
Ilustración 22: Prueba 1 imagen con buena calidad: Elaborado por el autor	31
Ilustración 23: Resultado obtenido de la prueba 2: Elaborado por el autor	32
Ilustración 24: Resultado obtenido de la prueba 3: Elaborado por el autor	33
Ilustración 25: Resultado obtenido de la prueba 4: Elaborado por el autor	34
Ilustración 26: Resultados obtenidos de la prueba 5: Elaborado por el autor.....	35
Ilustración 27: Resultados obtenidos de la prueba 6: Elaborado por el autor.....	36
Ilustración 28: Resultados obtenidos de la prueba 7: Elaborado por el autor.....	37
Ilustración 29: Resultados obtenidos de la prueba 8: Elaborado por el autor.....	38
Ilustración 30: Resultados obtenidos de la prueba 9: Elaborado por el autor.....	39
Ilustración 31: Resultados obtenidos de la prueba 10: Elaborado por el autor.....	40
Ilustración 32: Resultados del entrenamiento: Elaborado por el autor	41

RESUMEN

El siguiente proyecto tiene como objetivo desarrollar un sistema de identificación de aves usando Inteligencia Artificial, enfocado principalmente en especies de aves de la comuna Dos Mangas (Santa Elena, Ecuador), para esto se utilizó una red neuronal convolucional basada en ResNet50 la cual trabajó con un dataset robusto de 4830 imágenes, divididas en 23 clases, una por cada especie considerada para este proyecto, el mismo que inicialmente contaba con un total de 30 imágenes por especie, pero que posteriormente fue aumentado usando técnicas de aumento de datos, por otra parte la interfaz de usuario fue desarrollada usando la herramienta Streamlit, esta nos permite cargar imágenes desde el navegador y tener predicciones en tiempo real sobre la especie que se intenta identificar, por ultimo para dar constancia del óptimo funcionamiento del sistema se plantearon 10 escenarios diferentes en cuanto a las imágenes usadas, dando resultados positivos en todos los escenarios propuestos.

Palabras claves: ResNet50, Red Neuronal Convolucional, Aumento de Datos.

ABSTRACT

The following project aims to develop a bird identification system using Artificial Intelligence, focused mainly on bird species from the Dos Mangas commune (Santa Elena, Ecuador), for this a convolutional neural network based on ResNet50 was used which worked with a robust dataset of 4830 images, divided into 23 classes, one for each species considered for this project, which initially had a total of 30 images per species, but was later increased using data augmentation techniques, on the other hand the user interface was developed using the Streamlit tool, this allows us to load images from the browser and have real-time predictions about the species being identified, finally, to prove the optimal functioning of the system, 6 different scenarios were proposed in terms of the images used, giving positive results in all the proposed scenarios.

Keywords: ResNet50, Convolutional Neural, Data Augmentation.

INTRODUCCIÓN

La biodiversidad de la región costera del Ecuador representa un patrimonio natural de enorme valor ecológico, social y económico, en particular, la Comuna Dos Mangas, ubicada en la provincia de Santa Elena, se destaca por su abundante fauna silvestre, entre la cual se resalta gran variedad de aves, estas especies no solo embellecen el entorno natural, sino que también cumplen funciones vitales en los ecosistemas locales, como la polinización, el control de plagas y la dispersión de semillas, sin embargo, la correcta identificación de estas aves aún depende del conocimiento especializado, lo que limita el acceso a esta información por parte de la comunidad local y de los turistas visitantes de la zona.

Esta situación representa una barrera para promover prácticas de conservación participativa, educación ambiental y turismo sostenible, además, la falta de recursos didácticos accesibles dificulta la sensibilización y el aprendizaje sobre la riqueza natural de la región, por ello, es fundamental buscar alternativas que integren el conocimiento científico con herramientas accesibles y fáciles de usar por cualquier persona interesada, la tecnología en este sentido, puede ofrecer soluciones eficaces, inclusivas e innovadoras.

En este contexto, la siguiente investigación propone el diseño e implementación de un sistema de identificación automático de especies de aves mediante el uso de redes neuronales convolucionales (CNN), el modelo fue entrenado utilizando un dataset personalizado, conformado por imágenes reales de aves captadas en la Comuna Dos Mangas, las cuales fueron etiquetadas manualmente y aumentadas artificialmente para mejorar la precisión del algoritmo, para el entrenamiento se empleó la arquitectura ResNet50, se obtuvo un modelo capaz de predecir con alta exactitud la especie de un ave a partir de una imagen, además, se desarrolló una plataforma interactiva usando Streamlit, que permite a los usuarios cargar fotos y recibir predicciones en tiempo real, esto no solo facilita a la identificación de aves para expertos, sino también para estudiantes, investigadores y turistas, así, se espera que este proyecto contribuya a la conservación ambiental, al fortalecimiento del conocimiento y al impulso del turismo responsable basado en la biodiversidad

CAPÍTULO I

1. FUNDAMENTACIÓN

1.1. Antecedentes

En el año de 1990 el informático Yann LeCun y su equipo de trabajo dieron un paso muy grande en el análisis y procesamientos de datos visuales creando las denominadas Redes Neuronales Convolucionales (CNN), las mismas que han transformado el reconocimiento de imágenes, analizando las pequeñas regiones de una imagen logrando de esta manera una captura más precisa de las características espaciales y jerárquicas [1].

Dos Mangas es una comunidad ubicada en la Provincia de Santa Elena a siete kilómetros al noreste de Manglaralto, cuenta con dos senderos denominados Sendero Las Pozas y Sendero Las Cascadas, gracias a las características que posee este bosque es usado como nicho ecológico convirtiéndose en hábitat de varias especies nativas, siendo de nuestro interés principal su avifauna, con un total de alrededor de 84 especies de aves entre las cuales podemos destacar el Caracara Jacana, las Tortolitas, Periquito del Pacífico, Cuco Ardilla, Tangará, Pájaro Brujo, entre otros, lo que hace que esta comunidad sea ideal para el aviturismo [2].

La comuna Dos Mangas es parte de la Red de Turismo Comunitario del Litoral “Spondylus”, actualmente la comuna cuenta con un centro de interpretación turística bajo la gestión de una persona y 25 habitantes, pero solo 4 cuentan con credencial de guía nativo, avalada por el Ministerio de Turismo, cabe recalcar que no se ofertan de forma única los productos y servicios a precios determinados por la misma comunidad [3]. Lo cual refleja una evidente desorganización y falta de interés en los habitantes hacia el trabajo mancomunado, dicha desorganización incita a las personas a explorar los senderos por su cuenta, sin recibir la información importante sobre la flora y fauna visible en la zona.

Ahora, el desconocimiento que esto puede ocasionar a la persona turista nos lleva a nuestro problema principal, el cual es el impacto del humano hacia las aves del sector debido a la desinformación del entorno y la especie producido por la falta de organización en la comuna en cuanto al ecoturismo, ya que las aves son una de las especies de animales en libertad más cercanas a los humanos [7].

Este proyecto está motivado a comprender cómo la urbanización impacta los ecosistemas puede ser clave para que los planificadores urbanos ayuden a preservar y mejorar la biodiversidad en áreas locales. Asimismo, la manera en que los ciudadanos perciben la naturaleza en entornos urbanos influye considerablemente en las decisiones de gestión de la biodiversidad no solo a nivel regional y nacional, sino también a escala global [8].

Shirley Cheng y Julie Wang del Departamento de Ciencias de la computación de la Universidad de Stanford en su estudio denominado “Detección de especies de aves a través de sonidos” se enfocan en el monitoreo de aves mediante grabaciones de audio, explorando como las redes neuronales convolucionales (CNN) ayudan a clasificar los cantos de aves en grabaciones para posteriormente ser usados para la identificación [4].

Liber Adrián Hernández Abad de Ingeniería en Computación de la Universidad Nacional Autónoma de México, en su tesis llamada “PreBird: Biblioteca Open Source de Python para el Procesamiento de Audio, Orientado a Canto de Aves”, se centra en crear una IA la cual reconoce especies de aves mediante el canto, el proyecto incluye la grabación de cantos de aves, procesamiento de audios para eliminar ruidos, generación de espectrogramas y el entrenamiento de redes neuronales (CNN) para identificar especies [5].

Erick Joel Pilay José de la carrera de Biología de la Universidad Estatal Península de Santa Elena (UPSE), en su proyecto de tesis denominado “Diversidad y Abundancia de Aves en la Comuna Dos Mangas, Parroquia Manglaralto, Provincia de Santa Elena, Ecuador”, analiza la diversidad y abundancia de aves en la Comuna Dos Mangas mediante la metodología de conteos por puntos en tres estaciones [6].

Podemos concluir que el uso de Inteligencia Artificial (IA) para la identificación de aves en la Comuna Dos Mangas, mediante el análisis y procesamiento de imágenes es posible gracias a las Redes Neuronales Convolucionales (CNN), las cuales han sido pieza fundamental en el desarrollo de varios proyectos similares con el único propósito de contribuir en el conocimiento sobre la fauna en distintas

partes del mundo.

1.2.Descripción del Proyecto

El presente proyecto tiene como objetivo diseñar un sistema de identificación de aves mediante el uso de Redes Neuronales Convolucionales (CNN) en la comuna Dos Mangas, ubicada en la provincia de Santa Elena, Ecuador, aprovechando el potencial de las CNN en el análisis de imágenes, se busca desarrollar una herramienta que permita clasificar de forma precisa las especies de aves locales a partir de imágenes capturadas en la zona, este sistema facilitará el reconocimiento de la avifauna, apoyando actividades turísticas como el aviturismo y promoviendo la conservación de las especies nativas.

El sistema propuesto constará de un algoritmo de inteligencia artificial que, alimentado por un conjunto de datos de imágenes etiquetadas, será capaz de identificar automáticamente las especies de aves, se desarrollará una plataforma interactiva de fácil uso, que permitirá a los usuarios cargar imágenes para la clasificación automática de especies, este proyecto no solo contribuirá al conocimiento de la biodiversidad local, sino que también ofrecerá una herramienta innovadora para los guías turísticos y aficionados a la observación de aves. Para cumplir los objetivos de este proyecto se han formulado 4 fases las cuales están basadas en la metodología top-down, y son las siguientes [10]:

Fase 1: Recopilación y Preparación de Datos

- **Recopilación de imágenes en plataformas especializadas**

Las imágenes que se usarán para la elaboración de este proyecto serán obtenidas mediante el uso de plataformas en línea abiertas a todo público especializadas en avistamientos tanto de flora como de fauna, como por ejemplo iNaturalistEc, estas plataformas son usadas por personas de todo el mundo, profesionales y aficionados documentan sus aportes, logrando una variedad de especies y ángulos de cada una.

- **Selección de imágenes**

Se seleccionarán imágenes de buena resolución para así poder capturar detalles muy importantes de las mismas, como plumaje, forma del pico

y los colores correspondientes a cada especie para facilitar el etiquetado y asegurar la precisión del sistema, también se tomarán en cuenta variaciones en las imágenes, como el fondo y el tipo de iluminación, para entrenar al modelo en distintos escenarios.

- **Herramienta de etiquetado**

Una vez seleccionadas las imágenes se procederá al etiquetado usando el programa Labelimg, en este paso se debe procurar una alta precisión ya que el resultado final será un dataset, el mismo que nos brindará la efectividad del sistema.

Fase 2: Desarrollo del Algoritmo de Clasificación

- **Diseño del Modelo de Redes Neuronales Convolucionales (CNN)**

El diseño de un modelo CNN es nuestro siguiente punto de partida para la elaboración del sistema de identificación de aves, gracias a su habilidad de identificar patrones complejos en imágenes, para esta etapa se ha elegido la arquitectura ResNet-50 ya que este nos permite extraer características de alto nivel en las imágenes.

- **Entrenamiento del modelo CNN**

Usando las imágenes recopiladas e la fase anterior, el modelo se entrenará con el dataset para que aprenda a distinguir patrones entre cada especie, refiriéndonos a las plumas, forma del pico, colores, entre otras características importantes que hacen única a la especie, aquí se monitoreará el rendimiento del modelo y se evitará el sobreajuste.

Fase 3: Desarrollo del Producto Mínimo Viable (MVP)

- **Sistema de identificación de aves**

Teniendo el modelo entrenado con el dataset procedemos a integrar una plataforma interactiva de fácil uso la cual permita a los usuarios subir sus fotografías tomadas en sus excursiones, que gracias a su interfaz gráfica presentará los resultados obtenidos, identificando de manera precisa al ave requerida por el usuario.

- **Interfaz de usuario**

Esta plataforma le será de utilidad tanto a personas con formación

académica en el área como a aficionados, contará con una interfaz gráfica intuitiva y fácil de usar, usará botones con indicaciones claras de una carga simple de imágenes para facilitar la navegación y que el usuario se sienta cómodo usándola, mejorando su experiencia.

- **Pruebas en entornos controlados**

Para esta parte se contará con un conjunto de imágenes con alta resolución de aves locales obtenidas en línea, esto nos permitirá comprobar si el sistema responde de manera correcta a los detalles que tienen dichas imágenes encontrando de manera acertada la especie solicitada y que la interfaz funcione de manera correcta.

Fase 4: Pruebas y Validación del Sistema

- **Pruebas de campo**

Aquí es donde se evaluará la efectividad del sistema en un entorno real, se cargarán en el sistema imágenes de aves de la comuna Dos Mangas capturadas con un celular común, para determinar si el modelo puede identificar de manera correcta la especie, cabe recalcar que al ser tomadas con un celular común tendrá distintos escenarios en cuanto a la variación de luz, distancia y ángulo.

1.3.Objetivos de Proyecto

1.3.1. Objetivo General

Diseñar un Sistema de Identificación de aves por imágenes mediante Redes Neuronales Convolucionales (CNN) para promover la conservación de la avifauna local, en la Comuna Dos Mangas, Ecuador.

1.3.2. Objetivos Específicos

- Entrenar un algoritmo de Redes Neuronales Convolucionales (CNN) capaz de clasificar con precisión imágenes de aves capturadas en la comuna Dos Mangas, mejorando la identificación automática de especies locales.
- Etiquetar imágenes de especies de aves en la comuna Dos Mangas para crear un conjunto de datos (dataset) robusto, empleando Labeling para el

etiquetado, con el fin de entrenar el modelo de clasificación.

- Desarrollar un Producto Mínimo Viable (MVP) de una plataforma interactiva que permita la identificación automática de especies de aves en la comuna Dos Mangas, con base en el modelo de clasificación entrenado.

1.4. Justificación del Proyecto

La Inteligencia Artificial (IA) es un excelente aliado en la lucha de la conservación del ecosistema, una de sus principales funciones es monitorear la biodiversidad, mediante el uso de algoritmos de Machine Learning pueden identificar fauna y flora en imágenes de cámaras trampa o de satélites con gran precisión, estos nos proveen información valiosa sobre las especies, la misma que es útil para la conservación o el conocimiento en general [9].

Este proyecto se plantea con el fin de aprovechar los avances en las tecnologías para promover la conservación y el conocimiento sobre la avifauna local en la Comuna Dos Mangas, ubicada en la Provincia de Santa Elena, la cual tiene desafíos en la gestión del ecoturismo y preservación de sus especies, debido a su bajo nivel de guías calificados y asimismo la falta de herramientas de apoyo, como resultado final obtenemos una limitación en el potencial turístico y educativo que nos ofrece esta comunidad.

Entre las ventajas que nos ofrece la implementación del proyecto es la mejora de la experiencia en el ecoturismo en la comuna, ya que los turistas al poder identificar las especies de aves por sí mismos tendrán una experiencia más enriquecedora aumentando el interés en la biodiversidad, logrando un ecoturismo responsable, donde no solo se trate de disfrutar de los bellos paisajes que ofrece la naturaleza, sino que también se aprende sobre las especies endémicas.

Por otro lado, el proyecto no solo busca beneficiar a turistas o estudiantes interesados en la avifauna, sino que también a la comunidad local, lograr que la información se encuentre disponible en todo momento, los residentes podrán conocer más sobre las especies que habitan su entorno, lo cual fomenta el aprecio al patrimonio natural, y al ofrecer un sistema digital accesible automáticamente este proyecto se convierte en un recurso valioso para impulsar la formación de

futuros guías turísticos locales.

En resumen, el proyecto propone una solución tecnológica simple pero efectiva en la identificación de aves, y al mismo tiempo mejorará la experiencia de los turistas fortaleciendo el conocimiento sobre la avifauna local incluso si en el momento no se cuenta con guías especializados, y a su vez promueve un ecoturismo responsable en preservar tanto fauna como flora de la comuna.

1.5. Alcance del Proyecto

El proyecto se centrará en la identificación de especies de aves presentes en Dos Mangas, según lo descrito en la investigación "Diversidad y abundancia de aves en la comuna Dos Mangas" de Erick Joel Pilay José [6]. Las especies seleccionadas serán aquellas de relevancia ecológica y turística en la región, el alcance del sistema se limitará a:

- Especies de aves seleccionadas: Se trabajará con un conjunto específico de aves identificadas en el estudio de Pilay José, lo que garantiza un enfoque preciso en la fauna local [6].
- Número de imágenes por especie: Se recopilarán aproximadamente 30 imágenes de cada especie desde diferentes ángulos para capturar características visuales distintivas.
- Fuentes de información visual: Las imágenes se obtendrán de diversas fuentes, incluyendo Google Imágenes e iNaturalistEc, asegurando una representación diversa de las especies.

El sistema estará orientado a proporcionar una identificación precisa para turistas y la comunidad, sin reemplazar el análisis detallado de expertos en ornitología, además, el alcance contempla el desarrollo de una interfaz de usuario intuitiva, que permita a personas sin conocimientos técnicos utilizar la herramienta de identificación de manera eficiente.

CAPÍTULO II

2. MARCO CONCEPTUAL, MARCO TEÓRICO Y METODOLOGÍA DEL PROYECTO

2.1. Marco Conceptual

2.1.1 Machine Learning

Es la ciencia que permite que los computadores tengan la capacidad de aprender mediante el uso de un conjunto de datos (Dataset), completamente diferente a programar paso a paso la solución de la situación a resolver como lo hace el enfoque de la programación convencional, en el caso de Machine Learning nos centramos en desarrollar algoritmos genéricos que tienen como propósito extraer patrones del conjunto de datos elegido [23].

Los algoritmos de Machine Learning sustancialmente no difieren uno del otro, ya que todos dependen directamente de un conjunto de datos previamente etiquetados, debido a esto bajo el mismo principio podemos tener un sistema de Machine Learning que clasifique números escritos a mano y por otro lado un sistema que identifica especies de aves, como es el ejemplo de este mismo proyecto [23].

2.1.1.1. Tipos de Machine Learning

Con el fin de poder realizar cualquier tipo de tarea en especifica debemos conocer sobre los diferentes tipos de Machine Learning:

- Aprendizaje supervisado: Este tipo de Machine Learning se aplica cuando cada dato o conjunto de datos de entrada tienen una etiqueta, como ejemplo podemos tener un conjunto de imágenes como dataset, una vez que el algoritmo entrene al modelo en base a ese dataset se obtendrá como resultado un modelo capaz de predecir la etiqueta correspondiente a una nueva imagen, es decir, aunque esta misma no esté dentro del conjunto de datos [23].
- Aprendizaje no supervisado: Este tipo de Machine Learning es todo lo opuesto al supervisado, en este caso contamos igualmente con un conjunto

de datos, pero con la diferencia de este no está etiquetado, el trabajo de estos algoritmos es encontrar patrones ocultos sin la necesidad de una intervención humana como si se hace en el tipo supervisado [24].

- **Aprendizaje semi – supervisado:** Este tipo de Machine Learning es una unión del tipo supervisado y no supervisado, su principal característica es que esta trata con un conjunto de datos en el cual una parte está etiquetada y otra sin etiquetar, dominando más la parte no etiquetada, en resumen, es una mezcla de métodos supervisados y no supervisados [23].
- **Aprendizaje por refuerzo:** Este tipo de Machine Learning tiene una visión más futuristas que los otros tipos mencionados anteriormente, esta está inspirada en los mecanismos naturales, es decir, su conjunto de datos proviene de un entorno real o simulado y este al realizar una acción dependiendo de los resultados obtenidos es recompensado o sancionado, tal como pasa en un entorno humano cotidiano, los algoritmos se denominan agentes y aprender estrategias las cuales se denominan políticas, las cuales están encargadas de maximizar las recompensas y minimizar las penalizaciones [23].

2.1.2. Dataset

Un dataset, o conjunto de datos, es una colección organizada y sistemática de información en la que los elementos están interrelacionados, puede representarse en diversos formatos, como tablas, archivos de texto, bases de datos o hojas de cálculo, y se emplea para capturar, almacenar, analizar y distribuir datos de manera eficiente, rápida y segura, los datos se estructuran en matrices compuestas por filas y columnas, permitiendo gestionar la información dentro de una base de datos de origen, la cual puede estar vinculada a otros conjuntos de datos relacionados [16].

2.1.2.1.El Dataset en Machine Learning

En Machine Learning el Dataset es un elemento fundamental, el mismo debe ser considerablemente robusto, ya que si contamos con un conjunto de información poco numeroso podemos obtener de resultado un modelo con aprendizaje poco efectivo, esto se debe a que cuando la información usada para el entrenamiento del

modelo es escasa este en vez de aprender de manera generalizada lo único que hace es memorizar los ejemplos en concreto, esto se denomina sobreajuste, y debemos evitarlo en nuestro modelo [23].

2.1.2.2.Datos Sintéticos (Data Augmentation)

El aumento de datos se refiere a métodos que tienen como finalidad brindar los datos que faltan para mejorar el análisis de nuestro conjunto de datos, esto se logra tras generar copias modificadas de nuestros datos ya existentes en nuestro dataset base, con el único fin de aportar variedad y mayor tamaño, en resumen, los datos sintéticos se pueden entender como la parte artificial de datos del mundo real potencialmente ausentes [25].

Debido al tiempo y esfuerzo requerido para hacer la recolección de datos necesarios manualmente, es decir uno por uno, nos puede resultar complicado y hasta cierto punto dependiendo de la cantidad de datos que necesitemos hasta imposible, justamente ahí es donde los datos sintéticos toman un papel importante dentro del Machine Learning proporcionando un medio eficaz para aumentar el tamaño y la variedad del conjunto de datos, el mismo que a su vez mejora la optimización y la generalización del modelo entrenado, de esta manera se reduce el sobreajuste dando como resultado un modelo sólido y estable [25].

2.1.2.3.Aumento de Imagen

Entre los tipos de datos que puede tener un dataset, uno de los más comunes son las imágenes, usados para diversas tareas relacionadas con la visión artificial, tareas que van desde la clasificación de imágenes, hasta la detección de objetos, podemos clasificar el aumento de imágenes en dos categorías, siendo estas Transformaciones Geométricas y Transformaciones fotométricas, estas tienen en común la manipulación de archivos de imagen, a continuación, una explicación de estas [25]:

- **Transformación Geométrica**

Esta primera categoría se refiere a las técnicas que modifican en el espacio y diseño de la imagen original de nuestro dataset, entre las modificaciones

más comunes usadas se encuentran el cambio de tamaño, aumento de zoom, o los cambios de orientación, haciendo que de esta manera por cada cambio aplicado a la imagen original se cree una nueva imagen [25].

- **Transformación fotométrica**

Esta categoría se centra en alterar los canales RGB (Rojo, Verde, Azul) de la imagen original de nuestro dataset, entre las modificaciones más comunes usadas se encuentran el ajuste de saturación y escalado de grises, a la igual manera que la primera categoría esta por cada modificación empleada en la imagen principal se crea una imagen nueva [25].

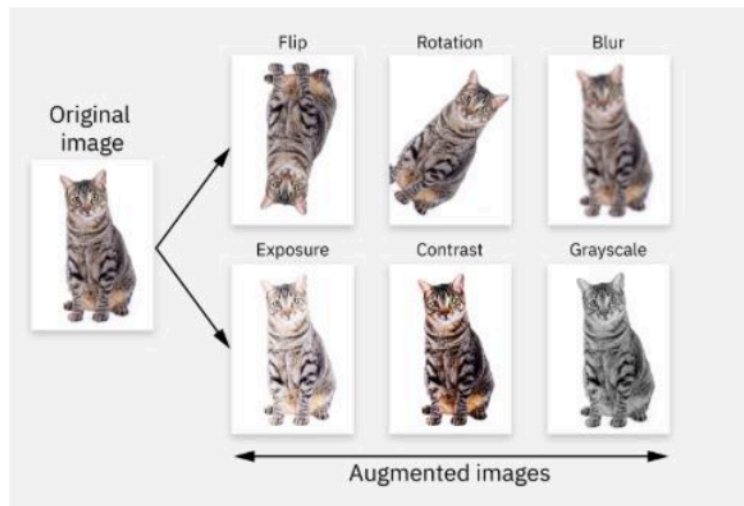


Ilustración 1: Transformación Geométrica y Fotométrica: Elaborado por [25]

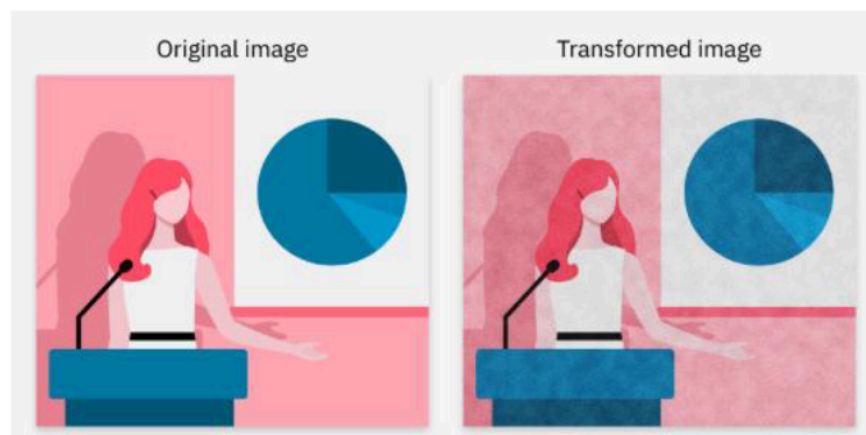


Ilustración 2: Transformación Fotométrica: Elaborado por [25]

2.1.3. Python

Python es un lenguaje interpretado de programación de alto nivel, creado a finales de los años 80 y a principios de los años 90 por el programador holandés Guido van Rossum cuando se encontraba trabajando en el sistema operativo Amoeba, la mayor fortaleza de de Python es su librería estándar, cubriendo las necesidades básicas de un programador e incluso mucho más, podemos afirmar que la librería de Python es una de las más completas en la actualidad, incluso comparable con la de Java y .NET [26].

Si hablamos de rendimiento, pese que es un lenguaje interpretado y que estos tienden a ser más lentos que los leguajes compilados, Python saca ventaja de otros ya que toda su librería estándar está implementada en el lenguaje C, logrando que sus funciones primitivas sean eficientes, teniendo como resultados procesos de interpretación rápidos y óptimos [26].

2.1.4. Streamlit

Streamlit es una biblioteca basada en Python, de código abierto para compartir aplicaciones web de machine learning y ciencia de datos, para que una aplicación tenga éxito es importante que su interfaz sea intuitiva, llamativa y fácil de usar, y de eso se encarga streamlit, permite a los desarrolladores crear interfaces llamativas para los usuarios sin la necesidad de poseer una gran experiencia en front-end [28].

2.1.5. Inteligencia Artificial (IA)

En 1956, se llevó a cabo un taller en Dartmouth en la que se reunió a diez de los investigadores más destacados en las áreas de teoría de autómatas, redes neuronales y estudios sobre inteligencia, durante este evento, que tuvo una duración de varios meses, se presentaron proyectos relacionados con aplicaciones particulares, juegos y 25 programas de razonamiento, los avances obtenidos no fueron especialmente significativos, lo más relevante fue la propuesta del término Inteligencia Artificial (IA), realizada por John McCarthy, quien es considerado por muchos como el padre de este campo de investigación, la Inteligencia Artificial se define como una rama de las ciencias computacionales que se centra

en desarrollar modelos capaces de realizar tareas propias de los seres humanos, basándose en dos características esenciales, el razonamiento y el comportamiento [14].

2.1.6. Redes Neuronales Convolucionales (CNN)

Las redes neuronales convolucionales (CNN) han ganado gran popularidad en el análisis de datos a gran escala, destacándose como herramientas prometedoras en diversos campos, especialmente en la clasificación de imágenes, debido a que estas redes simulan el funcionamiento de la visión humana, alcanzan altos niveles de precisión en tareas relacionadas con el reconocimiento visual, su desempeño ha llegado a ser comparable al humano en múltiples aplicaciones de visión artificial en el mundo real, como la clasificación de imágenes, la detección y reconocimiento de objetos, el diagnóstico médico mediante imágenes, la segmentación de escenas y el etiquetado, entre otras [13].

2.1.7. ResNet 50

Resnet 50 es un modelo de clasificación de imágenes, su principal característica es que se puede entrenar en grandes conjunto de datos, se divide en cuatro partes fundamentales, está las capas convolucionales, el bloque de identidad, el bloque convolucional y las capas completamente conectadas, las primeras capas convolucionales son las responsables de extraer características de las imágenes de entrada, las siguientes dos capas se encargan de procesar y transformar dichas características y por último las capas completamente conectadas se encargan de la clasificación final [27].

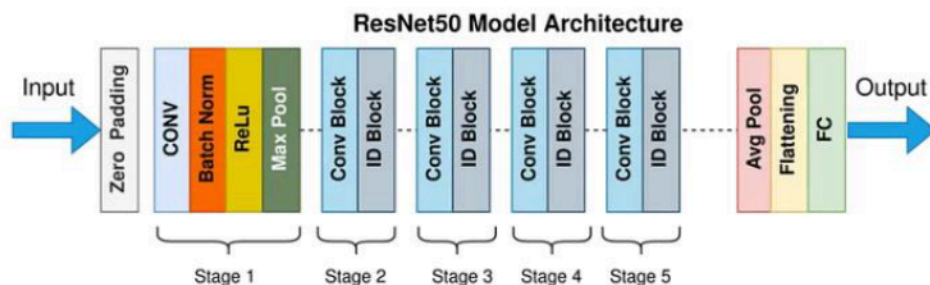


Ilustración 3: Arquitectura del modelo ResNet50: Elaborado por [27]

2.1.8. ¿Qué es LabelImg?

Es un programa de software gratuito y de código abierto diseñado para la anotación de imágenes mediante gráficos, lanzado en 2015 por Tzutalin, está escrito en Python y utiliza QT para su interfaz gráfica, esta herramienta es ligera y fácil de usar, ideal para etiquetar cuadros delimitadores alrededor de objetos en imágenes, para ello, seleccionar el software adecuado para la anotación de imágenes es importante para garantizar el éxito sostenido de las aplicaciones de visión por computadora [15].

2.1.9. Producto Mínimo Viable (MVP)

Un MVP (Producto Mínimo Viable) es un concepto introducido por Eric Ries en 2011, que se refiere a la versión más básica de un producto nuevo, diseñada con las funcionalidades esenciales, su objetivo es permitir que el equipo de desarrollo recopile información valiosa sobre la respuesta de los clientes, optimizando al máximo los recursos disponibles, como tiempo, dinero y esfuerzo [17].

2.2. Marco Teórico

Importancia de la IA en la Actualidad

En la actualidad la cantidad de datos generados por parte de los humanos es descomunal, superando por mucho la capacidad que tienen las personas para almacenar, interpretar y tomar decisiones complejas basadas en esos datos, es ahí donde aparece el término de Inteligencia Artificial (IA), la cual es denominada en la actualidad como la base de todos los procesos complejos en la toma de decisiones, la inteligencia artificial (IA) está emergiendo como una fuerza transformadora que está revolucionando nuestro entorno, cada vez más compañías y entidades adoptan esta tecnología para optimizar sus procesos y servicios, la IA está marcando una nueva era en la manera de gestionar los negocios, por lo que resulta esencial comprender el impacto que está teniendo en el mundo [18].

Aplicaciones de la Visión Computacional en la Conservación Ambiental

La visión computacional busca como objetivo principal replicar la manera en que vemos los humanos, usando imágenes capturadas por cámaras para posteriormente

interpretarlas y así poder reconocer objetos, cabe recalcar la que la visión computacional es diferente al procesamiento de imágenes, el primero busca mejorar imágenes para que sean comprendidas por personas, mientras que esta última las interpreta automáticamente, dentro de este campo existen varios enfoques, como por ejemplo los Métodos Empíricos Basados en Matemáticas propuesto por Rosenfeld, o los Modelos con Bloques Geométricos propuesto por Waltz y Mackworth, siendo estos aportes importantes en esta disciplina [20].

Se ha destacado por ser una herramienta más que valiosa en cuanto a la protección del medio ambiente, así también como en la gestión de emergencias naturales causadas por el ser humano, al día de hoy es posible poder detectar incendios en zonas forestales, también permiten detectar fenómenos como flujos de lava, emisión de gases y cenizas durante erupciones volcánicas, nos facilita el monitoreo el zonas boscosas, al recopilar datos visuales usando UAVs (Unmanned Aerial Vehicles), que en su traducción significa Vehículos Aéreos No Tripulados, optimizando el monitoreo ambiental y contribuyendo a la toma de decisiones informadas [21].

Importancia del Dataset en el Modelo de IA

El correcto funcionamiento de los modelos de aprendizaje automático en tareas de reconocimiento de imágenes depende del tamaño y de la calidad del dataset, al tener un conjunto de datos de mayor tamaño permite que el modelo aprenda las características de las clases, para de esta manera reducir los errores y evitar el sobreajuste, el preprocesamiento también juega un papel importante para la obtención de resultados precisos, al integrar procesos como la normalización de valores y la incorporación de transformaciones controladas ayudan a que el modelo sea más estable ante variaciones no deseadas, como la iluminación, ruido o cambios de fondo [22].

Importancia de la Biodiversidad y la Conservación de Aves

Las aves son fundamentales para el funcionamiento del ecosistema en el trapezio amazónico, ya que sus interacciones ecológicas contribuyen al mantenimiento del medioambiente y de servicios ecosistémicos esenciales para los humanos, como la

regeneración del bosque y la provisión de recursos, su importancia destaca la necesidad de integrarlas en la enseñanza de la Biología y la Educación Ambiental para promover la valoración y conservación de la biodiversidad, las culturas y los territorios, fomentando acciones para proteger la selva húmeda tropical cercana [19].

2.3. Metodología del Proyecto

2.3.1. Metodología de la Investigación

Contexto de la Investigación

El proyecto se desarrolla en el contexto de la comuna Dos Mangas, ubicada en la provincia de Santa Elena, Ecuador, esta región es conocida por su biodiversidad y su creciente interés en el ecoturismo, lo que la convierte en un lugar ideal para el estudio de aves, Dos Mangas alberga una amplia variedad de especies de aves que son de interés para turistas, investigadores y habitantes de la comunidad, y se ha identificado una necesidad de herramientas tecnológicas que faciliten la identificación de especies locales para fortalecer el turismo y la conservación[2].

La metodología se basa en la obtención y análisis de imágenes de aves, las cuales servirán para entrenar un modelo de inteligencia artificial, este modelo será capaz de identificar especies en función de características visuales, el contexto abarca aspectos ecológicos, tecnológicos y sociales, con el objetivo de promover la educación ambiental y el ecoturismo a través de un sistema de fácil manejo.

Alcance de la Investigación

El proyecto se centrará en la identificación de especies de aves presentes en Dos Mangas, según lo descrito en la investigación "Diversidad y abundancia de aves en la comuna Dos Mangas" de Erick Joel Pilay José [6]. Las especies seleccionadas serán aquellas de relevancia ecológica y turística en la región, el alcance del sistema se limitará a:

- Especies de aves seleccionadas: Se trabajará con un conjunto específico de aves identificadas en el estudio de Pilay José, lo que garantiza un enfoque preciso en la fauna local [6].

- Número de imágenes por especie: Se recopilarán aproximadamente 30 imágenes de cada especie desde diferentes ángulos para capturar características visuales distintivas.
- Fuentes de información visual: Las imágenes se obtendrán de diversas fuentes, incluyendo Google Imágenes e iNaturalistEc, asegurando una representación diversa de las especies.

El sistema estará orientado a proporcionar una identificación precisa para turistas y la comunidad, sin reemplazar el análisis detallado de expertos en ornitología, además, el alcance contempla el desarrollo de una interfaz de usuario intuitiva, que permita a personas sin conocimientos técnicos utilizar la herramienta de identificación de manera eficiente.

Enfoque de Investigación

El enfoque de la investigación es cuantitativo [11], orientado a medir el impacto del sistema de identificación de aves en la precisión de clasificación de especies locales. Este enfoque permite comparar las métricas de desempeño, como la precisión y la tasa de error, antes y después de implementar el sistema basado en Redes Neuronales Convolucionales (ResNet-50). Así, se evalúa cómo la incorporación de inteligencia artificial mejora la identificación de la avifauna respecto a métodos tradicionales, brindando una herramienta más confiable para su estudio y conservación.

Tipo de Investigación

La investigación es de tipo descriptivo aplicado[12], integrando elementos que permiten tanto analizar las características específicas de las especies de aves en Dos Mangas como desarrollar una herramienta tecnológica práctica, se busca documentar detalladamente las características visuales y morfológicas de las aves locales, utilizando imágenes y modelos de inteligencia artificial, este enfoque descriptivo facilita la identificación precisa de cada especie mediante patrones visuales específicos, tales como color, tamaño, forma del pico y alas, además, creando una base de datos útil para investigadores y la comunidad en general, este análisis es crucial para ajustar y optimizar el modelo de IA, ya que permite entender

las variaciones en las imágenes y mejorar la precisión del sistema en el contexto particular de Dos Mangas.

El componente aplicado de la investigación se manifiesta en el desarrollo de un sistema de identificación de aves que tendrá aplicaciones prácticas en la educación y el ecoturismo, esta herramienta será accesible por medio de un sistema web, permitiendo a los usuarios, incluso aquellos sin conocimientos previos en ornitología, identificar fácilmente las aves locales, al servir como un recurso educativo y turístico, el sistema fomenta la conservación de la biodiversidad y aporta valor al turismo ecológico de Dos Mangas, además, se proyecta como un recurso de apoyo para la educación ambiental, incentivando el conocimiento y la valoración de la fauna local en la comunidad.

2.3.2. Técnicas e Instrumentos de Recolección de Datos

Fuente de Información y Selección de Especies

La selección de la avifauna a identificar de la Comuna Dos Mangas se basa en la tesis Diversidad y Abundancia de aves en la Comuna Dos Mangas de Erick Joel Pilay José [6], este estudio nos ofrece un inventario completo de las aves que habitan en este entorno, detallando sus características y hábitos, asegurando que nuestro enfoque se centre en especies locales relevantes.

Recopilación de Imágenes

Es necesario poder contar con un conjunto robusto de imágenes por cada especie, en este caso se emplearán 30 imágenes por cada especie, de todos los ángulos posibles y en diversas condiciones de luz para garantizar que el modelo pueda reconocer características claves para identificar a las aves, como por ejemplo el color del plumaje, la forma del pico, su postura.

Cabe recalcar que en los senderos Las Cascadas y Las Pozas se han documentado 84 especies de aves, para este proyecto solo se usarán 23 especies documentadas en el estudio de Pilay [6]. Para la obtención de dichas imágenes se emplea el método de búsqueda en páginas especializadas en aves como iNaturalistEc, la cual es muy conocida por la calidad de sus aportaciones hechas por Biólogos.

Creación del Dataset

Una vez recolectadas las imágenes se procede a hacer el etiquetado usando el programa Labelimg, el cual nos permite añadir etiquetas precisas a cada imagen y así clasificarla por cada especie, en total serán 23 especies y por cada una de ellas 30 imágenes, este es un paso vital en nuestro sistema, ya que afecta directamente en la manera en que nuestro sistema identificará cada especie, las etiquetas se asignan en base al estudio de Pilay José [6], esto nos garantiza una consistencia entre nuestro dataset y las aves locales.

2.3.3. Metodología de Desarrollo

Para el desarrollo de un sistema de reconocimiento de aves en Dos Mangas, se ha adoptado una metodología basada en un enfoque top-down, adecuada para proyectos que requieren un diseño estructurado y secuencial, este enfoque permite desglosar el proyecto en fases específicas, avanzando de manera jerárquica desde los elementos generales hasta los componentes detallados, la finalidad de este proyecto es crear una herramienta de identificación automática de especies de aves, utilizando imágenes y modelos de inteligencia artificial (IA) que faciliten el reconocimiento de la fauna local, beneficiando tanto a la comunidad como a los turistas interesados en el ecoturismo y la educación ambiental.

Este sistema busca aportar un valor práctico, permitiendo que usuarios sin conocimientos especializados puedan reconocer especies de aves autóctonas mediante una plataforma intuitiva y de fácil acceso, en su diseño, la metodología considera tanto aspectos técnicos, como el desarrollo y ajuste del modelo de IA, como aspectos prácticos relacionados con la funcionalidad y usabilidad del sistema final, de esta forma, se busca no solo cumplir los objetivos científicos de clasificación de especies, sino también fomentar la conservación y apreciación de la biodiversidad en Dos Mangas.

La metodología se organiza en cuatro fases principales, que abarcan desde la recopilación de datos hasta la validación del sistema, garantizando así que cada etapa esté completamente desarrollada antes de pasar a la siguiente, esto asegura que el proyecto avance de manera eficiente y permite realizar ajustes o mejoras de

manera sistemática en cada fase.

Fase 1: Recopilación y Preparación de Datos

En esta etapa, se procederá a la recolección de imágenes de aves presentes en la comuna Dos Mangas, estas imágenes serán obtenidas a través de fuentes locales, como observadores de aves o guías turísticos, también se usarán páginas web de avistamientos de aves, posteriormente, se utilizará la herramienta Labelimg para etiquetar las especies de aves en las imágenes, esta etapa es crucial, ya que de la calidad y cantidad de los datos dependerá el rendimiento del modelo de clasificación.

Fase 2: Desarrollo del Algoritmo de Clasificación

En esta fase, se diseñará y entrenará un algoritmo de Redes Neuronales Convolucionales (CNN) para la clasificación de las imágenes recopiladas, el algoritmo será entrenado utilizando el dataset previamente etiquetado, y se aplicarán técnicas de ajuste de hiper parámetros y validación cruzada para mejorar la precisión del modelo, el objetivo es que el modelo sea capaz de identificar con precisión las distintas especies de aves locales.

Fase 3: Desarrollo del Producto Mínimo Viable (MVP)

Una vez el algoritmo esté entrenado y validado, se procederá al desarrollo del Producto Mínimo Viable (MVP), que consistirá en una plataforma interactiva para la identificación automática de aves, esta plataforma permitirá a los usuarios subir imágenes y obtener una clasificación automática de las especies de aves, se trabajará en una interfaz intuitiva para que tanto expertos como aficionados puedan utilizarla de manera eficiente.

Fase 4: Pruebas y Validación del Sistema

Esta etapa estará enfocada en la evaluación del rendimiento del sistema en un entorno real, se realizarán pruebas con imágenes capturadas en la comuna Dos Mangas para verificar la precisión y eficacia del modelo en identificar las especies, y realizar ajustes en el sistema según sea necesario dependiendo de los resultados obtenidos en las distintas pruebas de las diferentes especies.

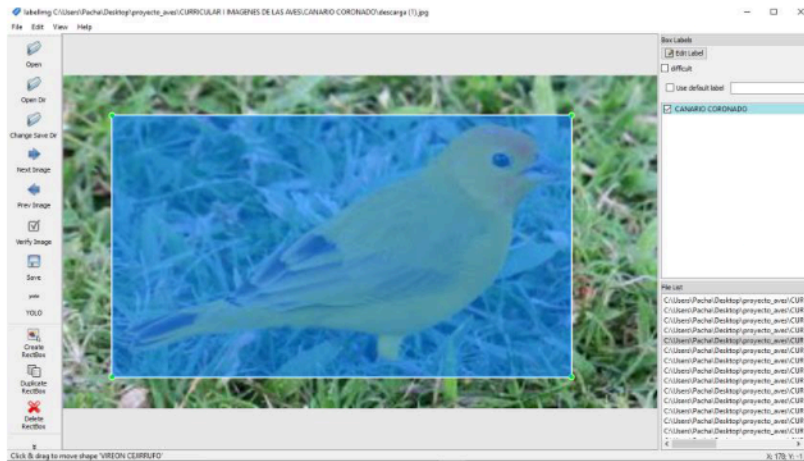


Ilustración 5: Etiquetado de imagen en Labeling: Elaborado por el autor

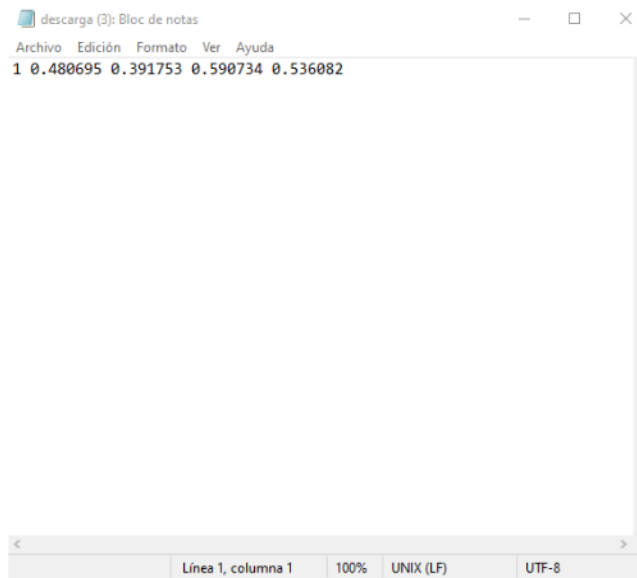


Ilustración 6: txt con la ubicación del objeto: Elaborado por el autor

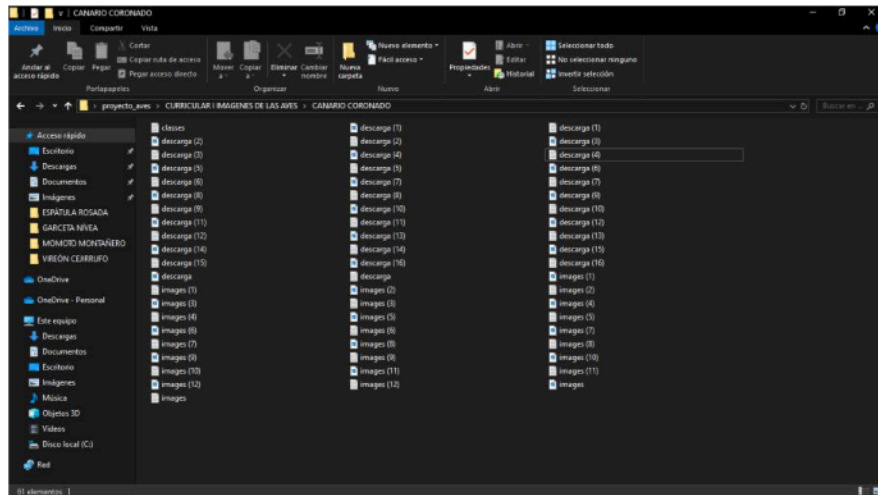


Ilustración 7: Etiquetado guardado en formato txt: Elaborado por el autor

Aumento del Dataset con Datos Sintéticos

Durante el proceso se pudo notar que la cantidad de imágenes que estábamos usando por especie nos quedaba muy corta en base a la arquitectura que redes neuronales convolucionales que se está usando en el proyecto, la cual es ResNet 50, una de las características que más resalta en esta arquitectura es que está hecha para soportar grandes volúmenes de datos, y en el caso de no contar con un dataset robusto nos podría resultar contraproducente y no se estarían aprovechando al máximo los recursos que nos ofrece, así que por ese motivo se creó un script en Python para el aumento de imágenes, para el desarrollo del mismo se usó Jupyter Notebook.

Básicamente el script fue desarrollado para que en base al dataset inicial genere más imágenes artificiales, para esto se usó la librería Albumentations que nos permite hacer este tipo de aumentos, como ya teníamos imágenes y etiquetas hechas de manera manual por decirlo así, ahora el código debía generar más imágenes con sus respectivas etiquetas YOLO, para esto se configuró que por cada imagen de nuestro primer dataset se generen 6 imágenes, dando un total de 180 imágenes nuevas y sumadas con las 30 ya existentes nos daban un total de 210 imágenes por especie, llegando a un dataset robusto de 4830 imágenes con su respectiva etiqueta YOLO.

```
[1]: pip install albumentations opencv-python tqdm

Requirement already satisfied: albumentations in c:\users\pacha\appdata\local\programs\python\python312\lib\site-packages (2.0.8)
Requirement already satisfied: opencv-python in c:\users\pacha\appdata\local\programs\python\python312\lib\site-packages (4.11.0.86)
Requirement already satisfied: tqdm in c:\users\pacha\appdata\local\programs\python\python312\lib\site-packages (4.67.1)
Requirement already satisfied: numpy>=1.24.4 in c:\users\pacha\appdata\local\programs\python\python312\lib\site-packages (from albumentations) (2.0.2)
Requirement already satisfied: scipy>=1.10.0 in c:\users\pacha\appdata\local\programs\python\python312\lib\site-packages (from albumentations) (1.14.1)
Requirement already satisfied: PyYAML in c:\users\pacha\appdata\local\programs\python\python312\lib\site-packages (from albumentations) (6.0.2)
Requirement already satisfied: pydantic>=2.9.2 in c:\users\pacha\appdata\local\programs\python\python312\lib\site-packages (from albumentations) (2.11.5)
Requirement already satisfied: albucore==0.0.24 in c:\users\pacha\appdata\local\programs\python\python312\lib\site-packages (from albumentations) (0.0.24)
Requirement already satisfied: opencv-python-headless>=4.9.0.80 in c:\users\pacha\appdata\local\programs\python\python312\lib\site-packages (from albumentations) (4.11.0.86)
Requirement already satisfied: stringzilla>=3.10.4 in c:\users\pacha\appdata\local\programs\python\python312\lib\site-packages (from albucore==0.0.24->albumentations) (3.12.5)
Requirement already satisfied: simevents==5.9.2 in c:\users\pacha\appdata\local\programs\python\python312\lib\site-packages (from albucore==0.0.24->albumentations) (6.4.7)
Requirement already satisfied: colorama in c:\users\pacha\appdata\local\programs\python\python312\lib\site-packages (from tqdm) (0.4.6)
Requirement already satisfied: annotated-types>=0.6.0 in c:\users\pacha\appdata\local\programs\python\python312\lib\site-packages (from pydantic>=2.9.2->albumentations) (0.7.0)
Requirement already satisfied: pydantic-core==2.33.2 in c:\users\pacha\appdata\local\programs\python\python312\lib\site-packages (from pydantic>=2.9.2->albumentations) (2.33.2)
Requirement already satisfied: typing-extensions>=4.12.2 in c:\users\pacha\appdata\local\programs\python\python312\lib\site-packages (from pydantic>=2.9.2->albumentations) (4.12.2)
Requirement already satisfied: typing-inspection>=0.4.0 in c:\users\pacha\appdata\local\programs\python\python312\lib\site-packages (from pydantic>=2.9.2->albumentations) (0.4.1)
Note: you may need to restart the kernel to use updated packages.
```

Ilustración 8: Instalación de la librería Albumentations: Elaborado por el autor

```
# Función para Leer etiquetas YOLO
def read_yolo_labels(label_path):
    bboxes = []
    class_labels = []
    with open(label_path, 'r') as file:
        for line in file.readlines():
            parts = line.strip().split()
            if len(parts) != 5:
                continue
            class_id, x_center, y_center, width, height = map(float, parts)
            bboxes.append([x_center, y_center, width, height])
            class_labels.append(int(class_id))
    return bboxes, class_labels

# Función para guardar nuevas etiquetas
def save_yolo_labels(path, bboxes, class_ids):
    with open(path, 'w') as f:
        for box, cls in zip(bboxes, class_ids):
            x_center, y_center, width, height = box
            f.write(f"{cls} {x_center:.6f} {y_center:.6f} {width:.6f} {height:.6f}\n")
```

Ilustración 9: Función para leer y guardar nuevas etiquetas YOLO: Elaborado por el autor

```
# Aumentaciones
transform = A.Compose([
    A.ShiftScaleRotate(shift_limit=0.05, scale_limit=0.1, rotate_limit=15, p=0.9, border_mode=cv2.BORDER_REFLECT),
    A.RandomBrightnessContrast(p=0.5),
    A.HorizontalFlip(p=0.5),
    A.VerticalFlip(p=0.2),
    A.HueSaturationValue(p=0.3)
], bbox_params=A.BboxParams(format='yolo', label_fields=['class_labels']))
```

Ilustración 10: Aumento de 5 imágenes por cada imagen original: Elaborado por el autor

3.1.2. Fase 2: Desarrollo del Algoritmo de Clasificación

Seguimos usando la herramienta de Jupyter Notebook para el desarrollo de nuestro algoritmo Python que se encargará de entrenar nuestro modelo usando RestNet 50, el nuevo dataset creado con el nombre de “DATASET_AUMENTADO” es el que nos servirá para el entrenamiento del modelo, el mismo que aún conserva su estructura de subcarpetas por cada especie, ya que así lo requiere

Flow_from_directory de Keras, también se procede a preparar los datos usando ImageDataGenerator el cual normaliza los píxeles en 1./255 y aplica aumento de imágenes como zooms, desplazamientos y rotaciones para que de esta manera el modelo generalice mejor.

```
# Ruta a tu dataset
base_dir = 'C:/Users/Pacha/Desktop/proyecto_aves/DATASET_AUMENTADO'

# Crear generadores con aumento de datos
train_datagen = ImageDataGenerator(
    rescale=1./255,
    validation_split=0.2,
    rotation_range=25,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
```

Ilustración 11: Carga de dataset y aumento de datos: Elaborado por el autor

Se divide de forma automática en entrenamiento con un 80% y en validación un 20% el conjunto de datos, usando el método Flow_from_directory se crean generadores lo cuales tienen la función de leer las imágenes y sus etiquetas categóricas, se ajusta su tamaño a 224x224 píxeles, tamaño requerido obligatoriamente por la arquitectura usada en el proyecto, ResNet 50.

```
train_generator = train_datagen.flow_from_directory(
    base_dir,
    target_size=(224, 224),
    batch_size=16,
    class_mode='categorical',
    subset='training'
)

val_generator = train_datagen.flow_from_directory(
    base_dir,
    target_size=(224, 224),
    batch_size=16,
    class_mode='categorical',
    subset='validation'
)
```

Ilustración 12: Entrenamiento y validación: Elaborado por el autor

Usando ResNet 50 implementamos el transfer learning, cabe recalcar que el Resnet se carga sin su parte superior, refiriéndome a las capas de clasificación originales debido a su parámetro `include_top=False`, el mismo que permite que se reutilicen las capas convolucionales como extractor de características, para mejorar la clasificación de aves de nuestro modelo se realiza un fine tuning descogelando las últimas 30 capas para que estas puedan ajustarse en el entrenamiento y las demás siguen congeladas para conservar el conocimiento general ya adquirido.

```
# Cargar modelo base ResNet50
base_model = ResNet50(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

# Descongelar últimas 30 capas para fine-tuning
for layer in base_model.layers[:-30]:
    layer.trainable = False
for layer in base_model.layers[-30:]:
    layer.trainable = True
```

Ilustración 13: Transfer learning: Elaborado por el autor

También se hace uso de los llamados Callbacks durante el entrenamiento, `EarlyStopping` se encarga de detener el proceso si la pérdida de validación no mejora cada cierto rango, en este caso es cada 5 épocas consecutivas, y el `ModelCheckpoint` que se encarga de guardar el mejor modelo en formato keras, y por último se colocó un límite de 50 épocas como máximo, con la opción de poder detenerse antes si el algoritmo lo ve como necesario, en este caso nuestro entrenamiento finalizó al completarse alrededor de 9 épocas.

```
# Callbacks
early_stop = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
checkpoint = ModelCheckpoint("modelo_resnet50_aves_finalfinalfinalfinal.keras", save_best_only=True)

# Entrenamiento
history = model.fit(
    train_generator,
    epochs=50,
    validation_data=val_generator,
    callbacks=[early_stop, checkpoint]
)
```

Ilustración 14: Callbacks y Entrenamiento: Elaborado por el autor

```

Epoch 1/50
200/200 ----- 0s 6s/step - accuracy: 0.0987 - loss: 2.9867
C:\Users\Pahe\AppData\Local\Programs\Python\Python312\Lib\site-packages\keras\src\trainers\data_adapters\py_dataset_adapter.py:121: UserWarning: Your 'PyDataset' class should call 'super().__init__(**kwargs)' in its constructor. '**kwargs' can include 'workers', 'use_multiprocessing', 'max_queue_size'. Do not pass these arguments to 'fit()', as they will be ignored.
  self._warn_if_super_not_called()
Epoch 2/50
200/200 ----- 1335s 7s/step - accuracy: 0.0987 - loss: 2.9864 - val_accuracy: 0.0892 - val_loss: 2.8696
Epoch 3/50
200/200 ----- 1098s 6s/step - accuracy: 0.1286 - loss: 2.7867 - val_accuracy: 0.1005 - val_loss: 3.2400
Epoch 4/50
200/200 ----- 1075s 5s/step - accuracy: 0.1449 - loss: 2.7535 - val_accuracy: 0.0867 - val_loss: 3.1833
Epoch 5/50
200/200 ----- 1004s 5s/step - accuracy: 0.1814 - loss: 2.6630 - val_accuracy: 0.1633 - val_loss: 2.7681
Epoch 6/50
200/200 ----- 1088s 5s/step - accuracy: 0.2128 - loss: 2.5457 - val_accuracy: 0.1118 - val_loss: 3.4106
Epoch 7/50
200/200 ----- 1118s 6s/step - accuracy: 0.2164 - loss: 2.4890 - val_accuracy: 0.1520 - val_loss: 2.9035
Epoch 8/50
200/200 ----- 1199s 6s/step - accuracy: 0.2309 - loss: 2.4456 - val_accuracy: 0.2010 - val_loss: 2.8301
Epoch 9/50
200/200 ----- 1423s 7s/step - accuracy: 0.2560 - loss: 2.3633 - val_accuracy: 0.1784 - val_loss: 2.7685
Epoch 9/50
200/200 ----- 1494s 7s/step - accuracy: 0.2737 - loss: 2.3167 - val_accuracy: 0.1382 - val_loss: 3.7238

```

Ilustración 15: Número de épocas realizadas: Elaborado por el autor

3.1.3. Fase 3: Desarrollo del Producto Mínimo Viable (MVP)

Una vez entrenado el modelo tenemos como resultado un archivo en formato keras, el cual posee todos los parámetros requeridos para nuestro sistema de identificación de aves, como siguiente paso se procede a instalar en nuestro ordenador un editor de texto cualquiera, en este caso se eligió el editor de texto Sublime Text, siendo ahí donde se desarrollará el código de nuestra interfaz gráfica.

En primer lugar, se importan todas las bibliotecas que se van a usar en nuestro código para la elaboración de nuestra interfaz, como es el caso de streamlit y PIL, para la parte de manipulación de datos tenemos numpy y json, y para ejecutar el modelo tenemos tensorflow, con esto tenemos las herramientas fundamentales para cargar y procesar imágenes.

```

import streamlit as st
import tensorflow as tf
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
from PIL import Image
import json

```

Ilustración 16: Importación de librerías: Elaborado por el autor

Luego se define una función para cargar el modelo anteriormente entrenado, el cual es “modelo_resnet50_aves_finalfinalfinalfinal.heras”, pero antes de eso se agrega @st.cache_resource, lo que hace que el modelo solo se cargue una vez por cada sesión evitando las cargas innecesarias

```
# Cargar modelo
@st.cache_resource
def load_model_custom():
    return load_model("modelo_resnet50_aves_finalfinalfinalfinal.keras")

model = load_model_custom()
```

Ilustración 17: Carga del modelo: Elaborado por el autor

Aquí se carga el archivo json el cual contiene un diccionario creado para el mapeo de las clases por especie y sus respectivos índices numéricos, y luego se invierte el mismo para poder convertir predicciones numéricas en nombres legibles de las clases, que en este caso son el nombre de las especies de aves.

```
# Cargar diccionario de clases
with open("class_indicesfinalfinalfinalfinal.json", "r") as f:
    class_indices = json.load(f)
index_to_class = {v: k for k, v in class_indices.items()}
```

Ilustración 18: Carga de diccionario de clases: Elaborado por el autor

A continuación se crea una función que se encargará de transformar los nombres de las clases en un formato un poco más legible para el usuario, por ejemplo, en caso de ser canario_colorado la identificación se mostrará en pantalla la respuesta de Canario Colorado, esto se logra gracias al uso de `replace` y `title()`.

```
# Función para embellecer el nombre de la especie
def formatear_nombre(nombre):
    nombre_bonito = nombre.replace("_", " ").title()
    return nombre_bonito
```

Ilustración 19: Función para mejorar el nombre del resultado: Elaborado por el autor

Seguimos mejorando lo que se muestra en la pantalla con títulos y emojis relacionados al sistema como tal, luego se realiza la parte donde se le permite al usuario subir una imagen en formato jp, jpeg o png, luego que la imagen sea cargada se procede a convertir en formato RGB lo cual garantiza la compatibilidad, se muestra en pantalla la imagen cargada correctamente junto al nombre de la especie detectada.

```
if uploaded_file is not None:
    img = Image.open(uploaded_file).convert("RGB")
    st.image(img, caption="Imagen cargada", use_column_width=True)

    # Preprocesar imagen
    img = img.resize((224, 224))
    img_array = image.img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0) / 255.0

st.success(f"🐦 Especie detectada: **{predicted_class_bonito}**")
```

Ilustración 20: Subir la imagen del ave: Elaborado por el autor

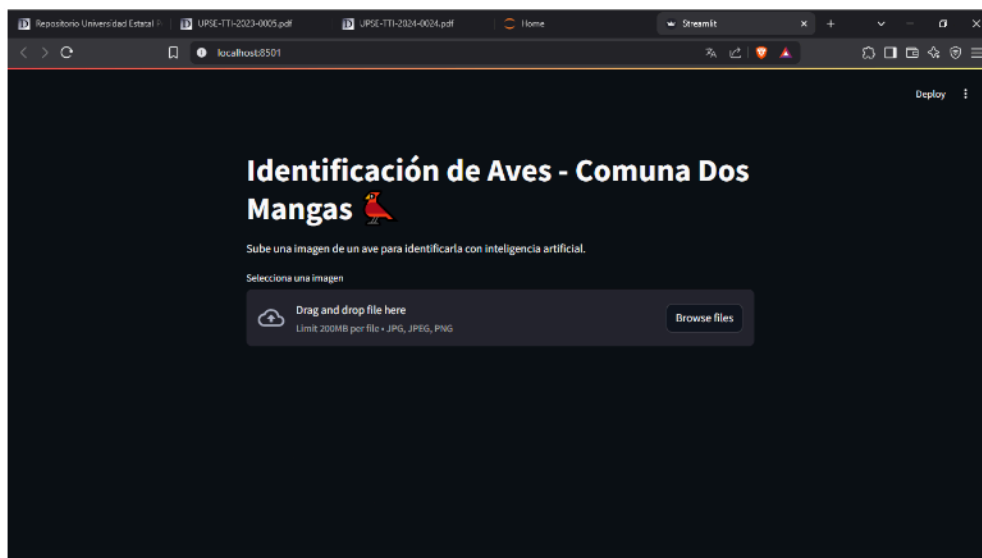


Ilustración 21: Interfaz del sistema en streamlit: Elaborado por el autor

3.1.4. Fase 4: Pruebas y Validación del Sistema

Para asegurarnos que el sistema de identificación de aves funcione de correcta manera es necesario hacer pruebas en diferentes escenarios, para evaluar la efectividad de los resultados a obtener, además estas mismas nos ayudarán a detectar cualquier fallo presentado durante las distintas pruebas, para esto se crearon 10 escenarios diferentes especificados en las siguientes tablas:


DATOS DEL CASO DE PRUEBA	
CAMPO	DETALLE
NÚMERO DE LA PRUEBA	1
OBJETIVO	Verificar si el sistema es capaz de identificar de manera correcta la especie de un ave proporcionándole una imagen con buena calidad, iluminación y enfoque.
ENTRADA	Imagen en formato jpg de un Picamaderos de Guayaquil
RESULTADO ESPERADO	El sistema debe detectar: Picamaderos de Guayaquil
RESULTADO OBTENIDO	Picamaderos de Guayaquil
ESTADO	

Tabla 1: Prueba 1 imagen con buena calidad: Elaborado por el autor

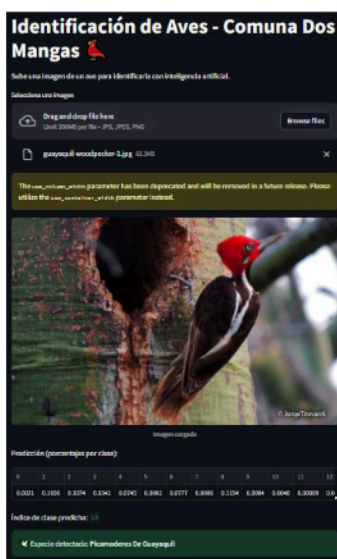


Ilustración 22: Prueba 1 imagen con buena calidad: Elaborado por el autor


DATOS DEL CASO DE PRUEBA	
CAMPO	DETALLE
NÚMERO DE LA PRUEBA	2
OBJETIVO	Verificar si el sistema es capaz de identificar de manera correcta la especie de un ave proporcionándole una imagen con calidad no muy buena.
ENTRADA	Imagen en formato jpeg de una Garza Encapuchada
RESULTADO ESPERADO	El sistema debe detectar: Garza Encapuchada
RESULTADO OBTENIDO	Garza Encapuchada
ESTADO	

Tabla 2: Prueba 2 imagen con mala calidad: Elaborado por el autor

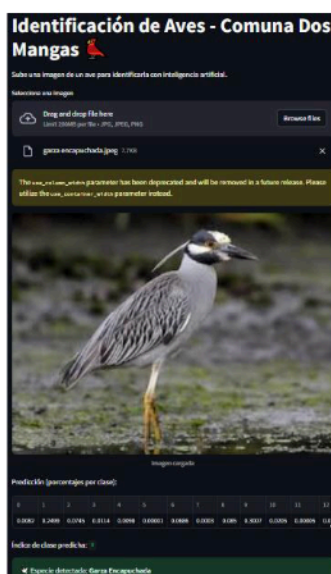


Ilustración 23: Resultado obtenido de la prueba 2: Elaborado por el autor


DATOS DEL CASO DE PRUEBA	
CAMPO	DETALLE
NÚMERO DE LA PRUEBA	3
OBJETIVO	Verificar si el sistema es capaz de identificar de manera correcta la especie de un ave proporcionándole una imagen en la que se encuentren varias aves de la misma especie.
ENTRADA	Imagen en formato jpeg de un Picamaderos de Guayaquil
RESULTADO ESPERADO	El sistema debe detectar: Picamaderos de Guayaquil
RESULTADO OBTENIDO	Picamaderos de Guayaquil
ESTADO	

Tabla 3: Prueba 3 varias aves de la misma especie en una foto: Elaborado por el autor

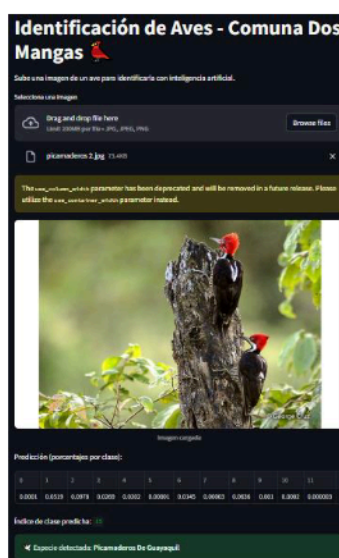


Ilustración 24: Resultado obtenido de la prueba 3: Elaborado por el autor


DATOS DEL CASO DE PRUEBA	
CAMPO	DETALLE
NÚMERO DE LA PRUEBA	4
OBJETIVO	Verificar si el sistema es capaz de identificar de manera correcta la especie de un ave proporcionándole una imagen con poco brillo
ENTRADA	Imagen en formato png de un Ibis Blanco
RESULTADO ESPERADO	El sistema debe detectar: Ibis Blanco
RESULTADO OBTENIDO	Ibis blanco
ESTADO	

Tabla 4: Prueba 4 imagen con poco brillo: Elaborado por el autor

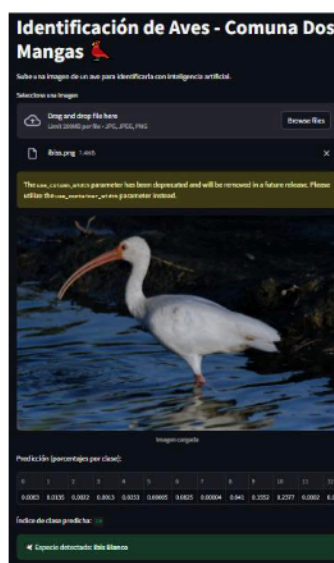


Ilustración 25: Resultado obtenido de la prueba 4: Elaborado por el autor


DATOS DEL CASO DE PRUEBA	
CAMPO	DETALLE
NÚMERO DE LA PRUEBA	5
OBJETIVO	Verificar si el sistema es capaz de identificar de mandar un mensaje de advertencia cuando una especie no corresponde a las del dataset.
ENTRADA	Imagen en formato jpg de un Cernícalo Americano
RESULTADO ESPERADO	El sistema debe detectar: La especie que intenta identificar no está dentro de las consideradas en este proyecto
RESULTADO OBTENIDO	La especie que intenta identificar no está dentro de las consideradas en este proyecto
ESTADO	

Tabla 5: Prueba 5 especie que no consta en el dataset: Elaborado por el autor

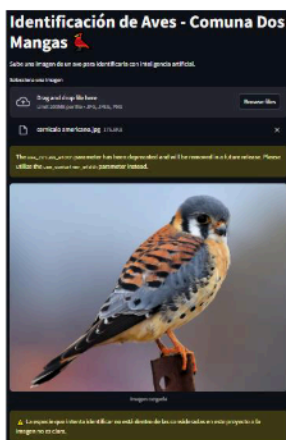


Ilustración 26: Resultados obtenidos de la prueba 5: Elaborado por el autor


DATOS DEL CASO DE PRUEBA	
CAMPO	DETALLE
NÚMERO DE LA PRUEBA	6
OBJETIVO	Verificar si el sistema es capaz de dar un error cuando se sube un archivo con un formato que no soporta el sistema
ENTRADA	Imagen en formato html de un Picamaderos de Guayaquil
RESULTADO ESPERADO	El sistema debe detectar: Error
RESULTADO OBTENIDO	Error
ESTADO	

Tabla 6: Prueba 6 error en el formato del archivo: Elaborado por el autor



Ilustración 27: Resultados obtenidos de la prueba 6: Elaborado por el autor


DATOS DEL CASO DE PRUEBA	
CAMPO	DETALLE
NÚMERO DE LA PRUEBA	7
OBJETIVO	Verificar si el sistema es capaz de identificar la especie con una imagen desenfocada
ENTRADA	Imagen en formato png de un Ibis Blanco
RESULTADO ESPERADO	El sistema debe detectar: Ibis Banco
RESULTADO OBTENIDO	Ibis Blanco
ESTADO	

Tabla 7: Prueba 7 imagen desenfocada: Elaborado por el autor

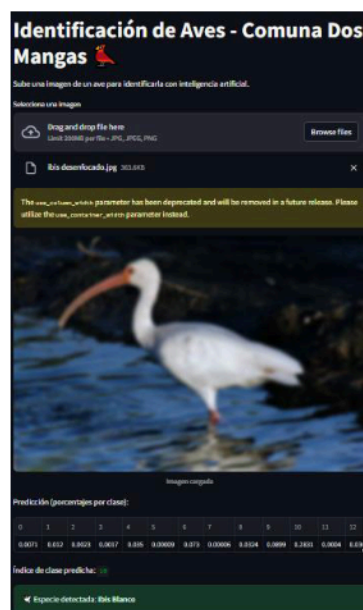


Ilustración 28: Resultados obtenidos de la prueba 7: Elaborado por el autor


DATOS DEL CASO DE PRUEBA	
CAMPO	DETALLE
NÚMERO DE LA PRUEBA	8
OBJETIVO	Verificar si el sistema es capaz de identificar la especie con una imagen recortada
ENTRADA	Imagen en formato jpg de un Garrapatero Asurcado
RESULTADO ESPERADO	El sistema debe detectar: Garrapatero Asurcado
RESULTADO OBTENIDO	Garrapatero Asurcado
ESTADO	

Tabla 8: Prueba 8 imagen recortada: Elaborada por el autor

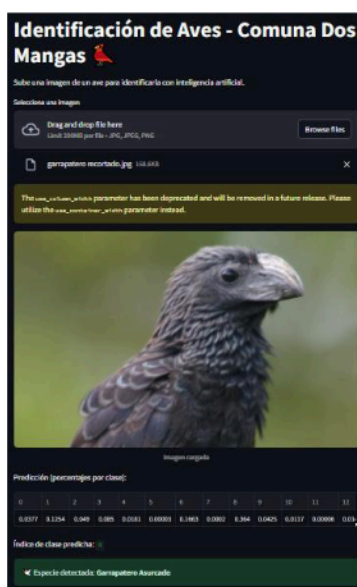


Ilustración 29: Resultados obtenidos de la prueba 8: Elaborado por el autor


DATOS DEL CASO DE PRUEBA	
CAMPO	DETALLE
NÚMERO DE LA PRUEBA	9
OBJETIVO	Verificar si el sistema es capaz de identificar un ave Tirano Tropical
ENTRADA	Imagen en formato jpg de un Tirano Tropical
RESULTADO ESPERADO	El sistema debe detectar: Tirano Tropical
RESULTADO OBTENIDO	Garrapatero Asurcado
ESTADO	

Tabla 9: Prueba 9 imagen de Tirano Tropical: Elaborado por el autor

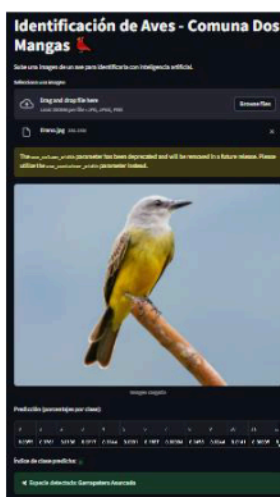


Ilustración 30: Resultados obtenidos de la prueba 9: Elaborado por el autor


DATOS DEL CASO DE PRUEBA	
CAMPO	DETALLE
NÚMERO DE LA PRUEBA	10
OBJETIVO	Verificar si el sistema es capaz de identificar un ave Tangara Azuleja
ENTRADA	Imagen en formato jpg de una Tangara Azuleja
RESULTADO ESPERADO	El sistema debe detectar: Tangara Azuleja
RESULTADO OBTENIDO	Viudita Enmascarada
ESTADO	

Tabla 10: Prueba 10 imagen de Tangara Azuleja: Elaborado por el autor

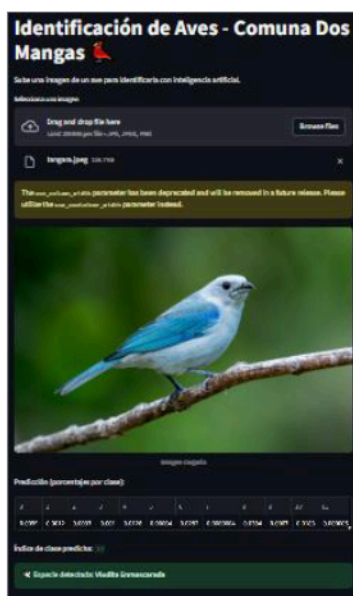


Ilustración 31: Resultados obtenidos de la prueba 10: Elaborado por el autor

Durante el entrenamiento del modelo se realizaron 9 épocas con la arquitectura Resnet 50, se activó fine-tuning que es una técnica que consiste en usar un modelo pre entrenado y adaptarlo a una tarea específica, sobre el conjunto de datos aumentado artificialmente, con un total de 4830 imágenes, la partición usada consiste en un 80% para el entrenamiento y un 20% para la validación usando la clase ImageDataGenerator con validation_split, los resultados obtenidos en el entrenando del modelo son los siguientes:

MÉTRICA	ENTRENAMIENTO	VALIDACIÓN
PRECISIÓN (accuracy)	27.37%	20.10%
PÉRDIDA (loss)	2.3167	3.7238

Tabla 11: Resultados del entrenamiento del modelo: Elaborado por el autor

Durante las épocas el modelo empezó a mostrar signos de sobreajuste desde la época número 5, deteniéndose por completo una vez terminada la época 9 gracias al EarlyStopping, se notó esto debido a que la pérdida de validación iba en aumento y por otra parte la precisión de entrenamiento subía, concluyendo que el modelo empezaba a memorizar en lugar de generalizar, una vez realizadas las diferentes pruebas en la etapa 4 del proyecto llegamos a una cantidad de 8 aciertos en un conjunto de 10 pruebas, lo que nos muestra una precisión empírica del 80% en condiciones de uso real.

```

C:\Users\Pacha\AppData\Local\Programs\Python\Python312\Lib\site-packages\keras\src\trainers\data_adapters\py_dataset_adapter.py:121: UserWarning: Your 'PyDataset' class should call 'super().__init__(**kwargs)' in its constructor. '**kwargs' can include 'workers', 'use_multiprocessing', 'max_queue_size'. Do not pass these arguments to 'fit()', as they will be ignored.
  self._warn_if_super_not_called()
Epoch 1/50
200/200 --- 0s 6s/step - accuracy: 0.0987 - loss: 2.9867
C:\Users\Pacha\AppData\Local\Programs\Python\Python312\Lib\site-packages\keras\src\trainers\data_adapters\py_dataset_adapter.py:121: UserWarning: Your 'PyDataset' class should call 'super().__init__(**kwargs)' in its constructor. '**kwargs' can include 'workers', 'use_multiprocessing', 'max_queue_size'. Do not pass these arguments to 'fit()', as they will be ignored.
  self._warn_if_super_not_called()
Epoch 2/50
200/200 --- 1335s 7s/step - accuracy: 0.0987 - loss: 2.9864 - val_accuracy: 0.0892 - val_loss: 2.8696
Epoch 3/50
200/200 --- 1098s 6s/step - accuracy: 0.1286 - loss: 2.7867 - val_accuracy: 0.1005 - val_loss: 3.2400
Epoch 4/50
200/200 --- 1075s 5s/step - accuracy: 0.1449 - loss: 2.7535 - val_accuracy: 0.0867 - val_loss: 3.1833
Epoch 5/50
200/200 --- 1004s 5s/step - accuracy: 0.1814 - loss: 2.6630 - val_accuracy: 0.1633 - val_loss: 2.7681
Epoch 6/50
200/200 --- 1088s 5s/step - accuracy: 0.2120 - loss: 2.5457 - val_accuracy: 0.1118 - val_loss: 3.4106
Epoch 7/50
200/200 --- 1118s 6s/step - accuracy: 0.2164 - loss: 2.4890 - val_accuracy: 0.1520 - val_loss: 2.9035
Epoch 8/50
200/200 --- 1199s 6s/step - accuracy: 0.2309 - loss: 2.4450 - val_accuracy: 0.2010 - val_loss: 2.8301
Epoch 9/50
200/200 --- 1423s 7s/step - accuracy: 0.2500 - loss: 2.3633 - val_accuracy: 0.1784 - val_loss: 2.7685
Epoch 9/50
200/200 --- 1494s 7s/step - accuracy: 0.2737 - loss: 2.3167 - val_accuracy: 0.1382 - val_loss: 3.7238

```

Ilustración 32: Resultados del entrenamiento: Elaborado por el autor

CONCLUSIONES

- Se entrenó un modelo con las características para poder identificar aves de la comuna Dos Mangas, el cual trabajó con una arquitectura Resnet50, que se especializa en manejar grandes volúmenes de datos, y siendo una parte fundamental en el entrenamiento del modelo, después de pasar por varias pruebas en diferentes escenarios se evidenció que el modelo tiene un 80% de precisión empírica, debido a que en las 10 pruebas realizadas acertó 8 veces y erró 2 veces en los resultados.
- Se elaboró un dataset estructurado, una carpeta por cada especie, 23 en total, y por cada carpeta 30 imágenes con su respectiva etiqueta YOLO realizada con la herramienta Labelimg, dentro de nuestro algoritmo estas carpetas se interpretaron como clases, inicialmente se tenía un conjunto pequeño de imágenes, pero a medida que se avanzaba en el proyecto se pudo notar que era necesario aplicar técnicas de aumento de datos desarrollando un script que nos ayudaría con esta tarea, llegando así a un dataset de 4830 imágenes, dado que el algoritmo fue hecho para que aumente una cantidad de 6 imágenes sintéticas por cada imagen real encontrada en el dataset, de esta manera se obtuvo un dataset robusto e ideal para nuestra arquitectura.
- Gracias al uso de Streamlit se pudo diseñar en Python una interfaz intuitiva, fácil de usar y que aparte se vea atractiva en los colores, básicamente su uso se resume en un botón para la carga de una imagen (ave a identificar), puede ser el formato jpg, png o jpeg, el sistema se encargará de procesarla y en base a su modelo entrenado se encargará de identificar la especie de ave de la foto proporcionada.

RECOMENDACIONES

- Para este proyecto se usó la arquitectura ResNet50, pero eso no significa que sea la única que existe para este tipo de tareas, también se pueden explorar otras arquitecturas como EfficientNet, que ha sido diseñada para optimizar el rendimiento reduciendo el consumo de recursos computacionales, InceptionV3, que destaca por su estructura que permite capturar características a múltiples escalas dentro de una imagen, o DenseNet, que tiene un mejor flujo de información gracias a su conectividad entre las capas anteriores, para así aumentar la precisión del sistema.
- Para un desarrollo escalable, se recomienda ir aumentando el tamaño del dataset, de manera proporcional para todas las clases, así nos aseguramos de que haya un equilibrio en el entrenamiento del modelo, otra buena opción y la que se aplicó en este proyecto es el incremento de imágenes sintéticas en base a las imágenes originales ya existentes en el dataset.
- En este proyecto se usó Streamlit para el desarrollo de una interfaz intuitiva y amigable para el usuario, y para el desarrollador, ya que el objetivo era crear un producto mínimo viable, pero en caso se quiera hacer una mejora significativa en la interfaz se puede usar frameworks de desarrollo web más completos como lo son HTML, CSS y JavaScript.
- Se sugiere considerar el uso de métricas como Recall que mide la capacidad del modelo para encontrar de manera correcta las instancias reales de una clase, Precisión que evalúa que tan preciso es un modelo al predecir una clase determinada, F1-score que es la media entre Precisión y Recall, la matriz de confusión que es una tabla que muestra la distribución de las predicciones de modelo y las compara con las verdaderas clases, todas estas métricas ayudan con la evaluación del modelo.

ANEXOS

ANEXO 1 – MAPA DE AMBAS RUTAS: “LAS POSAS” Y “LAS CASCADAS”



Las Cascadas

1. Paja Toquilla
2. Caña Guadua
3. Fernan Sánchez
4. Heliconias
5. Guarumo
6. Guayacán
7. Figueroa
8. Paja Toquilla
9. Bosque Húmedo Tropical Occidental

Las Piscinas

1. Los Cerros
2. Palmito
3. Flora y Fauna
4. Gigantes de Sumpa
5. Voces en el Bosque



**ANEXO 2 – IMÁGENES RECORRIENDO LOS SENDEROS “LAS POSAS”
Y “LAS CASCADAS”**





Bibliografía

- [1] B. D. Rubio Constante and G. A. Montúfar Burbano, "Desarrollo de una aplicación para la detección del tipo de mascota utilizando machine learning," Tesis de Ingenieros en Telecomunicaciones, Universidad Politécnica Salesiana, Quito, Ecuador, 2024.
- [2] "Visita a Santa Elena," Santa Elena, Ecuador, [<https://www.santaelena.gob.ec/index.php/noticias-boletines-varias/17-turismo/visita?start=8>]. [Accedido: 25/09/2024].
- [3] E. G. Beltrán Prudente, "La problemática del turismo comunitario frente al desarrollo turístico de la comuna Dos Mangas, cantón Santa Elena," Proyecto de investigación, Universidad Estatal Península de Santa Elena, Facultad de Ciencias Administrativas, Carrera Ingeniería en Gestión y Desarrollo Turístico, La Libertad, Ecuador, febrero 2017.
- [4] S. Cheng and J. Wang, "Detection of bird species through sounds," Department of Computer Science, Stanford University. [Online]. Available: <https://www.stanford.edu>. [Accessed: 25-Sep-2024].
- [5] L. A. Hernández Abad, "PreBird: biblioteca open source de Python para el preprocesamiento de audio, orientado a canto de aves," Universidad Nacional Autónoma de México, Facultad de Ingeniería, 2021.
- [6] E. J. Pilay José, "Diversidad y abundancia de aves en la comuna Dos Mangas, parroquia Manglaralto, provincia de Santa Elena, Ecuador," Universidad Estatal Península de Santa Elena, Facultad de Ciencias del Mar, 2022.
- [7] J. D. Tàbara, Las aves como naturaleza y la conservación de las aves como cultura. Universitat Autònoma de Barcelona, Institut de Ciència i Tecnologia Ambientals, 1964. [En línea]. Disponible en: <https://ddd.uab.cat/record/16725>
- [8] L. M. Leveau y C. M. Leveau, "Comunidades de aves en un gradiente urbano de la ciudad de Mar del Plata, Argentina," Revista del Museo Argentino de Ciencias Naturales, vol. 6, no. 1, pp. 1-10, 2004. [En línea]. Disponible en: https://www.scielo.org.ar/scielo.php?pid=s0073-34072004000100003&script=sci_arttext
- [9] C. A. Correa, "La inteligencia artificial para conservar la biodiversidad," Hoy en la Javeriana, Ed. 1390, ago. 2023. [Enlace]. Disponible en: <https://www.javeriana.edu.co/repositorio-hoy-en-la-javeriana/la-inteligencia-artificial-para-conservar-la-biodiversidad/>
- [10] J. C. Castillo Orrala, "Reestructuración de cableado horizontal empleando los estándares; ANSI/TIA/EIA 568-B, TIA/EIA 569-B, ANSI/TIA 606-C y ISO/IEC 14763-1 para el laboratorio 06-07 de la Universidad Estatal Península de Santa Elena", 19-abr-2023. [Enlace]. Disponible en: <https://repositorio.upse.edu.ec/handle/46000/9261>.
- [11] Revista Latino-Americana de Enfermagem, "Revisión de diseños de investigación resaltantes para enfermería. Parte 1: diseños de investigación cuantitativa," Revista Latino-Americana de Enfermagem, Jun. 2007. [En línea]. Disponible en: <https://www.scielo.br/j/rlae/a/7zMf8XypC67vGPrXVrVFGdx/?lang=es&for>.
- [12] N. T. E. Nieto, Tipos de Investigación. [En línea]. Disponible en: <https://d1wqtxts1xzle7.cloudfront.net/99846223/250080756-libre.pdf>.
- [13] F. López Saca, Clasificación de imágenes usando redes neuronales convolucionales, Universidad Autónoma Metropolitana, Unidad Azcapotzalco, 30 de mayo de 2019.
- [14] B. L. Takeyas, Introducción a la inteligencia artificial. Instituto Tecnológico de

- Nuevo Laredo, Reforma Sur 2007, C.P. 88250, Nuevo Laredo, Tamps., México.
- [15] Viso.ai, "LabelImg for Image Annotation," [Online]. Available: <https://viso.ai/computer-vision/labelimg-for-image-annotation/>.
- [16] The Information Lab, "Qué es un dataset," 27 de noviembre de 2023. [Online]. Available: <https://www.theinformationlab.es/blog/que-es-un-dataset/>.
- [17] Universidad ORT Uruguay, Facultad de Ingeniería, El proceso de creación y evolución del Producto Mínimo Viable en las startups de software, Tutor: G. Maturro.
- [18] NetApp, "¿Qué es la inteligencia artificial?," NetApp. [En línea]. Disponible en: <https://www.netapp.com/es/artificial-intelligence/what-is-artificial-intelligence/>.
- [19] J. D. Baquero Gómez y A. C. Cuellar Velásquez, Reconocimiento de la importancia ecológica de las aves y su contribución al cuidado de la biodiversidad en el trapecio amazónico con estudiantes de preescolar y primero en San Antonio (Perú).
- [20] L. Páez, "Visión Computacional, ¿qué es y por dónde comenzar?," Ana2LP, [En línea]. Disponible en: <https://www.ana2lp.mx/inteligencia-artificial/vision-computacional-donde-comenzar/>. [Accedido: 15-abr-2025].
- [21] H. Cruz, "La visión por computadora y las futuras aplicaciones tecnológicas en diversos escenarios," *Revista de la Academia de Guerra del Ejército Ecuatoriano*, vol. 12, núm. 1, pp. 116–120, abr. 2019. [En línea]. Disponible en: <https://journal.espe.edu.ec/ojs/index.php/Academia-de-guerra/article/view/VOL12ART13/html>. [Accedido: 15-abr-2025].
- [22] D. N. Quinto Rodríguez, Algoritmo de Machine Learning en el Reconocimiento de Imágenes, Universidad Técnica de Babahoyo, 2024. [En línea]. Disponible en: <https://dspace.utb.edu.ec/handle/49000/16984>. [Accedido: 15-abr-2025].
- [23] Bobadilla, *Machine learning y deep learning: usando Python, Scikit y Keras*. Bogotá, Colombia: Ediciones de la U, 2021.
- [24] IBM, "¿Qué es el aprendizaje no supervisado?," *IBM Think*, 23-sep-2021. [En línea]. Disponible en: <https://www.ibm.com/es-es/think/topics/unsupervised-learning>
- [25] IBM, "¿Qué es el aumento de datos?," *IBM Think*, 07-may-2024. [En línea]. Disponible en: <https://www.ibm.com/es-es/think/topics/data-augmentation>
- [26] I. Challenger-Pérez, Y. Díaz-Ricardo y R. A. Becerra-García, "El lenguaje de programación Python," *Ciencias Holguín*, vol. 20, no. 2, pp. 1–13, 2014. [En línea]. Disponible en: <https://www.redalyc.org/pdf/1815/181531232001.pdf>
- [27] N. Kundu, "Explorando ResNet50: Una Visión In-Depth de la Arquitectura Modelo y la Implementación del Código," *Medium*, 23 de enero de 2023. [En línea]. Disponible en: <https://medium.com/@nitishkundu1993/exploring-resnet50-an-in-depth-look-at-the-model-architecture-and-code-implementation-d8d8fa67e46f>
- [28] DataCamp, "Streamlit," *DataCamp*, [En línea]. Disponible en: <https://www.datacamp.com/es/tutorial/streamlit>. [Accedido: 11-jun-2025].

