

UNIVERSIDAD MAYOR REAL Y PONTIFICIA DE SAN
FRANCISCO XAVIER DE CHUQUISACA
FACULTAD DE TECNOLOGÍA
SIS (330)



TÍTULO DEL PROYECTO:

Identificación de Plantas Medicinales de Sucre Basado
en Visión de Computador

ESTUDIANTE: Colque García Ariel Rodrigo.

CARRERA: Ing. Ciencias de la Computación.

MATERIA: SIS (330) Desarrollo de aplicaciones Inteligentes.

ÍNDICE

RESUMEN	3
ANTECEDENTES	3
PROBLEMÁTICA	3
ABORDAJE O PROPUESTA DE SOLUCIÓN	3
OBJETIVO GENERAL	3
OBJETIVOS ESPECÍFICOS	3
FUNDAMENTOS TEÓRICOS CONSIDERADOS EN EL TRABAJO	3
A. Ámbito de la inteligencia artificial, técnicas, algoritmos, modelos base, entre otros	3
B. Ámbito al que se aplicó la inteligencia artificial	3
METODOLOGÍA	3
A. Sustento metodológico	3
B. Técnicas de recolección de datos	3
C. Materiales & herramientas	3
2) Esquema y descripción de componentes del software.	4
3) Esquema y descripción de componentes de hardware.	4
4) Esquema y descripción de modelos o componentes inteligentes (esquemas y/o gráficas).	4
5) Valores de parámetros e hiper parámetros aplicados	4
6) Técnicas de depuración aplicadas	4
7) Especificaciones técnicas.	4
8) Lenguajes de programación, frameworks, entre otros.	4
9) Otros aspectos importantes	4
D. Plan de trabajo	4
E. Cronograma	4
RESULTADOS	4
A. Dataset	4
1. Descripción y preprocesamiento realizado (Evidencia del antes y del después)	4
2. Conjunto de entrenamiento, evaluación y validación	4
3. Técnicas, criterio y/o métodos aplicados para la conformación de los conjuntos de datos de entrenamiento, prueba y validación.	4
4. Información adicional que considere importante incluir.	4
B. Resultados de entrenamiento y prueba (Métricas de rendimiento).	4
C. Resultados de aplicación y utilización.	4
D. Otros resultados.	4
CONCLUSIONES	4
RECOMENDACIONES	4
REFERENCIAS BIBLIOGRÁFICAS (Harvard o APA)	4
ANEXOS	4

RESUMEN

La presente propuesta de proyecto se enfoca en el desarrollo de una aplicación de Inteligencia Artificial (IA) diseñada para identificar plantas medicinales de la ciudad de Sucre, Bolivia. Utilizando técnicas de visión por computadora para reconocer de manera precisa y eficiente las diversas especies de plantas medicinales presentes en la región. El objetivo principal es abordar la falta de conocimiento en la ciudad de Sucre, donde el conocimiento sobre las plantas medicinales, que ha sido transmitido principalmente de forma oral, se encuentra en riesgo de perderse con el tiempo, especialmente entre las generaciones más jóvenes. La aplicación tiene como finalidad proporcionar información precisa sobre la identificación, propiedades medicinales y usos de estas plantas, beneficiando así a la población sucreña al mejorar su comprensión y acceso a la salud natural.

ANTECEDENTES

En Sucre, Bolivia, abundan las plantas medicinales utilizadas para tratar diversas dolencias por la población local, sobre todo, por personas [provenientes de las provincias](#). No obstante, la transmisión oral de este conocimiento conlleva el riesgo de pérdida y distorsión con el tiempo.

Además, la mayoría de la población carece de la capacidad para identificar y utilizar correctamente estas plantas, debido a la falta de información o acceso a las mismas, lo que puede resultar en errores, confusiones y riesgos para la salud. Sin embargo, los avances recientes en visión por computadora y aprendizaje automático ofrecen una solución prometedora. Al aprovechar modelos avanzados, es factible desarrollar una aplicación que permita a los usuarios capturar imágenes de plantas medicinales y recibir información instantánea en forma de texto sobre su nombre, descripción, propiedades y usos medicinales.

PROBLEMÁTICA

En Sucre, Bolivia, y en muchos lugares del país, existe una rica tradición de uso de plantas medicinales para tratar enfermedades y promover la salud. Sin embargo, esta tradición se enfrenta a desafíos significativos. La transmisión oral del conocimiento sobre estas plantas a menudo conduce a la pérdida de información y a la confusión en su identificación y uso, lo que puede representar riesgos para la salud. Además, a medida que las generaciones más jóvenes avanzan, se corre el riesgo de que este conocimiento se pierda.

ABORDAJE O PROPUESTA DE SOLUCIÓN

Para abordar este desafío, se desarrollará un modelo de visión por computadora avanzado utilizando YoloV8 específicamente el tamaño YoloV8l-seg, que ofrece un buen equilibrio

entre la precisión y la velocidad. A este modelo se le aplicará Fine Tuning con una extensa base de datos de plantas medicinales de Sucre. Este modelo se encargará de la identificación y clasificación de plantas medicinales cuando los usuarios capturen imágenes de las plantas utilizando una aplicación móvil complementaria.

La aplicación permitirá a los usuarios tomar fotos de plantas o capturarlas en tiempo real y enviarlas al modelo. YOLOv8l procesará estas imágenes y devolverá el nombre de la planta identificada. La aplicación móvil mostrará, junto con el nombre de la planta identificada, una serie de informaciones útiles y relevantes sobre la misma, tales como:

- Una descripción detallada de la planta, incluyendo su aspecto, hábitat, distribución, etc.
- Las propiedades medicinales de la planta, explicando sus efectos, beneficios y mecanismos de acción.
- Las contraindicaciones y precauciones de la planta, advirtiéndole sobre sus posibles efectos secundarios, interacciones, alergias, etc.
- Los usos de la planta, indicando cómo se debe preparar, administrar y consumir la planta para aprovechar sus propiedades medicinales.
- La forma de recolección, conservación y almacenamiento de la planta, dando consejos prácticos para mantener la calidad y la frescura de la planta.
- La forma de cultivo y cuidado de la planta, proporcionando instrucciones para sembrar, regar, podar, abonar, etc., la planta en casa o en un huerto.

La aplicación también permitirá a los usuarios buscar plantas medicinales para tratar dolencias específicas, ingresando el nombre de la dolencia y activando la cámara para buscar la planta en tiempo real.

OBJETIVO GENERAL

Implementar un modelo de Visión por Computador con la finalidad fundamental de identificar las plantas medicinales que prosperan en la ciudad de Sucre. Este enfoque tiene como propósito principal proporcionar información detallada y relevante sobre las propiedades medicinales y usos de estas plantas, contribuyendo así al bienestar y la salud de la comunidad sucrense.

OBJETIVOS ESPECÍFICOS

- ☐ Estudiar el estado del arte en la identificación de plantas medicinales, considerando la precisión, la eficiencia y la innovación de los diferentes enfoques de inteligencia artificial existentes en el proyecto.
- ☐ Realizar un Análisis y Verificación Exhaustiva de Proyectos Previos, Investigar y evaluar proyectos similares que hayan sido implementados para extraer información relevante y aplicar las mejores prácticas en la construcción del proyecto..
- ☐ Desarrollar el modelo de detección y clasificación de plantas medicinales utilizando YOLOv8l-seg.

- ☐ Validación y Evaluación del modelo, diseñando y ejecutando las pruebas de funcionalidad, y evaluando su desempeño con medidas como la precisión, el error y la sensibilidad.

FUNDAMENTOS TEÓRICOS CONSIDERADOS EN EL TRABAJO

- A. Ámbito de la inteligencia artificial, técnicas, algoritmos, modelos base, entre otros.

Modelo de Visión por Computadora: YOLOv8 (You Only Look Once versión 8) para la segmentación de instancias de plantas medicinales. YOLOv8 es un modelo de visión por computadora y una arquitectura de red neuronal profunda que puede realizar la segmentación de instancias, que consiste en separar y etiquetar cada planta individual en una imagen, así como la identificación y la clasificación de objetos, que consiste en reconocer y categorizar las plantas según sus características.

Aprendizaje Automático (Machine Learning): El proyecto se basa en técnicas de transfer learning y fine-tuning, aplicadas a modelos preentrenados de Visión por Computadora. Esto permite aprovechar el conocimiento de modelos ya entrenados en grandes conjuntos de datos para mejorar la precisión en la identificación de plantas medicinales.

Modelos Pre Entrenados: Se aprovechará la ventaja de utilizar modelos base pre entrenados con grandes conjuntos de datos como YoloV8. Esto acelera el proceso de entrenamiento y mejora la precisión del sistema, ya que estos modelos ya han aprendido características generales de imágenes.

- B. Ámbito al que se aplicó la inteligencia artificial.

La implementación de **YOLOv8** en este proyecto se centra en la identificación de plantas medicinales de Sucre. **YOLOv8** se utiliza en el proceso de entrenamiento mediante la alimentación de un extenso conjunto de imágenes de plantas medicinales autóctonas de Sucre. Durante este proceso, el modelo va ajustando sus parámetros internos para aprender y reconocer con precisión las características específicas de estas plantas en las imágenes.

Una vez entrenado, **YOLOv8** es capaz de procesar imágenes, detectando las plantas medicinales en ellas. El modelo identifica y clasifica estas plantas, proporcionando información detallada sobre sus atributos, propiedades medicinales y aplicaciones tradicionales

METODOLOGÍA

A. Sustento metodológico

La metodología se ha diseñado meticulosamente para alcanzar los objetivos específicos del proyecto y garantizar la precisión en la identificación de las plantas medicinales. Está compuesta por las siguientes etapas fundamentales:

1. Recopilación de Datos Fotográficos:

Se llevará a cabo una recolección exhaustiva de datos visuales, en forma de fotografías de plantas medicinales autóctonas de Sucre. Estas imágenes formarán el conjunto de datos principal que será empleado en el entrenamiento del modelo YOLOv8. Se prestará una atención especial a la calidad de las imágenes y a la diversidad de especies representadas.

2. Preprocesamiento de Datos Visuales:

Las imágenes recolectadas pasarán por un proceso de preprocesamiento que incluye estandarización del tamaño, eliminación de imágenes borrosas y la aplicación de técnicas de Data Augmentation. Esto tiene como objetivo mejorar la calidad y consistencia de las imágenes, lo que permitirá al modelo YOLOv8 generalizar y reconocer variaciones en las plantas medicinales.

3. Entrenamiento del Modelo YOLOv8:

El modelo YOLOv8 será empleado para llevar a cabo el entrenamiento de una red neuronal profunda. Durante esta fase, el modelo aprenderá de manera autónoma a identificar y clasificar las plantas medicinales presentes en las imágenes. El entrenamiento se repetirá hasta alcanzar un nivel óptimo de precisión.

4. Validación y Ajustes Finales:

Después de finalizar el entrenamiento, se llevará a cabo una validación del modelo utilizando un conjunto de datos independiente. Se realizarán ajustes necesarios para mejorar la precisión y reducir la posibilidad de falsos positivos o negativos.

5. Desarrollo de la Aplicación Móvil:

Luego del entrenamiento del modelo, se avanzará en el desarrollo de la aplicación móvil. Esta aplicación permitirá a los usuarios capturar imágenes de plantas medicinales en su entorno y enviarlas al modelo entrenado para su identificación. La interfaz de la aplicación se diseñará priorizando la intuición y la facilidad de uso.

B. Técnicas de recolección de datos

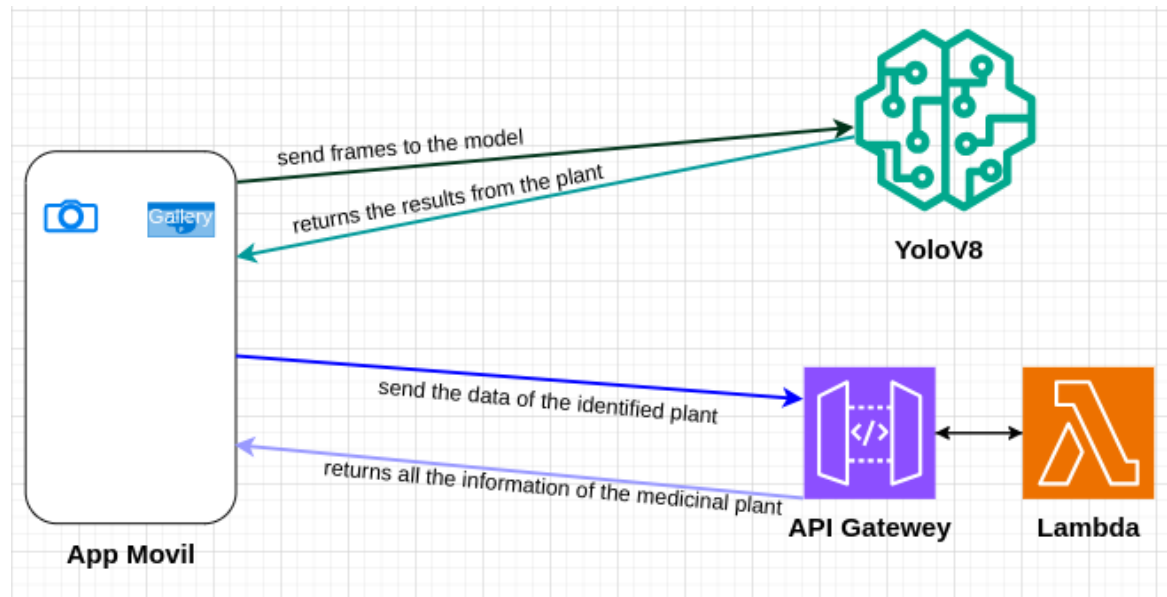
En la ejecución de este proyecto, se emplearán diversas técnicas de recolección de datos con el propósito de obtener información valiosa sobre las plantas medicinales de Sucre. Estas técnicas se han centrado en la obtención de datos visuales y descriptivos relacionados con las plantas medicinales. Entre las técnicas utilizadas se incluyen:

- **Fotografía de Campo:** Se llevará a cabo la captura de imágenes de plantas medicinales en su entorno natural. Esto involucra la toma de fotografías de plantas específicas, incluyendo sus hojas, flores, tallos y otros atributos identificativos.
- **Investigación Documental:** Se consultará una variedad de fuentes bibliográficas y recursos en línea para recopilar información detallada sobre las plantas medicinales de la región, incluyendo sus propiedades, usos tradicionales y características distintivas.
- **Entrevistas y Encuestas:** Se llevará a cabo entrevistas y encuestas a nivel local para recopilar información anecdótica y experiencial sobre el uso de plantas medicinales. Esto ha permitido enriquecer la base de datos con conocimiento práctico transmitido de generación en generación.
- **Bases de Datos Existentes:** Se recurrirá a bases de datos existentes sobre plantas medicinales, para enriquecer el conjunto de datos.

Estas técnicas de recolección de datos se combinarán para obtener un conjunto de información sobre las plantas medicinales de Sucre.

C. Materiales & herramientas

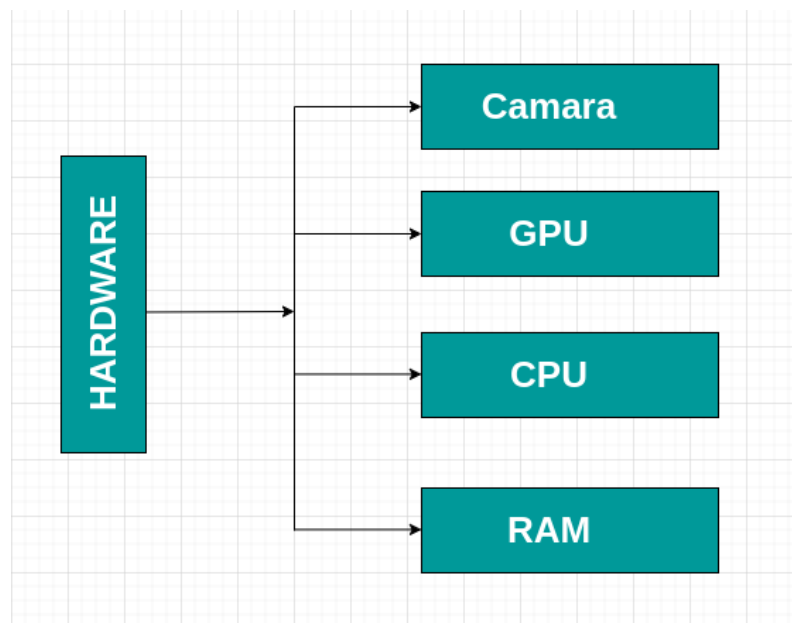
1) Arquitectura software desarrollado.



- El componente **aplicación móvil** es el software desarrollado con **Flutter**, un framework de código abierto para crear aplicaciones nativas para iOS y Android. Esta aplicación recibe la imagen desde la cámara o galería, la procesa y la envía al modelo YoloV8. Además, muestra los resultados y la información de las plantas medicinales al usuario.

- El componente del modelo **YoloV8** está integrado en la aplicación móvil y es una red neuronal convolucional profunda. Recibe las imágenes procesadas desde la aplicación y devuelve los resultados de detección y clasificación de las plantas medicinales.
- La API, construida con servicios de **AWS, Lambda y API Gateway**, almacena información sobre las plantas medicinales, como sus nombres, descripciones, formas de uso, contraindicaciones, entre otros. La API recibe el o los nombres de las plantas desde la aplicación Flutter y proporciona información detallada sobre la planta solicitada.

2) Esquema y descripción de componentes de hardware.

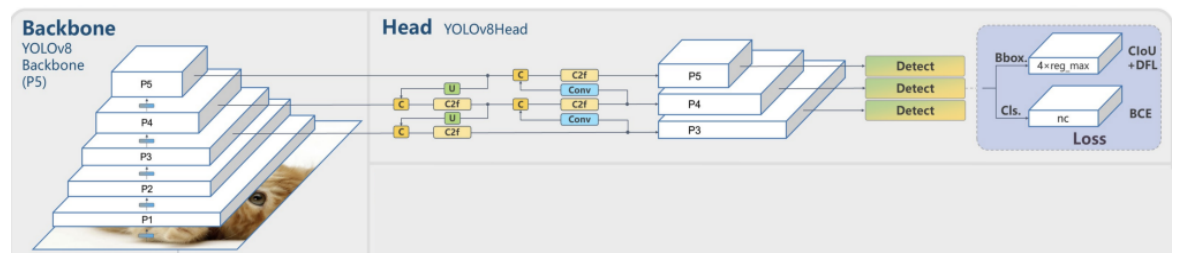


- **Cámara (Cam):** La cámara es el componente de entrada que captura imágenes en tiempo real o imágenes estáticas desde la galería. La aplicación utiliza la cámara para adquirir los datos visuales de la planta medicinal que se va a analizar.
- **CPU (Unidad Central de Procesamiento):** La CPU, el cerebro de nuestro dispositivo, es responsable de procesar los datos capturados por la cámara. Realiza tareas como la conversión de imágenes en datos digitales, el control de flujo de la aplicación y el procesamiento inicial de las imágenes.

- **GPU (Unidad de Procesamiento Gráfico):** La GPU es un componente esencial para acelerar la detección de objetos en tiempo real. Aprovecha su capacidad de procesamiento paralelo para ejecutar operaciones intensivas de análisis de imágenes, como la detección de plantas medicinales. Esto permite una respuesta rápida y suave al usuario al mostrar resultados en tiempo real.
- **RAM (Memoria de Acceso Aleatorio):** La RAM proporciona un espacio de memoria temporal para almacenar datos críticos durante el procesamiento. Los datos de imágenes procesados, los resultados de la detección y otra información relevante se almacenan en la RAM para su acceso rápido. Esto es esencial para la eficiencia de la aplicación en tiempo real.

3) Esquema y descripción de modelos o componentes inteligentes (esquemas y/o gráficas).

Arquitectura de YoloV8



Backbone (Espina dorsal):

El backbone se refiere a la parte de la red neuronal que extrae características de la entrada (como una imagen) y crea una representación de características.

En el contexto de YOLOv8, el backbone es responsable de procesar la entrada y aprender características a diferentes niveles de abstracción.

Los modelos preentrenados, como ResNet, Xception o MobileNet, a menudo se utilizan como backbones en YOLOv8 para aprovechar su capacidad de extracción de características.

Head (Cabeza):

La cabeza es la parte de la red que sigue al backbone y se encarga de realizar tareas específicas, como detección de objetos o segmentación.

En YOLOv8, la cabeza incluye capas adicionales que procesan las características extraídas por el backbone.

La cabeza puede variar según la tarea: por ejemplo, en detección de objetos, la cabeza genera cajas delimitadoras y puntajes de confianza para cada objeto detectado.

4) Valores de parámetros e hiper parámetros aplicados

- Tazas de aprendizaje
 - $lr_0 = 0.01$ → es el punto de partida para la tasa de aprendizaje.
 - $lr_f = 0.01$ → es el punto final hacia el cual se reduce gradualmente la tasa de aprendizaje.
 -
- epochs = 20: Define el número de épocas de entrenamiento, en este caso, se establece en 20.
- batch = 16: Especifica el tamaño del lote (batch size) utilizado durante el entrenamiento. En este caso, se utiliza un tamaño de lote de 16.
- imgsz = 640: Indica el tamaño de las imágenes de entrada durante el entrenamiento (640x640 píxeles en este caso)..
- pretrained = True: Indica que se utilizará una versión preentrenada del modelo como punto de partida.

5) Especificaciones técnicas.

Para desarrollar este proyecto se utilizan las siguientes dependencias:

➤ **Bibliotecas y Dependencias:**

Para entrenar el modelo:

- ☐ Ultralytics 8.0.28
- ☐ Python 3.10.12
- ☐ Roboflow
- ☐ Flutter 3.13.6
- ☐ torchvision 0.16.0
- ☐ onnx 1.15.0

Para desarrollar la aplicación:

- ☐ cupertino_icons: ^1.0.2
- ☐ camera: ^0.10.5+5
- ☐ flutter_vision: ^1.1.4
- ☐ image_picker: ^1.0.4
- ☐ flutter_speed_dial: ^7.0.0

➤ **Google Colab:** Versión actual en la nube (se accede vía navegador)

➤ **Otros:**

- Se utiliza Google Drive para el almacenamiento y acceso a los datos.
- Se integra AWS específicamente con los servicios de Lambda y API Gateway para almacenar informaciones de las plantas medicinales.

6) Lenguajes de programación, frameworks, entre otros.

Lenguaje de Programación y Marcos de Trabajo:

- *Python* es el lenguaje principal utilizado en el desarrollo del proyecto. Python se destaca en el campo de la inteligencia artificial y el

aprendizaje automático debido a su amplia gama de bibliotecas y frameworks disponibles.

- Flutter es un marco de desarrollo de código abierto de Google que utiliza el lenguaje de programación Dart para crear aplicaciones multiplataforma con interfaces de usuario atractivas y fluidas. Permite desarrollar aplicaciones para Android, iOS, web y escritorio, todo desde una base de código común.

Framework de Aprendizaje Profundo: El proyecto se basa en PyTorch, un framework de aprendizaje profundo ampliamente reconocido en Python. PyTorch es conocido por su flexibilidad y facilidad de uso, lo que lo convierte en una elección ideal para desarrollar modelos de identificación de plantas medicinales.

Entorno de Desarrollo:

- Para el desarrollo y entrenamiento de los modelos, se empleará [Google Colab](#). Colab es una plataforma de cuaderno colaborativo en línea que permite ejecutar código Python en los servidores de Google, proporcionando recursos computacionales necesarios para proyectos de aprendizaje automático.
- Visual Studio Code (VS Code) es un editor de código fuente gratuito y altamente personalizable desarrollado por Microsoft. Es ampliamente utilizado por programadores y desarrolladores para escribir, editar y depurar código en una variedad de lenguajes de programación.

Gestión de Datos: [Roboflow](#) es la herramienta clave para organizar y preparar los datos de imágenes del proyecto. Roboflow simplifica la carga, el etiquetado y el preprocesamiento de datos, lo que resulta fundamental en el desarrollo de modelos de detección de plantas medicinales en Sucre.

Plan de trabajo

1. **Desarrollar el Dataset de Plantas Medicinales:** Recopilar, organizar y etiquetar un conjunto de datos que contenga imágenes de plantas medicinales autóctonas de Sucre, junto con sus correspondientes etiquetas de clasificación.
2. **Preparación de Datos:** Utilizar la herramienta Roboflow para las imágenes, especificando las clases de las plantas en cada imagen, y aplicar técnicas de aumento de datos para enriquecer el dataset.
3. **Implementación del Modelo YoloV8 para el Entrenamiento:** Configurar e implementar el modelo YoloV8 en un entorno de entrenamiento como Google Colab, preparando el sistema para aprender a identificar plantas medicinales en las imágenes.

4. **Realización de Pruebas al Modelo Entrenado:** Evaluar el modelo entrenado mediante pruebas exhaustivas, verificando su capacidad para identificar con precisión plantas medicinales en imágenes de prueba.
5. **Diseño de una Interfaz para Android:** Desarrollar una interfaz de usuario en el entorno de Visual Studio Code (VSCode) que sea compatible con dispositivos Android, permitiendo a los usuarios interactuar con la aplicación.
6. **Creación e Implementación de una API con Especificaciones de Plantas Medicinales:** Diseñar, Integrar y desarrollar una API que contenga información detallada sobre las plantas medicinales presentes en el dataset, incluyendo sus propiedades medicinales y usos.
7. **Realización de Pruebas en la APP Desarrollada:** Ejecutar pruebas en la aplicación móvil completa, asegurándose de que funcione correctamente, tanto en términos de identificación de plantas como en la interacción con la API de plantas medicinales.

D. Cronograma

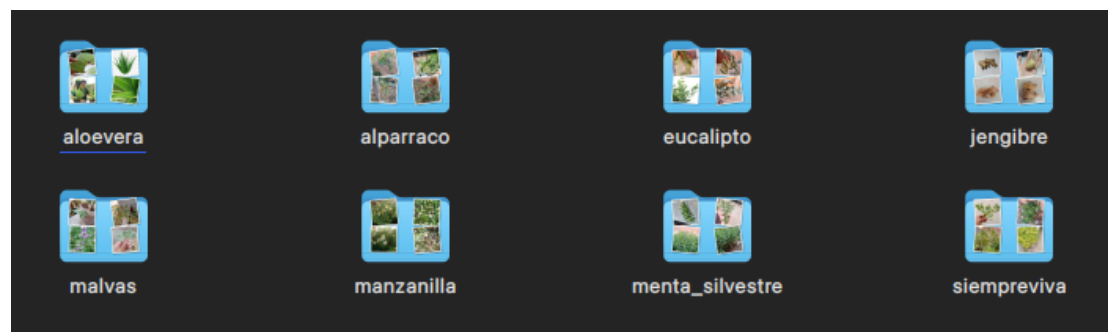
Fases del Proyecto	Se1	Se2	Se3	Se4	Se5	Se6	Se7
Desarrollar el Dataset de Plantas Medicinales	X						
Preparación de Datos		X					
Implementación del Modelo YoloV8 para el Entrenamiento			X				
Realización de Pruebas al Modelo Entrenado				X			
Diseño de una Interfaz en VSCode para Android					X		
Creación e Implementación de una API con Especificaciones de Plantas Medicinales						X	
Realización de Pruebas en la APP Desarrollada							X

RESULTADOS

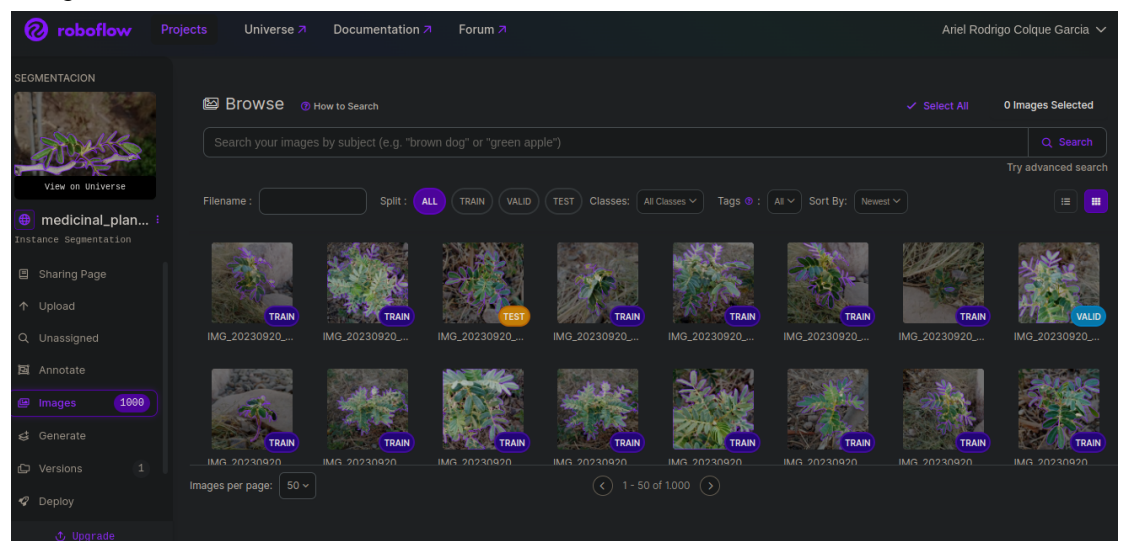
A. Dataset

1. Descripción y preprocesamiento realizado (Evidencia del antes y del después)

El preprocesamiento de los datos se llevaron a cabo en varias etapas. Inicialmente, se realizó una exhaustiva investigación para identificar las plantas medicinales de la ciudad de Sucre, que formarían parte del dataset. Posteriormente, se emprendió la tarea de adquirir al menos 1000 imágenes por clase de planta necesarias para el proyecto, visitando zonas rurales y jardines locales para fotografiar estas plantas. Una vez recopiladas las imágenes, se procedió a su clasificación según las diferentes clases de plantas medicinales, lo que marcó el inicio del proceso de preprocesamiento de los datos.



Clases de imágenes obtenidas, cada una cuenta con más de 1000 imágenes fotografiadas.

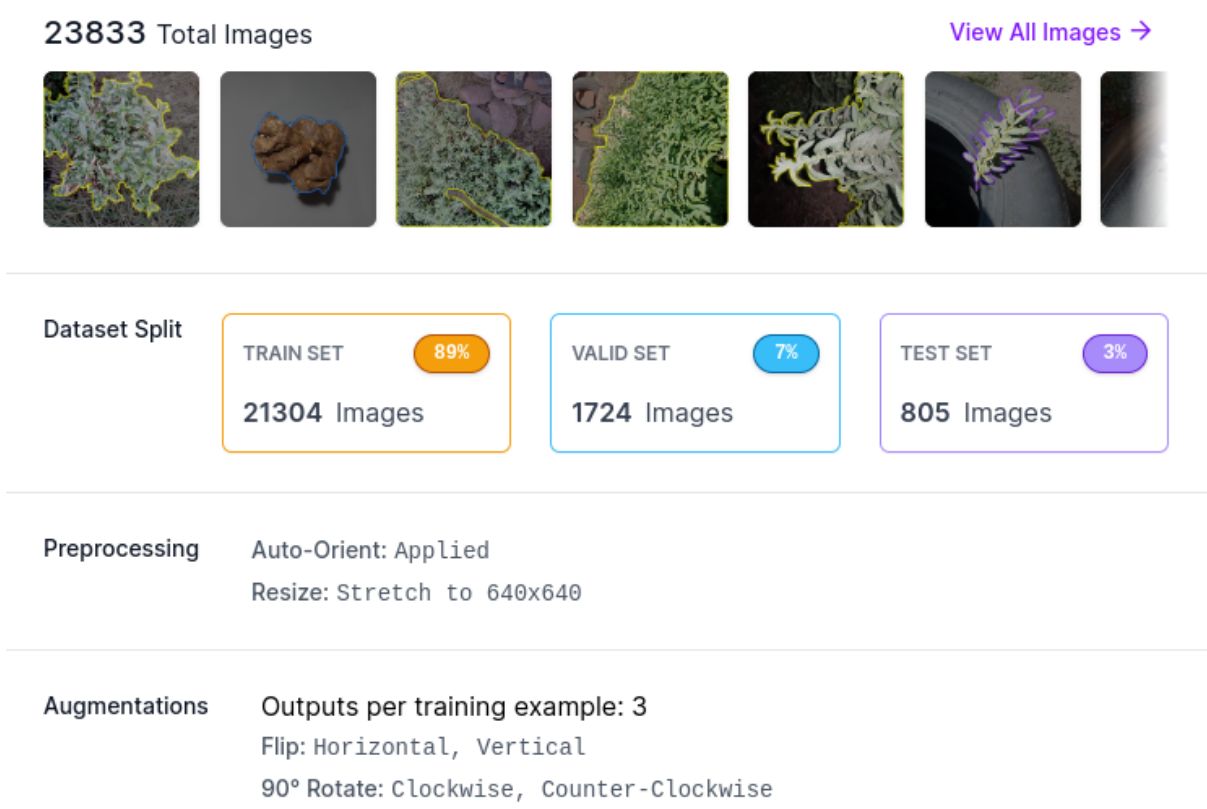


Una vez obtenidas las imágenes, se optó por emplear la herramienta de **Roboflow** para llevar a cabo la segmentación de las mismas. Esta elección se basó en su capacidad para facilitar el proceso de anotación de objetos en las

imágenes, permitiendo una segmentación precisa de las diferentes partes de las plantas medicinales garantizando que el conjunto de datos estuviera adecuadamente etiquetado y listo para su uso en el entrenamiento del modelo de visión por computadora.

2. Conjunto de entrenamiento, validación y prueba

En total se logró obtener 23833 imágenes para el dataset de lo cual se divide en tres conjuntos distintos: entrenamiento, validación y prueba. Se asigna un 87% (14313) de las imágenes al conjunto de entrenamiento, un 8% (1364) al conjunto de validación y un 4% (681) al conjunto de prueba. Posteriormente se aplicó Data Augmentation logrando así obtener más de **85000** imágenes.



3. Técnicas, criterio y/o métodos aplicados para la conformación de los conjuntos de datos de entrenamiento, prueba y validación.

Se incluyeron el uso de las opciones de Roboflow para realizar el aumento de datos. A través de la versión gratuita de esta herramienta, logramos duplicar nuestro conjunto inicial de 1500 imágenes, generando un total de 3000 imágenes. Esta estrategia de aumento de datos contribuyó significativamente a enriquecer y diversificar el conjunto de datos, mejorando así la capacidad del modelo para generalizar y reconocer las variaciones en las plantas medicinales.

Preprocessing Auto-Orient: Applied
Resize: Stretch to 640x640

Augmentations Outputs per training example: 3
Flip: Horizontal
90° Rotate: Clockwise, Counter-Clockwise
Crop: 0% Minimum Zoom, 20% Maximum Zoom
Shear: $\pm 15^\circ$ Horizontal, $\pm 15^\circ$ Vertical

4. Información adicional que considere importante incluir.

Se recolectaron inicialmente 1500 imágenes por cada clase de plantas medicinales, como se había planificado. Sin embargo, dado que este conjunto inicial de imágenes no proporcionaba suficiente diversidad para un entrenamiento efectivo del modelo, se aplicaron técnicas de Data Augmentation de manera uniforme a todas las clases. Esto permitió enriquecer el conjunto de datos, generando un total de al menos 3000 imágenes por clase después de la aplicación de Data Augmentation. Este enfoque aseguró la disponibilidad de un conjunto de datos más completo y equilibrado para el entrenamiento del modelo.

B. Resultados de entrenamiento y prueba (Métricas de rendimiento).

Class	Images	Instances	Box(P)	Box(R)	mAP50	mAP50-95	Mask(P)	Mask(R)	mAP50	mAP50-95
all	1341	2489	0.819	0.774	0.836	0.754	0.826	0.776	0.839	0.703
alparrac o	1341	324	0.795	0.779	0.855	0.737	0.811	0.787	0.857	0.637
eucalipt o	1341	284	0.886	0.905	0.934	0.871	0.892	0.905	0.933	0.85
jengibre	1341	192	0.985	0.984	0.994	0.987	0.985	0.984	0.994	0.984
malva	1341	528	0.678	0.532	0.638	0.51	0.695	0.538	0.656	0.44

manzanilla	1341	645	0.64	0.438	0.559	0.425	0.644	0.437	0.557	0.363
menta	1341	230	0.884	0.891	0.927	0.861	0.878	0.883	0.923	0.784
siempre viva	1341	286	0.866	0.892	0.946	0.885	0.873	0.899	0.952	0.86

Box(P, R, mAP50, mAP50-95): Estas métricas se refieren a la detección de cajas delimitadoras que rodean a los objetos de interés. P es la precisión, que mide la proporción de cajas detectadas que son correctas. R es el recall, que mide la proporción de cajas verdaderas que son detectadas. mAP50 es el promedio de precisión para un umbral de IoU de 0.5, que es una medida de la superposición entre las cajas detectadas y las verdaderas. mAP50-95 es el promedio de precisión para diferentes umbrales de IoU desde 0.5 hasta 0.95, con un incremento de 0.05.

Mask(P, R, mAP50, mAP50-95): Estas métricas se refieren a la segmentación de instancias, que consiste en asignar un píxel a una clase y a una instancia de objeto. Las métricas son las mismas que para las cajas, pero se aplican a las máscaras de segmentación en lugar de a las cajas delimitadoras.

Precisión media de Cajas:

$$\text{Precisión media} = \frac{\sum_{i=1}^n \text{Precisión de la clase } i}{n}$$

$$\text{Precisión media} = \frac{0.795 + 0.886 + 0.985 + 0.678 + 0.64 + 0.884 + 0.866}{7} = 0.819$$

Recall media de Cajas:

$$\text{Recall medio} = \frac{\sum_{i=1}^n \text{Recall de la clase } i}{n}$$

$$\text{Recall medio} = \frac{0.779 + 0.905 + 0.984 + 0.532 + 0.438 + 0.891 + 0.892}{7} = 0.774$$

Precisión media de las Máscaras:

$$\text{Precisión media} = \frac{0.811 + 0.892 + 0.985 + 0.695 + 0.644 + 0.878 + 0.873}{7} = 0.826$$

Recall media de las Máscaras:

$$\text{Recall medio} = \frac{0.787 + 0.905 + 0.984 + 0.538 + 0.437 + 0.883 + 0.899}{7} = 0.776$$

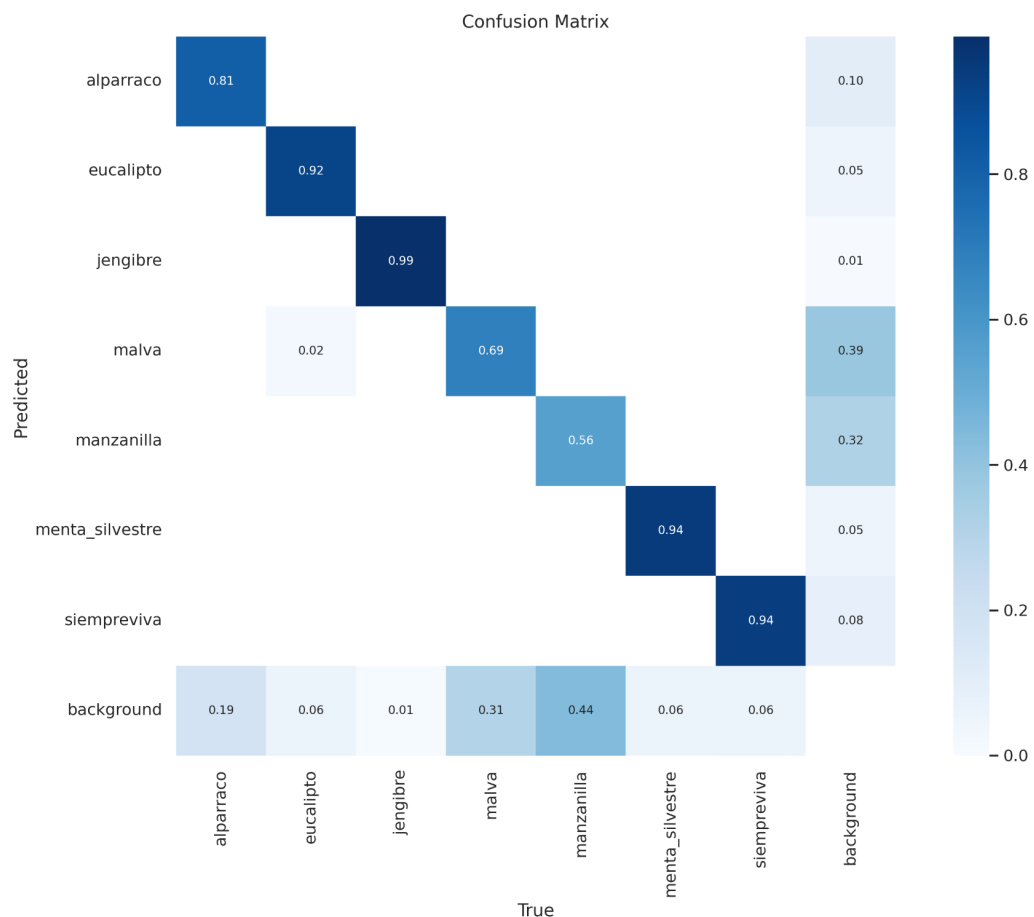
Precisión Total:

$$\text{Precisión total} = \frac{\text{Precisión de cajas} + \text{Precisión de máscaras}}{2} = \frac{0.819 + 0.826}{2} = 0.823$$

Recall Total:

$$\text{Recall total} = \frac{\text{Recall de cajas} + \text{Recall de máscaras}}{2} = \frac{0.774 + 0.776}{2} = 0.775$$

i. Matriz de confusión

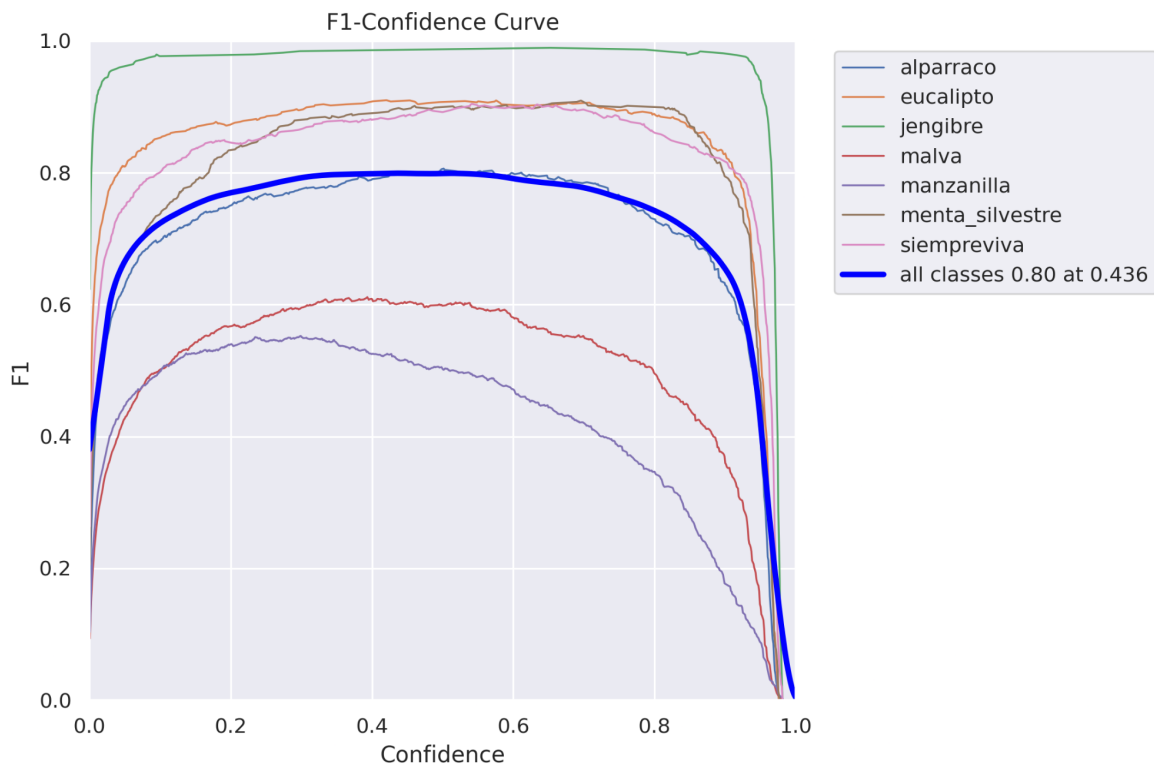


La **matriz de confusión** muestra cómo de bien clasifica el modelo las imágenes según las clases que tienen. Cada clase tiene una fila y una columna en la matriz. Los valores de la diagonal son los que el modelo acierta, y los valores fuera de la diagonal son los que el modelo se equivoca.

Resumen de los resultados de el modelo para cada clase:

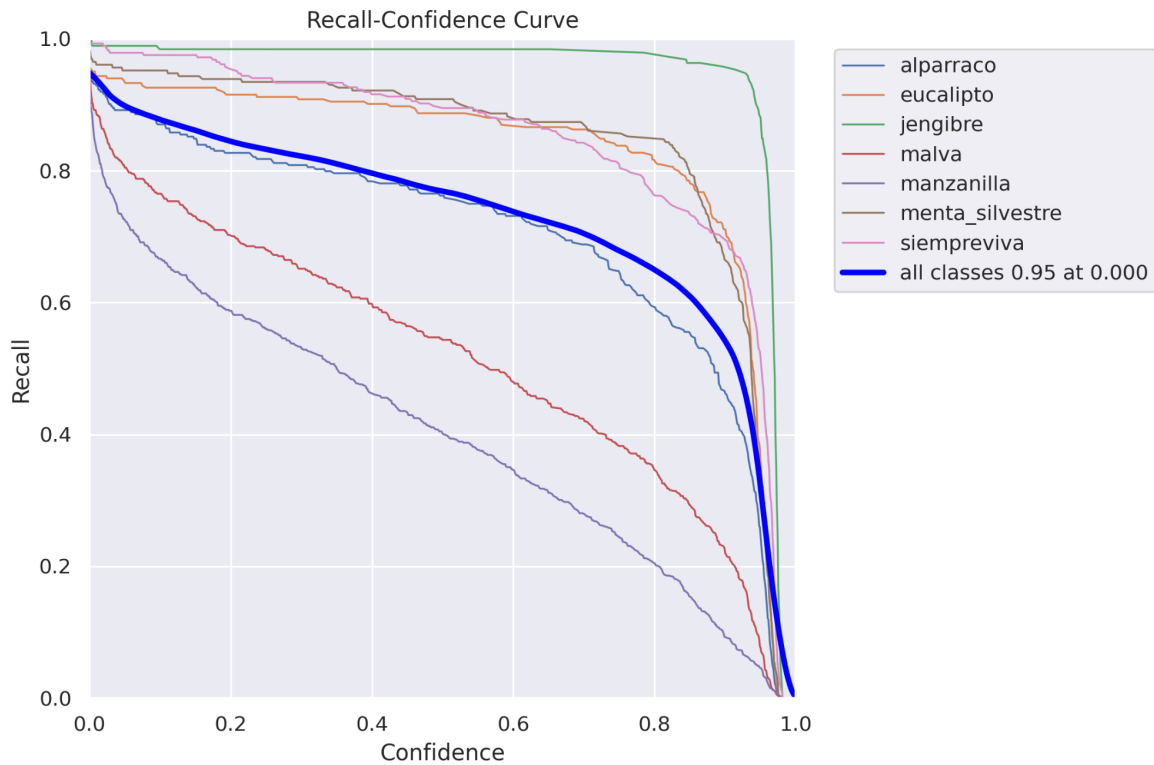
- **Alparraco:** El modelo acierta el 81% de las imágenes que tienen alparracos, y se equivoca el 19% de las que no los tienen. También se equivoca el 10% de las que los tienen y dice que no los tienen.
- **Eucalipto:** El modelo acierta el 92% de las imágenes que tienen eucaliptos, y se equivoca el 6% de las que no los tienen. También se equivoca el 5% de las que los tienen y dice que no los tienen.
- **Jengibre:** El modelo acierta el 99% de las imágenes que tienen jengibre, y se equivoca el 1% de las que no lo tienen. También se equivoca el 1% de las que lo tienen y dice que no lo tienen.
- **Malva:** El modelo acierta el 69% de las imágenes que tienen malvas, y se equivoca el 31% de las que no las tienen. También se equivoca el 39% de las que las tienen y dice que no las tienen.
- **Manzanilla:** El modelo acierta el 56% de las imágenes que tienen manzanillas, y se equivoca el 44% de las que no las tienen. También se equivoca el 32% de las que las tienen y dice que no las tienen.
- **Menta Silvestre:** El modelo acierta el 94% de las imágenes que tienen menta, y se equivoca el 6% de las que no la tienen. También se equivoca el 5% de las que la tienen y dice que no la tienen.
- **Siempreviva:** El modelo acierta el 94% de las imágenes que tienen siempreviva, y se equivoca el 6% de las que no la tienen. También se equivoca el 8% de las que la tienen y dice que no la tienen.
- **Background:** El modelo no acierta ninguna imagen que tenga background, y se equivoca el 100% de las que no lo tienen. También se equivoca el 80% de las que lo tienen y dice que no lo tienen.

ii. **F1 confidence curve**



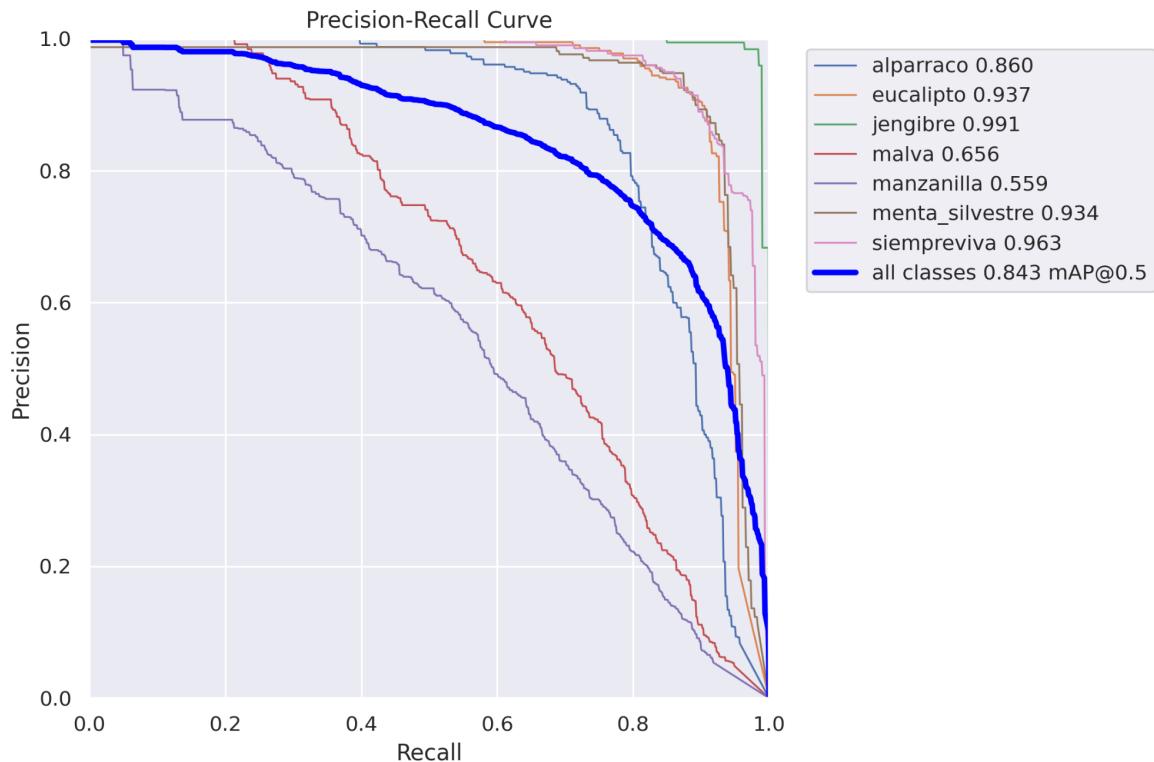
Esta métrica muestra cómo varía el valor **F1** del modelo al cambiar el umbral de confidencialidad. El valor **F1** es una medida de la calidad de la detección y segmentación de instancias, que combina la precisión y el recall del modelo. La confidencialidad es la probabilidad de que el modelo detecte correctamente los objetos. El punto óptimo de la curva **F1** confidence es el que maximiza el valor F1 del modelo. En mi caso, el punto óptimo se alcanza con una confidencialidad de 0.436 y un valor F1 de 0.80. Esto significa que tu modelo tiene una precisión y un recall de 0.80.

iii. Recall confidence curve



La **métrica recall confidence curve** es una forma de visualizar cómo varía el recall del modelo al cambiar el umbral de confidencialidad. El recall es la proporción de objetos reales que detecta y segmenta correctamente el modelo. La confidencialidad es la probabilidad de que el modelo detecte correctamente los objetos. El punto óptimo de la curva recall confidence es el que maximiza el recall del modelo. En tu caso, el punto óptimo se alcanza con una confidencialidad de 0.000 y un recall de 0.95. Esto significa que mi modelo detecta y segmenta correctamente el 95% de los objetos reales en la imagen, sin importar la confidencialidad de las predicciones.

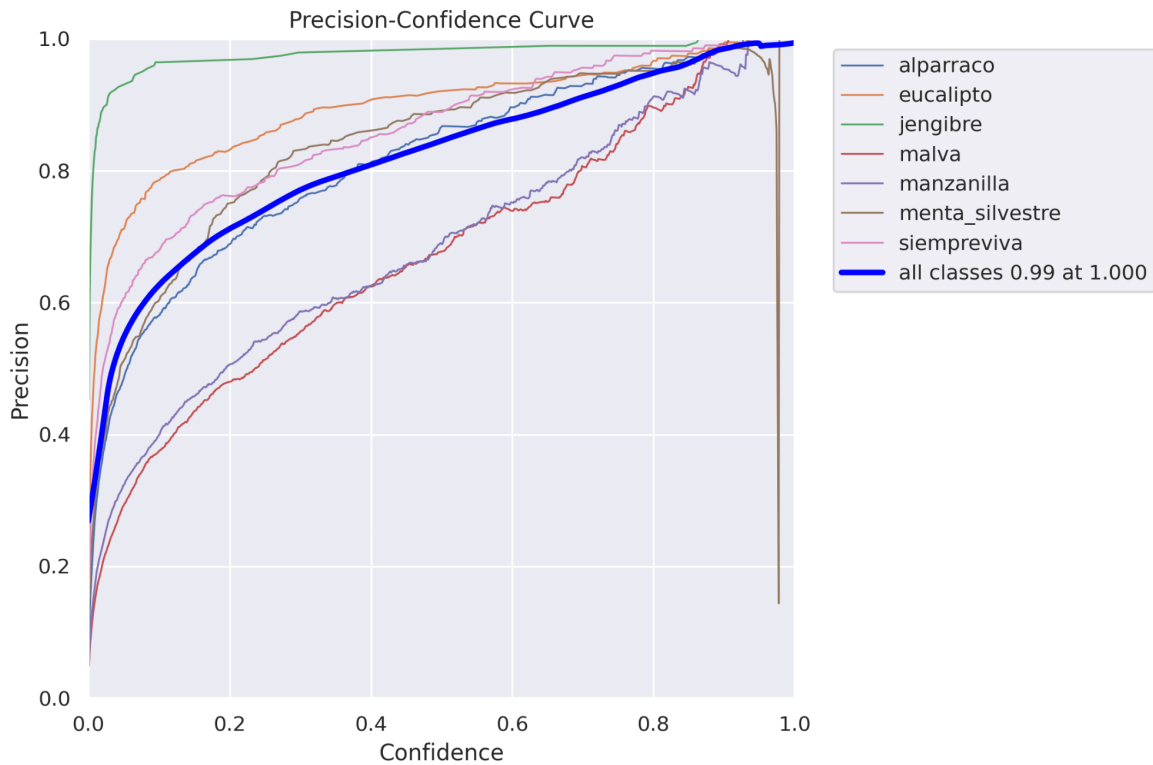
iv. Precision recall curve



Esta métrica muestra la precisión y el recall del modelo para diferentes umbrales de confidencialidad. La precisión es la proporción de objetos correctos entre los predichos. El recall es la proporción de objetos correctos entre los reales. La curva Precision recall muestra cómo se relacionan la precisión y el recall a medida que se cambia el umbral de confidencialidad. El área bajo la curva (AUC) es una medida que resume la calidad de la curva. Un AUC alto indica que el modelo tiene una buena precisión y un buen recall para diferentes umbrales de confidencialidad.

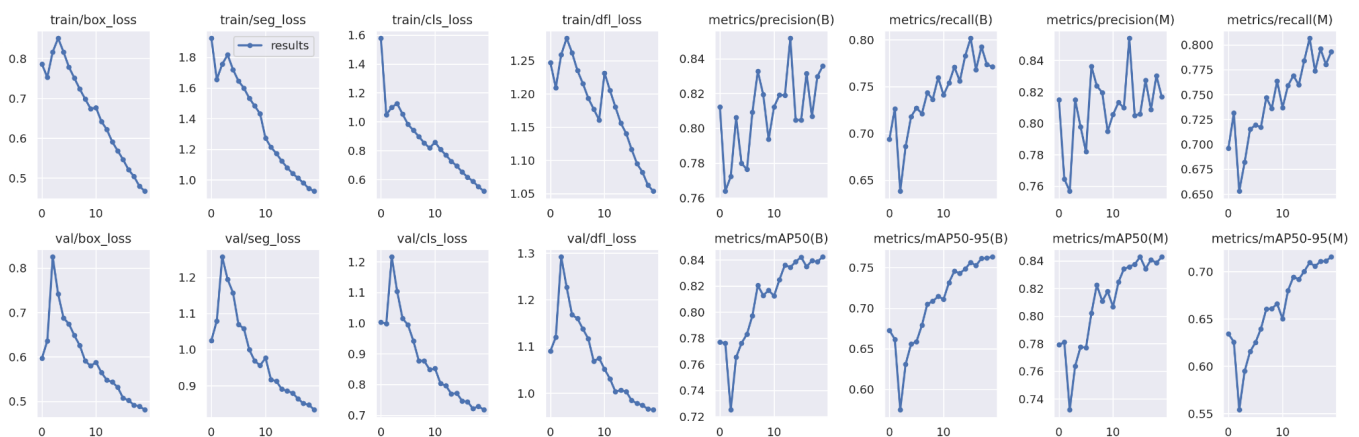
El valor 0.843 mAP@0.5 indica que el modelo tiene un AUC de 0.843 con un umbral de confidencialidad de 0.5. Esto significa que el modelo tiene una buena precisión y un buen recall con ese umbral.

v. Precision-Confidence curve



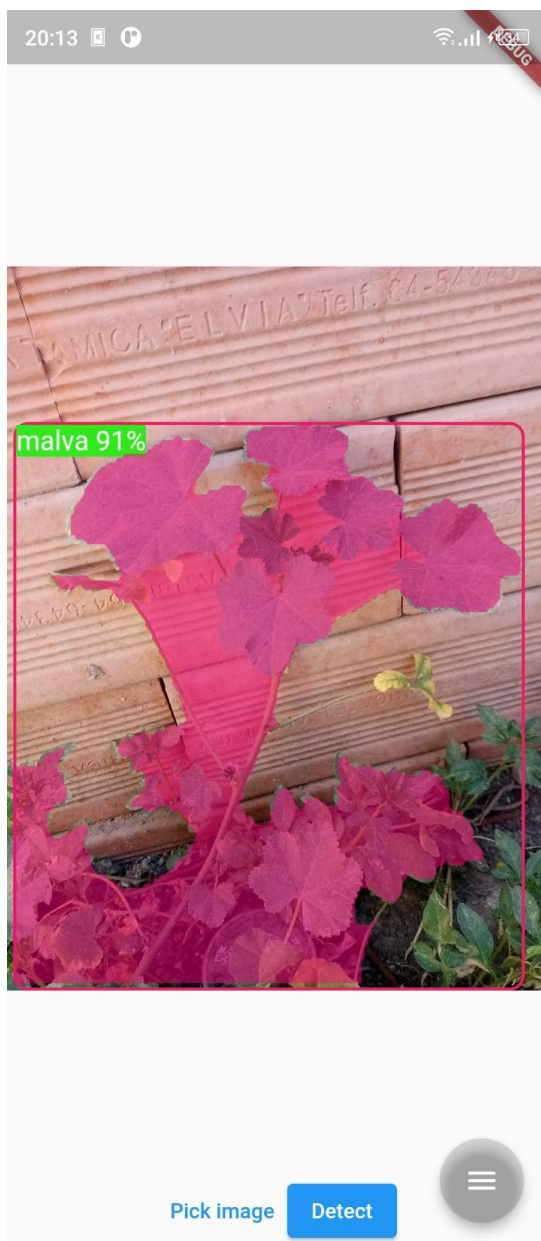
Esta métrica muestra la precisión del modelo para diferentes umbrales de confidencialidad. La precisión es la proporción de objetos correctos entre los predichos. La confidencialidad es la probabilidad de que el modelo detecte correctamente los objetos. El punto óptimo es el que maximiza la precisión. En mi caso, el punto óptimo se alcanza con una confidencialidad de 1.000 y una precisión de 0.99. Esto significa que el modelo tiene una precisión casi perfecta con ese umbral.

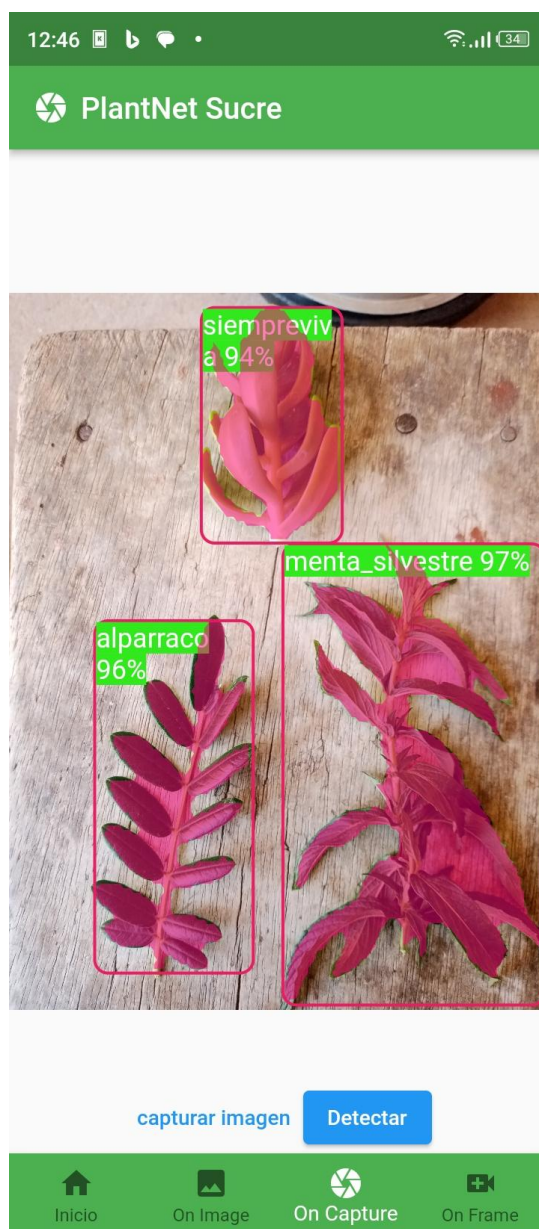
vi. Resultados



C. Resultados de aplicación y utilización.

Se realizaron pruebas con el modelo entrenado e implementado a Flutter con imágenes cargadas desde la galería, imágenes capturadas con la cámara y en tiempo real.





CONCLUSIONES

El desarrollo de este proyecto se enfocó en alcanzar cuatro objetivos fundamentales:

- 1. Estudio del Estado del Arte en la Identificación de Plantas Medicinales:**
 Durante la investigación y revisión del estado del arte en la identificación de plantas medicinales, se ha logrado una comprensión profunda de las técnicas existentes en la aplicación de inteligencia artificial para este propósito. Esto proporcionó una base sólida para el desarrollo del modelo.

- **2. Creación de un Dataset de Plantas Medicinales de Sucre:**

La creación de un conjunto de datos específico de plantas medicinales de Sucre ha sido un logro esencial. La recopilación y preparación de imágenes de estas plantas sienta las bases para el entrenamiento efectivo de nuestro modelo.

- **3. Desarrollo del Modelo de Detección y Clasificación de Plantas Medicinales con YOLOv8:**

La implementación y entrenamiento del modelo YOLOv8 para la detección y clasificación de plantas medicinales es un paso significativo en el proyecto. El modelo ha demostrado su capacidad para procesar imágenes y realizar tareas precisas de identificación y clasificación.

- **4. Validación y Evaluación del Modelo:**

Los resultados obtenidos al evaluar el modelo son prometedores. La métrica de precisión muestra un alto rendimiento, destacando la efectividad del modelo en su tarea.

RECOMENDACIONES

Para optimizar el rendimiento del sistema y garantizar su utilidad efectiva, se sugieren las siguientes recomendaciones:

1. *Ajuste Continuo y Entrenamiento del Modelo:*

Implementar una estrategia de ajuste constante de hiperparámetros y entrenamiento del modelo YOLOv8 para mejorar su precisión y eficiencia en la detección de plantas medicinales.

2. *Ampliación del Dataset con Variedad de Escenarios:*

Incrementar gradualmente el dataset, incorporando imágenes de plantas medicinales en diversas condiciones y entornos. Una mayor diversidad en el conjunto de datos puede mejorar la capacidad del modelo para adaptarse a situaciones variadas.

3. *Investigación Continua sobre Integración con Flutter:*

Explorar continuamente bibliotecas y actualizaciones para la integración de YOLOv8m con Flutter, priorizando la mejora en la calidad de la segmentación en tiempo real y la optimización de recursos, especialmente en dispositivos Android.

4. *Experimentación con Técnicas de Preprocesamiento de Imágenes:*

Realizar experimentos con diversas técnicas de preprocesamiento de imágenes para mejorar la calidad de las entradas del modelo, especialmente cuando se enfrenta a imágenes de baja calidad.

REFERENCIAS BIBLIOGRÁFICAS (Harvard o APA)

- Ultralytics. (2021). Train. Recuperado el 10 de noviembre de 2021, de <https://docs.ultralytics.com/modes/train/>

- Ultralytics. (2021). Command Line Interface. Recuperado el 10 de noviembre de 2021, de <https://docs.ultralytics.com/usage/cli/>
- Ultralytics. (2021). YOLOv8 - Ultralytics/yolov8. Recuperado el 10 de noviembre de 2021, de <https://github.com/ultralytics/yolov8/issues/1325>
- Flutter. (2021). tflite | Flutter Package. Recuperado el 10 de noviembre de 2021, de [\[https://pub.dev/packages/tflite/install\]](https://pub.dev/packages/tflite/install)
- Faisal, M. M. (2021). MuhammadMoinFaisal (Muhammad Moin Faisal) · GitHub. Recuperado el 10 de noviembre de 2021, de [\[https://github.com/MuhammadMoinFaisal\]](https://github.com/MuhammadMoinFaisal)
- Nielsen, N. (2021). niconielsen32/YOLOv8-Class: A class for running YOLOv8 inference on images and videos. Recuperado el 10 de noviembre de 2021, de [\[https://github.com/niconielsen32/YOLOv8-Class/blob/main/YOLOv8InferenceClass.py\]](https://github.com/niconielsen32/YOLOv8-Class/blob/main/YOLOv8InferenceClass.py)
- Ultralytics. (2021). YOLOv8 - Ultralytics/yolov8. Recuperado el 10 de noviembre de 2021, de [\[https://github.com/ultralytics/yolov8/issues/2317\]](https://github.com/ultralytics/yolov8/issues/2317)
- Rath, S. (2023). Train YOLOv8 Instance Segmentation on Custom Data - LearnOpenCV. Recuperado el 10 de noviembre de 2021, de <https://learnopencv.com/train-yolov8-instance-segmentation/>
- Roboflow. (2023). How to Train YOLOv8 Instance Segmentation on Custom Data. Recuperado el 10 de noviembre de 2021, de <https://blog.roboflow.com/how-to-train-yolov8-instance-segmentation/>
- AprendeIngenia. (2021). Deteccion-y-Segmentacion-de-Instancias-personalizado-con-YoloV8. Recuperado el 10 de noviembre de 2021, de <https://github.com/AprendeIngenia/Deteccion-y-Segmentacion-de-Instancias-personalizado-con-YoloV8>