



Universidad
Nacional
de Loja

Universidad Nacional de Loja

Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables

Carrera de Telecomunicaciones

Desarrollo de un Sistema de Reconocimiento de Especies Vegetales para el Jardín Botánico de la UNL Basado en Inteligencia Artificial

Trabajo de Integración Curricular
previo a la obtención del Título de
Ingeniero en Telecomunicaciones

AUTOR:

Hernán Alexander Rivera Merchán.

DIRECTOR:

Ing. Marcelo Valdiviezo Condolo. Mg. Sc.

Loja – Ecuador

2024

Certificación

Loja, 06 de junio de 2024

Ing. Marcelo Valdiviezo Condolo. Mg. Sc

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICO:

Que he revisado y orientado todo el proceso de elaboración del Trabajo de Integración Curricular denominado: **Desarrollo de un Sistema de Reconocimiento de Especies Vegetales para el Jardín Botánico de la UNL Basado en Inteligencia Artificial**, previo a la obtención del título de **Ingeniero en Telecomunicaciones**, de la autoría del estudiante **Hernán Alexander Rivera Merchán**, con **cédula de identidad Nro. 1105337875**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja para el efecto, autorizo la presentación del mismo para su respectiva sustentación y defensa.

Ing. Marcelo Fernando Valdiviezo Condolo. Mg. Sc

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Autoría

Yo, **Hernán Alexander Rivera Merchán**, declaro ser autor del presente Trabajo de Integración Curricular, y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos, de posibles reclamos y acciones legales, por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja la publicación de mi Trabajo de Titulación en el Repositorio Digital Institucional – Biblioteca Virtual.

Firma:

Cédula de identidad: 1105337875

Fecha: 06/06/2024

Correo electrónico: hernan.rivera@unl.edu.ec

Teléfono: 0979218763

Carta de autorización por parte del autor, para consulta, reproducción parcial o total y/o publicación electrónica del texto completo del Trabajo de Integración Curricular.

Yo, **Hernán Alexander Rivera Merchán**, declaro ser autor del Trabajo de Integración Curricular denominado: **Desarrollo de un Sistema de Reconocimiento de Especies Vegetales para el Jardín Botánico de la UNL Basado en Inteligencia Artificial**, como requisito para optar el título de **Ingeniero en Telecomunicaciones**, autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que, con fines académicos, muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido en el Repositorio Digital Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del Trabajo de Integración Curricular que realice un tercero.

Para constancia de esta autorización, suscribo en la ciudad de Loja a los seis días del mes de junio del dos mil veinticuatro.

Firma:

Autor: Hernán Alexander Rivera Merchán

Cédula: 1105337875

Dirección: Cdla. Julio Ordoñez

Correo electrónico: hernan.rivera@unl.edu.ec

Teléfono: 0979218763

DATOS COMPLEMENTARIOS:

Director de Trabajo de Integración Curricular: Ing. Marcelo Fernando Valdiviezo Condolo. Mg. Sc.

Dedicatoria

A mis queridos padres, Hernán y Deice, cuyo amor incondicional y apoyo constante han sido mi roca durante este viaje académico.

A mis hermanos, Daniela y David, por su aliento y comprensión durante todos estos años.

A mis amigos y profesores, que han sido fundamentales en mi formación académica. Este logro no habría sido posible sin su influencia positiva en mi vida.

Hernán Alexander Rivera Merchán

Agradecimientos

Quiero expresar mi más sincero agradecimiento a la Universidad Nacional de Loja por haberme brindado la oportunidad de formarme en el campo de la Ingeniería de Telecomunicaciones. Especialmente, deseo extender mi gratitud a los miembros administrativos y del plantel docente de la carrera de Ingeniería en Telecomunicaciones. Su apoyo ha sido fundamental en mi trayectoria académica, permitiéndome no solo adquirir nuevos conocimientos, sino también desarrollar habilidades y destrezas que serán de invaluable importancia en mi carrera profesional.

Agradezco también al Ing. Marcelo Gutiérrez Riofrio y al personal del Jardín Botánico que han sido de gran ayuda en el desarrollo del trabajo, al asistirme en la recopilación de la información y fotografías de las diferentes especies vegetales utilizadas en el presente trabajo.

Finalmente agradezco al Ing. Marcelo Valdiviezo Condolo, mi director del trabajo de Integración Curricular, quien con sus conocimientos del campo de la inteligencia artificial me ha ido asesorando y orientando a lo largo del desarrollo del trabajo de titulación.

Hernán Alexander Rivera Merchán

Índice de Contenidos

Portada	i
Certificación	ii
Autoría.....	iii
Carta de autorización.....	iv
Dedicatoria	v
Agradecimientos	vi
Índice de Contenidos	vii
Índice de tablas	ix
Índice de figuras	x
Índice de anexos	xii
1. Título	1
2. Resumen	2
Abstract	3
3. Introducción.....	4
4. Marco Teórico	6
4.1. Antecedentes	6
4.2. Visión por computador	7
4.2.1. Adquisición de imágenes	7
4.2.2. Preprocesamiento.....	7
4.2.3. Segmentación.....	8
4.2.4. Extracción de características	8
4.3. Inteligencia Artificial	9
4.3.1. Machine Learning.....	9
4.3.2. Redes Neuronales Artificiales	10
4.4. Arquitecturas	13
4.4.1. MobileNet.....	13
4.4.2. ResNet	14
4.4.3. VGG19.....	15
4.5. Tecnologías	16

4.5.1.	Teachable Machine.....	16
4.5.2.	Pytorch.....	17
4.6.	Parámetros de evaluación	17
4.6.1.	La curva ROC.....	17
4.6.2.	Matriz de confusión.....	18
4.6.3.	Métricas de rendimiento	19
5.	Metodología.....	21
5.1.	Área de estudio	21
5.2.	Base de Datos.....	22
5.2.1.	Información de las características de las plantas	22
5.2.2.	Adquisición de las imágenes	22
5.3.	Modelos de entrenamiento	23
5.3.1.	Primer algoritmo (MobileNet).....	23
5.3.2.	Segundo Algoritmo (ResNet)	25
5.3.3.	Tercer algoritmo (VGG19).....	27
5.4.	Desarrollo de la Aplicación	29
6.	Resultados	36
6.1.	Obtención de la base de datos	36
6.2.	Modelo MobileNet.....	37
6.3.	Modelo ResNet	39
6.4.	Modelo VGG19	40
6.5.	Modelo final.....	41
7.	Discusión	45
8.	Conclusiones	47
9.	Recomendaciones	48
10.	Bibliografía	49
11.	Anexos	52

Índice de tablas

Tabla 1. Recursos de la red por tipo de capa.....	14
Tabla 2. Métricas para la evaluación de rendimiento de un modelo de clasificación.	19
Tabla 3. Parámetros de Entrenamiento del primer algoritmo	24
Tabla 4. Parámetros de Entrenamiento del segundo algoritmo.....	25
Tabla 5. Librerías utilizadas para el desarrollo de la aplicación	31
Tabla 6. Distribución de la base de datos por planta.....	36
Tabla 7. Resultados de los parámetros de evaluación del primer algoritmo.....	38
Tabla 8. Resultados de los parámetros de evaluación del primer algoritmo.....	40
Tabla 9. Resultados de los parámetros de evaluación del primer algoritmo.....	41
Tabla 10. Resultados obtenidos de las pruebas realizadas con el algoritmo final.....	42
Tabla 11. Resultados de los parámetros de evaluación del algoritmo final	44

Índice de figuras

Figura 1. Arquitectura típica de una red neuronal.....	11
Figura 2. Ejemplo de arquitectura de una CNN para la clasificación de imágenes	12
Figura 3. Capas que componen a la arquitectura de MobileNet. A la izquierda se presenta las capas que componen una red neuronal típica y a la derecha se presenta las capas que componen la red de MobileNet.....	13
Figura 4. Construcción del bloque de una red de aprendizaje residual.....	15
Figura 5. Configuración de la red VGG.....	16
Figura 6. Ejemplo de una curva ROC, donde el eje y es la sensibilidad y el eje x la especificidad.....	18
Figura 7. Ejemplo de una Matriz de confusión	19
Figura 8. Área del Jardín Botánico Reinaldo Espinoza.....	21
Figura 9. Diagrama del proceso de entrenamiento de Teachable Machine.....	23
Figura 10. Interfaz de la herramienta de Teachable Machine	24
Figura 11. Diagrama del proceso de entrenamiento de ResNet	25
Figura 12. Comandos para la selección del dispositivo de ejecución	26
Figura 13. Comandos para la configuración del algoritmo	26
Figura 14. Comandos para el ingreso de los datos de entrenamiento y validación.....	27
Figura 15. Comandos para la exportación del modelo con Pytorch.....	27
Figura 16. Diagrama del proceso de entrenamiento de VGG-19	28
Figura 17. Comandos para establecer los parámetros de entrada a la red neuronal... ..	28
Figura 18. Comandos para la configuración de la red.....	28
Figura 19. Comandos para la exportación del modelo con Pytorch.....	29
Figura 20. Ejecución del comando flutter doctor en la consola de Windows	29
Figura 21. Activación de las opciones de desarrollador en Android.....	30
Figura 22. Activación de la depuración por USB e instalación vía USB.....	31
Figura 23. Función del botón de cámara que permite importar imagen desde la galería o desde la cámara.....	33
Figura 24. Elementos de la interfaz de la aplicación.....	34
Figura 25. Interfaz de la ventana secundaria, donde como ejemplo se muestra la información etimológica del nombre de la planta	35
Figura 26. Distribución de la base de datos según el uso dado a las imágenes.....	36
Figura 27. Gráficas que representan valores generados durante el entrenamiento.	37

Figura 28. Matriz de Confusión del primer algoritmo	38
Figura 29 Gráficas que representan valores generados durante el entrenamiento del segundo algoritmo	39
Figura 30. Matriz de Confusión del segundo algoritmo.....	39
Figura 31. Gráficas que representan valores generados durante el entrenamiento del tercer algoritmo	40
Figura 32. Matriz de Confusión del tercer algoritmo	41
Figura 33. Gráficas que representan valores generados durante el entrenamiento del algoritmo final	42
Figura 34. Matriz de Confusión del algoritmo final.....	43

Índice de anexos

Anexo 1. Características de las plantas obtenidas del Jardín Botánico	52
Anexo 2. Certificado de traducción del resumen	68

1. Título

Desarrollo de un Sistema de Reconocimiento de Especies Vegetales para el Jardín Botánico de la UNL Basado en Inteligencia Artificial

2. Resumen

La inteligencia artificial ha experimentado un destacado avance en los últimos años, encontrando aplicaciones en diversas áreas para automatizar y optimizar tareas específicas. En el contexto de este proyecto, se enfoca en la aplicación de inteligencia artificial en el ámbito de la botánica, específicamente para la identificación de especies vegetales en el Jardín Botánico Reinaldo Espinosa de la ciudad de Loja. Este trabajo implica el desarrollo de tres algoritmos de redes neuronales, basados en las arquitecturas predefinidas MobileNet, VGG19 y ResNet18. Para acelerar el proceso de entrenamiento, se ha utilizado una base de datos reducida, utilizando aproximadamente el 12% de la base de datos original. Se obtuvieron distintos niveles de precisión para cada uno de los algoritmos: MobileNet alcanzó un 94,83%, VGG19 un 62,25%, y ResNet18 un 60,42%. Luego de evaluar los resultados, se seleccionó la arquitectura MobileNet por presentar un rendimiento superior comparado con los demás. Posteriormente, MobileNet fue sometida a un entrenamiento completo con la base de datos original, que consta de 1354 imágenes de 20 especies vegetales distintas, logrando una precisión final del 88,59%. Como parte de la implementación práctica, se desarrolló una aplicación para Android utilizando el Framework de Flutter, permitiendo la utilización del algoritmo de identificación de especies vegetales directamente desde dispositivos móviles.

Palabras clave: *Redes Neuronales, VGG19, MobileNet, ResNet, Flutter, Identificación Especies de Plantas*

Abstract

Artificial intelligence has experienced an outstanding advance in recent years, finding applications in various areas to automate and optimize specific tasks. In the context of this project, the focus is on the application of artificial intelligence in the field of botany, specifically for the identification of plant species in the Reinaldo Espinosa Botanical Garden in the city of Loja. This work involves the development of three neural network algorithms, based on the predefined architectures MobileNet, VGG19 and ResNet18. In order to speed up the training process, a reduced database has been used, using approximately 12% of the original database. Different levels of accuracy for each of the algorithms were obtained: MobileNet achieved 94.83%, VGG19 62.25%, and ResNet18 60.42%. After evaluating the results, the MobileNet architecture was selected for its superior performance compared to the others. Subsequently, MobileNet was subjected to a complete training with the original database, which consists of 1354 images of 20 different plant species, achieving a final accuracy of 88.59%. As part of the practical implementation, an Android application was developed using the Flutter Framework, allowing the use of the algorithm for plant identification directly from mobile devices.

Keywords: *Neural Networks, VGG19, MobileNet, ResNet, Flutter, Plant Species Identification*

3. Introducción

Los jardines botánicos desempeñan un papel crucial en la preservación de la biodiversidad y se consideran espacios fundamentales en este sentido. Estos lugares se convierten en verdaderos laboratorios vivos donde se investigan y estudian las distintas especies de flora, al mismo tiempo que se protegen a aquellas que se encuentran en peligro. Además, los jardines botánicos han evolucionado para convertirse en centros educativos, brindando a las personas la oportunidad de aprender sobre la importancia de la conservación y el papel vital que desempeñan las plantas en nuestro entorno (Vovides et al., 2013).

El reconocimiento manual de especies vegetales en el Jardín Botánico de la UNL puede ser un proceso lento y propenso a errores, lo que dificulta la catalogación precisa de la diversidad vegetal presente. Además, la falta de una herramienta automatizada para el reconocimiento de especies impide el acceso rápido a información detallada sobre las plantas, limitando la divulgación científica y la gestión eficiente del jardín. La identificación manual no es escalable y puede resultar difícil para aquellos que no tienen conocimientos botánicos especializados. Con el uso de un sistema de reconocimiento de la flora del jardín botánico mediante inteligencia artificial, permitirá de manera automática identificar la especie a la que pertenezca una planta en específico, contribuyendo directamente al campo científico de la biodiversidad, facilitando a los científicos la identificación de especies de forma rápida y precisa, especialmente en zonas remotas y poco accesibles. Por otro lado, esto también beneficiará a las personas que no tengan un gran conocimiento en la botánica como agricultores o estudiantes, facilitando la comprensión de la biodiversidad local.

Para el desarrollo de este proyecto se tomará en cuenta el diseño, desarrollo e implementación de un sistema de identificación de especies de plantas basado en IA específicamente adaptado para el jardín botánico en cuestión. Esto implica la selección y entrenamiento de algoritmos de IA adecuados, la recopilación y preparación de datos de muestra, la configuración del sistema, y la validación de su precisión y eficiencia. Sin embargo, debido a la gran cantidad de especies presentes, se tomará únicamente un número determinado para el desarrollo del prototipo. Además, se llegará a considerar la evaluación de los resultados obtenidos por el sistema de identificación de especies de plantas. Esto implica realizar pruebas para determinar la precisión y confiabilidad del sistema.

Para el desarrollo óptimo del trabajo de titulación, se ha considerado los siguientes objetivos:

❖ Objetivo General

- Desarrollar e implementar un sistema de reconocimiento de especies vegetales en el Jardín Botánico de la UNL utilizando técnicas de inteligencia artificial

❖ Objetivos específicos

- Recopilar y construir una base de datos que contenga imágenes de las especies vegetales presentes en el Jardín Botánico de la UNL, junto con su correspondiente información taxonómica y características relevantes.
- Estudiar a profundidad las técnicas de IA y el software relacionado con la finalidad de determinar la plataforma tecnológica adecuada para el desarrollo de un algoritmo capaz de realizar la extracción de características y la identificación precisa de las especies vegetales con ayuda de la base de datos.
- Verificar y analizar el rendimiento del sistema con el fin de determinar su viabilidad y eficiencia

4. Marco Teórico

4.1. Antecedentes

El uso de la IA en el reconocimiento de especies vegetales representa un hito importante dentro del área de la biología, ya que brinda la capacidad de catalogar y comprender la biodiversidad de manera eficiente y precisa. Este mecanismo puede llegar a agilizar los procesos de identificación, además de presentarse como una herramienta valiosa para la investigación, educación y la conservación. En esta sección de antecedentes, exploraremos diversos mecanismos de inteligencia artificial aplicados a la clasificación de las especies vegetales.

Una de las aplicaciones más comunes para la clasificación de imágenes orientadas a las especies vegetales, es la identificación de enfermedades. Tal como se presenta en (Ai y otros, 2020), cuyo objetivo fue la detección temprana de posibles plagas que azotan a los cultivos con el fin de reducir eficazmente las pérdidas económicas. Para ello se usó la base de datos de AI Challenger 2018, que cuenta con 47363 imágenes de 27 enfermedades de 10 cultivos comunes. Para el entrenamiento se hizo uso de la arquitectura Inception-ResNet-v2 logrando alcanzar una precisión general del 86.1%.

En (Quoc y otros, 2019) también se propone un método de identificación temprana de enfermedades. Se hace uso de dos bases de datos, *Swedish leaf dataset* con un total de 1125 imágenes y *Flavia dataset* con 1907 imágenes en total. Para el entrenamiento del modelo se presentan dos métodos propios. El primer método es una combinación del Histograma Orientado a Gradiente (HOG) y Máquina de Vectores de Soporte (SVM), logrando obtener una precisión del 92%. El segundo método está basado en una Red Neuronal Convolutiva (CNN) compuesta con un total de 6 capas, logrando alcanzar una precisión del 95.2%.

Otro método para la detección de enfermedades en imágenes se presenta en (Ashwinkumar y otros, 2022), donde se presenta una red neuronal convolutiva óptima basada en una red móvil (OMNCNN) utilizando filtrado bilateral (BF) y una segmentación de imágenes basada en umbrales Kapur que facilita la identificación de las regiones afectadas de una hoja de la planta. Para la extracción de características, se hace uso de la arquitectura de MobileNet. Finalmente, con un clasificador basado en aprendizaje de máquina extremo, se asignan etiquetas correspondientes a cada imagen. La precisión máxima obtenida en este algoritmo es del 98,7%.

(Kumar y otros, 2019) propuso un sistema de detección de especies vegetales basándose principalmente en la extracción de características morfológicas enfocadas principalmente en las hojas de las plantas, ya que estas suelen ser más accesibles en

comparación con otras partes. Entre las características utilizadas se puede encontrar la longitud del eje mayor y del eje menor, solidez, perímetro y orientación, centroide, etc. Para el desarrollo del modelo se utilizó una base de datos pública que cuenta con 1907 imágenes y se hizo uso de diferentes clasificadores: k-NN (vecinos más próximos), árbol de decisión y perceptrón multicapa. Además, empleó la metodología de AdaBoost alcanzando una precisión del 94,42%.

4.2. Visión por computador

La visión por computadora puede ser considerada como un conjunto de conceptos que tienen como finalidad la replicación del sistema visual humano, donde toma en cuenta las funcionalidades de una cámara, las proyecciones y la fotogrametría (Kakani et al., 2020). Como humanos tenemos una manera particular de percibir nuestro alrededor, al poseer dos ojos nos permite visualizar nuestro medio en tres dimensiones con aparente facilidad. Así mismo, somos capaces de procesar lo que vemos, identificando los objetos que entran en nuestro campo de visión, como el caso una flor, en la que podemos llegar a observar los patrones sutiles de sus pétalos.

Además, somos capaces de encontrar las diferencias en los rasgos faciales de cada persona e identificar sus emociones por medio de los gestos que estos llegan a exhibir. En la visión por computadora, se intenta digitalizar al mundo que vemos a través de una o más imágenes mediante la reconstrucción de sus propiedades, la forma, la iluminación y la distribución de colores (Szeliski, 2022),

4.2.1. Adquisición de imágenes

La adquisición de las imágenes es el primer paso a realizar para la detección de algún elemento en particular usando la visión por computadora. La calidad de las mismas determinará el rendimiento del sistema, debido a que, si las imágenes disponen de un gran tamaño, el tiempo necesario para procesar cada una será mayor (Yang et al., 2021). Algunos dispositivos de adquisición de imágenes pueden llegar a ser:

- Cámaras monoculares 2D
- Cámaras monoculares 3D
- Cámaras de infrarrojo cercano
- Sensores de luz estructurados

4.2.2. Preprocesamiento

Al momento de haber adquirido la imagen, el siguiente paso necesario es el preprocesamiento, este está compuesto por un conjunto de técnicas cuyo fin es mejorar la

extracción de la información relevante de la imagen. Estas técnicas pueden ayudar a la reducción del ruido de la imagen y mejorar la calidad de la misma, facilitando al algoritmo la extracción de las características más relevantes (Castro, 2018). Según (Yang et al., 2021) las técnicas más utilizadas en el preprocesamiento son:

- Conversión a escala de grises. Convertir la imagen a escala de grises incrementa la velocidad de procesamiento y se logra detectar la información más fácilmente
- Removedor de ruido. El ruido afecta el rendimiento general del sistema por lo que es necesario el uso de un filtro para mantener únicamente la información relevante
- Recorte y cambio de tamaño. Se puede recortar la imagen para eliminar la información redundante del fondo. Además, se fija el tamaño a uno fijo para evitar posibles errores.
- Adición de datos. Es posible que a la imagen sea necesario voltearla, desenfocarla o modificar la intensidad de luz para mejorar su interpretación
- Reducción de dimensionalidad. Se puede reducir las dimensiones de sus características aplicando técnicas de PCA y LDA.

4.2.3. Segmentación

La segmentación es el tercer paso de la visión por computadora, consiste en dividir la imagen en zonas disjuntas e individualizadas. El objetivo de esta etapa es mejorar la representación de una imagen haciéndola más simple y facilitando su análisis. Desde otro punto de vista se puede definir como etiquetar cada píxel relevante de la imagen, de modo que la agrupación de varias etiquetas representa un objeto en particular dentro de la imagen (Guerrero, 2015).

4.2.4. Extracción de características

Las características se pueden definir como propiedades individuales y relevantes particulares que describen y definen un objeto. Dentro del procesamiento de imágenes es recomendable trabajar con las características ya que permiten la codificación de información de alto nivel. Podemos encontrar características locales y globales, las primeras se centran en puntos clave, representando una pequeña área de la imagen. Las globales por otro lado, definen a la imagen en su totalidad, tomando en cuenta todos sus píxeles, consideraras como un atributo particular de la imagen (Alcántara et al., 2019).

4.3. Inteligencia Artificial

La inteligencia no es algo propio de los humanos ni los animales, según (Mathivet, 2018) todo sistema, ya sea un ser vivo o no, capaz de adaptarse y responder a su entorno en base a la interacción entre ambos, puede llegar a considerarse como algo inteligente. El término de inteligencia artificial data desde la década de 1950 y gracias al avance en las ciencias computacionales de los últimos años, esta ha ido avanzando progresivamente, hasta llegar a convertirse en una herramienta a la que casi cualquier persona tiene acceso.

Sin embargo, (Mathivet, 2018) también detalla que no todo lo relacionado con las computadoras puede llegar a considerarse como un sistema inteligente. Es difícil determinar cuándo un sistema es inteligente, ya que la propia palabra tiene muchas interpretaciones. Por eso, se puede definir a la inteligencia artificial como algo que llegue a imitar el comportamiento humano o sea capaz de implementar estrategias flexibles para la resolución de problemas que la programación clásica no es capaz de realizar, en otras palabras, que dé la impresión de que posee cierta adaptabilidad. Adicionalmente (Mathivet, 2018) nos presenta dos grandes corrientes de la inteligencia artificial

- **Enfoque simbólico:** el sistema es capaz de realizar una descripción lo más detallada posible del entorno, así como de las leyes que son aplicadas, de tal manera que es capaz de seleccionar la mejor decisión posible. Se suele utilizar en sistemas expertos o de lógica difusa.
- **Enfoque conexionista:** se da las herramientas necesarias al sistema para determinar si las acciones realizadas son correctas o no y de la misma manera se le incita a encontrar las soluciones por sí mismo. Se encuentra presente en las redes neuronales o sistemas multi-agentes

La inteligencia artificial posee una amplia gama de aplicaciones, por lo que presenta una gran variedad de técnicas

4.3.1. *Machine Learning*

Conocido también como ‘aprendizaje automático’ en español, es una de las principales corrientes utilizadas en el campo de la inteligencia artificial. Se trata de un aspecto de la informática en el cual las máquinas son capaces de adquirir conocimiento y mejorar su rendimiento sin la necesidad de ser programadas de forma explícita para cada tarea. El aprendizaje de máquina se basa en el uso de algoritmos diseñados para analizar datos y, de esta manera, identificar patrones y tendencias (Rouhiainen, 2018).

A través del Machine Learning (ML), las máquinas tienen la capacidad de identificar y comprender relaciones complejas en los datos, lo que les permite tomar decisiones, realizar predicciones y llevar a cabo tareas específicas con mayor precisión y eficiencia. Este enfoque se ha vuelto cada vez más relevante en el campo de la inteligencia artificial, impulsado por la disponibilidad de enormes volúmenes de datos en la actualidad y el poder de procesamiento de las computadoras modernas (Bonaccorso, 2017). Existen 3 enfoques en los que se basa el aprendizaje de máquina:

6.3.1.1. Aprendizaje supervisado. Es un escenario caracterizado por el concepto de profesor o supervisor, el cual se basa en etiquetar los datos mostrando que salidas se llegan a obtener de las posibles entradas. Al empezar con esta información, el algoritmo puede ajustar y corregir sus parámetros para disminuir las probabilidades de errores.

6.3.1.2. Aprendizaje no supervisado. Este algoritmo no necesita intervención humana, ya que presenta un escenario donde no se cuenta con información de partida, por lo que la propia máquina debe aprender cómo realizar la agrupación de los datos de acuerdo a sus similitudes o algún otro parámetro de clasificación.

6.3.1.3. Aprendizaje por refuerzo. Se fundamenta en la retroalimentación y se aplica en escenarios donde la información tiene un carácter predominantemente cualitativo. Estos algoritmos se caracterizan por requerir una mayor participación humana, ya que aprenden a partir de la experiencia. Se emplean dos elementos esenciales: la recompensa y la penalización. En caso de obtener una salida acorde a lo esperado, se otorga a la máquina una recompensa, mientras que, si la salida es errónea, se le impone una penalización. De tal manera que la forma en la que aprender se asemeja al entrenamiento de una mascota.

4.3.2. *Redes Neuronales Artificiales*

Es un sistema computacional formado por nodos interconectados que imitan a las neuronas y pueden interactuar, transmitir y procesar información en ciclos computacionales, generando uno o más datos reales como resultado. No obstante, la efectividad y exactitud de este sistema dependen de varios factores, como el entrenamiento de la red, su arquitectura, la cantidad de datos, entre otros. Para entrenar la red, es esencial emplear algoritmos de entrenamiento, como backpropagation (que busca replicar el funcionamiento de redes neuronales biológicas) y feedforward (redes neuronales sin ciclos), los cuales pueden proporcionar una alta precisión (Brayan et al., 2020).

La idea general de una red neuronal artificial es que las neuronas trabajen juntas para resolver un problema y se mantengan en constante comunicación. Estas a su vez están

organizadas en capas, la capa de entrada se encarga de la recepción de la información, mientras la capa de salida genera el resultado final. Adicionalmente existe una o más capas intermedias las cuales son las encargadas de realizar la predicción o el aprendizaje (Dastres y Soori, 2021). La arquitectura típica de una red neuronal se presenta en la **Figura 1**.

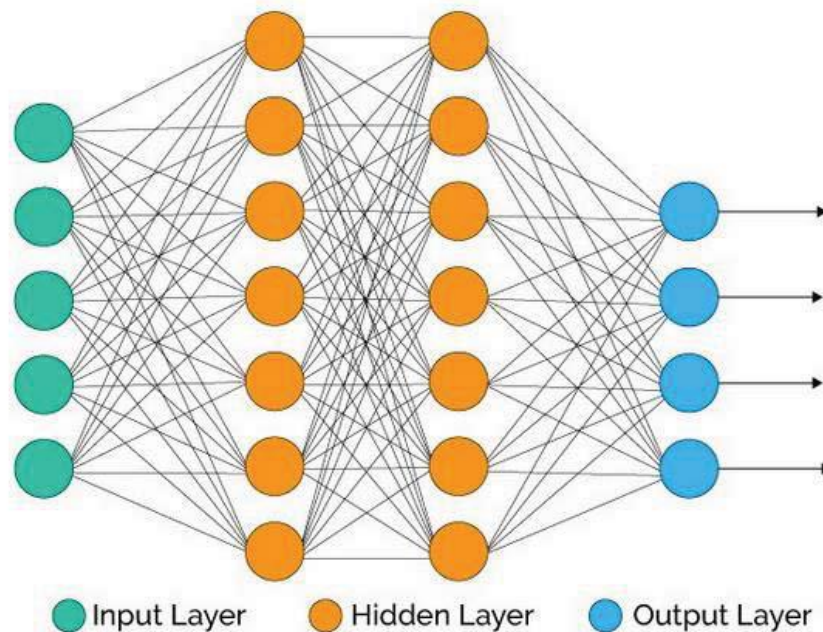


Figura 1. Arquitectura típica de una red neuronal

Fuente: (Dastres y Soori, 2021)

4.3.2.1. Redes Neuronales convolucionales. Las redes neuronales convolucionales (CNN, del inglés Convolution Neural Network) fueron diseñadas con el objetivo de trabajar con estradas estructuradas en cuadrícula, la más común son las imágenes 2D. Las entradas también presentan dependencias espaciales, donde cada localización espacial adyacente presenta valores de color similares a los píxeles individuales. Con ello se crea una tercera dimensión en los datos, la cual se encarga de capturar cada uno de los colores de las entradas. De las características más importantes de este tipo de redes, se puede considerar la conocida como convolución, que es un producto punto entre un conjunto de pesos (parámetros ajustables de entrenamiento de la red) y entradas, ambos valores estructurados en cuadrícula (Aggarwal, 2018).

Las CNNs son una de las técnicas más frecuentemente empleadas en el ámbito del aprendizaje profundo. Se destacan por su capacidad para identificar de manera automática las características más relevantes de la entrada, sin la necesidad de la supervisión humana. Estas redes poseen aplicaciones en diversos campos como la visión por computadora, el

reconocimiento de voz, reconocimiento facial, entre otros (Alzubaidi et al., 2021). La manera en la que estructura una CNN se presenta en la **Figura 2**.

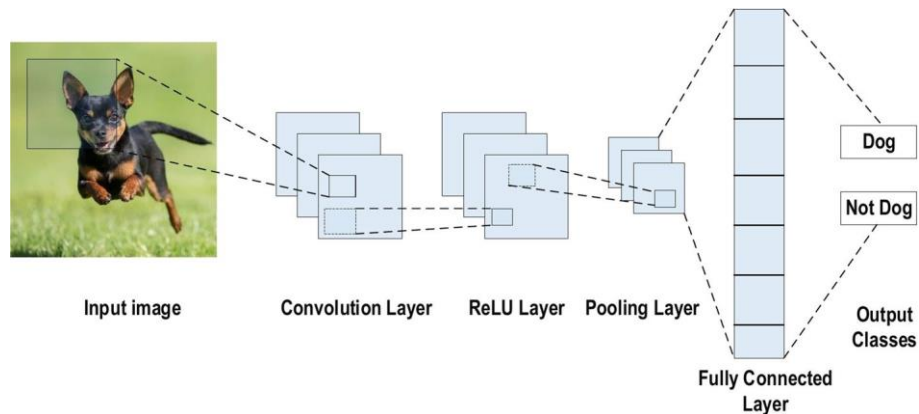


Figura 2. Ejemplo de arquitectura de una CNN para la clasificación de imágenes

Fuente: (Alzubaidi et al., 2021)

Según (Alzubaidi et al., 2021) una CNN está compuesta por diversas capas, en las cuales podemos destacar:

- **Convolution Layer (Capa convolucional).** Es la capa más importante de la red, consiste en un conjunto de filtros convolucionales, los cuales convierten a una entrada de N-dimensiones en un mapa de características de salida.
- **Pooling Layer (Capa de agrupación).** Es la encargada de reducir el tamaño del mapa de características, generando mapas más pequeños. Hay varios métodos de agrupación que se pueden utilizar como la agrupación de árboles, la agrupación cerrada, la agrupación promedio, la agrupación mínima, entre otros. Esta capa puede tener un impacto en el rendimiento global de la CNN, ya que no solamente contribuye a identificar si una característica está presente o ausente en una imagen particular, sino que también prioriza la tarea de definir la ubicación precisa de dicha característica. Esto puede dar lugar a que la información relevante en otras áreas se vea omitida.
- **Función de activación.** Es una operación matemática aplicada a la salida de cada neurona para introducir no linealidad al modelo, permitiendo a la red aprender patrones y relaciones más complejas en los datos. ReLU es la función de activación más común, convierte los valores enteros de la entrada en números positivos. Su baja demanda computacional es una de sus principales ventajas, su representación matemática está dada en la **Ecuación 1**. Se puede encontrar otras funciones tales como Tanh, Sigmoid, Noisy ReLU, Leaky ReLU, etc.

$$f(x)_{ReLU} = \max(0, x) \quad (1)$$

- **Fully Connected Layer (Capa completamente conectada).** Comúnmente está ubicada en la parte final de una CNN. Cada una de las neuronas de esta capa está conectada a todas las neuronas de la capa anterior. Se encarga de realizar la clasificación y la entrada tiene la forma de un vector creado a partir de los mapas de características. La salida de la capa es la salida de toda la red.
- **Función de pérdida.** Adicionalmente se utilizan funciones de pérdida en la capa de salida de la red con el objetivo de calcular el error previsto durante el entrenamiento. Para ello utiliza dos parámetros para su cálculo, la salida estimada y la salida real.

4.4.Arquitecturas

4.4.1. MobileNet

La estructura de MobileNet está basada en convoluciones separables en profundidad, a excepción de la primera capa la cual es una convolución completa. Cada una de las capas va seguida de una función batchnorm (BN), que se encarga de normalizar la entrada de cada capa oculta de una red neuronal, y por una función de no linealidad ReLU a excepción de la capa final completamente conectada. La misma que se encargada de alimentar una capa softmax en la clasificación (Howard y otros, 2017). En la **Figura 3** se presenta como se diferencia una red neuronal convolucional regular.

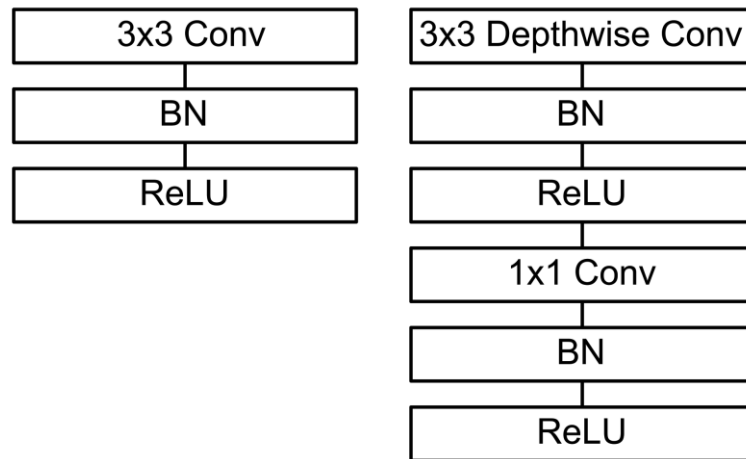


Figura 3. Capas que componen a la arquitectura de MobileNet. A la izquierda se presenta las capas que componen una red neuronal típica y a la derecha se presenta las capas que componen la red de MobileNet

Fuente: Howard y otros, 2017

Este modelo de red está compuesto principalmente por convoluciones densas de 1x1, gracias a la implementación de funciones de multiplicación de matrices generales (GEMM, del inglés general matrix multiply) con una alta optimización. Las convoluciones de 1x1 no requieren un reordenamiento en la memoria, a diferencia de las regulares, y pueden ser

implementadas directamente en la GEMM. En MobileNet el 95% de su tiempo está dedicado al cálculo de las convoluciones 1x1, que componen el 75% de los parámetros. Mientras el resto de ellos están casi completamente en la capa completamente conectada (Howard y otros, 2017). Tal como se observa en la **Tabla 1**. Adicionalmente, la versión 2 de MobileNet fue presentada en 2018 e incrementó el número de capas de la red de 28 a 53.

Tabla 1. Recursos de la red por tipo de capa

Tipo	Tiempo de cómputo	Parámetros
Conv 1x1	94,86%	74,59%
Conv DW 3x3	3,06%	1,06%
Conv 3x3	1,19%	0,02%
Capa totalmente conectada	0,18%	24,33%

4.4.2. ResNet

Es una red neuronal desarrollada por un equipo de Microsoft para el Desafío de reconocimiento visual a gran escala de ImageNet (ILSVRC, ImageNet Large Scale Visual Recognition Challenge) del año 2015, en el cual fue el ganador. El objetivo de esta red fue evitar el problema de degradación que se presenta al aumentar el número de capas de una red neuronal, lo que llega a provocar una degradación del entrenamiento, que aumenta al agregar un número mayor de capas. Para resolver propone un aprendizaje residual, donde el mapeo subyacente de la red está denotado por $H(x)$, siendo x la representación de la entrada de la red (He y otros, 2016).

El objetivo de esta red es cambiar el enfoque de aprendizaje, donde las capas no lineales apiladas en vez de aprender directamente de $H(x)$, aprendan la diferencia entre la salida deseada y la entrada, presentando un nuevo mapeo presentado como $F(x) = H(x) - x$. Este proceso puede ser realizado mediante conexiones directas o también llamados atajos, las que permite realizar una copia de la entrada original x y sumarla a la salida de las capas apiladas $F(x)$ como se observa en la **Figura 4**. Adicionalmente estas conexiones no suponen una mayor carga computacional ni parámetros adicionales (He y otros, 2016).

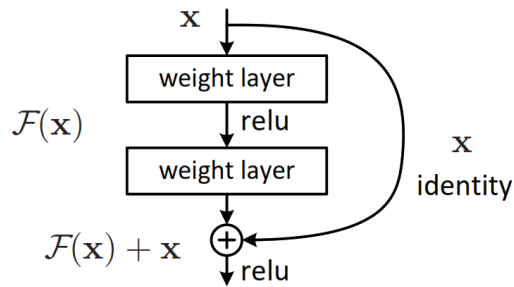


Figura 4. Construcción del bloque de una red de aprendizaje residual

Fuente: He y otros, 2016

(He y otros, 2016) también han realizado pruebas de arquitecturas de redes simples/residuales que se presentan en la y están descritas a continuación

Red simple. Está basado en la arquitectura VGG. Las capas convolucionales están conformadas principalmente por filtros 3x3 y se establece dos reglas en su diseño. La primera establece que, para los mapas de características del mismo tamaño, las capas poseen la misma cantidad de filtros. En la segunda regla nos dice que cuando el tamaño del mapa de características es reducido a la mitad, la cantidad de filtros se duplica permitiendo que la complejidad temporal de cada capa se mantenga. Al final de la red, hay una capa de agrupación promedio global y una capa totalmente conectada.

Red residual. Es la contraparte residual de la red simple mediante el uso de conexiones directas. Cuando la entrada y la salida de la capa son las mismas se utilizan los atajos de identidad. En cambio, cuando nos encontramos que las dimensiones no son las mismas, podemos encontrar dos escenarios, podemos usar el atajo de identidad con la diferencia de que se añaden ceros adicionales para igualar la dimensión de ambas capas o hacer uso del atajo de proyección para hacer coincidir las dimensiones. En ambos casos las conexiones directas se realizan saltando dos capas.

4.4.3. VGG19

Al igual que ResNet, esta arquitectura también fue presentada en el ILSVRC del 2014 logrando obtener el segundo lugar. La entrada para esta red son imágenes de tamaño fijo de 224 x 224 píxeles y no es necesario un preprocesamiento complejo, ya que únicamente se resta el valor RGB medio. Esto ayuda a centrar los datos alrededor de cero y a mejorar el rendimiento del modelo durante el proceso de aprendizaje. Una vez que la imagen ingresa dentro de la red, esta pasa a través de capas convolucionales donde se utilizan filtros muy pequeños de campo receptivo con una dimensión de 3x3. Adicionalmente también se puede llegar a utilizar filtros 1x1 que actúan como transformadores lineales para el canal de entrada, para luego hacer uso de una función de no linealidad, generalmente la función ReLU (Simonyan, 2014).

El paso de convolución se fija en 1 píxel. Esto significa que el filtro se desplaza una posición (un píxel) a la vez mientras se realiza la convolución. Después de las capas de convolución se aplican 5 capas de máxima agrupación (no todas las capas de convolución van seguidas de capas de agrupación), para la agrupación máxima es necesario realizarla en una ventana de 2x2 píxeles, con un desplazamiento de 2 píxeles a la vez. Al final de todas las capas de convolución, nos encontramos con tres capas totalmente conectadas, dos de las cuales poseen un total de 4096 canales cada una mientras que la tercera posee 1000 canales. Al final de la red nos encontramos con la capa Softmax. La configuración de la capa se presenta en la **Figura 5**.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figura 5. Configuración de la red VGG.

De izquierda a derecha (A-B) incrementa el número de capas. El número de capas convolucionales en cada pila inicia en 64 e incrementa el doble luego de una capa de máxima agrupación hasta un máximo de 512

Fuente: Simonyan, 2014

4.5.Tecnologías

4.5.1. Teachable Machine

Es una herramienta desarrollada por Google para brindar una opción fácil de aprendizaje de máquina. Permite entrenar modelos propios utilizando la cámara web, imágenes o sonidos;

generando un modelo de clasificación en cuestión de segundos. Teachable Machine está desarrollado a partir del Framework TensorFlow y surgió como una herramienta para facilitar el uso de la inteligencia artificial accesible a todo público (Forchhammer et al., 2022).

Esta herramienta utiliza aprendizaje por transferencia que permite utilizar una base de datos propia y volver a entrenar el modelo sobre un modelo pre-entrenado. Para el caso de la clasificación de imágenes, se utilizó al modelo MobileNet como base y se entrenó para reconocer hasta 1000 clases diferentes. Estos permiten mejorar la identificación de nuevas clases mejorando la capacidad de identificación y reducir considerablemente el tiempo necesario para el entrenamiento (Carney et al., 2020).

4.5.2. Pytorch

Es un framework de código abierto destinado al Machine Learning. Proporciona una ejecución basada en gráficos contando con una fuerte aceleración con la GPU, convirtiéndose en una opción asequible para el desarrollo de redes neuronales artificiales. Desarrollado por Facebook AI Research, Pytorch ha sido utilizado en importantes proyectos de inteligencia artificial como el Autopilot de Tesla o Pyro de Uber (Martel, 2020). Una de sus principales características es el uso de Tensores, objetos matemáticos que almacenan valores numéricos, lo que le permite trabajar con datos multidimensionales. Lo mismo que puede ocasionar que se requiera cierta potencia para un rendimiento adecuado, pero esto se puede solventar utilizando recursos en la nube (De Luca, D, s.f.).

4.6. Parámetros de evaluación

4.6.1. La curva ROC

Es una herramienta utilizada para representaciones gráficas en estadística y en aprendizaje automático con el fin de realizar una evaluación de un modelo de clasificación en diversos umbrales de decisión. Del Valle (2017) establece que: “La curva ROC poblacional representa 1-especificidad frente a la sensibilidad para cada posible valor umbral o punto de corte en la escala de resultados de la prueba en estudio”. La misma que puede ser definida por la Ecuación 2.

$$ROC(c) = \begin{cases} y = S(c) \\ x = 1 - E(c) \end{cases} \quad (1)$$

Sin embargo, debido a que la obtención de datos poblacionales precisos resulta de gran dificultad, se puede hacer uso de la curva ROC muestral para la obtención de un resultado aproximado. Este resultado presenta a la fracción de los falsos positivos en el eje de las abscisas

contra la fracción de verdaderos positivos que se encuentran representados en el eje de las ordenadas. Esto viene presentado por la Ecuación 3.

$$ROC_p(c) = \begin{cases} FVP(c) \\ FFP(c) \end{cases} \quad (2)$$

Este tipo de gráfica es de suma importancia en el análisis de algoritmos de clasificación de imágenes, ya que facilita la evaluación de la separación entre las distribuciones de sensibilidad y especificidad de un test. Esta gráfica ofrece varias ventajas significativas, como su facilidad de interpretación al mostrar visualmente la relación entre la sensibilidad y la tasa de falsos positivos, lo que brinda una visión clara del desempeño del modelo en distintos puntos de corte, como se observa en la **Figura 6**. Además, no requiere la elección de un umbral de decisión específico, ya que exhibe el rendimiento del modelo a lo largo de todo el rango de posibles umbrales de clasificación. Asimismo, presenta independencia de la prevalencia, ya que no se ve afectada por la proporción de la población total que posee la condición en cuestión (Torres, 2010).

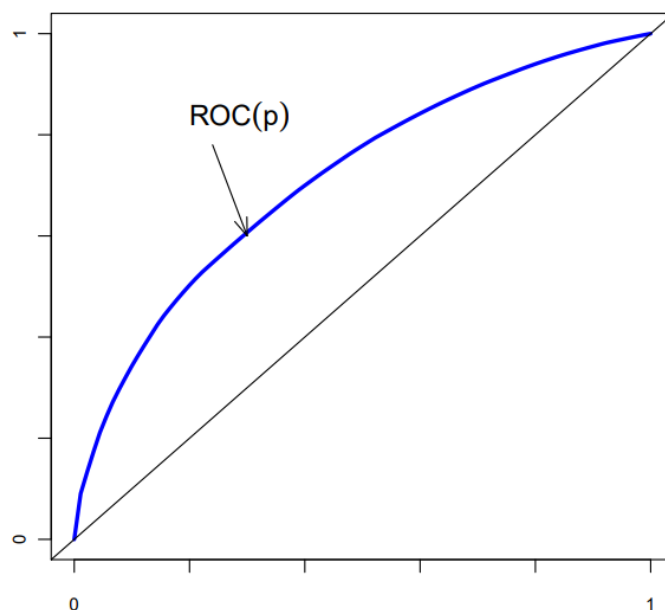


Figura 6. Ejemplo de una curva ROC, donde el eje y es la sensibilidad y el eje x la especificidad

Fuente: Torres, 2010

4.6.2. *Matriz de confusión*

Es una herramienta utilizada para evaluar el rendimiento de un algoritmo de clasificación. En esta matriz, cada columna viene a representar las clases reales, mientras las filas vienen a representar las clases predichas por el algoritmo. En otras palabras, permite identificar el número de aciertos y errores que presenta el algoritmo al evaluar su rendimiento, como se presenta en la **Figura 7**. Adicionalmente dentro de la matriz tendremos un valor binario que llegará a representar 4 posibles escenarios (Godoy, 2021). Estos escenarios son:

- **Verdadero Positivo (TP):** Es el número de valores positivos clasificados correctamente por el modelo
- **Verdadero Negativo (TN):** Es el número de valores negativos clasificados correctamente por el modelo
- **Falso Positivo (FP):** Es el número de valores positivos clasificados incorrectamente por el modelo
- **Falso Negativo (FN):** Es el número de valores negativos clasificados incorrectamente por el modelo

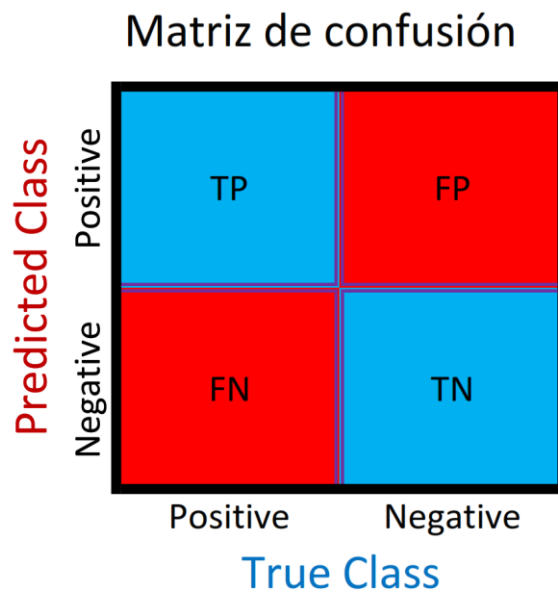


Figura 7. Ejemplo de una Matriz de confusión
Fuente: Elaboración propia

4.6.3. Métricas de rendimiento

En un modelo de clasificación, es esencial cuantificar la información recopilada para facilitar la evaluación de su rendimiento. Esto implica el uso de métricas con el objetivo de identificar el algoritmo que presente el mejor desempeño según las necesidades de nuestra investigación. Estas métricas son descritas en la **Tabla 2**.

Tabla 2. Métricas para la evaluación de rendimiento de un modelo de clasificación

Métrica	Descripción	Fórmula
Exactitud	Es la proporción de elementos predichos de manera satisfactoria en relación al total	$A = \frac{TP + TN}{TP + TN + FP + FN}$
Sensibilidad	Es la tasa de casos positivos clasificados correctamente	$R = \frac{TP}{TP + FN}$

Métrica	Descripción	Fórmula
Especificidad	Es la tasa de casos negativos clasificados correctamente	$E = \frac{TN}{TN + FP}$
1-especificidad	Es la tasa de casos positivos clasificados incorrectamente	$1 - E = \frac{FP}{TN + FP}$
Puntuación F1	Es la fusión de las métricas de Exactitud y Sensibilidad, con el objetivo de presentar diferencias en el rendimiento del algoritmo	$F1 = 2 \frac{A \cdot R}{P + R}$
Índice kappa	Se usa para verificar la concordancia entre un estándar de referencia y un clasificador, o entre dos observadores humanos. $Po =$ proporción de exactitud observado	$k = \frac{Po - Pe}{1 - Pe}$
Pe	Es la proporción de exactitud esperada de acuerdo al azar	$Pe = \frac{F1_T \cdot f1_T + Fo_T \cdot fo_T}{N^2}$
Área bajo la curva	Área bajo la curva ROC y representa la probabilidad de clasificar correctamente una clase positiva frente a una clase negativa	$AUC = \frac{R - (1 - E) + 1}{2}$

Fuente: Borja y otros, 2020

5. Metodología

En este capítulo se detallará los diferentes procedimientos y métodos utilizados durante el desarrollo de este tema

5.1. Área de estudio

El Jardín Botánico Reinaldo Espinoza es un centro de conservación de flora, fue fundado en el año 1949 y es considerado como el más antiguo de nuestro país. Gracias a su ubicación geográfica, encontrándose en el punto de convergencias de las corrientes bioclimáticas húmedas de la amazonia y cálidas secas de la vertiente del pacifico, da origen a una gran variedad de especies, que en su área que se extiende hasta aproximadamente 7 hectáreas, llegando a albergar cerca de 1385 especies vegetales tanto nativas como exóticas (UNL, s.f.).

Este jardín botánico está ubicado a las afueras de la ciudad de Loja en las coordenadas $4^{\circ} 2' 13.82''$ S y $79^{\circ} 11' 56.02''$ O. En la **Figura 8** se puede observar el área correspondiente al jardín botánico

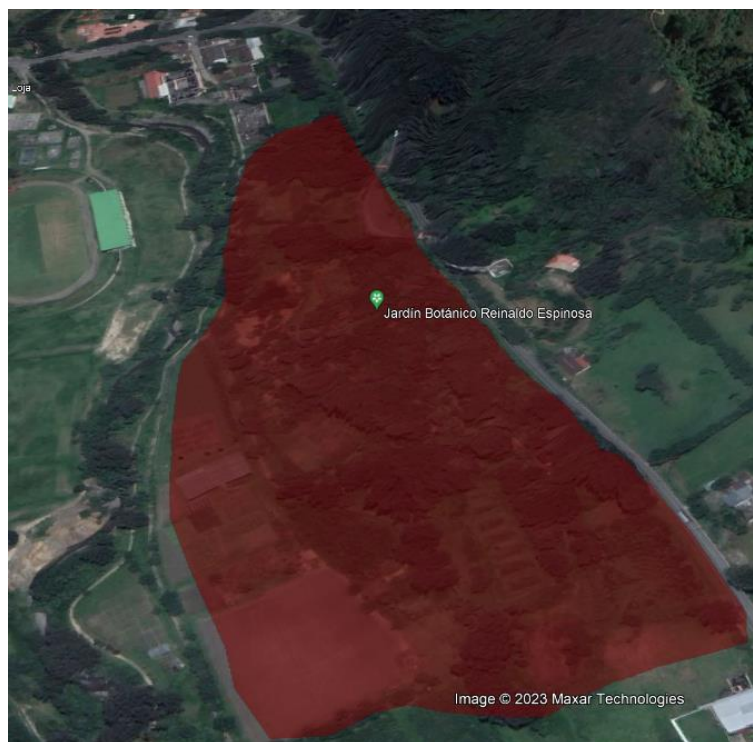


Figura 8. Área del Jardín Botánico Reinaldo Espinoza
Fuente: Google Earth

5.2. Base de Datos

La base de datos constituye un componente fundamental en el algoritmo, desempeñando un papel crítico en términos de eficiencia y precisión. Se ha considerado incluir dos elementos clave en la base de datos.

En primer lugar, se han incorporado muestras fotográficas de cada una de las plantas necesarias para el entrenamiento. Estas imágenes proporcionan una representación visual de las especies y permiten al algoritmo aprender a reconocer y distinguir las diferentes características de cada planta. Adicionalmente se ha incluido información descriptiva correspondiente a cada planta en la base de datos. Esta información detalla las particularidades más relevantes de cada especie.

5.2.1. Información de las características de las plantas

Al identificar una especie vegetal no solo se busca la detección sino también que la persona que haga uso del algoritmo conozca más acerca de las plantas, para ello se ha recopilado la información más relevante de cada especie. Esta información ya estaba presente en una base de datos, para ello se ha realizado la petición al Ing. Marcelo Gutiérrez Riofrio, técnico encargado de gestionar y supervisar el Jardín Botánico Reinaldo Espinoza y se presenta en el . De la información obtenida se han seleccionado los datos correspondientes a las plantas de muestra y se ha guardado en un archivo csv para facilitar su manipulación al momento de desarrollar el algoritmo. Los parámetros que se han considerado son:

- Nombre común
- Nombre científico
- Distribución
- Etimología
- Características
- Propagación
- Usos

La información obtenida se presenta en el [Error! No se encuentra el origen de la referencia.](#)

5.2.2. Adquisición de las imágenes

Debido a la falta de disponibilidad de imágenes de cada una de las plantas, se ha llevado a cabo un proceso de obtención de muestras fotográficas de cada especie. Para este propósito, se ha utilizado la cámara de un teléfono celular con una resolución de 3072 x 4080 píxeles. Esta resolución es suficientemente alta para capturar detalles importantes de las plantas. Con el fin

de mejorar la confiabilidad en el entrenamiento, se ha considerado obtener cada imagen con un valor fijo de exposición de la cámara. Para lograr esto, se realizaron ajustes en la velocidad de obturación y la sensibilidad ISO de la cámara. Estos ajustes permiten que todas las imágenes tengan una exposición consistente y evitan variaciones en la iluminación que podrían afectar la precisión del entrenamiento.

Para llevar a cabo este proceso, contamos con el apoyo del Ing. Marcelo Gutiérrez Riofrio, quien fue responsable de identificar cada una de las plantas. Su experiencia y conocimiento en botánica fueron fundamentales para asegurar que las especies fueron correctamente identificadas en las fotografías. Esta colaboración facilitó y aceleró el procedimiento de recopilación de las imágenes necesarias.

5.3. Modelos de entrenamiento

Para el desarrollo de la tesis, se ha considerado el uso de algunos modelos de entrenamiento, para realizar una comparación entre ellos y determinar cuál el más eficiente para el caso de estudio.

5.3.1. Primer algoritmo (MobileNet)

El primer Algoritmo está basado en MobileNet, presentado por (Howard y otros, 2017), el cual se describe en la sección 4.4.1. Teachable Machine fue la herramienta seleccionada para el desarrollo del primer algoritmo, esta herramienta nos permite entrenar un modelo de manera fácil y rápida. El proceso necesario para ello se detalla en la **Figura 9**, teniendo en primera instancia la entrada la cual estará compuesta por las imágenes de entrenamiento y las etiquetas que vincula a cada imagen a un elemento específico.

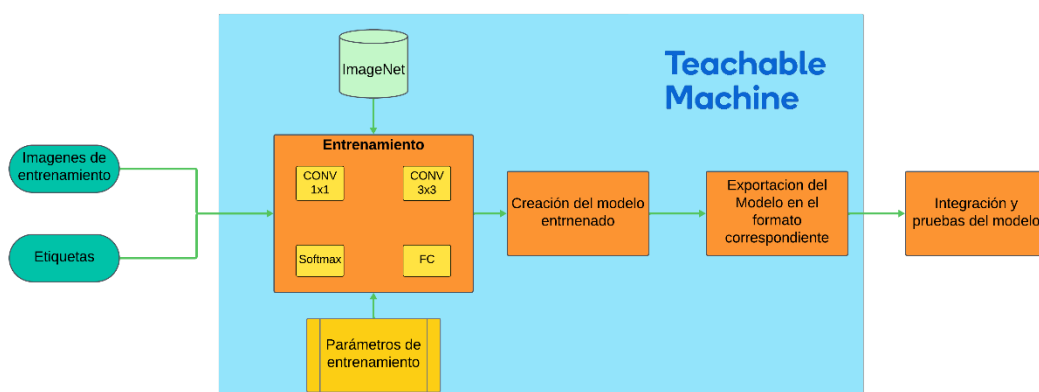


Figura 9. Diagrama del proceso de entrenamiento de Teachable Machine

Fuente: El autor

El preprocesamiento de las imágenes consta de dos fases, la primera fase consiste en el redimensionamiento de las imágenes, ajustando el tamaño a 224 x 224 píxeles. Para disminuir

la pérdida de la calidad de las imágenes, se utiliza interpolación de área. Este método consiste en convertir bloques de píxeles en un solo píxel, para ello se calcula el valor promedio de todos los píxeles que lo componen. En la segunda fase convertimos la imagen en un array de datos para que el modelo sea capaz de procesar la información, luego se realiza la normalización que consiste en la estandarización de los valores de cada píxel para que estén en una escala específica, facilitando así el entrenamiento del modelo. Este proceso generalmente se realiza dividiendo los valores de píxeles por un factor constante, en este caso se aplica la fórmula presentada en la **Ecuación 2**.

$$N = \frac{X}{127,5} - 1 \quad (2)$$

De tal manera que los valores presentes en cada pixel estarán en el rango de valor de [-1,1]. Las etiquetas no se realizan en cada imagen, sino que la interfaz de Teachable Machine nos permite la creación de instancias por cada clase donde subiremos las imágenes correspondientes a cada clase como se observa en la **Figura 10**.

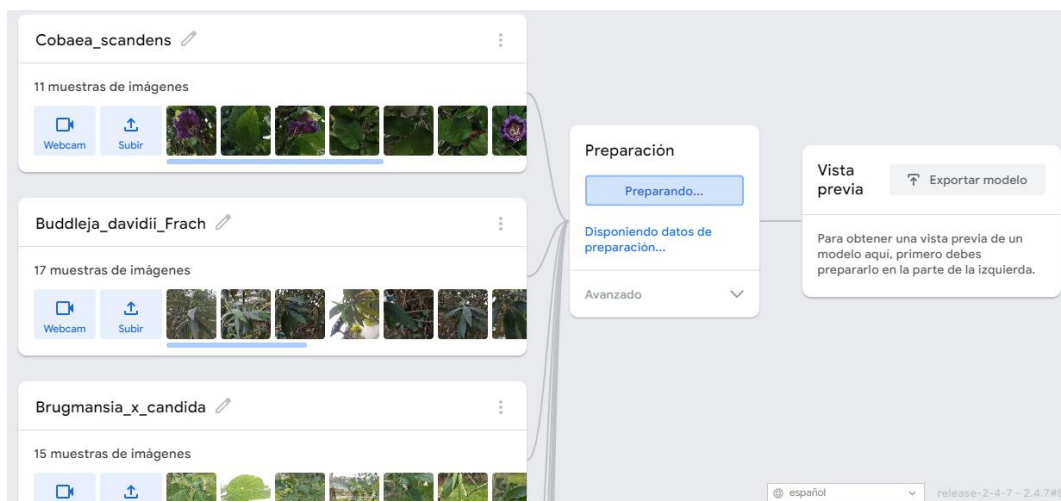


Figura 10. Interfaz de la herramienta de Teachable Machine

Fuente: Teachable Machine

Una vez subida la base de datos necesaria para el entrenamiento se procede a ajustar los parámetros necesarios, dados en la **Tabla 3**. A continuación, se procede a ejecutar el entrenamiento, el cual partirá de un algoritmo pre-entrenado con la base de datos ImageNet, este procedimiento tardará únicamente unos minutos debido a que todo el proceso se ejecuta en la nube.

Tabla 3. Parámetros de Entrenamiento del primer algoritmo

Épocas	200
Tamaño de lote	16

Tasa de aprendizaje	0,001
----------------------------	-------

Fuente: El autor

Al finalizar, nos permitirá exportar el modelo en diferentes formatos de TensorFlow ya sea para trabajar en java, Python o en su versión lite para dispositivos móviles. MobileNet al ser un algoritmo diseñado específicamente para ser liviano y eficiente, el modelo entrenado que se obtuvo no ocupa más de 5 MB.

5.3.2. Segundo Algoritmo (ResNet)

Pytorch nos presenta diversos modelos que pueden ser usados para la creación de un clasificador de imágenes. Para el caso del segundo algoritmo, se ha hecho uso del modelo ResNet-18, el cual es una red convolucional residual de 18 capas. Este modelo se explica en la sección 4.4.2, puede ser entrenado desde cero o también es posible usarlo pre-entrenado. En este caso se ha usado no pre-entrenado, por lo que el tiempo de ejecución del mismo fue mayor y se necesitó una mayor demanda de recursos del equipo, por lo que se trabajó con un entorno de desarrollo en la nube, Google Colab. El proceso de entrenamiento se presenta en la **Figura 11**.

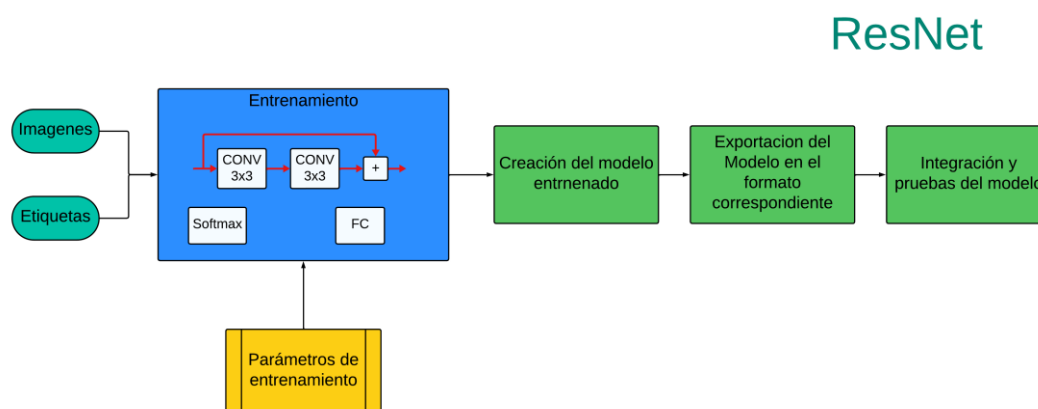


Figura 11. Diagrama del proceso de entrenamiento de ResNet

Fuente: El autor

Los parámetros establecidos para el entrenamiento del segundo algoritmo se presentan en la **Tabla 4**.

Tabla 4. Parámetros de Entrenamiento del segundo algoritmo

Épocas	200
Tamaño de lote	10
Tasa de aprendizaje	0,001

Fuente: El autor

Pytorch también ofrece un aceleramiento de procesamiento mediante el uso de la GPU, por lo que en primera instancia estableceremos el dispositivo de ejecución. Para ello lo seleccionamos mediante el comando presentado en la **Figura 12**. Donde al detectar que hay una GPU disponible devolverá el valor 'cuda' en caso contrario nos dará 'cpu'.

```
import torch

dispositivo = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
dispositivo

device(type='cuda')
```

Figura 12. Comandos para la selección del dispositivo de ejecución

Fuente: El autor

En la siguiente instancia, configuraremos el modelo con el que se trabajará, para ello se carga la red ResNet18 sin valores pre-entrenados, además se modificará la primera capa convolucional para aceptar imágenes RGB, el número de canales, así como el tamaño del kernel, el padding y el stride. Finalmente se modifica la última capa de la red, la capa totalmente conectada igualando el tamaño de la salida al número de clases como se presenta en la **Figura 13**.

```
self.model = torchvision.models.resnet18(weights=None)
self.model.conv1 = nn.Conv2d(3,64,kernel_size=(128,128),stride=(2,2),padding=(3,3),bias=False)
self.model.fc = nn.Linear(in_features=self.model.fc.in_features, out_features = num_clases)
```

Figura 13. Comandos para la configuración del algoritmo

Fuente: El autor

A continuación, se realiza la carga de las imágenes y sus etiquetas, para ello cargamos dos archivos txt uno para los datos de entrenamiento y el otro para los datos de validación. Dentro de estos archivos encontramos las etiquetas y la ruta de la ubicación de cada imagen. Además, es necesario redimensionar cada imagen a un tamaño de 128x128 y convertirla en un Tensor, mismo procedimiento se realiza en la función 'transformacion'. Adicionalmente se establece el tamaño del batch en 10 y se activa la función shuffle que se encarga de cargar los datos de manera aleatoria en cada época de entrenamiento. Los comandos utilizados se presentan en la **Figura 14**.

```

ruta = '/content/'

datos_trains = data_reader('train_files.txt', ruta, transform=transformacion)
datos_vals = data_reader('val_files.txt', ruta, transform=transformacion)

cargadores = {
    'entrenamiento': torch.utils.data.DataLoader(datos_trains,
                                                  batch_size= 10,
                                                  shuffle=True,
                                                  num_workers=1),
    'validacion': torch.utils.data.DataLoader(datos_vals,
                                              batch_size= 10,
                                              shuffle=True,
                                              num_workers=1)
}

```

Figura 14. Comandos para el ingreso de los datos de entrenamiento y validación

Fuente: El autor

Una vez finalizada la configuración de la red y los datos, se empieza el entrenamiento de la misma, para ello se establece un número de 200 épocas a una tasa de aprendizaje de 0.001. Al finalizar el entrenamiento, se puede exportar el modelo para hacer uso del mismo en cualquier momento o en otro dispositivo, como se presenta en la **Figura 15**. El mismo se exportará con el formato pt, que es el formato utilizado por la librería Pytorch para el almacenamiento de modelos entrenados.

```

import torch

PATH = '/content/model.pt'
torch.save(model.state_dict(), PATH)

```

Figura 15. Comandos para la exportación del modelo con Pytorch

Fuente: El autor

Al extraer el modelo, se procede a hacer uso de los datos de validación para determinar la precisión de la red, para ello se genera una matriz de convolución presentada en la **Figura 30**.

5.3.3. Tercer algoritmo (VGG19)

Para el desarrollo de este algoritmo se hizo uso de las librerías keras y TensorFlow, además de utilizar la red VGG19 explicada en la sección 4.4.3. El proceso de desarrollo de este modelo, se presenta en la **Figura 16**.

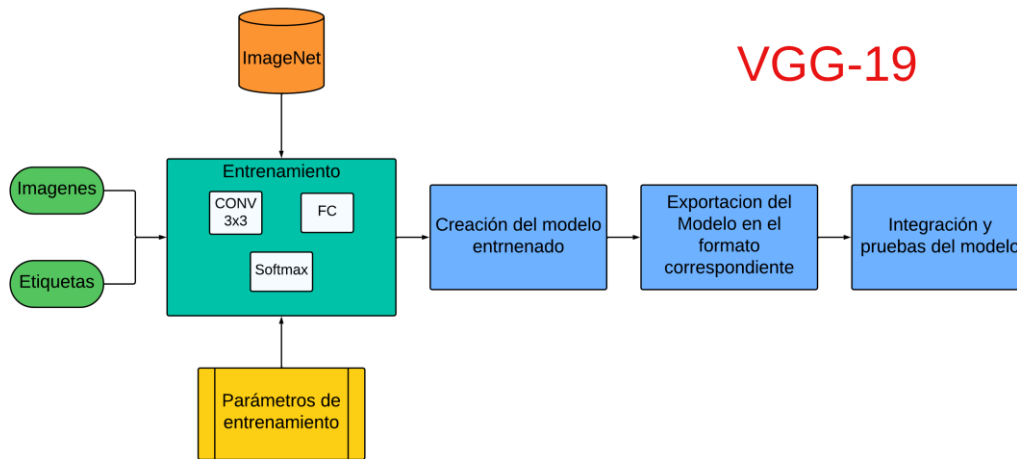


Figura 16. Diagrama del proceso de entrenamiento de VGG-19

Fuente: El autor

Para el entrenamiento de la red es necesario definir los parámetros de entrada. Se ha establecido el tamaño de lote a 10 y estableciendo el modo de clasificación ‘sparse’, utilizado en etiquetas con valores enteros, los comandos utilizados se presentan en la **Figura 17**.

```
def data_class(data_path, size, class_mode):
    datagen = ImageDataGenerator(rescale = 1./255)
    classes = datagen.flow_from_directory(data_path,
                                        target_size = size,
                                        batch_size = 10,
                                        class_mode = class_mode)

    return classes
```

Figura 17. Comandos para establecer los parámetros de entrada a la red neuronal

Fuente: El autor

En el siguiente paso, se importa el modelo VGG19 y se establece el tamaño de la entrada a 250x250 píxeles con 3 canales para imágenes RGB. Además de utilizar el algoritmo pre-entrenado con la base de datos ImageNet. Para mejorar el rendimiento de la red, se deshabilita la capacidad de entrenamiento de las capas, de tal manera que se congela la parte convolucional de la red para retener los conocimientos aprendidos durante el entrenamiento previo en la tarea de clasificación de imágenes. Adicionalmente, se añade a la salida una capa Flatten para la conversión de los datos tridimensionales a unidimensionales. Finalmente se crea una capa densa con la función Softmax con una dimensión igual al número de clases, los comandos se presentan en la **Figura 18**.

```
for layer in vgg.layers:
    layer.trainable = False

x = Flatten()(vgg.output)
prediction = Dense(8, activation='softmax')(x)
```

Figura 18. Comandos para la configuración de la red

Fuente: El autor

Al finalizar la configuración del modelo, se procederá a iniciar el entrenamiento del mismo, estableciendo el número de épocas en 200. Al finalizar el entrenamiento se generará un modelo entrenado, el cual se exportará para almacenar los valores aprendidos. El mismo se guardará con el formato h5 de la librería de TensorFlow, el comando necesario se presenta en la **Figura 19**.

```
model.save("/content/Classificatin_model.h5")
```

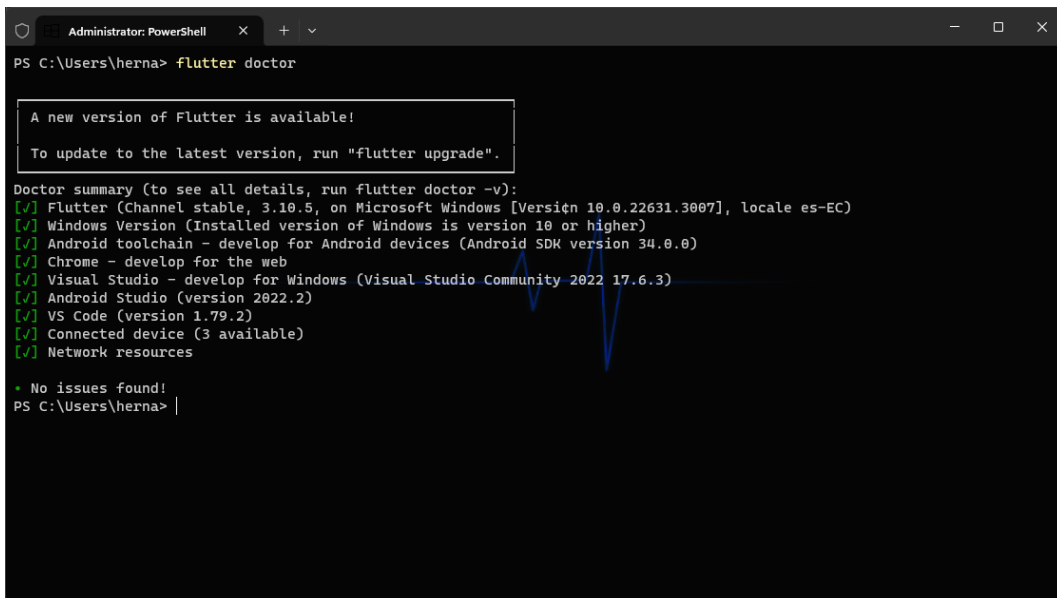
Figura 19. Comandos para la exportación del modelo con Pytorch

Fuente: El autor

Con el modelo guardado, se procederá a cargarlo para evaluar el rendimiento de la red, por lo que a su vez se ha generado una matriz de confusión presentada en la

5.4. Desarrollo de la Aplicación

Para el desarrollo de la aplicación se ha hecho uso del framework Flutter, el cual funciona con el lenguaje Dart y nos permite el desarrollo de páginas web y aplicaciones multiplataforma. Luego de la instalación de Flutter es necesario la instalación de software externo para su correcto funcionamiento, estos requerimientos se pueden comprobar mediante el comando 'flutter doctor' en la consola de Windows, tal como se muestra en la **Figura 20**.



```
Administrator: PowerShell x + v
PS C:\Users\herna> flutter doctor

A new version of Flutter is available!
To update to the latest version, run "flutter upgrade".

Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.10.5, on Microsoft Windows [Version 10.0.22631.3007], locale es-EC)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[✓] Android toolchain - develop for Android devices (Android SDK version 34.0.0)
[✓] Chrome - develop for the web
[✓] Visual Studio - develop for Windows (Visual Studio Community 2022 17.6.3)
[✓] Android Studio (version 2022.2)
[✓] VS Code (version 1.79.2)
[✓] Connected device (3 available)
[✓] Network resources

• No issues found!
PS C:\Users\herna> |
```

Figura 20. Ejecución del comando flutter doctor en la consola de Windows

Fuente: El autor

En la imagen anterior podemos observar que todos los elementos necesarios para su correcto funcionamiento están instalados correctamente. Además, en Flutter podemos probar

nuestras aplicaciones en tiempo real ya sea mediante un dispositivo físico o uno virtual y cualquier cambio que se haga en la aplicación, este cambiará automáticamente sin necesidad de volverlo a compilar. Una de las mayores ventajas de Flutter es que no requiere una gran capacidad de hardware del equipo de desarrollo a diferencia de Android Studio, lo que facilita su uso en computadores de bajos recursos.

Para ejecutar las aplicaciones dentro de un dispositivo físico es necesario brindar los permisos necesarios dentro del dispositivo. Para ello el primer es la activación de las opciones de desarrollador, por lo cual se debe ir a la sección de información del dispositivo y hacer clic en la parte de Versión del sistema operativo hasta que aparezca un mensaje flotante que las opciones de desarrollador están activas, tal como se observa en la **Figura 21**.



Figura 21. Activación de las opciones de desarrollador en Android

Fuente: El autor

Una vez activadas, se accede a estas nuevas configuraciones, donde nos brinda una gran número de nuevas opciones, pero en este caso únicamente nos interesa aquellas relacionadas al desarrollo de la aplicación, por lo que únicamente se toman en consideración dos: la depuración por USB y la Instalación vía USB. Estas opciones deben estar activas como se observa en la **Figura 22**.



Figura 22. Activación de la depuración por USB e instalación vía USB

Fuente: El autor

Una vez asegurada la instalación del framework y la configuración del mismo para la ejecución en un dispositivo virtual, es necesario la instalación de las librerías necesarias para el desarrollo de la aplicación, mismas que se detallan en la **Tabla 5**.

Tabla 5. Librerías utilizadas para el desarrollo de la aplicación

Nombre de la librería	Versión	Descripción
image_picker	1.0.7	Es una librería compatible con iOS y Android que es utilizada para importar imágenes desde la galería o tomar nuevas fotografías con la cámara.
image_cropper	4.0.1	Permite el recorte de imágenes. Está basada en tres librerías y cuenta con una interfaz propia.
image	3.3.0	Es una librería que permite la manipulación de imágenes de diferentes formatos.
tflite_flutter_plus	0.0.1	Permite el manejo flexible y rápido al intérprete de TensorFlow Lite y brinda una gran gama de comandos para diferentes aplicaciones como la clasificación de imágenes, detección de texto, etc.
tflite_flutter_helper_plus	0.0.2	Es una biblioteca dedicada al procesamiento y manipulación de entradas y salidas de modelos de TensorFlow Lite.

collection	1.17.1	Contiene funciones de utilidad y clases para facilitar el trabajo.
csv	5.1.1	Proporciona funcionalidades para leer y escribir datos en formato CSV

Fuente: El autor

Para el desarrollo de la aplicación se ha desarrollado en tres partes: MainApp, Second y Clasificador.

En el script MainApp se encuentra el código necesario para el desarrollo de la interfaz de la página principal. Aquí podemos encontrar un botón en la parte inferior derecha que al seleccionarlo nos dará la opción de cargar una imagen desde la galería o tomar una nueva imagen desde la cámara, como se observa en la **Figura 23**.

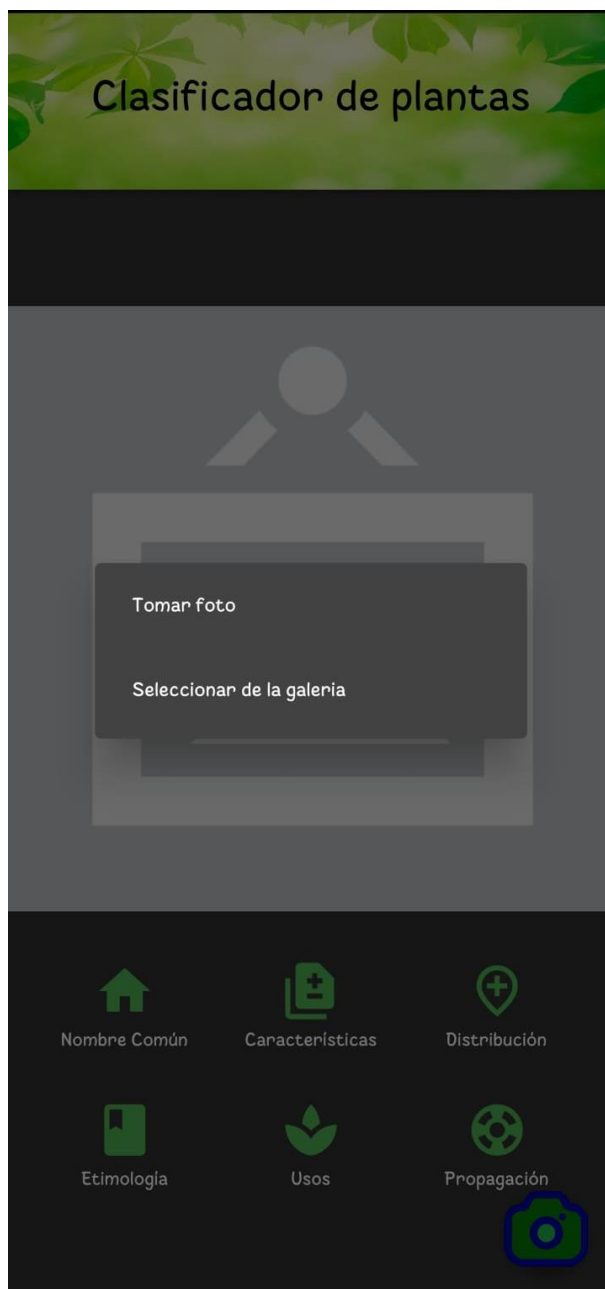


Figura 23. Función del botón de cámara que permite importar imagen desde la galería o desde la cámara

Fuente: El autor

La imagen seleccionada se podrá visualizar en una sección en la parte superior de la aplicación y encima de esta aparecerá el nombre científico de la planta que el algoritmo haya detectado, así como el porcentaje de precisión. Adicionalmente se han agregado 6 botones para mostrar información adicional acerca de la planta que se ha detectado, observados en la **Figura 24**. Estos botones tienen la función de acceder a la información relacionada con la planta como el nombre común, las características, la etimología, las zonas geográficas donde se distribuye, el uso y la forma en la que se propagan.

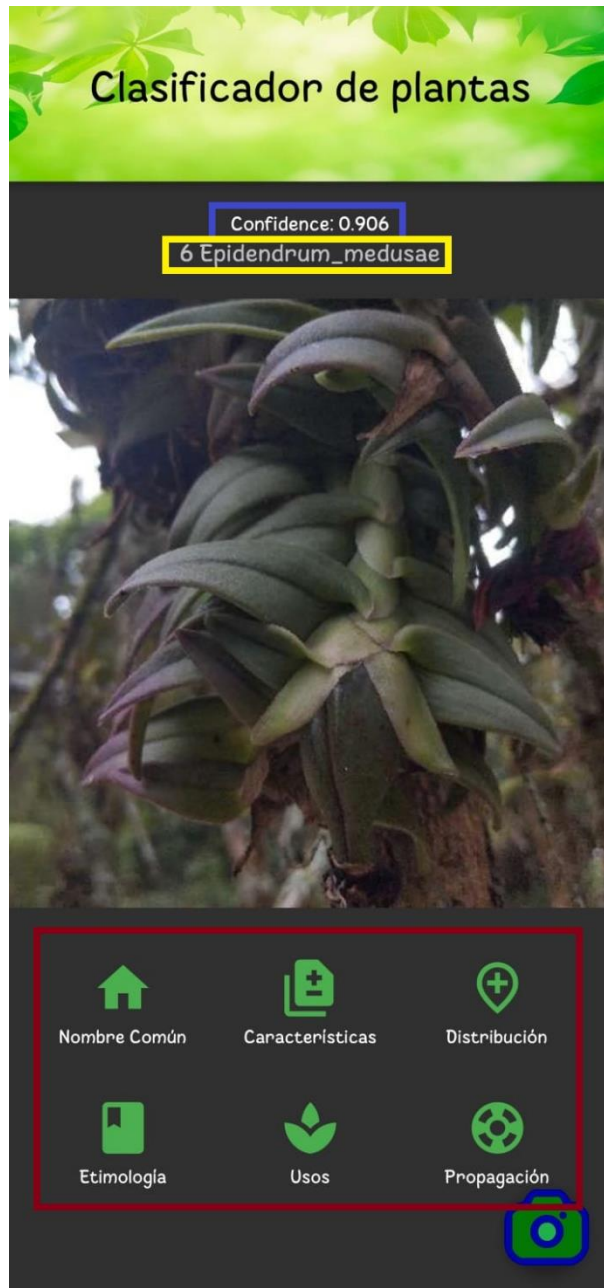


Figura 24. Elementos de la interfaz de la aplicación.

En el cuadro rojo se muestra los botones que permiten acceder a más información de la planta, en el cuadro azul muestra la confiabilidad de la identificación de la planta y en el cuadro amarillo el nombre científico de la misma.

Fuente: El autor

El script Second tiene la funcionalidad de cargar la información adicional contenida en un archivo csv y mostrarla en una nueva ventana, esta se presenta en la **Figura 25**.

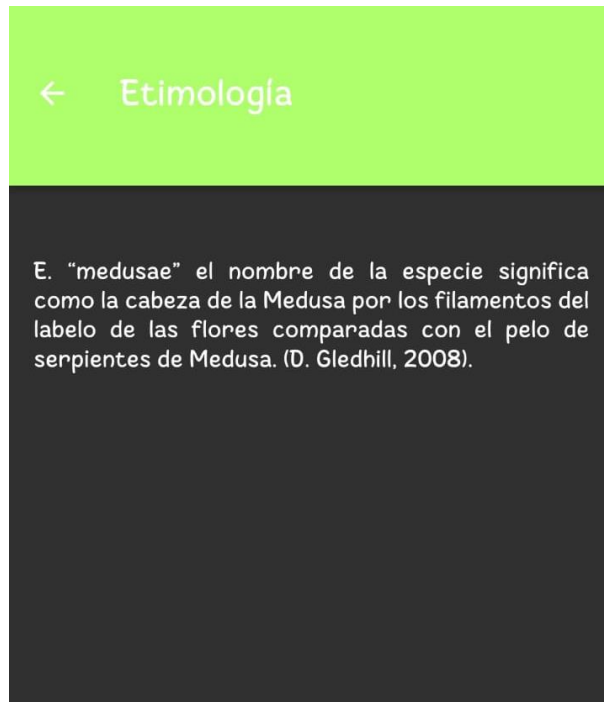


Figura 25. Interfaz de la ventana secundaria, donde como ejemplo se muestra la información etimológica del nombre de la planta

Fuente: El autor

Finalmente, en el script Clasificador está todo el código relacionado al proceso de identificación de la planta. Se encarga de cargar tanto el modelo generado como las respectivas etiquetas. Además de realizar el procesamiento de clasificación de imágenes y devolviendo el nombre científico de la planta identificada, así como el valor de exactitud. También se ha establecido un umbral de 0.5 para determinar cuando la detección sea correcta o no, si al cargar una imagen el valor de confiabilidad es menor a este umbral, la aplicación devolverá un mensaje que la planta no se encuentra en la base de datos.

6. Resultados

En este capítulo se presentarán los resultados de cada uno de los modelos evaluados, mismos que servirán para determinar el modelo que presente el mejor rendimiento. Posteriormente se evaluará el rendimiento general del modelo con toda la base de datos recopilada

6.1. Obtención de la base de datos

La base de datos recopilada consta de 1354 imágenes. Las cuales se han dividido para el entrenamiento de la red, para la validación y finalmente para evaluar la precisión del modelo. La cantidad asignada a cada una se presenta en la **Figura 26**.

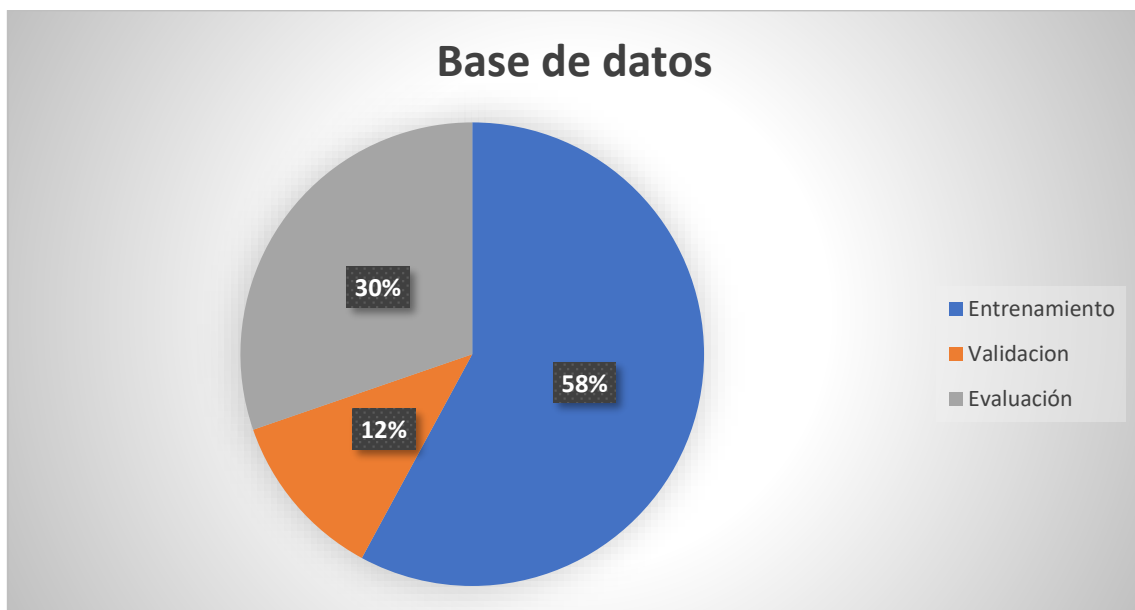


Figura 26. Distribución de la base de datos según el uso dado a las imágenes

Fuente: El autor

Estas imágenes fueron obtenidas como se describió en la sección 7.2. Además, la cantidad de imágenes de cada una de las plantas no es similar, la distribución de la base de datos por planta se presenta en la **Tabla 6**.

Tabla 6. Distribución de la base de datos por planta

Nombre	Entrenamiento	Validación	Evaluación
Acineta superba	38	8	20
Brassia aurantiaca	38	8	20
Brugmansia x candida	45	8	23
Brunfelsia grandiflora	38	8	20
Buddleja americana L	45	8	23
Buddleja davidii Frach	38	8	20

Nombre	Entrenamiento	Validación	Evaluación
Cobaea scandens	38	8	20
Epidendrum medusae	38	8	20
Epidendrum porphyreum	38	8	20
Fuchsia boliviana	38	8	20
Piper crassinervium	38	8	20
Pouteria lucuma	38	8	20
Roupala monosperma	38	8	20
Ruta graveolens L	45	8	23
Salix pyramidalis	38	8	20
Solandra maxima	38	8	20
Solanum albidum	38	8	20
Solanum americanum	38	8	20
Solanum cajanumense	38	8	20
Triplaris cumingiana	41	8	21
Total	784	160	410

6.2. Modelo MobileNet

Para la evaluación de cada uno de los modelos se han obtenido los parámetros presentados en la sección 4.6. Adicionalmente durante el entrenamiento de los modelos, se han generado gráficas que presentan las pérdidas y la exactitud por época. Así estas gráficas vienen presentadas en la **Figura 27**.

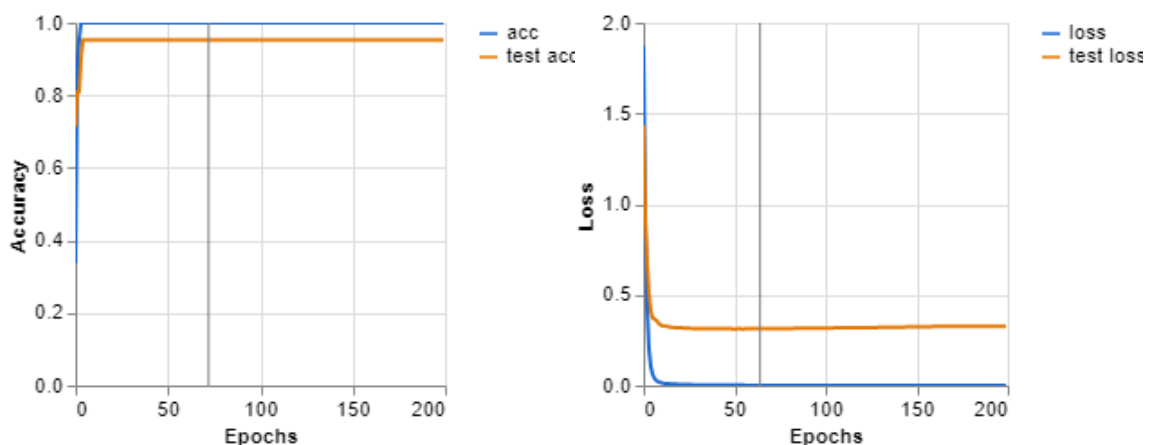


Figura 27. Gráficas que representan valores generados durante el entrenamiento.

A la izquierda nos encontramos con la exactitud por época, donde la parte azul representa la exactitud de los datos de entrenamiento y el naranja de los datos de validación. En la derecha por otro lado, nos presenta las pérdidas generadas, siendo el azul para las pérdidas de entrenamiento, y el naranja para las pérdidas de validación

Fuente: Elaborado por el autor

Así mismo, se genera una matriz de confusión que nos da una mejor apreciación de que tan confiable es el algoritmo, la misma se presenta en la **Figura 28**.

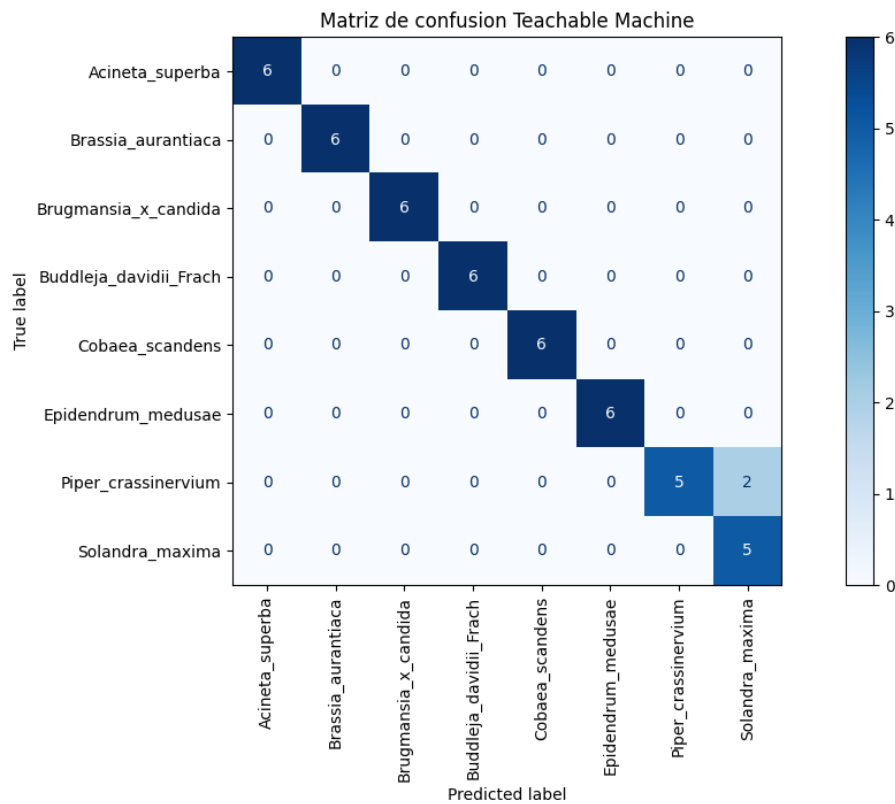


Figura 28. Matriz de Confusión del primer algoritmo

Fuente: Elaborado por el autor

Finalmente obtenemos las métricas de evaluación de rendimiento del primer algoritmo, mismas que se presentan en la **Tabla 7**.

Tabla 7. Resultados de los parámetros de evaluación del primer algoritmo

Parámetro de evaluación	Valor (%)
Exactitud	95,8333
Sensibilidad	96,4286
Especificidad	96,4286
1-especificidad	03,5714
Puntuación F1	95,8333
Área bajo la curva	96,4286

Fuente: El autor

6.3. Modelo ResNet

En este modelo también se tomó en cuenta las mismas consideraciones, por lo que las gráficas de pérdida y exactitud se presentan en la **Figura 29**.

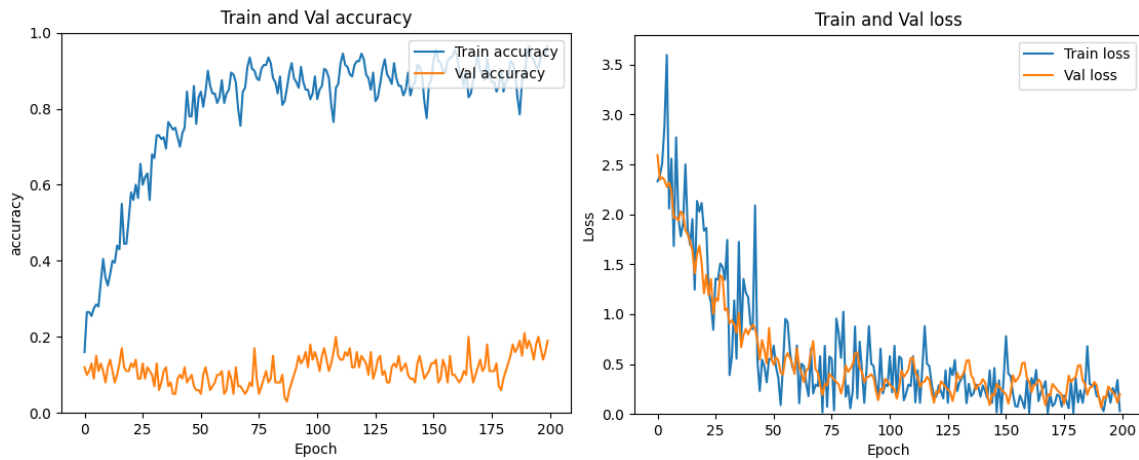


Figura 29 Gráficas que representan valores generados durante el entrenamiento del segundo algoritmo. A la izquierda nos encontramos con la exactitud por época, donde la parte azul representa la exactitud de los datos de entrenamiento y el naranja de los datos de validación. En la derecha por otro lado, nos presenta las pérdidas generadas, siendo el azul para las pérdidas de entrenamiento, y el naranja para las pérdidas de validación.

Fuente: Elaborado por el autor

Así mismo, se genera una matriz de confusión que nos da una mejor apreciación de que tan confiable es el algoritmo, la misma se presenta en la **Figura 30**.

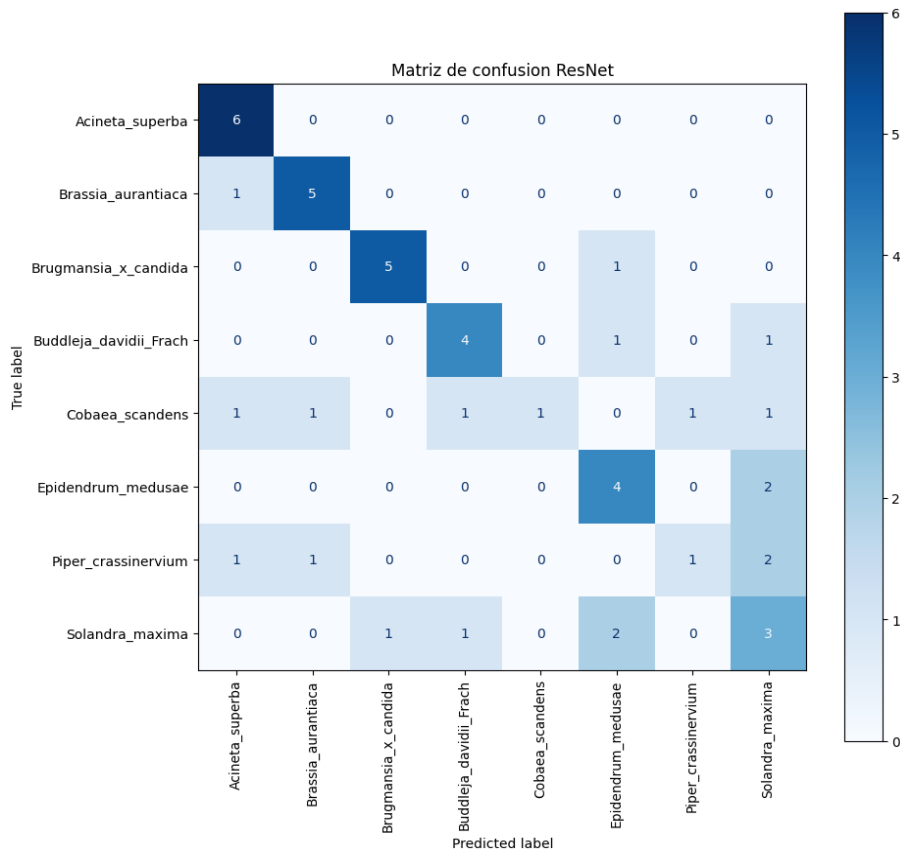


Figura 30. Matriz de Confusión del segundo algoritmo

Fuente: Elaborado por el autor

Finalmente obtenemos las métricas de evaluación de rendimiento del tercer algoritmo, mismas que se presentan en la **Tabla 8**.

Tabla 8. Resultados de los parámetros de evaluación del primer algoritmo

Parámetro de evaluación	Valor (%)
Exactitud	60,4167
Sensibilidad	59,9404
Especificidad	65,1785
1-especificidad	34,8214
Puntuación F1	57,8874
Área bajo la curva	62,5595

Fuente: El autor

6.4. Modelo VGG19

En este modelo también se tomó en cuenta las mismas consideraciones, por lo que las gráficas de pérdida y exactitud se presentan en la **Figura 31**.

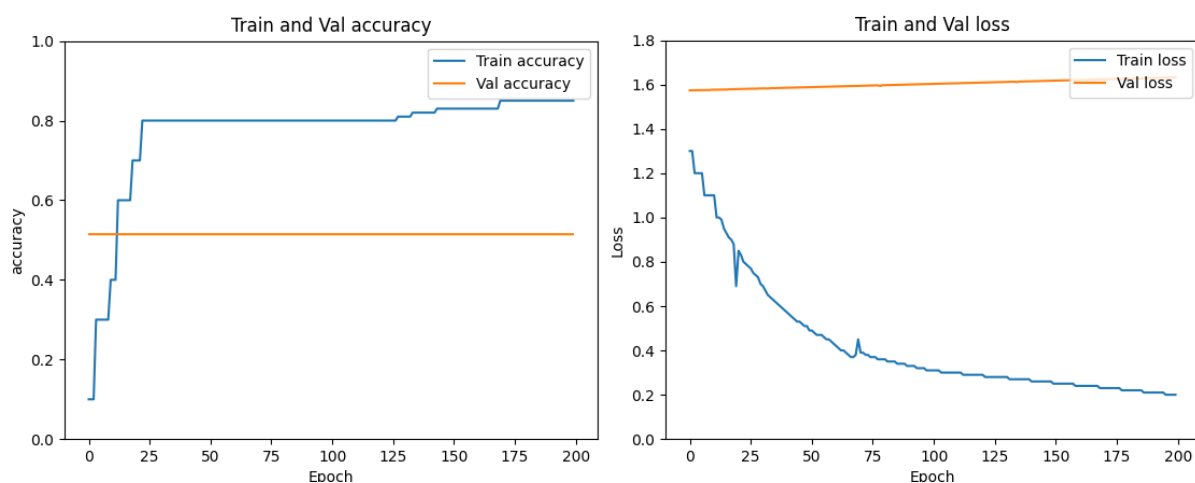


Figura 31. Gráficas que representan valores generados durante el entrenamiento del tercer algoritmo. A la izquierda nos encontramos con la exactitud por época, donde la parte azul representa la exactitud de los datos de entrenamiento y el naranja de los datos de validación. En la derecha por otro lado, nos presenta las pérdidas generadas, siendo el azul para las pérdidas de entrenamiento, y el naranja para las pérdidas de validación.

Fuente: Elaborado por el autor

Así mismo, se genera una matriz de confusión que nos da una mejor apreciación de que tan confiable es el algoritmo, la misma se presenta en la **Figura 32**.

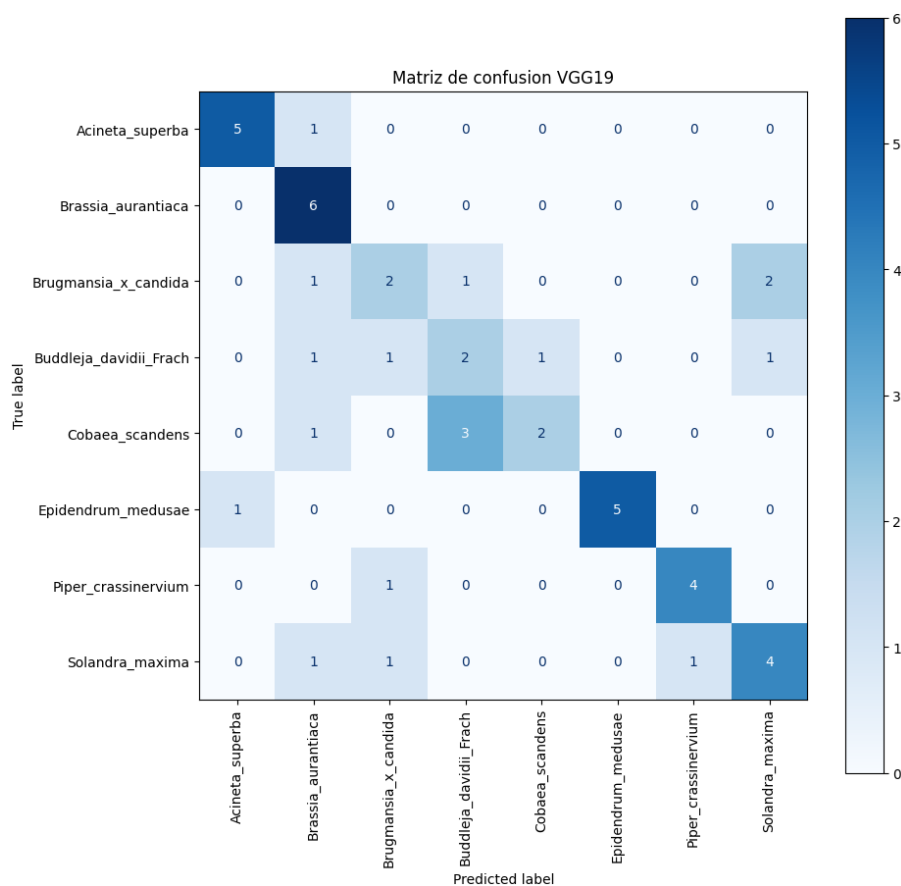


Figura 32. Matriz de Confusión del tercer algoritmo

Fuente: Elaborado por el autor

Finalmente obtenemos las métricas de evaluación de rendimiento del tercer algoritmo, mismas que se presentan en la **Tabla 9**.

Tabla 9. Resultados de los parámetros de evaluación del tercer algoritmo

Parámetro de evaluación	Valor (%)
Exactitud	62,2500
Sensibilidad	62,9762
Especificidad	64,3777
1-especificidad	35,6223
Puntuación F1	61,5650
Área bajo la curva	63,6770

Fuente: El autor

6.5. Modelo final

Al revisar cada una de las métricas de los 3 algoritmos, se puede llegar a observar que tanto el algoritmo 2 y 3, presentan un rendimiento similar. Por otro lado, el primer modelo es significativamente mejor que los otros dos modelos, por lo que es el utilizado para la

elaboración del algoritmo final. Para ello se hace uso de la base de datos completa y así mismo, se obtienen las gráficas de pérdidas, presentadas en la **Figura 33**.

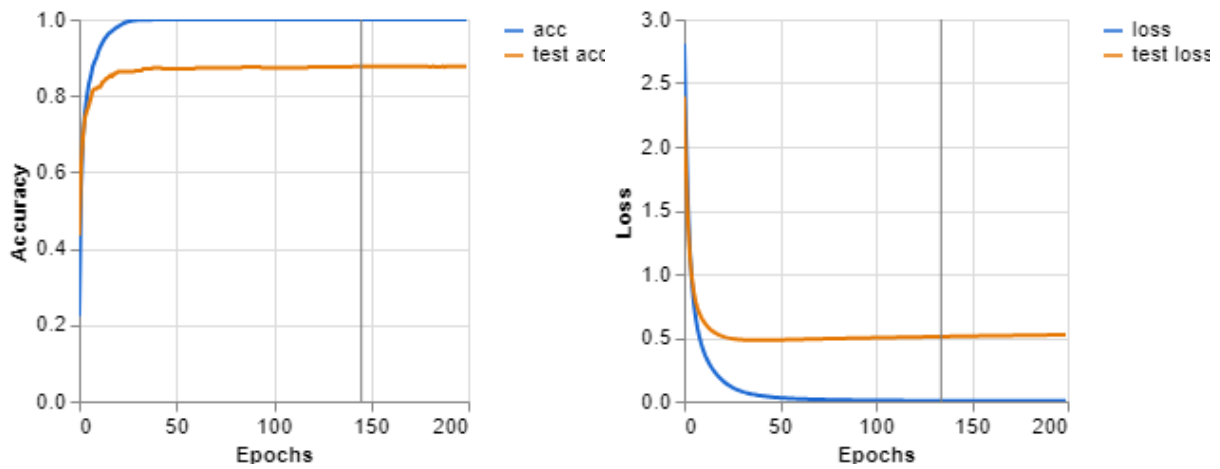


Figura 33. Gráficas que representan valores generados durante el entrenamiento del algoritmo final. A la izquierda nos encontramos con la exactitud por época, donde la parte azul representa la exactitud de los datos de entrenamiento y el naranja de los datos de validación. En la derecha por otro lado, nos presenta las pérdidas generadas, siendo el azul para las pérdidas de entrenamiento, y el naranja para las pérdidas de validación.

Fuente: Elaborado por el autor

Además, se presenta el número de positivos que ha detectado el algoritmo por cada tipo de planta, mismo que se presenta en la **Tabla 10**.

Tabla 10. Resultados obtenidos de las pruebas realizadas con el algoritmo final

Nombre	N° de muestras	positivos	Porcentaje (%)
Acineta superba	20	18	90.00
Brassia aurantiaca	20	18	90.00
Brugmansia x candida	23	19	82.61
Brunfelsia grandiflora	20	18	90.00
Buddleja americana L	23	18	78.26
Buddleja davidii Frach	20	20	100.00
Cobaea scandens	20	18	90.00
Epidendrum medusae	20	20	100.00
Epidendrum porphyreum	20	17	85.00
Fuchsia boliviana	20	18	90.00
Piper crassinervium	20	20	100.00
Pouteria lucuma	20	14	70.00
Roupala monosperma	20	18	90.00
Ruta graveolens L	23	23	100.00

Nombre	N° de muestras	positivos	Porcentaje (%)
Salix pyramidalis	20	20	100.00
Solandra maxima	20	18	90.00
Solanum albidum	20	16	80.00
Solanum americanum	20	15	75.00
Solanum cajanumense	20	18	90.00
Triplaris cumingiana	21	17	80.95
Total	410	363	88,59

Fuente: El autor

Así mismo, se genera una matriz de confusión que nos da una mejor apreciación de que tan confiable es el algoritmo, la misma se presenta en la **Figura 34**.

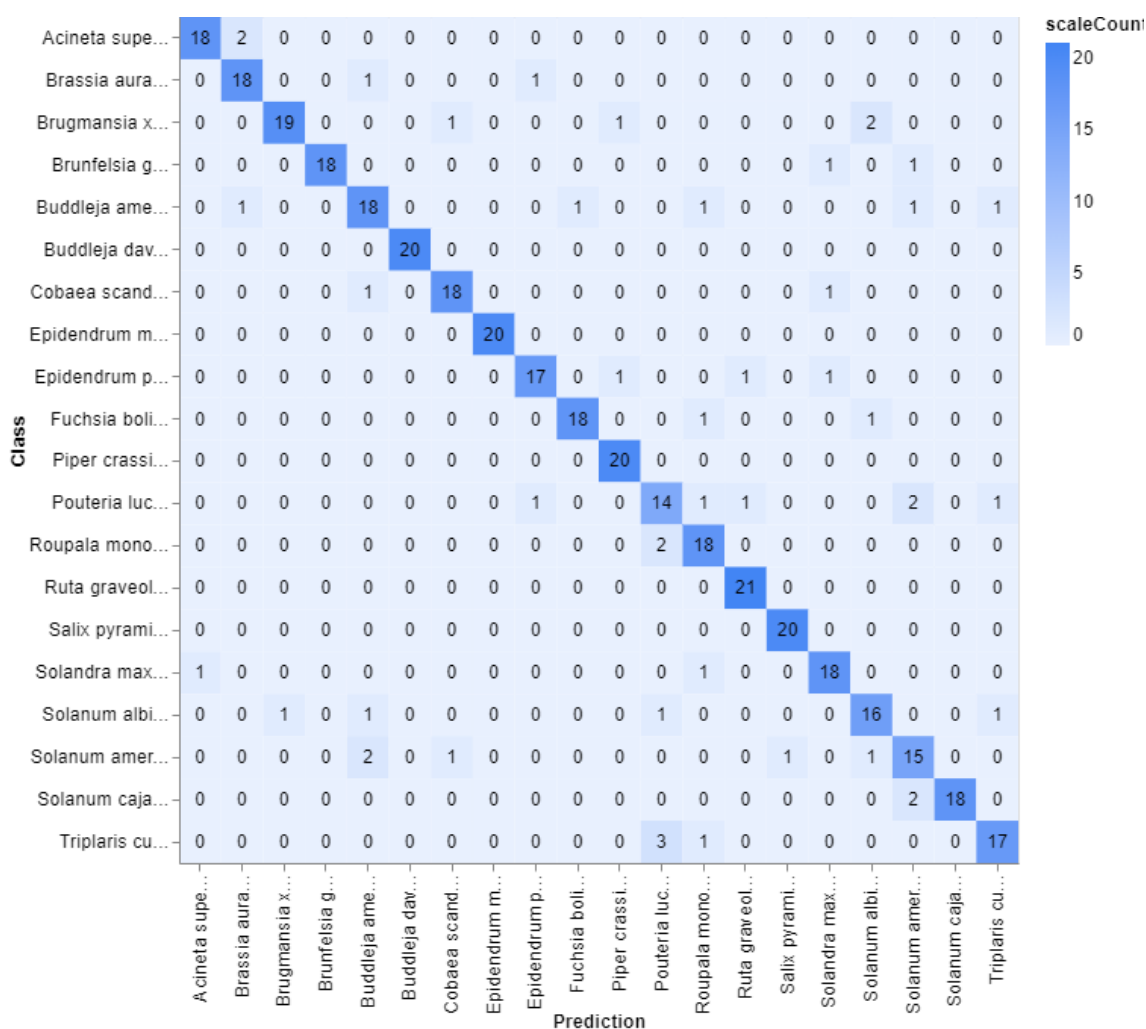


Figura 34. Matriz de Confusión del algoritmo final

Fuente: Elaborado por el autor

Finalmente obtenemos las métricas de evaluación de rendimiento del tercer algoritmo, mismas que se presentan en la **Tabla 11**.

Tabla 11. Resultados de los parámetros de evaluación del algoritmo final

Parámetro de evaluación	Valor (%)
Exactitud	88,5911
Sensibilidad	89,1781
Especificidad	88,7889
1-especificidad	11,2111
Puntuación F1	88,7848
Área bajo la curva	88,9835

Fuente: El autor

7. Discusión

Este proyecto tuvo como objetivo principal el desarrollo de un sistema de reconocimiento de especies vegetales para el jardín botánico Reinaldo Espinoza, ubicado en la ciudad de Loja.

La cercanía de la ciudad de Loja con la Amazonía ecuatoriana, y su ubicación dentro de la cordillera de los andes, ha presentado un clima adecuado que permite al Jardín Botánico contar con una gran variedad de especies vegetales, tanto nativas como exóticas. Sin embargo, debido a ello es un procedimiento muy complicado el realizar un modelo de reconocimiento de todas las especies, sobre todo a la limitación de tiempo que presentó este proyecto. Por lo que, para su desarrollo, únicamente se consideró un total de 20 especies vegetales.

El enfoque que se tomó en cuenta en este proyecto es el desarrollo de varios algoritmos de clasificación de imágenes basados en 3 de las arquitecturas de redes neuronales más utilizadas actualmente: ResNet, MobileNet y VGG. Las mismas fueron entrenadas utilizando una base de datos reducida con el objetivo de disminuir el tiempo y la carga computacional necesaria.

En lo que respecta a la arquitectura ResNet, se ha logrado obtener un rendimiento del 60,42%. Al analizar la matriz de confusión, se destaca que el modelo logra un buen desempeño en tres de las ocho clases de entrenamiento. Sin embargo, en las demás clases se logra evidenciar un notable porcentaje de error, lo que llega a afectar considerablemente el rendimiento global del sistema. Por otro lado, en el caso de la arquitectura de VGG, presenta un caso similar respecto a ResNet, ya que al igual que el modelo anterior, este también presenta un buen desempeño únicamente en tres clases. No obstante, se aprecia una ligera mejoraría en la precisión del modelo, logrando alcanzar un rendimiento del 62,25%.

Para el caso de la tercera y última arquitectura de MobileNet, llega a presentar una precisión del 94,83%. Exhibiendo una considerable mejora en el desempeño comparado con las arquitecturas anteriores. Debido a ello, este último ha sido considerado como el modelo final del proyecto. Siendo entrenado con la base de datos completa, la cual cuenta con un total de 1354 imágenes, de las cuales el 30% se ha reservado para el testeo de la red, mientras que del 70% restante un aproximado del 15% se utilizó para la validación y el 85% para el entrenamiento de la red. Resultando en una precisión general del sistema del 88,59%

Varios estudios previos, como los de los autores Quoc y colaboradores (2019), Ashwinkumar y otros (2022), y Kumar y otros (2019); han implementado diversos mecanismos que han resultado en una considerable mejoría en la precisión y eficacia de sus algoritmos. A diferencia de la metodología de Ai y colegas (2020) y en contraste con la metodología aplicada

en este proyecto, ambos centrados exclusivamente en la utilización de una red neuronal convolucional para la clasificación de imágenes. Como resultado, se observa que este trabajo logra una eficacia comparable a la obtenida por Ai y sus colegas en 2020, incluso superándola ligeramente. Además, los resultados se sitúan dentro del rango de precisión de los trabajos presentados en los antecedentes, que oscilan entre el 86% y el 98%.

8. Conclusiones

En el desarrollo del Sistema de Reconocimiento de Especies Vegetales Basado en Inteligencia Artificial, luego del análisis de los resultados, se ha llegado a obtener las siguientes conclusiones:

- El proyecto logró el cumplimiento de los objetivos planteados para el desarrollo de un Sistema de Reconocimiento de Especies Vegetales para el Jardín Botánico Reinaldo Espinoza en la ciudad de Loja. A pesar de no haber desarrollado el sistema tomando en cuenta todas las especies disponibles en el Jardín Botánico. Se cumplió con lo planteado al considerar 20 especies vegetales.
- Se ha logrado la creación de un conjunto de datos compuesto por 1354 imágenes que abarcan 20 especies vegetales, así como la recopilación de cada una de las particularidades más importantes de cada especie.
- En el proyecto se planteó el uso de tres arquitecturas de redes neuronales ampliamente utilizadas: ResNet, MobileNet y VGG. Cada una fue evaluada en términos de rendimiento, destacando que MobileNet superó significativamente a las otras dos, alcanzando una precisión del 94,83%.
- La implementación final del algoritmo, considerando la totalidad de la base de datos, ha demostrado una precisión del 88,59%. A pesar de su enfoque exclusivo en el desarrollo de redes neuronales, el sistema logró alcanzar un nivel de rendimiento satisfactorio.
- La incorporación del modelo desarrollado a la aplicación creada se llevó a cabo sin inconvenientes, resaltando la rapidez de respuesta de la aplicación al realizar la identificación de especies vegetales. Este rendimiento eficiente se atribuye a la optimización del modelo MobileNet, diseñado para operar de manera efectiva en dispositivos con limitaciones computacionales, lo cual contribuye al tiempo de respuesta reducido durante el proceso de identificación en la aplicación.
- El desarrollo del proyecto en el jardín botánico, que sirve como un punto de conservación de especies vegetales, puede llegar a presentarse como una herramienta en la conservación de la biodiversidad y sugiere la posibilidad de adaptar el modelo a otras ubicaciones botánicas.

9. Recomendaciones

En base a los resultado y conclusiones obtenidas, también se presentan recomendaciones a tomar en cuenta para trabajos futuros

- Para la creación de la base de datos, es importante tomar en consideración que las plantas pueden variar ligeramente su aspecto dependiendo de la época en la que se encuentre, por lo que para mayor precisión es recomendable construir la base de datos con fotografías recopiladas a lo largo de un año.
- Si se busca la implementación del proyecto a una mayor escala, también es importante tomar en consideración la cantidad de fotografías por planta que se disponga, ya que al aumentar el número de clases también es necesario incrementar el número de muestras. Por lo que se sugiere tomar en consideración las limitaciones computacionales disponibles al momento de realizar el entrenamiento
- La tecnología es una área que se encuentra en constante evolución, por lo que a pesar del éxito de MobileNet, se sugiere mantenerse al tanto de nuevas arquitecturas y técnicas en el campo del aprendizaje profundo, explorando su aplicabilidad en futuras investigaciones.
- El sistema no solo debe ser capaz de identificar las especies presentes en el jardín botánico, sino que también sería recomendable agregar una base de datos de especies invasoras que afecten la flora local, de tal manera que su rápida detección permita una pronta intervención para su eliminación.

10. Bibliografía

- Aggarwal, C. (2018). *Neural Networks and Deep Learning* (Primera ed.). Springer International Publishing AG.
- Ai, Y., Sun, C., Tie, J., & Cai, X. (2020). Research on Recognition Model of Crop Diseases and Insect Pests Based on Deep Learning in Harsh Environments. *IEEE Access*, 8, 171686 - 171693. <https://doi.org/10.1109/ACCESS.2020.3025325>
- Alcántara, C., Pedraza, J., Ramos, J., Gorrostieta, E., Tovar, S., & Vargas, J. (2019). Detección efectiva de rostros en imágenes utilizando descriptores basados en HOG. *Research in Computing Science*, 148(8), 371–385.
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., . . . Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1). <https://doi.org/10.1186/s40537-021-00444-8>
- Ashwinkumar, S., Rajagopal, S., Manimaran, V., & Jegajothi, B. (2022). Automated plant leaf disease detection and classification using optimal MobileNet based convolutional neural networks. *Materials Today: Proceedings*, 51(1), 480-487. <https://doi.org/10.1016/j.matpr.2021.05.584>
- Bonaccorso, G. (2017). *Machine Learning Algorithms*. Birmingham: Pack Publishing.
- Borja, R., Monleón, A., & Rodellar, J. (2020). Estandarización de métricas de rendimiento para clasificadores Machine y Deep Learning. *Risti: Revista Ibérica de Sistemas e Tecnologías de Información*(E30), 184-196.
- Brayan, C., Carlos, C., David, E., & Yuli, G. (2020). Redes neuronales artificiales y estado del arte aplicado en la ciberseguridad. *Revista Matices Tecnológicos*, 12, 58-63.
- Carney, M., Webster, B., Alvarado, I., PhillipsM, K., Phillips, K., Griffith, J., . . . Chen, A. (2020). Teachable Machine: Approachable Web-Based Tool for ExploringMachine Learning Classification. *CHI: Conference on Human Factors in Computing*, 1-8. <https://doi.org/10.1145/3334480.3382839>
- Castro, J. (2018). Diseño de un sistema de visión artificial de bajo coste para el control fitosanitario de cultivos de lulo. [Tesis de Maestría, Universidad de Pamplona]. Repositorio Institucional Universidad de Pamplona
- Dastres, R., & Soori, M. (2021). Artificial Neural Network Systems. *International Journal of Imaging and Robotics*, 21(2), 12-25.
- De Luca, D. (s.f.). *Qué es PyTorch*. Retrieved 23 de noviembre de 2023, from <https://damiandeluca.com.ar/que-es-pytorch>

- del Valle, A. (2017). Curvas ROC y sus aplicaciones. [Tesis de Grado, Universidad de Sevilla].
 Depósito de Investigación Universidad de Sevilla
- Forchhammer, S., Abu-Ghazaleh, A., Metzler, G., Garbe, C., & Eigentler, T. (2022). Development of an Image Analysis-Based Prognosis Score Using Google's Teachable Machine in Melanoma. *Cancers*, 14(2243). <https://doi.org/10.3390/>
- Godoy, F. (2021). Métodos Clásicos de Clasificación: Comparación y Aplicación. [Tesis de Grado. Universidad Nacional de Córdoba]. Repositorio Digital UNC
- Guerrero, J. (2015). Sistema de visión para agricultura de precisión: identificación en tiempo real de líneas de cultivo y malas hierbas en campos de maíz. [Tesis Doctoral, Universidad Complutense de Madrid]. ResearchGate
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778. <https://doi.org/10.1109/CVPR.2016.90>
- Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., . . . Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv preprint*. <https://doi.org/10.48550/arXiv.1704.04861>
- Kakani, V., Nguyen, V. H., Kumar, B. P., Kim, H., & Pasupuleti, V. R. (2020). A critical review on computer vision and artificial intelligence in food industry [Una revisión crítica de la visión por computadora y la inteligencia artificial en la industria alimenticia]. *Journal of Agriculture and Food Research*, 2. <https://doi.org/https://doi.org/10.1016/j.jafr.2020.100033>
- Kumar, M., Gupta, S., Gao, X.-Z., & Singh, A. (2019). Plant Species Recognition Using Morphological Features and Adaptive Boosting Methodology. *IEEE Access*, 7, 163912-163918. <https://doi.org/10.1109/ACCESS.2019.2952176>
- Martel, J. (26 de agosto de 2020). ¿Conoces PyTorch? La herramienta de open source con la que puedes crear redes neuronales. <https://itelligent.es/conoces-pytorch-herramienta-open-source-la-puedes-crear-redes-neuronales/>
- Mathivet, V. (2018). *Inteligencia artificial para desarrolladores: Conceptos e implementación en C#*. Ediciones eni.
- Quoc, T., Thanh, N., Quoc, T., & Xuan, H. (2019). Plant species identification from leaf patterns using histogram of oriented gradients feature space and convolution neural networks. *Journal of Information and Telecommunication* , 4(2), 140-150 . <https://doi.org/0.1080/24751839.2019.1666625>

- Rouhiainen, L. (2018). *Inteligencia artificial 101 cosas que debes saber hoy sobre nuestro futuro*. Barcelona: Editorial Planeta, S.A.
- Simonyan, K. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations*.
<https://doi.org/10.48550/arXiv.1409.1556>
- Szeliski, R. (2022). *Computer Vision [Visión por computadora]* (Segunda ed.). Springer.
https://books.google.com.ec/books?id=QptXEAAAQBAJ&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false
- Torres, A. (2010). Curvas ROC para Datos de Supervivencia. [*Tesis de Maestría, Universidad de Santiago de Compostela*]. Repositorio del Departamento de Estadística, Análisis Matemático y Optimización
- UNL. (s.f.). *Jardín Botánico Reinaldo Espinosa*. Retrieved 23 de mayo de 2023, from <https://www.unl.edu.ec/investigacion/jardin-botanico-reinaldo-espinosa>
- Vovides, A. P., Iglesias, C., Luna, V., & Balcázar, T. (2013). Los jardines botánicos y la crisis de la biodiversidad. *Botanical Sciences*, 91(3), 239-250.
- Yang, L., Liu, Y., Yu, H., Fang, X., Song, L., Li, D., & Chen, Y. (2021). Computer Vision Models in Intelligent Aquaculture with Emphasis on Fish Detection and Behavior Analysis: A Review [Modelos de visión por computadora en acuicultura inteligente con énfasis a la detección de peces y análisis de comportamiento: Una revisión]. *Archives of Computational Methods in Engineering*, 28, 2785–2816.
<https://doi.org/10.1007/s11831-020-09486-2>

11. Anexos

Anexo 1. Características de las plantas obtenidas del Jardín Botánico

#	Nombre común	Nombre científico	Distribución	Etimología	Características	Propagación	Usos
1	Periquito.	<i>Acineta superba</i>	Se la encuentra en Venezuela, Colombia, Ecuador y Perú. Epífita nativa de la Costa y los Andes, en el Ecuador se la encuentra en la provincia de El Oro, entre los 500-1500 msnm.	El nombre del género " <i>Acineta</i> " proviene del griego <i>akinetos</i> = <i>inmóvil</i> refiriéndose al labelo rígido y articulado. El nombre de la especie " <i>superba</i> " significa <i>arrogante, orgulloso, magnífico</i> , por lo fabuloso de su inflorescencia.	Epífita grande, con pseudobulbos agrupados de forma ovoide a cilíndricos comprimidos lateralmente. Hojas con una vaina basal y con 2 a 4 hojas en la parte apical del pseudobulbo, son oblanceoladas, coriáceas, glabras, agudas, pecioladas. Inflorescencia un racimo colgante que puede llegar a medir 30 cm de largo. Flores fragantes cerosas y de un color crema con tonalidades rojizas oscuras.	Se propaga por división de matas, necesita buena sombra, agua abundante y en aumento, mientras que un breve descanso en seco en la época fría, que provocará fuertes brotes.	Ornamental.
2		<i>Brassia aurantiaca</i>	Originaria de Suramérica. Nativa del noroeste de Venezuela, Colombia y Ecuador. Epífita de los Andes ecuatorianos,	<i>B. "aurantiaca"</i> , el nombre de la especie alude a que se asemeja a <i>una naranja</i> , de color <i>naranja</i> , por el color de sus flores.	Epífita, con pseudobulbos elíptico y piriformes, verde oscuro, por cada pseudobulbo se aprecia generalmente 2 hojas apicales, lanceoladas y ápice acuminado.	Estas orquídeas no tienen un cultivo fácil debido a sus grandes requerimientos.	Ornamental

			presente en la provincia de Loja.		Inflorescencia en racimo axilar arqueado, que puede medir hasta 30 cm de largo. Las flores son agrupadas y de un color naranja brillante.		
3	Floripondio. Guando blanco. Trompeta de los ángeles.	<i>Brugmansia x candida</i> Pers	Es originaria de los Andes del Ecuador, Perú y países vecinos. Arbusto o árbol pequeño nativo y cultivado en Ecuador en Galápagos, los Andes y la Amazonía entre 0-3000 msnm, en las provincias de Carchi, Chimborazo, Cotopaxi, Galápagos, Guayas, Imbabura, Loja, Pichincha y Zamora.	B. "candida" el nombre de la especie significa <i>muy blanca</i> .	Arbusto de hasta 3 m de altura de hábito desordenado y ramificación baja. Hojas alternas de forma aovada lanceolada, de borde entero, y pubescentes, Flores acampanadas, péndulas, de hasta 30 cm de longitud, con la boca ancha. Fruto una baya inerte. Propagación: Se propaga por semillas y esquejes.		Planta de uso ritual, se le atribuye virtudes de protección contra los espíritus. Tiene carácter tóxico y alucinógeno. Medicinalmente el tallo y ramas alivian los dolores. Las hojas y flores en baños o en emplastos se usan para tratar salpullidos, gripe, fiebre, cólicos, afecciones de la menstruación, afecciones, reumáticas "nacidos", heridas y fracturas. Es usada como analgésico y anestésico. También se usa como ornamental.
4	Budelia. Budleia.	<i>Buddleja davidii</i> Frach	Originaria del noreste de China y	B. "davidii" el nombre de la especie	Arbusto de hoja caduca o semicaduca	Por esquejes	Se utiliza como ejemplar único, en

	<p>Budleja. Arbusto de las mariposas. Arbusto de la mariposa. Flor de las mariposas. Lilo de verano. Lila de verano. Bailey.</p>		<p>Japón. Arbusto introducido y cultivado en los Andes ecuatorianos, entre los 2500-3000 msnm, presente en las provincias de Loja, Pichincha y Tungurahua.</p>	<p>honra a Armand David un misionero naturalista francés que la descubrió en 1869</p>	<p>(según el clima), rústico, vigoroso y de rápido desarrollo. Tiene aspecto arqueado pudiendo alcanzar hasta 5 m de altura. Ramas colgantes. Requiere un lugar espacioso. Su follaje se caracteriza por su color verde glauco. Hojas: opuestas y decusadas, oblongo lanceoladas, borde ligeramente dentado, base y ápice agudos, miden 10,5 de largo x 2,5 cm de ancho. Inflorescencia en racimo. Flores pequeñas de color lila y perfumadas.</p>		<p>grupos o en alineación en caminos, las mariposas toman el néctar de sus flores</p>
5	<p>Campanilla morada. Funeraria.</p>	<p><i>Cobaea scandens</i> Cav</p>	<p>México, comúnmente cultivada como especie anual y herbácea, particularmente en climas fríos. Liana introducida en los Andes ecuatorianos entre los 1500-2500 msnm, presente en las provincias de Azuay, Bolívar,</p>	<p><i>El nombre del género "Cobaea" honra al cronista, científico y sacerdote español Bernabé Cobo (1582-1657). Y la especie "scandens" significa escandente, o que trepa.</i></p>	<p>Vigorosa enredadera perenne, que crece hasta 4,5 m de altura o más. Hojas alternas, compuestas formadas de 4-6 folíolos ovales, terminadas en un zarcillo ramificado. Flores con el cáliz estriado y generalmente de color violeta, con la corola acampanada, al</p>	<p>Por semillas.</p>	<p>De la Torre <i>et al.</i>, 2008, dice que la flor es visitada por abejas. Puede volverse una planta invasiva, aunque también se puede considerar una planta ornamental, para cubrir muros, troncos de árbol, pérgolas, etc</p>

			Carchi, Imbabura, Loja, Napo y Tungurahua.		principio de color verdoso y posteriormente violáceo o blanquecino. Fruto una cápsula cilíndrica.		
6	Medusa	<i>Epidendrum medusae</i>	Se la encuentra en los bosques nubosos húmedos, Epífita herbácea endémica de los Andes, en el Ecuador se la encuentra en las provincias de Azuay y Loja entre los 2500-3000 msnm.	E. “medusae” el nombre de la especie significa como la <i>cabeza de la Medusa</i> por los filamentos del labelo de las flores comparadas con el pelo de serpientes de Medusa.	Epífita de tamaño medio que se desarrolla con un rizoma corto, los tallos son colgantes, agrupados, en forma de caña, basalmente ramificadas, tallos completamente cubiertos de hojas. Hojas conduplicadas, carnosas, coriáceas, estrechamente oblongo ovadas, agudas, envainadoras. Inflorescencias pocas flores grandes que emergen de una axila, solitarias. Flores irregulares grandes de 7, 5 cm de diámetro, carnosa de color violáceo con el labelo periféricamente terminando en filamentos.	Como epífita en temperaturas frescas en condiciones intermedias de luz y humedad constante.	Ornamental
7	Guabiduca de sal.	<i>Piper crassinervium</i> Kunth	Nativa, en Centro América, Costa Rica, Panamá,	P. “crassinervium” el nombre de la especie se deriva de	Pequeño arbusto, con fuste nudoso. Hojas simples, alternas y	Por semillas, esquejes y acodos.	De la Torre et al., 2008 dice que las hojas preparadas son

			<p>Suramérica de Venezuela al Perú. Arbusto o árbol pequeño nativo de los Andes y la Amazonía entre los 0-2500 msnm, en Ecuador presente en las provincias de Carchi, Cotopaxi, Guayas, Morona, Napo, Pastaza, Pichincha, Sucumbíos y Zamora.</p>	<p>las palabras latinas <i>crassi</i> = <i>grueso</i>, y <i>nervi</i> = <i>nervio</i>; con nervios secundarios gruesos, o en resalto, en este caso particularmente.</p>	<p>dísticas, de color verde intenso en la superficie adaxial, y verde pálido en la superficie abaxial, ampliamente ovadas a elípticas, el ápice agudo a acuminado, la base obtusa o rotunda, nerviación pinnada; láminas papiráceas, con haz y envés glabro, el nervio principal y secundario moderadamente pubescente. Inflorescencias en espigas de color verde pálido, erguidas, pequeñas, rodeadas por brácteas angostamente triangulares, corta pero densamente fimbriadas en los márgenes.</p>		<p>comestibles. Las hojas aromáticas se usan como condimento. La cocción de la flor con miel de abeja se usa para tratar el “pasma” (enfermedad de la piel. Las hojas tratan afecciones indeterminadas.</p>
8	Copa de oro.	<i>Solandra maxima</i>	<p>México hasta Colombia y Venezuela. Liana introducida y cultivada en la Costa y los Andes, en Ecuador se la cultiva en las provincias de Guayas, Napo y</p>	<p>El nombre de la especie “<i>Solandra</i>” honra a Daniel Carlson Solander (1736-1782), naturalista sueco, discípulo de Linneo. La especie “<i>máxima</i>” significa</p>	<p>Bejuco gigante, de hasta 9 m o más, necesita de un soporte fuerte. Hojas alternas, oval oblongas, grabras, borde entero, ápice y base agudos. Las flores grandes de 25 cm de diámetro son amarillas con una</p>	Por esquejes	<p>Muy valorado en jardinería por sus enormes flores.</p>

			Pichincha, entre los 500-3000 msnm.	<i>más grande, la más grande.</i>	franja morada en el centro de cada pétalo.		
9	Fernán Sánchez. Roblón.	<i>Triplaris cumingiana</i>	Común y ampliamente distribuido en el bosque húmedo tropical, incluyendo llanuras de ríos y bosques tropofíticos y secos. Se encuentra en los bosques del Canal de Panamá y Ecuador. Árbol nativo de Galápagos, la Costa y los Andes ecuatorianos, entre los 0-1500 msnm, Se encuentra en las provincias de Bolívar, Chimborazo, El Oro, Esmeraldas, Galápagos, Guayas, Loja, Los Ríos, Manabí, Morona y Pichincha.	El género " <i>Triplaris</i> " deriva del latín <i>triplus-a-um</i> = <i>triple</i> , por las 3 alas del fruto. Y la especie " <i>cumingiana</i> " honra al naturalista inglés Hugh Cuming (1791-1865), considerado el príncipe de los recolectores.	Árbol de gran tamaño entre 22 – 28 m y 30 a 45 cm de diámetro, corteza de color pardo claro, ramitas gruesa verde castaño. Hojas alternas grandes y brillantes, la lámina oblonga u ovada, ápice puntiagudo y borde entero. Cuando la planta es joven el follaje es denso y la forma de las hojas ovalada, en estado adulto es semitransparente y de forma irregular. Inflorescencias en racimos, de color rojo cuando son jóvenes y luego amarillentas, flores masculinas y femeninas en diferentes árboles son dioicas. Frutos cubiertos por tres alas oblongas rosadas y vistosas.	Por semilla.	El fruto es alimento de loros, las flores las visitan las mariposas, posee uso apícola. Su madera sirve para vigas, tablas y postes en construcción. El tallo es refugio de hormigas.
10	Pepino de cerro	<i>Solanum cajanumense</i> Kunth	Arbusto nativo de la Costa y los Andes entre los 300-3500 msnm,	: <i>S. "cajanumense"</i> el nombre de la especie posiblemente se	Arbusto de hasta unos 2, 5 m de altura, poco ramificado. Hojas alternas, lanceoladas,	Por semillas	Su fruto es comestible, con las hojas quemadas se ahuman ollas de

			presente en las provincias de Bolívar, Carchi, Chimborazo, Cotopaxi, Loja y Pichincha.	refiere a que la especie fue determinada con material proveniente del nudo de <i>Cajanuma</i> en Loja, Ecuador.	de borde un tanto ondulado base cordiforme y ápice acuminado. Inflorescencias cimosas. Flores pentámeras, hermafroditas. Fruto una baya		arcilla donde se almacena chichi. El fruto cocido se usa para tratar afecciones del corazón. Las hojas se usan para tratar la inflamación. La infusión de las hojas es útil para bajar la fiebre en los niños.
11	Hierba mora. Mortiño.	<i>Solanum americanum</i> Mill	Nativa, desde el sur de Canadá hasta Sudamérica. Herbácea o subarbusto nativo de Galápagos, Costa, Andes y la Amazonía, en Ecuador se la puede encontrar en las provincias de Azuay, Bolívar, Cañar, Carchi, Chimborazo, Cotopaxi, Esmeraldas, Galápagos, Guayas, Imbabura, Loja, Los Ríos, Manabí, Napo, Pichincha, Sucumbíos, Tungurahua y Zamora, entre los 0-4000 msnm.	S. "americanum" alude a que la planta es oriunda o se publicó con material proveniente de este continente americano.	Hierba, erecta de hasta 1,5 m de alto. Tallo ramificado, o casi sin pelos. Hojas a menudo en pares, siendo una más grande que la otra, sobre peciolo hasta de 3 cm de largo, lanceolada a ovada, hasta 15 cm de largo y 5,5 cm de ancho, ápice agudo a acuminado, borde entero asinuado dentada, base cuneada, con pelos o sin pelos. Inflorescencia en cimas umbeliformes laterales. Flores: hermafroditas y regulares, con cáliz con 5 lóbulos, más o menos del mismo largo que el tubo,	Por semillas	El fruto es comestible se usa para preparar chicha morada. La planta es usada por los brujos para curar el susto. El zumo de las hojas se usa como insecticida. El fruto se exprime en el ojo para tratar la nube del ojo. La infusión de la planta se usa para tratar el resfrío, infecciones y el dolor de cabeza. En floraciones da calor al pecho, trata la gastritis, la gripe, golpes, el "chuchaqui" y afecciones del hígado y los nervios. Es purgante,

					<p>ovados a oblongos, doblados hacia abajo en fruto; corola morada o blanca, sus lóbulos triangulares, mucho más largos que el tubo. Fruto una baya globosa, negra en la madurez; semillas lenticulares</p>		<p>narcótico y diaforético. El jugo o la infusión de hojas frescas provocan transpiración y se utilizan para limpiar el intestino, y para tratar la fiebre y el dolor. El tallo y las hojas machacadas se usan para tratar las liendres (piojos), irritaciones de la piel, paperas y viruela. La decocción de la planta con la raíz se usa para tratar infecciones. La infusión de las hojas se usa para realizar baños que tratan hinchazones y el tabardillo. Las hojas y las semillas se usan para tratar la irritación de las muelas. La infusión del tallo y hojas se bebe para tratar dolores estomacales. La raíz el fruto y las hojas contienen alcaloides como la</p>
--	--	--	--	--	---	--	--

							hiocimina y atropina que son antiespasmódicos. Las hojas quemadas se usan para tratar el salpullido. Las hojas se preparan con alcohol alcanforado para tratar la erisipela. Es útil para tratar el acné, heridas, granos, infecciones, afecciones del colon y de los riñones. La infusión se usa para tratar el cólera y dolores fueres del parto
12	Roble andino. Taro blanco.	<i>Roupala monosperma</i>	Sudamérica Perú y Ecuador. Arbusto o árbol nativo de los Andes ecuatorianos, está presente en las provincias de Carchi, Loja, Napo y Tungurahua, entre los 1000-3500 msnm,	El nombre del género " Roupala " es el nombre común usado en Las Guyanas para la planta <i>Roupala montana</i> . El nombre de la especie " monosperma " deriva de los términos <i>mono = uno</i> y <i>sperma = semilla</i> .	Árbol de 25 m de altura y 60 cm de diámetro en el tronco, el cual es duro y de color blanco, y su madera es parecida a la del roble. Hojas de 15 cm de largo por 6 cm de ancho, simples, alternas, ovado – oblongas, dispuestas en forma de hélice, borde aserrado, superficie áspera, textura apergaminada. Posee un olor similar	Por semillas.	La madera se usa en estructuras de construcción como madera de aserríos y para elaborar yugos y herramientas.

					al atún. Flores de 1.5 cm de largo, de color blanco, agrupadas en inflorescencias terminales		
13	Calavera.	<i>Brunfelsia grandiflora</i>	Arbusto o árbol, nativo y cultivado en la Costa, los Andes y la Amazonía entre los 0-1500 msnm. En Ecuador presente en las provincias de Loja, Los Ríos, Manabí, Morona, Napo, Pastaza, Pichincha, Sucumbíos y Zamora.	El nombre del género " Brunfelsia " honra la memoria del teólogo y botánico alemán Otto Brunfels (1488-1534). Carlos Linneo, consideraba a Brunfels como el padre de la botánica. Y la especie " grandiflora " significa <i>con flores grandes</i> .	Arbusto de hasta 5 m de altura. Tallos cilíndricos ramificados, con follaje denso. Hojas alternas oblongo lanceoladas de borde entero, ápice acuminado, base aguda, miden 12,5 cm de largo por 4,8 cm de largo, de un tono verde claro, glabras, pinnatinervadas, con el nervio medio prominente en la superficie abaxial. Inflorescencias terminales racimosas. Flores hermafroditas, regulares y tubulares, cáliz tubular con cuatro lóbulos, corola con hipantio tubular 5 lobada, lóbulos ondulados, de color violeta y blanco. Frutos una baya globosa de 1,6 cm de	Por semillas y recolección de plántulas	La decocción de las hojas se usa en baños de fresco. Las hojas y el tallo son usados para cazar animales. Es utilizada en ceremonias rituales. Las hojas y la raíz en infusión se usan para inducir el vómito. La planta es usada para preparar bebidas alucinógenas que se ingieren en rituales. Las hojas, raíz, tallo y corteza se usan como anticonceptivos. En asocio con otras plantas combate el mal de ojo. El tallo sin corteza se usa para aliviar el cuerpo hinchado y adolorido. Las hojas sirven para tratar quemaduras. El zumo de la corteza

					diámetro con el cáliz persistente.		machacada trata la gripe se aplica una gota en la nariz. La planta es usada para tratar dolores de cabeza. La raíz se usa para tratar dolores corporales. La raíz y la cocción de las hojas se usan para tratar la gripe. La infusión de las hojas trata el cuerpo débil. Se emplea en el tratamiento del reumatismo, artritis, escalofríos y bajar la fiebre.
14	Hoja de oso. Tacure. Tululuche. Tulupe.	<i>Solanum albidum</i> Dunal	Nativa de América del Sur occidental, desde el sur de Ecuador hasta el norte de Argentina, y crece bien en elevaciones medias en los Andes. Arbusto o pequeño árbol nativo de la Costa y los Andes, en Ecuador lo encontramos en las provincias de El Oro, Loja, Morona, Pichincha y	El nombre del género tiene su origen en el género " <i>Solanum</i> " según algunos autores se deriva de la palabra latina <i>solari</i> , que significa <i>consuelo, alivio al miedo</i> ; aludiendo a las propiedades calmantes atribuidas a algunas de las especies. Otros creen que con ese nombre llamaban los romanos a <i>Solanum nigrum</i> L., planta	Arbusto o árbol pequeño de hasta 5 m de altura. Tallo cilíndrico, ramificado, armado con espinas. Hojas alternas, oblongo lanceoladas, pecioladas de 23 cm de largo por 12,5 cm de ancho, lobuladas, superficie adaxial verde oscuro liso, abaxial blanquecino ligeramente pubescente. Inflorescencia cimosa, con unas 24 flores.	Por semillas	Sus frutos son comestibles para algunas variedades de animales silvestres..

			Zamora entre los 500-2000 msnm.	considerada altamente tóxica. Y la especie " <i>albidum</i> " significa <i>blanco</i> .	Flores regulares, hermafroditas, pentámeras, cáliz tubular cónico, corola tubular, cinco lobada de color blanco. Fruto una baya globosa de unos 0,8 cm de diámetro.		
15	Sauce lápiz. Sauce largo. Sauce real.	<i>Salix pyramidalis</i> Budischtschew ex Trautv	Nativa del sur de México, América Central, Costa Rica, El Salvador, Guatemala, Honduras, Nicaragua y Panamá) y Sudamérica (Brasil, Bolivia, Colombia, Ecuador, Perú, Argentina y Chile).	<i>S. "pyramidalis"</i> la especie alude a que posee una <i>forma piramidal</i>	Árbol alto y delgado crece hasta 15 m de altura. Su tronco principal suele ser liso y de color grisáceo, mientras que sus ramas delgadas son verticales, flexibles y se agrieta con facilidad. Son de color verde amarillento o rojizo verdoso cuando son jóvenes, con frecuencia se vuelve de color bronce a medida que envejecen. Las hojas largas y estrechas alternas a lo largo de los tallos, borde finamente dentadas. Superficie adaxial de color verde brillante, y abaxial por lo general de un verde pálido. Flores cuando están presentes,	Por estacas o esquejes	Sirve para elaborar artesanías y como cerca viva.

					pequeñas que se disponen en racimos alargados amentiformes		
16	Ruda.	<i>Ruta graveolens</i> L	Es nativa del sur de Europa, Asia menor. Arbusto introducido en los Andes ecuatorianos entre los 2500-3500 msnm, se lo encuentra en las provincias de Bolívar, Chimborazo, Loja, Pichincha y Tungurahua.	El nombre del género " Ruta " deriva del griego <i>reuo</i> y significa <i>que salva, que protege</i> ; otros autores lo relacionan con el termino griego <i>ruomai</i> , que significa <i>refrenar</i> , en alusión a la supuesta acción afrodisiaca. La especie " graveolens " significa <i>de olor fuerte, maloliente, muy perfumadas</i>	Arbusto perenne, pequeño de hasta 1 m de altura. Tallos duros poco ramificados. Posee un aroma característico. Hojas verdeazuladas alternas, divididas en pares de foliolos y con foliolo solitario en la punta, compuestas por varios segmentos de los cuales los laterales son alargados y el terminal ovalado o espatulado, consistencia algo carnosa. Inflorescencias cimas corimbiformes. Flores de color amarillo intenso, hermafroditas. Fruto capsula.	Recolección de plántulas, esquejes y semillas.	Planta de uso místico, para baños, limpiar cuartos y animales ojeados, curar el mal de aire. Medicinalmente el té caliente de las hojas trata los calambres estomacales, las hojas se mastican para calmar los nervios, el vértigo, en asocio con otras hierbas se usa como purgante, se debe usar con cuidado, en dosis alta pueden provocar la muerte; es usada como repelente de insectos y como aromatizante.
17	Luma. Lukma	<i>Pouteria lucuma</i> (Ruiz & Pav.) Kuntze	Originaria del Perú. Árbol nativo de los Andes ecuatorianos entre los 500-1000 y 1500-3500 msnm, presente en las provincias de Azuay, Bolívar,	El nombre del género " Pouteria " tiene su origen en la abreviación del nombre vernáculo <i>Pourama pouteri</i> , con el que se conoce a <i>Pouteria guianensis</i> Aubl., en	Árbol de fuste recto y cilíndrico crece hasta 12 m de altura. Su madera es de color claro, grano fino y resistente. Posee látex pegajoso. Copa densa y esférico. Hojas alternas, obovadas,	Por semilla.	Se cultiva por su fruto empleado en la gastronomía en la preparación de dulces, postres y helados. Su madera también sirve para leña.

			Cañar, Cotopaxi, Loja y Tungurahua.	la Guayana francesa. Y la especie " <i>lucuma</i> ", el término español proviene del quechua rukma con el significado fruto del lúcumo.	que se concentran en el ápice de las ramas tiernas, poseen una ligera pubescencia y son color verde oscuro en la superficie abaxial. Flores solitarias, axilares y tubulares de color amarillo o verdoso hermafroditas; poseen cinco a siete pétalos. Fruto una baya redondeada de 6 – 10 cm de diámetro, color verde y café con pulpa amarillenta muy dulce		
18	Epidendro púrpura.	<i>Epidendrum porphyreum</i> Lindl.	Se encuentra en Colombia y Ecuador en bosques montanos muy húmedos. Epífita nativa herbácea de los Andes ecuatorianos presente en las provincias de Carchi, Imbabura y Pichincha entre los 1500 – 3500	<i>E. porphyreum</i> , el nombre de la especie significa <i>morado, sangriento, de color cálido, rojizo, porfúreo, (Porphyrion</i> , uno de los 24 hijos de Ge y Urano, era un gigante en Mitología romana).	Epífita de crecimiento mediano a grande, de clima frío, o terrestre en terraplenes empinados; con tallos delgados como bastón envueltos por vainas foliáceas que llevan coriáceas, delgadas, hojas lineales y agudas que se juntan basalmente. Inflorescencia racimosa, terminal arqueada y ramificada, con flores de color púrpura de 1 cm de diámetro. Es muy	Sobre sustrato medio para orquídeas en pots de helecho arbóreo, o como epífita.	Ornamental

					similar a <i>E. hypoporphyreum</i> , pero se distingue fácilmente por la superficie abaxial púrpura de las hojas.		
19	Salvia real.	<i>Buddleja americana</i> L	Originaria de América desde México hasta el sur de Bolivia. Arbusto, árbol pequeño o árbol nativo de Galápagos, Costa, Andes y Amazonía entre 0-3000 msnm, en Ecuador presente en las provincias de Azuay, Chimborazo, Cotopaxi, Galápagos, Guayas, Imbabura, Loja, Los Ríos, Manabí, Morona, Pastaza, Pichincha, Tungurahua y Zamora.	El nombre del género " Buddleja " honra a Adam Buddle botánico en Essex Inglaterra. El nombre de la especie " americana " alude a que la planta es originaria, o se publicó con material procedente del continente <i>americano</i> .	Arbusto que alcanza un tamaño de 2–5 m de alto; hermafroditas o a veces dioicas o ginodioicas. Hojas lanceoladas, elípticas u ovado - lanceoladas, de 10–15 cm de largo y 5–8 cm de ancho, acuminadas en el ápice, atenuadas en la base, frecuentemente decurrentes, subglabras en la superficie adaxial, tomentosas en la superficie abaxial. Inflorescencias de 8–25 cm de largo, las primeras cúlulas pedunculadas. Flores con cáliz 2.5–4 mm de largo, corola infundibuliforme, 4–5 mm de largo, amarilla, internamente con una línea de tricomas en forma semicircular en los lobos; estambres	Semillas, esquejes	Las abejas visitan las flores de esta especie. Se usa para elaborar tabaco. Medicinalmente la infusión de las hojas se utiliza como diurético y para desinflamar partes del cuerpo golpeadas, trata infecciones indeterminadas, alivia dolencias reumáticas y se usa para tratar resfríos.

					insertos en los senos o apenas por debajo. Cápsulas 3.5–5 mm de largo; semillas aladas		
20	Poncho de indio.	Fuchsia boliviana Carrière	Nativa del sur de Perú, Bolivia y norte de Argentina. Arbusto o pequeño árbol introducido y cultivado en los Andes entre los 2000-3000 msnm, en Ecuador está presente en las provincias de Loja y Pichincha.	El nombre del género <i>“Fuchsia”</i> honra a Leonard Fuchs (1501-1566), botánico alemán renacentista. El nombre de la especie <i>“boliviana”</i> hace referencia a que es originario o se publicó con material procedente de <i>Bolivia</i> .	Arbusto de hoja perenne crece hasta 4 m de altura, Hojas grandes y pubescentes, opuestas, elíptico ovadas y pecioladas, verdes y pecíolos rojos. Inflorescencias corimbos péndulos. Flores tubo y sépalos rojo intenso; pétalos más o menos crespos, longitudinalmente acanalados, caedizos antes del tubo floral, rojos. Fruto baya cilíndrica purpúreo y comestible.	Se multiplican por semillas o esquejes.	Sus frutos son de alimento humano y de avifauna, también para tratar enfermedades venéreas. Por lo general es una planta ornamental de jardines.

Anexo 2. Certificado de traducción del resumen

María Eduarda Ludeña Piedra
Traductora oficial Español <> Inglés
Código de Registro 12297908
JUDICATURA ECUADOR

Por medio de la presente certifico que:

- a) Habiendo obtenido 1 documento con fines de registros educativos para publicación de tesis del Sr. **Hernán Alexander Rivera Merchán**, con cédula de identidad número **1105337875**, y nacionalidad ecuatoriana el 22 de abril del 2024 para que sea traducido.
- b) Habiendo traducido los documentos de español al inglés.
- c) Habiendo desarrollado dicha traducción de acuerdo con las normas de la traducción profesional académica y jurídica, que es mi área de especialización; de modo que los documentos finales sean una traducción exacta y precisa de los originales.
- d) Habiendo entregado el documento traducido a la persona interesada el día 23 de abril del 2024

Para todos los propósitos legales, certifico que, a mi leal saber y entender, lo anteriormente expuesto es cierto.

Para efectos de verificación, la página traducida contiene un pie de página con mi código de registro e identificación, de igual manera la página lleva mi firma digital.

Si tiene alguna pregunta, no dude en contactarme.

Loja, 23 de abril del 2024



MARÍA EDUARDA LUDEÑA PIEDRA
Traductor e Interprete
C. I.: 1105333825

Dirección: Juan de Salinas 156-10 y José Antonio de Sucre. Edificio Lagunas del Compadre. Dto. 4. Loja Ecuador. Teléfono: +593959279769.
Email: eduardaludena@gmail.com