

Desarrollo de un Sistema de Monitoreo Inteligente del Crecimiento de Cultivos Mediante  
Tecnología de Drones e Inteligencia Artificial

Daniel Felipe Suarez Blanco y Paola Andrea Moreno Díaz

Trabajo de Grado para optar al título de Ingenieros Electrónicos

Director

Jaime Guillermo Barrero Pérez

Magister en Potencia Eléctrica

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingeniería Eléctrica, Electrónica y de Telecomunicaciones

Ingeniería Electrónica

Bucaramanga

2024

**Tabla de Contenido**

<b>Introducción</b>	<b>9</b>
<b>1 Objetivos</b>	<b>11</b>
1.1 Objetivo General . . . . .	11
1.2 Objetivos Específicos . . . . .	11
<b>2 Marco Teórico</b>	<b>12</b>
2.1 Planta de Frijol . . . . .	12
2.2 Agricultura de Precisión . . . . .	13
2.3 Inteligencia Artificial . . . . .	13
2.3.1 Aprendizaje Automático (ML) . . . . .	13
2.3.2 Redes Neuronales Profundas . . . . .	14
2.3.3 Detección de objetos mediante YOLO . . . . .	15
<b>3 Metodología</b>	<b>16</b>
3.1 DataSet . . . . .	16
3.1.1 Obtención de las imágenes . . . . .	17
3.1.2 Procesamiento de Imágenes . . . . .	19
3.2 Clasificación por etapa de crecimiento . . . . .	20
3.3 Modelos detección de plantas utilizando YOLO . . . . .	23
3.4 Clasificación segun el Tamaño de la planta . . . . .	24
3.5 Clasificación segun Frondosidad . . . . .	25
3.6 Desarrollo de la GUI . . . . .	26
<b>4 Análisis y resultados</b>	<b>28</b>
4.1 Desempeño red neuronal etapa de crecimiento . . . . .	28
4.2 Desempeño detección de plantas con YOLO . . . . .	30
4.3 Funcionamiento de la GUI . . . . .	33

MONITOREO INTELIGENTE DE CULTIVOS DE FRIJOL	3
<b>5 Conclusiones</b>	<b>35</b>
<b>6 Recomendaciones</b>	<b>36</b>
<b>Apéndices</b>	<b>39</b>

**Lista de Figuras**

Figura 1.	Planta de Frijol. . . . .	12
Figura 2.	Funcionamiento YOLO . . . . .	15
Figura 3.	Dron empleado en la recolección de datos . . . . .	17
Figura 4.	Ciclo de crecimiento del cultivo de frijol . . . . .	18
Figura 5.	Comparación entre Imagen Original y Imagen Mejorada . . . . .	19
Figura 6.	Imagen original junto a una sub imagen extraida de la misma. . . . .	19
Figura 7.	Resumen del modelo clasificador de etapa de crecimiento del cultivo. . . . .	22
Figura 8.	Proceso detección de Plantas con YOLO . . . . .	24
Figura 9.	Clasificación por tamaño . . . . .	25
Figura 10.	Clasificación por frondosidad . . . . .	26
Figura 11.	Diagrama de Flujo de la GUI . . . . .	27
Figura 12.	Matriz de confusión Fold 1 . . . . .	28
Figura 13.	Gráfica ROC Fold 1 . . . . .	28
Figura 14.	Gráfica Accuracy y Loss Fold 1 . . . . .	28
Figura 15.	Matriz de confusión Fold 2 . . . . .	29
Figura 16.	Gráfica ROC Fold 2 . . . . .	29
Figura 17.	Gráfica Accuracy y Loss Fold 2 . . . . .	29
Figura 18.	Matriz de confusión Fold 3 . . . . .	29
Figura 19.	Gráfica ROC Fold 3 . . . . .	29
Figura 20.	Gráfica Accuracy y Loss Fold 3 . . . . .	30
Figura 21.	Pantalla de inicio . . . . .	33
Figura 22.	Cuadro del programa con la predicción . . . . .	33
Figura 23.	Cuadro imagen mejorada y dividida . . . . .	34
Figura 24.	Cuadro con las plantas detectadas y clasificadas . . . . .	34

**Lista de Tablas**

Tabla 1.	Distribución de Imágenes por Clases en el Conjunto de Datos . . . . .	18
Tabla 2.	Parámetros de la red neuronal utilizada . . . . .	22
Tabla 3.	Resultados del clasificador Datos de Test . . . . .	30
Tabla 4.	Comparación de métricas modelos de YOLO para Frijol 0 . . . . .	31
Tabla 5.	Comparación de métricas modelos de YOLO para Frijol 1 . . . . .	31
Tabla 6.	Comparación de métricas modelos de YOLO para Frijol 2 . . . . .	32
Tabla 7.	Comparación de métricas modelos de YOLO para Frijol 3 . . . . .	32

**Lista de Apéndices**

Apéndice A: Repositorio en GitHub ..... 39

Apéndice B: Videos del Cultivo ..... 40

Apéndice C: Funcionamiento de la GUI ..... 41

## Resumen

**Título:** Desarrollo De Un Sistema De Monitoreo Inteligente Del Crecimiento De Cultivos Mediante Tecnología De Drones e Inteligencia Artificial <sup>1</sup>

**Autor:** Daniel Felipe Suarez Blanco y Paola Andrea Moreno Díaz <sup>2</sup>

**Palabras Clave:** Agricultura, IA, redes neuronales profundas, procesamiento de imágenes, monitoreo de cultivos.

**Descripción:** La agricultura desempeña un papel fundamental en la seguridad alimentaria y económica de diversas comunidades. No obstante, la eficiencia y productividad de los cultivos pueden verse comprometidas por múltiples factores, como la escasez de mano de obra, condiciones meteorológicas impredecibles, enfermedades, plagas, estrés hídrico y la falta de información en tiempo real sobre el estado del cultivo. En este contexto, el presente proyecto se enfoca en el desarrollo de un sistema inteligente para la supervisión del crecimiento del cultivo de frijol, utilizando tecnología de drones e inteligencia artificial. Para llevar a cabo esta iniciativa, se emplearon herramientas de software libre y de código abierto, como Google Colaboratory. Se recopiló una base de datos de imágenes de cultivos de frijol mediante un dron equipado con cámara. Cada planta fue etiquetada utilizando la plataforma Roboflow para facilitar su detección y conteo. Posteriormente, clasificar las plantas detectadas según su tamaño y frondosidad, lo que dio como resultado una interfaz gráfica de usuario (GUI) al servicio del usuario.

---

<sup>1</sup>Trabajo de Grado

<sup>2</sup>Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y telecomunicaciones. Director: Jaime Guillermo Barreo Pérez, Magíster en potencia eléctrica.

**Abstract**

**Title:** Development of an Intelligent Monitoring System for Crop Growth Using Drone Technology and Artificial Intelligence <sup>1</sup>

**Author:** Daniel Felipe Suarez Blanco y Paola Andrea Moreno Díaz <sup>2</sup>

**Keywords:** Agriculture, AI, Deep Neural Network, Image Processing, Crop Monitoring.

**Description:** Agriculture plays a fundamental role in the food and economic security of various communities. However, the efficiency and productivity of crops can be compromised by multiple factors, such as labor shortages, unpredictable weather conditions, diseases, pests, water stress, and the lack of real-time information about the crop's status. In this context, this project focuses on the development of an intelligent system for monitoring the growth of bean crops, utilizing drone technology and artificial intelligence. To carry out this initiative, open-source software tools, such as Google Colaboratory, were employed. A database of images of bean crops was collected using a drone equipped with a camera. Each plant was labeled using the Roboflow platform to facilitate its detection and counting. Subsequently, the detected plants were classified according to their size and foliage, resulting in a graphical user interface (GUI) for user support.

---

<sup>1</sup>Degree Project

<sup>2</sup>Faculty of Physical-Mechanical Engineering. School of Electrical, Electronic and Telecommunications Engineering. Director: Jaime Guillermo Barrero Pérez, Master in Electrical Power.

## Introducción

Colombia, con una vasta extensión de tierras cultivables que abarca cerca de 114 millones de hectáreas, lo que representa más del 30 % de su territorio (Econexia, s.f.), se destaca como un productor alimenticio por excelencia. Entre sus principales cultivos, el fríjol ocupa un lugar destacado, según cifras de las Evaluaciones Agropecuarias Municipales, en el país se siembran 101.733 hectáreas de fríjol, con los departamentos de Santander y Boyacá liderando la producción, con aproximadamente 6.857 y 4.696 hectáreas sembradas, respectivamente. (Ministerio de Agricultura Colombia, 2024)

Sin embargo actualmente nos enfrentamos a un escenario global donde el cambio climático se ha convertido en un desafío crucial que impacta en gran medida la agricultura. Organismos internacionales como la FAO y la UIT han advertido sobre la urgencia de encontrar soluciones innovadoras para incrementar la producción de alimentos en un 70 % para el año 2050, a fin de satisfacer las necesidades de una población en crecimiento (Radoglou-Grammatikis y cols., 2020). Además de lo ya mencionado, existen otros factores que afectan la agricultura, como la migración de trabajadores agrícolas hacia las ciudades debido a las difíciles condiciones de vida en las zonas rurales (Comisión de la Verdad, 2024).

Es por tanto que la capacidad de control y monitoreo remoto, potenciada por innovaciones revolucionarias, ha trascendido las barreras convencionales. La introducción de nuevas tecnologías en el campo colombiano es un paso necesario hacia la modernización de las técnicas de monitoreo y la mejora de la productividad en los cultivos. Es aquí donde surge el concepto de agricultura de precisión, que permite la incorporación de drones para la captura aérea de imágenes, el uso de inteligencia artificial y redes neuronales para evaluar el estado de crecimiento de las plantas, como el frijol, proporcionando información crucial sobre su salud mediante el análisis de factores como color, tamaño y frondosidad. Esta herramienta permite revolucionar la agricultura al brindar datos precisos y oportunos a los agricultores y productores, dando la oportunidad de tomar decisiones informadas.

”El éxito en los cultivos lo dan las buenas prácticas que se pongan a disposición para lograr

un buen rendimiento por hectárea”(Agronegocios, 2024), es por ello que el presente proyecto de investigación se enfoca en la creación de un sistema de monitoreo inteligente que permita analizar las imágenes aéreas capturadas por un dron y determinar la etapa de crecimiento en la que se encuentra el cultivo. Adicionalmente, este sistema utiliza inteligencia artificial (IA) para contar, identificar el tamaño y evaluar la frondosidad de cada planta, presentando los resultados a través de una aplicación interactiva destinada al uso de agrónomos o encargados del cultivo. Esta herramienta permitirá alertar sobre posibles anomalías en los cultivos, optimizando los recursos económicos de los productores y facilitando una atención más temprana, lo que contribuirá al aumento de la productividad. A su vez, esto ayudará a mitigar la falta de personal disponible en ubicaciones remotas y abrirá la posibilidad de futuras aplicaciones y extensiones del proyecto mediante la incorporación de sensores que incrementen el número de variables controlables en el cultivo monitoreado.

La presente documento está estructurada en cuatro capítulos: el primero presenta los conceptos previos y revisión de la literatura existente, el segundo describe la metodología empleada para dar solución al problema, el tercero presenta los resultados obtenidos y respectivas conclusiones y, finalmente, el cuarto establece los desafíos futuros para futuros proyectos que busquen complementa o competir la solución implementada. Se espera que los resultados de esta investigación contribuyan a la modernización de las prácticas agrícolas y al desarrollo de herramientas tecnológicas aplicables en entornos de producción agrícola.

## **1. Objetivos**

### **1.1. Objetivo General**

Desarrollar un sistema de inteligencia artificial funcional para el monitoreo y análisis de los campos de cultivo de frijol, con el fin de mejorar la productividad agrícola, optimizar el uso de recursos y reducir los costos de mantenimiento.

### **1.2. Objetivos Específicos**

- Determinar y definir las variables fundamentales medibles en el cultivo de frijol, las cuales proporcionan información principalmente sobre el estado y salud de planta, además del ciclo de crecimiento para su posterior monitoreo.
- Recopilar y organizar una base de datos que contenga información relevante y representativa sobre el cultivo de frijol, incluyendo datos históricos e imágenes que permitan el correcto entrenamiento de la Inteligencia Artificial y eficiencia de los modelos de aprendizaje automático.
- Desarrollar una interfaz intuitiva y de fácil acceso para los agricultores y productores, permitiendo la visualización y comprensión de los datos recopilados, alertas y análisis proporcionados por el sistema de monitoreo.
- Realizar pruebas y evaluaciones del sistema implementado, midiendo su eficacia en la detección temprana de problemas, la toma de decisiones basadas en datos y su impacto en la productividad y reducción de costos

## 2. Marco Teórico

En esta sección se definirán conceptos que son fundamentales para comprender de manera más precisa el contenido del proyecto de investigación, facilitando así la interpretación de los métodos, resultados y discusiones que se presentan a lo largo del estudio.

### 2.1. Planta de Frijol

El frijol (*Phaseolus vulgaris* L.) es una leguminosa que posee características tanto nutricionales como agronómicas que lo hacen un cultivo de gran importancia a nivel mundial. En Colombia, es típicamente manejado por pequeños y medianos agricultores en la región Andina, y se siembra junto con otros cultivos de ciclo corto como el maíz y la arveja.

Entre las principales características del cultivo resalta un ciclo productivo (siembra a cosecha) de 80 días, aproximadamente; un grado de humedad de 20% para el éxito de la cosecha (Ospina Parra y cols., 2020). Se adapta a una amplia variedad de condiciones climáticas y suelos, aunque prefiere suelos bien drenados y climas templados. Desempeña un papel esencial en la nutrición, ya que es rico en vitaminas, proteínas, calcio y zinc. Tiene un contenido bajo en grasa, lo que lo hace adecuado para dietas saludables, y contiene compuestos antioxidantes, como los flavonoides, que ayudan a proteger las células del cuerpo contra el daño causado por los radicales libres (Akond y cols., 2011)



**Figura 1.** *Planta de Frijol.*

## **2.2. Agricultura de Precisión**

La agricultura de precisión es un enfoque moderno en la gestión agrícola que utiliza tecnologías avanzadas para optimizar la producción, mejorar el rendimiento y la eficiencia, y minimizar el uso de insumos agrícolas como fertilizantes, pesticidas y agua, mediante la recopilación y análisis de datos en tiempo real (Gawande y cols., 2023) . En los últimos años, el uso de inteligencia artificial ha sido investigado por su capacidad para mejorar la toma de decisiones precisas y rápidas en la automatización agrícola. Estas tecnologías se han probado exhaustivamente, demostrando su capacidad para mejorar el rendimiento de los cultivos (15-20%), reducir la inversión general (25-30%) y hacer que la agricultura sea más eficiente (20-25%) (Padhiary y cols., 2024).

Comúnmente se utiliza en terrenos amplios que requieren mayor precisión y, por tanto, un manejo más detallado y específico. Para ello, se emplean sensores que permiten recolectar datos sobre el estado del suelo, como la humedad, la salud de las plantas, el pH, la temperatura, entre otros factores. Por otra parte, el uso de vehículos no tripulados y drones para capturar imágenes aéreas reduce significativamente el tiempo y los recursos humanos necesarios, además de ser una opción más económica en comparación con el uso de helicópteros para sobrevolar amplias extensiones de cultivos.

## **2.3. Inteligencia Artificial**

La inteligencia artificial (IA) es un campo de la informática que se centra en el desarrollo de sistemas y algoritmos capaces de realizar tareas que normalmente requieren inteligencia humana. Esto incluye la capacidad de aprender de la experiencia, reconocer patrones, tomar decisiones, resolver problemas, comprender el lenguaje natural y adaptarse a nuevas situaciones.

### **2.3.1. Aprendizaje Automático (ML)**

El aprendizaje automático, un subconjunto de la inteligencia artificial, se dedica a establecer modelos y métodos que permitan a las computadoras aprender de los datos y mejorar sin necesidad de programación explícita. Analiza patrones de datos y se encarga de crear predicciones, aprender y tomar decisiones. (Gawande y cols., 2023)

Existen diferentes tipos de aprendizaje automático, como el aprendizaje supervisado, que se refiere a un modelo entrenado con un conjunto de datos etiquetados, donde cada entrada está asociada a una salida. El aprendizaje no supervisado, por otro lado, trabaja con datos no etiquetados, es decir, sin conocer de antemano la salida deseada. El aprendizaje por refuerzo utiliza interacciones con el entorno para aprender a alcanzar objetivos específicos mediante ensayo y error. El aprendizaje profundo, o deep learning, emplea redes neuronales profundas que aprenden representaciones complejas y abstractas de los datos (Géron, 2022). El aprendizaje automático ha transformado numerosas industrias al permitir la automatización de tareas complejas, mejorar la precisión en la toma de decisiones y ofrecer nuevas formas de interacción entre humanos y máquinas. A medida que la cantidad de datos disponibles continúa creciendo, el aprendizaje automático seguirá siendo un área de gran impacto en la innovación tecnológica.

### **2.3.2. *Redes Neuronales Profundas***

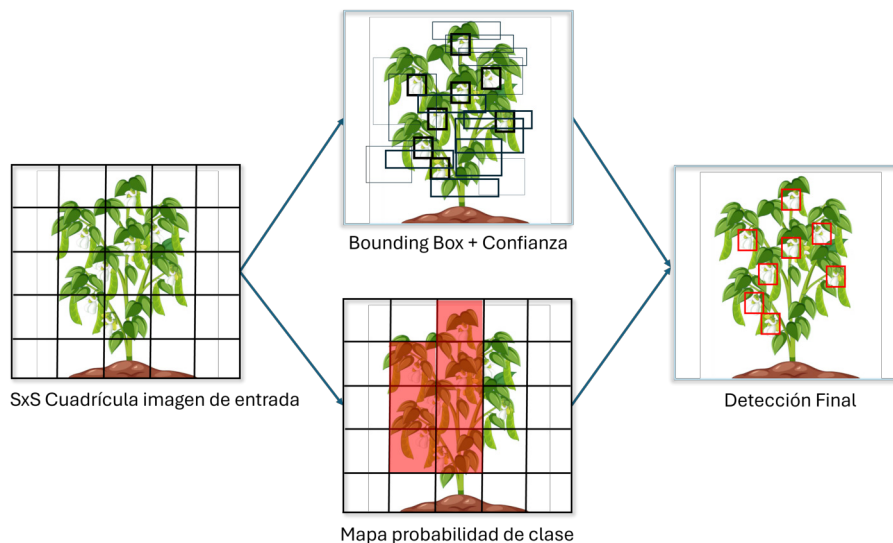
Las redes neuronales son modelos de inteligencia artificial inspirados en el funcionamiento del cerebro humano. Están compuestas por capas de neuronas artificiales, o nodos, que están interconectadas. Cada neurona recibe una serie de entradas, las procesa a través de una función de activación y genera una salida que es transmitida a la siguiente capa. Estas capas pueden ser capas de entrada, capas ocultas y capas de salida, dependiendo de su ubicación en la red. Las neuronas de las capas intermedias (capas ocultas) permiten que la red aprenda relaciones complejas y no lineales entre los datos de entrada y salida, lo que les da su capacidad para resolver problemas más allá de simples correlaciones.

Las redes neuronales se utilizan en una amplia gama de aplicaciones, desde el reconocimiento de imágenes y voz, hasta la detección de fraudes, la conducción autónoma y la predicción de comportamientos. Su principio de aprendizaje se basa en el ajuste de los pesos de las conexiones entre las neuronas mediante un proceso llamado retropropagación (backpropagation), que optimiza el rendimiento de la red. Al entrenar con grandes cantidades de datos, las redes neuronales pueden aprender patrones complejos y realizar tareas de clasificación, predicción y detección con alta precisión, convirtiéndolas en una herramienta clave en el campo del aprendizaje automático y la inteligencia artificial.

### 2.3.3. Detección de objetos mediante YOLO

En los últimos años, el algoritmo de detección de objetos de una sola etapa You Only Look Once (YOLO) ha emergido como una de las herramientas más avanzadas y efectivas en el campo de la agricultura. Su capacidad para combinar precisión, velocidad y tamaño de red le ha permitido posicionarse como una solución de vanguardia para tareas que requieren detección en tiempo real. YOLO ha sido implementado en una amplia variedad de operaciones agrícolas, como monitoreo de cultivos, vigilancia, automatización de procesos y en sistemas de robótica agrícola, debido a sus características que optimizan la precisión y velocidad del procesamiento de datos, factores críticos para la implementación tecnológica en este sector.

La arquitectura de YOLO se basa en una red neuronal convolucional (CNN) que divide la imagen en una cuadrícula y, para cada celda de esa cuadrícula, predice cajas delimitadoras (bounding boxes) y las probabilidades de que esas cajas contengan ciertos objetos (Badgujar y cols., 2024). Esta estrategia no solo permite detectar múltiples objetos en una imagen, sino que también lo hace a gran velocidad, lo que es crucial para aplicaciones en tiempo real.



**Figura 2.** *Funcionamiento YOLO*

Las versiones avanzadas como YOLOv8, YOLOv10 y YOLO-NAS han mejorado la optimización de recursos computacionales y la adaptabilidad del algoritmo, haciéndolo ideal para la agricultura de precisión, donde se requiere análisis rápido.

### 3. Metodología

En esta sección se analizan y describen los procedimientos realizados para llevar a cabo este proyecto investigativo. Se comienza con la explicación de lo que se proyectó realizar, el análisis de las opciones que se tenían a disposición, los métodos elegidos y una justificación posterior. Asimismo, se exponen los modelos construidos, su funcionamiento y la construcción de la GUI para abordar los objetivos propuestos, y posteriormente se entrega un análisis de los resultados, que se exponen en la siguiente sección.

#### 3.1. DataSet

Comenzando con el dataset, se realizó inicialmente una investigación exhaustiva para verificar en la red académica y general si existía un conjunto de datos que cumpliera con los requisitos mínimos establecidos para el proyecto. Estos requisitos consistían, idealmente, en imágenes relacionadas con el cultivo de frijol, que incluyeran fotos periódicas semanales de su crecimiento, con el fin de poder analizarlas. Se realizaron investigaciones en plataformas como Kaggle, UCI Machine Learning Repository, Google Dataset Search y Data.gov, ya que son las páginas más reconocidas para compartir, buscar y subir datasets al servicio de la comunidad. Sin embargo, al considerar los requisitos mencionados, ninguno de los conjuntos de datos disponibles cumplía completamente con lo necesario. Por lo tanto, se optó por la construcción de un dataset propio que cubriera las condiciones específicas, convirtiéndose así en uno de los objetivos principales del proyecto.

Tomada esta decisión se fijaron los requisitos del data set como:

- El dataset debía incluir imágenes del cultivo de frijol tomadas periódicamente. En este caso, las capturas se realizaron de manera semanal, con intervalos de 5 a 7 días entre ellas.
- Las imágenes debían tomarse a una altura promedio de 15 metros para asegurar que el tamaño de las plantas fuera un factor relevante a controlar en el proyecto.
- El ángulo de la cámara del dron debe ser paralelo a la superficie del cultivo para evitar que la perspectiva introduzca sesgos en la información.
- El conjunto de datos se dividió en cuatro clases, correspondientes a las principales etapas de crecimiento donde se observan los mayores cambios: "siembra a germinación (12 a 15

días), germinación a floración (27 a 45 días), floración a la aparición de la legumbre verde (7 a 15 días), y floración a la recolección de la semilla (37 a 38 días.)"(de Información Agroalimentaria y Pesquera, 2019).

### **3.1.1. Obtención de las imágenes**

Para la creación del dataset, se realizó la adquisición de un dron de la marca ZLL Beast, una marca china. El modelo del dron, específicamente, fue el Drone ZLL Beast SG906 Pro con cámara dual. El enlace al producto se encuentra en el repositorio.



**Figura 3.** Dron empleado en la recolección de datos

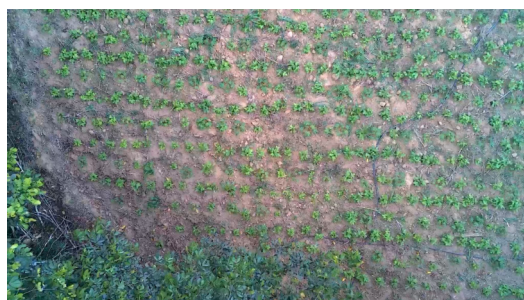
La elección del dron fue totalmente arbitraria; se buscó la mejor relación calidad-precio. Se procedió entonces a la creación del dataset cerca de la ciudad de Floridablanca en la zona de Ruitoque Bajo. A través del voz a voz, se logró contactar con agricultores de la zona, quienes muy amablemente pusieron a disposición sus cultivos de frijol y permitieron el ingreso del equipo de trabajo para las tomas del dron.

El procedimiento consistió en capturar fotografías de manera periódica, con un intervalo semanal. Durante un periodo de cuatro meses, se visitaron los cultivos cada fin de semana. La técnica empleada implicaba elevar el dron a una altura promedio de 15 metros y ajustar la cámara a un ángulo de 90 grados respecto a la horizontal, con el fin de obtener tomas planas. Posteriormente, se procedió a realizar las capturas fotográficas por zonas: el dron se desplazaba un metro hacia adelante, permitiendo tomar diez fotografías en cada área, lo que resultaba en aproximadamente 150 imágenes por visita al cultivo.

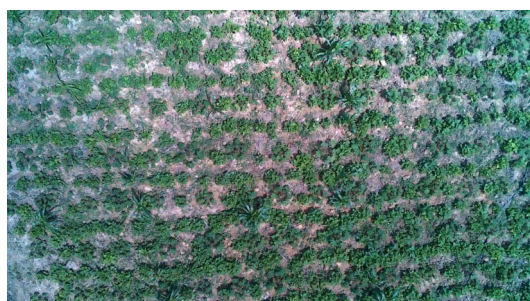
Finalmente, se recopiló una base de datos compuesta por un total de 2,294 imágenes en formato RGB, con una resolución de 5120 px por 2280 px. Estas imágenes abarcan las cuatro etapas de crecimiento: Siembra-Germinación, Germinación, Floración y Recolección. La distribución de estas clases se presenta en la Tabla 1.



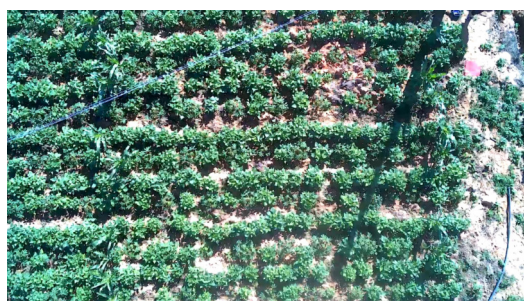
*a). Siembra a germinación*



*b). Germinación a floración*



*c). Floración*



*d). Recolección*

**Figura 4.** *Ciclo de crecimiento del cultivo de frijol*

Para más detalles, consulte el conjunto de datos disponible en GitHub Data Set.

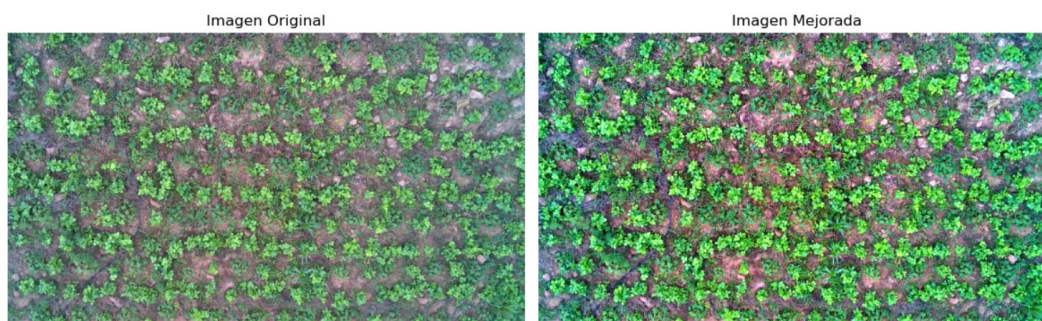
<b>Clase</b>	<b>Total de Imágenes</b>
Frijol 0	391
Frijol 1	793
Frijol 2	472
Frijol 3	638

**Tabla 1.** *Distribución de Imágenes por Clases en el Conjunto de Datos*

Adicionalmente, se creó la carpeta de pruebas que contiene unas pocas imágenes de cada etapa. Estas imágenes son separadas y no fueron usadas durante el entrenamiento.

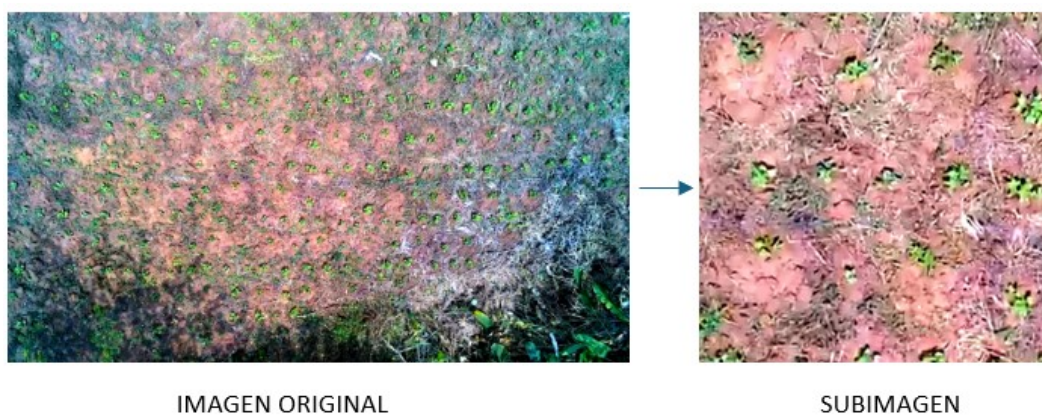
### 3.1.2. *Procesamiento de Imágenes*

Una vez obtenida la base de datos para las cuatro clases establecidas, se llevó a cabo un procesamiento de las imágenes mediante un script en lenguaje Python que utiliza la biblioteca PIL para manipular imágenes y la biblioteca OpenCV. Este proceso tuvo como objetivo aumentar la resolución y mejorar la nitidez, el contraste y la saturación, facilitando así la detección de las plantas y la distinción entre el ruido presente en su entorno, como maleza, otros tipos de plantas, ramas, piedras y otros factores a los que está expuesto el cultivo.



**Figura 5.** *Comparación entre Imagen Original y Imagen Mejorada*

Posteriormente, la imagen resultante con las mejoras fue dividida en imágenes cuadradas de 1280 x 1280 píxeles, completando con negro los espacios para aquellas subimágenes que no alcanzaban a formar el cuadrado con los píxeles mencionados, las cuales fueron almacenadas en una carpeta de salida. Todo esto se realizó con el fin de facilitar la detección de las plantas de frijol.



**Figura 6.** *Imagen original junto a una sub imagen extraida de la misma.*

### 3.2. Clasificación por etapa de crecimiento

Se implementó un modelo de red neuronal convolucional (CNN) para clasificar las imágenes mejoradas del cultivo, utilizando la técnica de validación cruzada K-Fold para evaluar su rendimiento en las 4 clases de crecimiento del frijol, las cuales por practicidad se denominaron Siembra a germinación como clase 0, Germinación a floración clase 1, Floración clase 2 y Recolección clase 3. El proceso comenzó configurando la validación cruzada con 3 particiones ( $n_{\text{splits}}=3$ ) mediante la técnica `MultilabelStratifiedKFold`, que garantiza una distribución balanceada de las clases en cada fold, mejorando así la consistencia de los resultados. El modelo secuencial de Keras empleó varias capas convolucionales, cada una con 64 filtros, un tamaño de kernel de  $2 \times 2$  y la función de activación ReLU. Además, se incluyeron capas de `BatchNormalization` para estabilizar y acelerar el entrenamiento, mientras que las capas de `MaxPooling` se encargaron de reducir la dimensionalidad de las imágenes con bloques de  $2 \times 2$ . Para prevenir el sobreajuste, se utilizaron capas de `Dropout` con tasas del 30 %, 40 % y 50 %. Finalmente, el modelo concluye con una capa densa que emplea una activación softmax para clasificar las imágenes en una de las 4 clases definidas (0,1,2,3). Esta configuración optimizó el proceso de clasificación de las imágenes en las distintas etapas de crecimiento del cultivo.

A continuación se analiza capa por capa el modelo, de igual manera el código y explicación del modelo se encuentran en el siguiente enlace al proyecto en GitHub: Repositorio en GitHub.

- **Conv2D (conv2d\_36):** Esta es la primera capa que "mira" la imagen de entrada. Aplica 64 filtros o "lentes" pequeños de  $3 \times 3$  píxeles que recorren la imagen para detectar características básicas como bordes o texturas. Al final, genera una imagen transformada con una profundidad de 64 canales.
- **Batch Normalization (batch\_normalization\_36):** Después de la primera capa de filtros, normaliza los valores para que el entrenamiento sea más estable y rápido, asegurándose de que las activaciones tengan una media y desviación estándar controladas.
- **Conv2D (conv2d\_37):** Aplica otro conjunto de 64 filtros  $2 \times 2$  para extraer características más complejas de la imagen. Los primeros filtros pueden detectar bordes, pero a medida que se aplican más, pueden empezar a capturar formas o patrones más detallados.

- **Batch Normalization (batch\_normalization\_37)**: Nuevamente, normaliza la salida para que los valores no crezcan o disminuyan demasiado durante el entrenamiento.
- **MaxPooling2D (max\_pooling2d\_27)**: Esta capa reduce la resolución de la imagen (de 200x200 a 100x100) seleccionando el valor más alto en pequeñas áreas 2x2, de forma similar a cómo podrías hacer zoom en las partes más importantes de una imagen, resumiendo la información sin perderla.
- **Dropout (dropout\_27)**: Actúa como una capa de "protección" contra el sobreajuste, apagando aleatoriamente algunas conexiones para que el modelo no dependa demasiado de ciertos caminos o características.
- **Conv2D (conv2d\_38)**: Otro conjunto de 64 filtros que analiza la imagen ya reducida, buscando características aún más complejas en la representación de menor resolución.
- **Batch Normalization (batch\_normalization\_38)**: Nuevamente, normaliza la salida de la capa anterior para mantener un entrenamiento estable.
- **MaxPooling2D (max\_pooling2d\_28)**: Reduce otra vez la resolución (de 100x100 a 50x50), extrayendo las características más importantes de la imagen.
- **Dropout (dropout\_28)**: Aplica otra vez la desconexión aleatoria de algunas conexiones para mejorar la generalización del modelo.
- **Conv2D (conv2d\_39)**: Otro conjunto de filtros se aplica sobre la imagen ahora más pequeña, identificando patrones aún más específicos.
- **Batch Normalization (batch\_normalization\_39)**: Normaliza las activaciones de esta capa convolucional para evitar que los valores se descontrolen.
- **MaxPooling2D (max\_pooling2d\_29)**: Reduce la imagen a un tamaño aún más pequeño (25x25), manteniendo la información esencial.
- **Dropout (dropout\_29)**: Una última capa de dropout antes de pasar a la fase de clasificación, desconectando aleatoriamente algunas conexiones para prevenir el sobreajuste.

- **Flatten (flatten\_9):** Aplana la imagen, transformando la salida tridimensional en un vector de 40,000 elementos para que las próximas capas (densas) puedan trabajar con ella como si fuera una lista de características.
- **Dense (dense\_9):** Finalmente, esta es la capa de decisión. Tiene 4 nodos que representan las 4 posibles clases en las que el modelo está tratando de clasificar. Aquí es donde el modelo emite su "veredicto" sobre a cuál de las 4 clases pertenece la imagen que ha procesado.

Layer (type)	Output Shape	Param #
conv2d_36 (Conv2D)	(None, 200, 200, 64)	832
batch_normalization_36 (BatchNormalization)	(None, 200, 200, 64)	256
conv2d_37 (Conv2D)	(None, 200, 200, 64)	16,448
batch_normalization_37 (BatchNormalization)	(None, 200, 200, 64)	256
max_pooling2d_27 (MaxPooling2D)	(None, 100, 100, 64)	0
dropout_27 (Dropout)	(None, 100, 100, 64)	0
conv2d_38 (Conv2D)	(None, 100, 100, 64)	16,448
batch_normalization_38 (BatchNormalization)	(None, 100, 100, 64)	256
max_pooling2d_28 (MaxPooling2D)	(None, 50, 50, 64)	0
dropout_28 (Dropout)	(None, 50, 50, 64)	0
conv2d_39 (Conv2D)	(None, 50, 50, 64)	16,448
batch_normalization_39 (BatchNormalization)	(None, 50, 50, 64)	256
max_pooling2d_29 (MaxPooling2D)	(None, 25, 25, 64)	0
dropout_29 (Dropout)	(None, 25, 25, 64)	0
flatten_9 (Flatten)	(None, 40000)	0
dense_9 (Dense)	(None, 4)	160,004

**Figura 7.** Resumen del modelo clasificador de etapa de crecimiento del cultivo.

Parámetros	Número de parámetros
Total parámetros	632,590 (2.41 MB)
Parámetros Entrenables	210,692 (823.02 KB)
Parámetros No Entrenables	512 (2.00 KB)
Parámetros Optimizador	421,386 (1.61 MB)

**Tabla 2.** Parámetros de la red neuronal utilizada

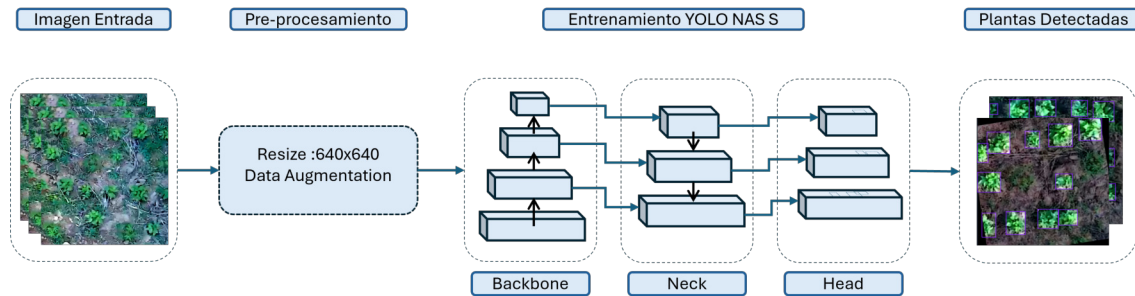
### 3.3. Modelos detección de plantas utilizando YOLO

Se diseñaron cuatro modelos para detectar las plantas de frijol en cada una de las etapas de crecimiento del cultivo, utilizando la plataforma Roboflow. Inicialmente, el etiquetado de las plantas se realizó de manera manual, identificando cada una en las imágenes. Una vez establecidas las etiquetas, se aprovechó la función de autolabeling del programa, lo que permitió automatizar y acelerar el proceso de etiquetado, ampliando de forma significativa y rápida el tamaño del dataset.

Para cada uno de los modelos, se dividió el conjunto de datos en tres subconjuntos: 80 % para entrenamiento, 20 % para validación y 10 % para test. Además, se implementó un redimensionamiento de las imágenes a una resolución de 640x640 píxeles para optimizar el proceso de entrenamiento. Se aplicaron técnicas de aumento de datos (data augmentation) para diversificar el conjunto de imágenes, incluyendo transformaciones como rotaciones horizontales y verticales, giros de 90° en ambos sentidos, y algunas rotaciones que varían entre -15° y 15°.

En el proceso de entrenamiento para cada uno de los modelos, se implementó el modelo YOLO-NAS S (You Only Look Once - Neural Architecture Search), una variante avanzada de la familia YOLO, diseñada para maximizar la eficiencia y precisión en tareas de detección de objetos. Esta arquitectura se basa en la búsqueda automática de arquitecturas óptimas, lo que permite adaptar la red de manera eficiente al problema específico de identificación de plantas de frijol en este proyecto.

La estructura del modelo YOLO-NAS se compone de tres partes fundamentales: el backbone, el neck, y el head. El backbone está encargado de extraer características clave de las imágenes, como texturas, bordes y patrones, mediante capas convolucionales que aplican activaciones no lineales como ReLU o SiLU. El neck se encarga de refinar estas características y prepara la información para la detección de objetos en múltiples escalas, utilizando técnicas como FPN (Feature Pyramid Network) o PANet (Path Aggregation Network). Finalmente, en el head, se generan las predicciones sobre las posiciones de los objetos y las clases correspondientes, ajustando las cajas delimitadoras a los objetos detectados mediante el uso de anclas predeterminadas.

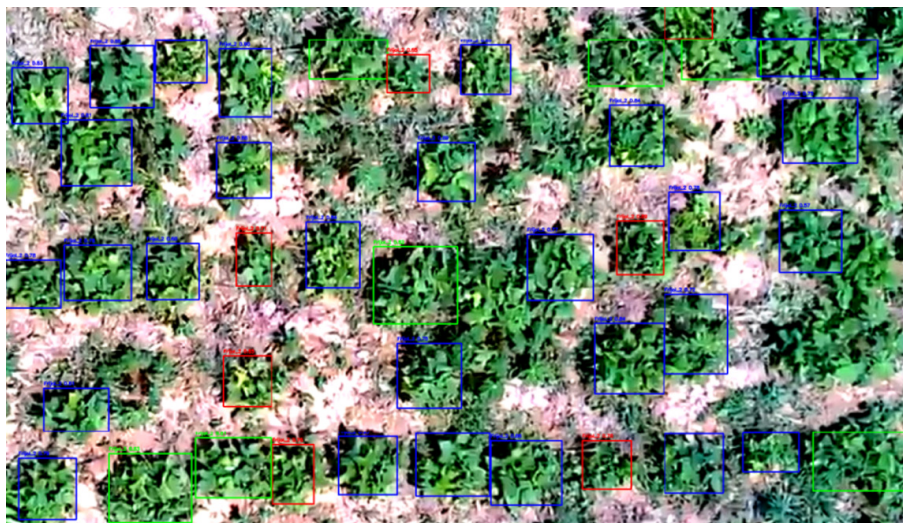


**Figura 8.** *Proceso detección de Plantas con YOLO*

### 3.4. Clasificación según el Tamaño de la planta

El código empleado comienza dividiendo la imagen de entrada del dataset con sus correspondientes mejoras de nitidez, saturación y contraste, en sub-imágenes más pequeñas para facilitar la detección. Cada una de estas sub-imágenes es analizada por un modelo previamente entrenado en Roboflow para identificar objetos de interés (en este caso, plantas de frijol). Una vez que el modelo procesa cada sub-imagen, genera predicciones de posibles objetos en las mismas. Estas predicciones contienen información relevante como la clase del objeto detectado, su posición en la imagen y un cuadro delimitador o bounding box que enmarca el área donde se encuentra el objeto. Para asegurar la fiabilidad de las predicciones, solo se consideran aquellas con una certeza superior a un umbral predeterminado de 45 %.

El siguiente paso consiste en extraer las dimensiones de cada cuadro delimitador. Para este fin, se registra el ancho de cada bounding box, que posteriormente se utiliza para calcular el tamaño promedio de los objetos detectados. Este promedio sirve como referencia para clasificar los objetos en tres categorías según su tamaño. Los objetos cuyo ancho es menor al 80 % del tamaño promedio son clasificados como pequeños, mientras que aquellos cuyo ancho se encuentra entre el 80 % y el 120 % del promedio se consideran de tamaño mediano. Los objetos cuyo ancho excede el 120 % del promedio se clasifican como grandes. Esta clasificación se refleja visualmente mediante la asignación de colores a los cuadros: rojo para los objetos pequeños, azul para los medianos y verde para los grandes. Por último, se reconstruye la imagen original que muestra los resultados de las predicciones, donde cada objeto detectado es enmarcado con su respectivo cuadro de color, indicando su tamaño.



**Figura 9.** *Clasificación por tamaño*

### 3.5. Clasificación según Frondosidad

Para la detección y clasificación de frondosidad, se implementó un modelo de red neuronal convolucional (CNN) entrenado utilizando imágenes derivadas de los cuadros delimitadores obtenidos del dataset original en Roboflow. Las imágenes fueron divididas en dos clases, Frondosa y Poco Frondosa, y se entrenó el modelo utilizando Stratified K-Fold con 3 particiones, preservando la proporción de las clases en cada subconjunto de datos. La arquitectura del modelo incluyó múltiples capas convolucionales (Conv2D) con 64 filtros y un tamaño de kernel de (2, 2), combinadas con normalización por lotes (BatchNormalization) y submuestreo (MaxPooling2D), con tasas de Dropout del 30%, 40%, y 50% para evitar el sobreajuste. El entrenamiento se llevó a cabo durante un máximo de 100 épocas, utilizando un tamaño de lote de 6, la función de pérdida categorical crossentropy, y el optimizador Adam. Se implementaron callbacks como EarlyStopping para detener el entrenamiento si no se observaba mejora en la pérdida de validación durante 5 épocas consecutivas, y ReduceLROnPlateau para reducir la tasa de aprendizaje en un 5% después de 4 épocas sin mejoras. El rendimiento del modelo fue evaluado mediante curvas ROC y matrices de confusión, mostrando una precisión robusta en la clasificación de frondosidad, y visualizado a través de gráficas de pérdida y precisión para cada Fold.



**Figura 10.** *Clasificación por frondosidad*

### 3.6. Desarrollo de la GUI

Una vez obtenido el conjunto de datos, se procedió a definir la estrategia para el desarrollo de la interfaz gráfica de usuario (GUI). Tras realizar varios análisis, se estableció la siguiente estructura operativa, cuyos modelos y configuraciones se describieron en detalle en este informe.

En el frontend, el usuario tiene la opción de cargar una imagen en formato JPG a través de la interfaz. Al hacer clic en el botón "Obtener análisis de cultivo", la aplicación procesa la imagen y devuelve un análisis que incluía: la etapa de crecimiento del cultivo, especificando el nombre y el rango de días correspondiente; el número total de plantas identificadas en la imagen, clasificación por tamaño. Las plantas que se encontraron por debajo del promedio reciben una alerta visual en color, mientras que las más grandes son resaltadas. Esta categorización se realizó mediante un sistema de codificación por colores, que también permitió la clasificación de las plantas en función de su frondosidad, diferenciando entre las más frondosas y las poco frondosas.

En el backend, se implementó un modelo de flujo secuencial. El proceso inicia con la recepción de la imagen por parte de la GUI, que luego la envió a un primer modelo de inteligencia artificial para clasificar la etapa de crecimiento del cultivo. A continuación, se lleva a cabo un preprocesamiento que resaltó los colores y mejora la calidad visual de la imagen. Posteriormente, la imagen es dividida en sub-imágenes cuadradas de 1280x1280 píxeles, lo que permite una obtención

más detallada de las plantas.

Se diseñaron cuatro modelos específicos, uno para cada clase, que se encargaron de identificar las plantas en cada sub-imagen. A continuación, un código calcula el tamaño del cuadro delimitador de cada planta en las sub-imágenes, generando un promedio global. Este promedio se utilizó para marcar las plantas de la siguiente manera: aquellas de tamaño promedio se indicaron en color azul, las pequeñas (20% o más por debajo del promedio) en rojo, y las grandes (20% o más por encima del promedio) en verde. Finalmente, un modelo adicional clasificó cada planta reconocida en función de su frondosidad, distinguiendo entre alta y baja frondosidad.

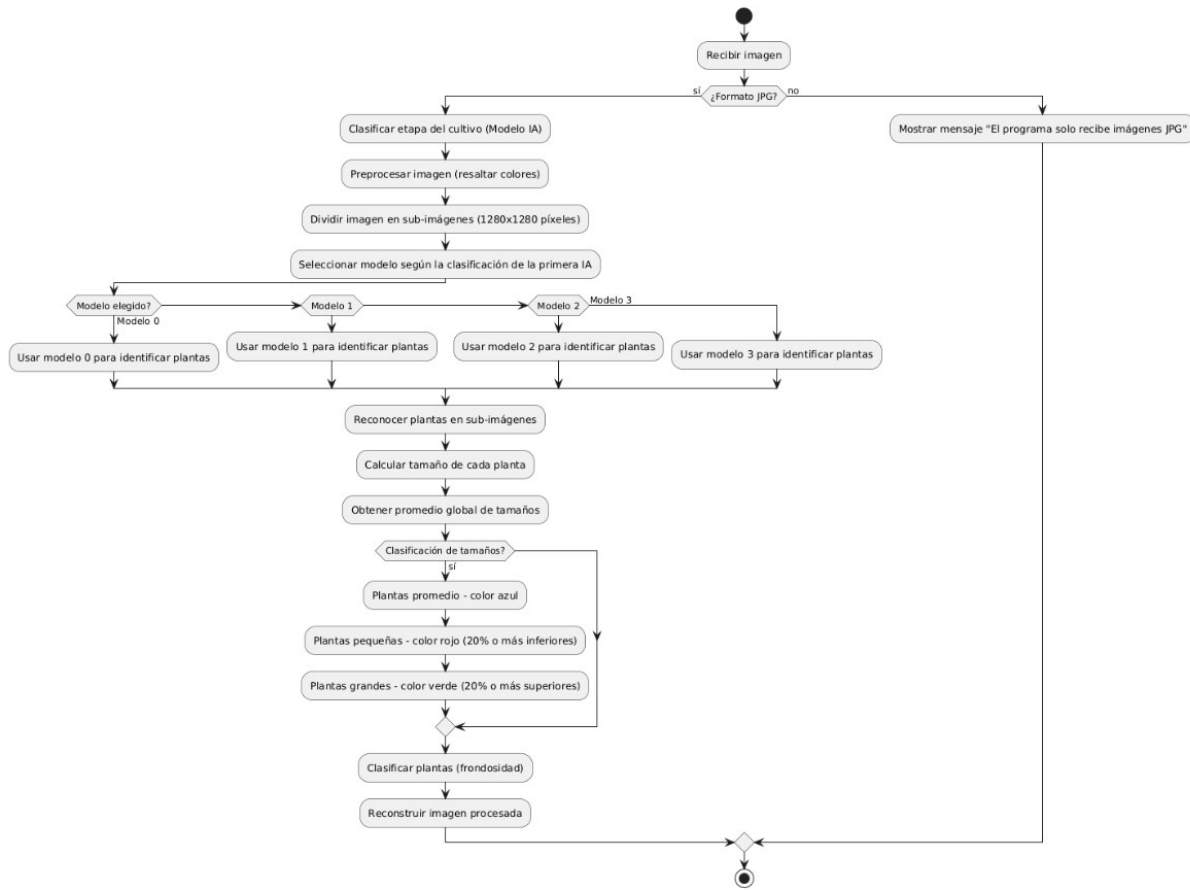


Figura 11. Diagrama de Flujo de la GUI

### 4. Análisis y resultados

En este capítulo se exponen los resultados obtenidos a lo largo del desarrollo del proyecto. Se realiza un análisis detallado de los datos recolectados durante la implementación, comparándolos con los objetivos propuestos al inicio. Entre los resultados presentados se incluyen las matrices de confusión, las curvas ROC, gráficos de precisión y pérdida, así como la comparación entre los diferentes modelos YOLO empleados. Además, se discuten los hallazgos obtenidos y su relevancia para los objetivos del estudio.

#### 4.1. Desempeño red neuronal etapa de crecimiento

Acontinuación se presentan las métricas de rendimiento del clasificador durante el entrenamiento y la validación en los distintos K-Folds .

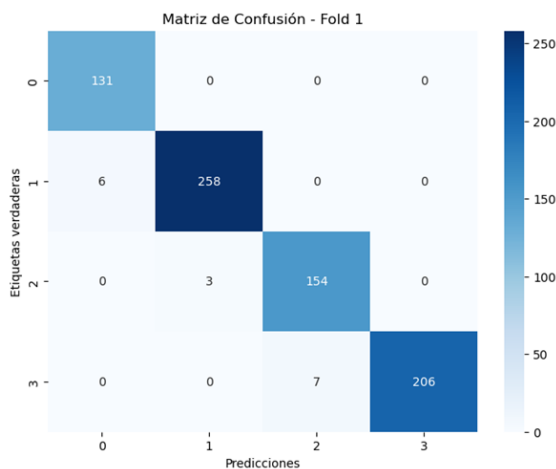


Figura 12. Matriz de confusión Fold 1

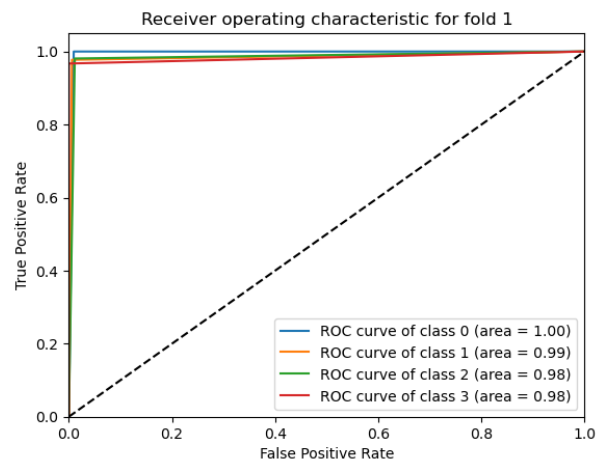


Figura 13. Gráfica ROC Fold 1

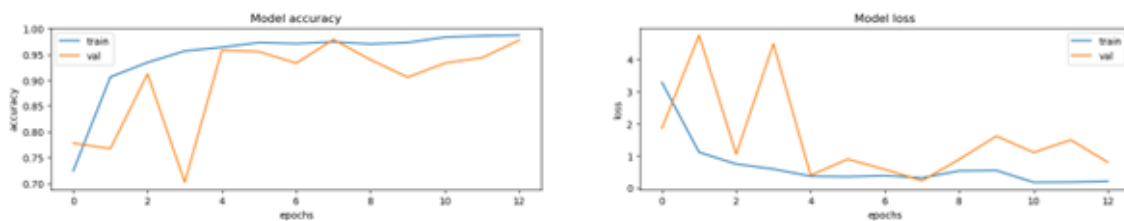


Figura 14. Gráfica Accuracy y Loss Fold 1

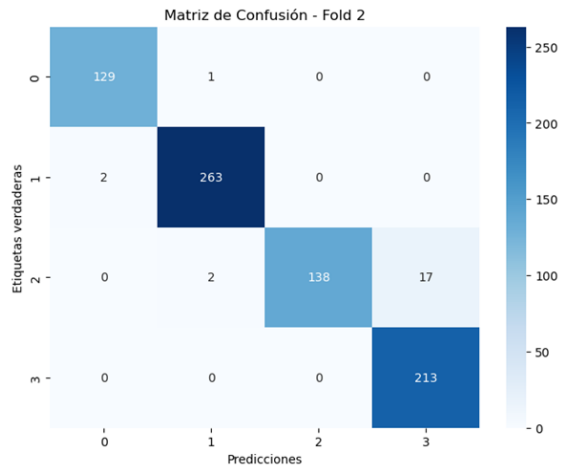


Figura 15. Matriz de confusión Fold 2

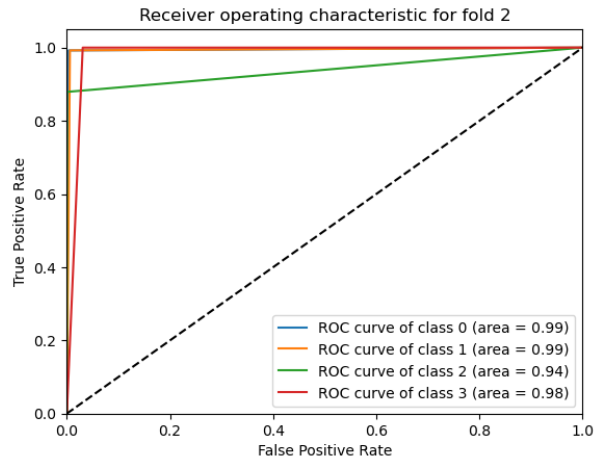


Figura 16. Gráfica ROC Fold 2

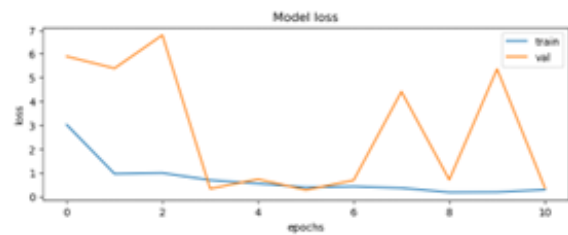
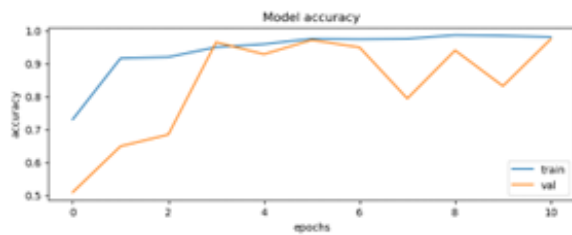


Figura 17. Gráfica Accuracy y Loss Fold 2

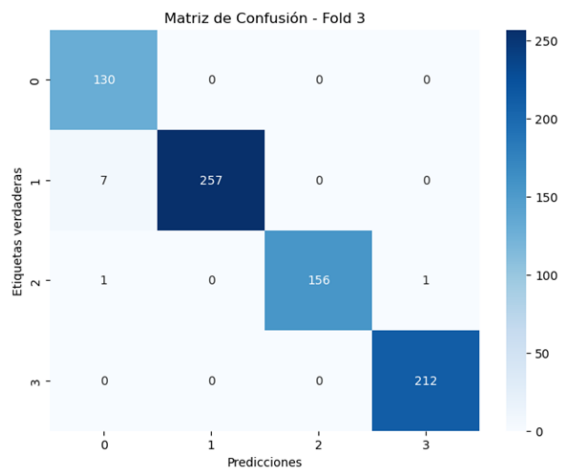


Figura 18. Matriz de confusión Fold 3

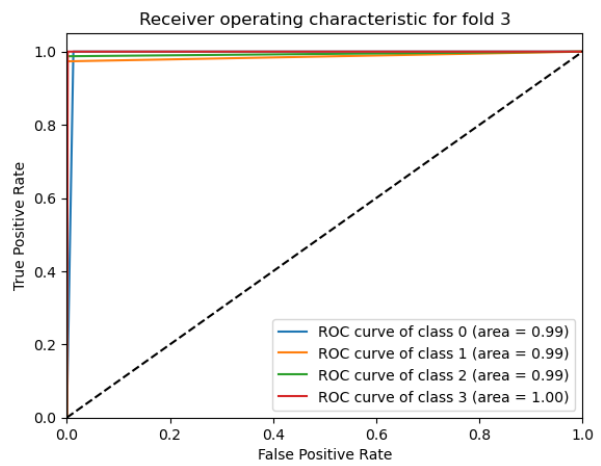
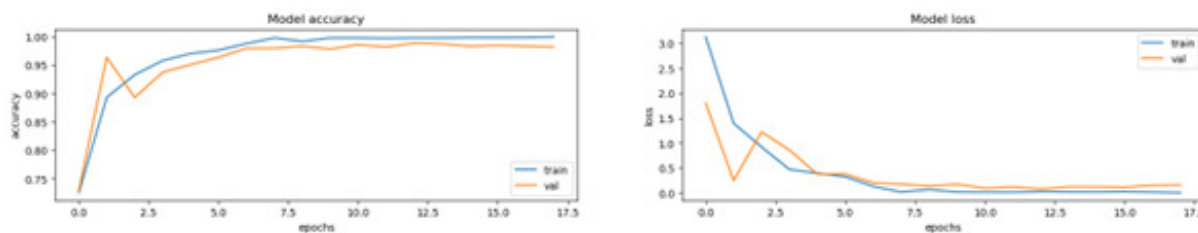


Figura 19. Gráfica ROC Fold 3



**Figura 20.** Gráfica Accuracy y Loss Fold 3

El clasificador fue entrenado utilizando un esquema de validación con 3 K-folds y los resultados en el conjunto de test muestran un rendimiento excelente. En cuanto a las cuatro clases, el modelo obtuvo una precisión y recall perfectos (1.0) para las clases 0 y 3, mientras que la clase 1 alcanzó una precisión de 0.97 y un recall de 1.0, resultando en un F1-score de 0.99. La clase 2 presentó una precisión de 1.0 y un recall de 0.97, con un F1-score de 0.98. El modelo demostró una exactitud global del 99 %, y tanto el macro promedio como el promedio ponderado de las métricas (precisión, recall y F1-score) fueron consistentemente 0.99, lo que indica que el modelo clasifica con gran precisión y equilibrio entre las diferentes clases.

Clase	Precision	Recall	F1-score	Support
Clase 0	1.0	1.0	1.0	24
Clase 1	0.97	1.0	0.99	34
Clase 2	1.0	0.97	0.98	29
Clase 3	1.0	1.0	1.0	45
<b>Accuracy</b>		0.99		132
<b>Macro avg</b>	0.99	0.99	0.99	132
<b>Weighted avg</b>	0.99	0.99	0.99	132

**Tabla 3.** Resultados del clasificador Datos de Test

## 4.2. Desempeño detección de plantas con YOLO

Para la detección de plantas de frijol en imágenes aéreas capturadas a una altura específica mediante dron, se desarrolló un modelo basado en YOLO para cada una de las cuatro etapas clave del crecimiento del frijol. Con el objetivo de evaluar la eficiencia en la detección en cada fase, se implementaron tres versiones del modelo YOLO: YOLOv7, YOLOv8 y YOLO-NASs. Cada

modelo fue entrenado y evaluado en términos de precisión, recall y mAP, lo que permitió comparar su rendimiento y seleccionar el más adecuado para identificar correctamente las plantas en cada etapa de crecimiento.

Modelo para la clase Frijol 0, Siembra a germinación:

<b>Modelo</b>	<b>mAP</b>	<b>Precisión</b>	<b>Recall</b>
YOLOv7	0.771	0.709	0.789
YOLOv8	0.794	0.751	0.761
YOLO-NASs	0.843	0.821	0.778

**Tabla 4.** Comparación de métricas modelos de YOLO para Frijol 0

En la tabla se comparan las métricas de los modelos YOLOv7, YOLOv10s y YOLO-NASs para la clase Frijol 0, que corresponde a la etapa Siembra a germinación. El modelo YOLOv7 obtiene un mAP de 0.771, lo que indica una efectividad moderada en la detección de las instancias de esta clase, con una precisión de 0.709 y un recall de 0.789. Esto sugiere que YOLOv7 es capaz de recuperar el 78.9% de las instancias reales, aunque con una tasa de falsos positivos más alta que en otros modelos. Por otro lado, el modelo YOLO-NASs muestra un rendimiento superior, con un mAP de 0.843, indicando un mejor desempeño global en la tarea de detección. Su precisión es notablemente superior, alcanzando 0.821, lo que significa que el modelo realiza predicciones correctas en un 82.1% de los casos.

Modelo para la clase Frijol 1, Germinación a floración:

<b>Modelo</b>	<b>mAP</b>	<b>Precisión</b>	<b>Recall</b>
YOLOv7	0.843	0.799	0.770
YOLOv8	0.855	0.831	0.788
YOLO-NASs	0.886	0.874	0.779

**Tabla 5.** Comparación de métricas modelos de YOLO para Frijol 1

La tabla presentada una comparación entre las métricas en la detección de la clase Frijol 1, correspondiente a la etapa Germinación a floración. El modelo YOLOv7 alcanza un mAP de 0.843, lo que indica una alta capacidad para detectar correctamente las instancias de la clase. Su precisión es de 0.799, lo que significa que casi el 80% de sus predicciones son correctas. Por

otro lado, el modelo YOLO-NASs obtiene un rendimiento superior con un mAP de 0.886 y una precisión de 0.874, lo que indica un mejor desempeño global en términos de detección y una mayor confiabilidad al hacer predicciones correctas, con menos falsos positivos que YOLOv7.

Modelo para la clase Frijol 2, Floración: El modelo YOLOv7 muestra un desempeño sólido, al-

<b>Modelo</b>	<b>mAP</b>	<b>Precisión</b>	<b>Recall</b>
YOLOv7	0.811	0.757	0.745
YOLOv8	0.826	0.769	0.742
YOLO-NASs	0.849	0.863	0.673

**Tabla 6.** Comparación de métricas modelos de YOLO para Frijol 2

canzando un mAP de 0.811, una Precisión de 0.757 y un Recall de 0.745. Estos resultados indican que YOLOv7 tiene un buen balance entre la precisión y la capacidad de recuperar las instancias correctamente, lo que sugiere un rendimiento eficiente para la detección de la clase "Frijol 2.<sup>en</sup> floración. Nuevamente, YOLO-NASs logra el mAP más alto con 0.849. Sin embargo, su Recall es notablemente inferior (0.673), lo que significa que tiene una menor capacidad para detectar todas las instancias de la clase. A pesar de esto, su Precisión es la más alta entre los modelos utilizados, alcanzando 0.863, lo que implica que cuando YOLO-NASs realiza una predicción, es más probable que esta sea correcta.

Modelo para la clase Frijol 3, Recolección: Con base en la tabla se observa que YOLOv7 logra un

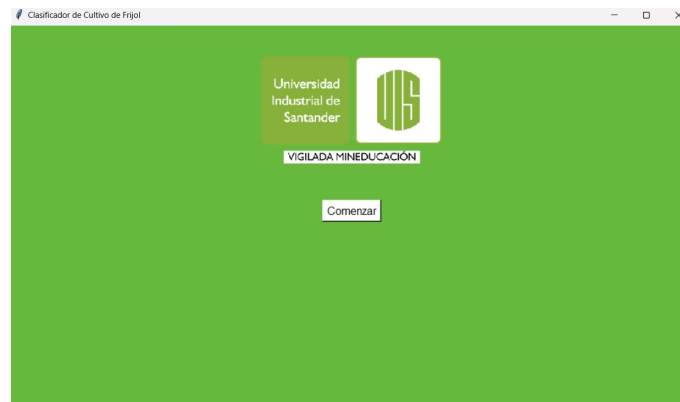
<b>Modelo</b>	<b>mAP</b>	<b>Precisión</b>	<b>Recall</b>
YOLOv7	0.841	0.831	0.781
YOLOv8	0.860	0.835	0.781
YOLO-NASs	0.875	0.877	0.734

**Tabla 7.** Comparación de métricas modelos de YOLO para Frijol 3

mAP de 0.841 con una Precisión de 0.831 y un Recall de 0.781, mostrando un buen equilibrio entre precisión y detección. Por su parte, YOLO-NASs, con el mejor mAP (0.875) y la mayor Precisión (0.877), ofrece predicciones más exactas, pero su Recall de 0.734 es menor, lo que indica que detecta menos instancias.

### 4.3. Funcionamiento de la GUI

La creación de la GUI fue realizada con la herramienta Tkinter disponible en Python. Se optó por un ejecutable para PC que facilita la conectividad con los servidores de Roboflow y permite usar los modelos para hacer predicciones. A continuación, se exponen partes específicas de la GUI para mostrar sus resultados; el código de la misma se encuentra en el repositorio. GUI



**Figura 21.** *Pantalla de inicio*

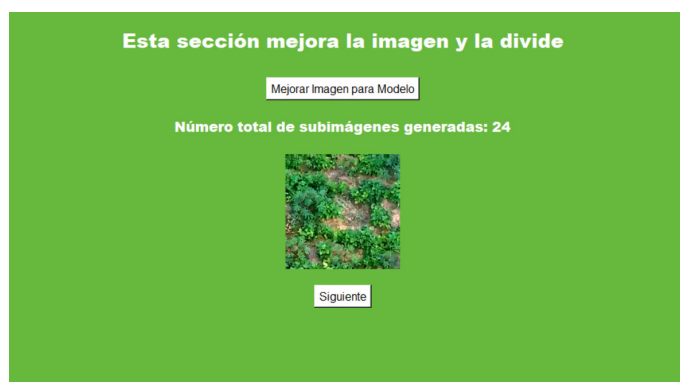
El usuario carga la imagen, tras lo cual se inicia el análisis con una primera clasificación, que determina la etapa de crecimiento del cultivo. En esta, se proporciona una explicación detallada sobre dicha etapa, junto con una estimación de los días aproximados en los que el cultivo se encuentra:



**Figura 22.** *Cuadro del programa con la predicción*

Una vez cargada y clasificada la imagen original, se aplican mejoras en contraste, color y

saturación. Luego, la imagen se divide en subimágenes de 1280 x 1280 píxeles para facilitar la detección de las plantas de frijol.



**Figura 23.** Cuadro imagen mejorada y dividida

A continuación, el programa detecta las plantas en cada una de las subimágenes utilizando el modelo correspondiente de Roboflow. Posteriormente, reconstruye la imagen original y clasifica las plantas detectadas según su tamaño, de mayor a menor, destacándolas en recuadros de color verde, azul y rojo.



**Figura 24.** Cuadro con las plantas detectadas y clasificadas

De igual manera, el sistema genera un archivo emergente que muestra la imagen final procesada, en la que se destacan las plantas detectadas y clasificadas. Este archivo emergente permite al usuario examinar con mayor precisión los resultados obtenidos, observando tanto la clasificación por tamaños como los recuadros de color verde, azul y rojo que resaltan las plantas.

## 5. Conclusiones

- El desarrollo de este proyecto de investigación ha sido un proceso integral que abarcó diversas etapas, comenzando con la recolección del conjunto de datos. Esta fase presentó múltiples desafíos, como el ruido introducido por la maleza, la falta de orden en los surcos, la presencia de plantas no deseadas y la calidad variable de las imágenes. Para abordar estas dificultades, se realizó un procesamiento de imágenes que mejoró la saturación, nitidez y contraste, facilitando así una base más sólida para el análisis posterior.
- El clasificador basado en una red neuronal convolucional (CNN) ha demostrado ser eficaz para la clasificación de imágenes en las etapas de crecimiento del cultivo de frijol. La implementación de validación cruzada K-Fold con tres particiones (`n_splits=3`) utilizando `MultilabelStratifiedKFold` garantizó una distribución balanceada de las clases, mejorando la consistencia de los resultados.
- La incorporación de herramientas y técnicas en la arquitectura de la red neuronal convolucional (CNN) es esencial para optimizar su rendimiento. Las capas de Dropout son fundamentales para prevenir el sobreajuste. Asimismo, los callbacks como Early Stopping permiten interrumpir el entrenamiento cuando no se observa mejora en la pérdida de validación durante varias épocas consecutivas, evitando entrenamientos innecesarios. Además, `ReduceLROnPlateau` ajusta la tasa de aprendizaje tras ciertas épocas sin mejoras, facilitando así la convergencia del modelo en etapas avanzadas. La implementación de estas técnicas refuerza la eficacia y eficiencia del modelo en la clasificación de imágenes.
- Con el fin de optimizar la detección y el conteo de las plantas, cada imagen se dividió en múltiples subimágenes. Gracias a la plataforma Roboflow, se crearon modelos específicos para cada una de las clases correspondientes a las etapas de crecimiento del frijol. Se implementaron modelos utilizando YOLOv7, YOLOv8 y YOLO-NASs, de los cuales se determinó que YOLO-NASs proporcionó el mejor rendimiento en términos de precisión y eficiencia, gracias al uso de Neural Architecture Search (NAS), que permite encontrar la mejor arquitectura de manera automática, optimizando tanto la velocidad como la precisión del modelo.

## 6. Recomendaciones

- Para mejorar la calidad del dataset, se sugiere invertir en un dron de mayor calidad que proporcione imágenes más nítidas y con mejor resolución. Además, es fundamental organizar el cultivo para reducir la presencia de maleza, lo cual facilitará un etiquetado más preciso y mejorará la calidad del dataset. Esto, a su vez, contribuirá a un rendimiento superior del modelo de detección y conteo.
- Se recomienda establecer un entorno de ejecución que incluya unidades de procesamiento gráfico (GPU) para mejorar significativamente la velocidad y eficiencia en el entrenamiento de los modelos de inteligencia artificial. Aunque Google Colab es una herramienta útil que proporciona acceso gratuito a GPUs, presenta limitaciones en cuanto al tiempo de ejecución y a los recursos disponibles.
- Se recomienda la incorporación de sensores de Internet de las Cosas (IoT) en el campo para recopilar datos en tiempo real sobre condiciones ambientales como la temperatura, humedad del suelo y niveles de luz. Estos datos podrían integrarse con las imágenes capturadas por los drones, proporcionando un análisis más completo del estado del cultivo y facilitando una toma de decisiones más informada.
- Mejora de la Interfaz de Usuario: Se recomienda desarrollar una interfaz más intuitiva y accesible que facilite la interpretación de los resultados y datos del sistema. Esto podría incluir visualizaciones interactivas, paneles de control personalizables y alertas automáticas para ayudar a los agricultores a monitorear el estado de sus cultivos de manera más eficiente.
- Detección de Anomalías: Se recomienda incorporar técnicas avanzadas de análisis de imágenes, como la detección de anomalías mediante algoritmos de inteligencia artificial, para identificar rápidamente plantas enfermas o en mal estado. Esta funcionalidad mejoraría la capacidad del sistema para ofrecer recomendaciones de intervención específicas.

### Referencias

- Agronegocios. (2024). *La fazenda, el negocio que cumplió 10 años*. Descargado de <https://www.agronegocios.co/finca/la-fazenda-el-negocio-que-cumplio-10-anos-2622463> (Accedido: 2024-09-26)
- Akond, A. S. M. G. M., Khandaker, L., Berthold, J., Gates, L., Peters, K., y Delong, H. (2011). Anthocyanin, total polyphenols and antioxidant activity of common bean. *Journal of Food Science*, 76(8), C1222–C1227.
- Badgujar, C. M., Poulouse, A., y Gan, H. (2024). Agricultural object detection with you only look once (yolo) algorithm: A bibliometric and systematic literature review. *Computers and Electronics in Agriculture*, 223, 109090. Descargado de <https://www.sciencedirect.com/science/article/pii/S0168169924004812> (Accessed: 2024-09-25) doi: 10.1016/j.compag.2024.109090
- Comisión de la Verdad. (2024). *Los campesinos le damos de comer al país, pero el país no reconoce la precariedad de la vida en el campo*. Descargado de <https://comisiondelaverdad.co/> (Accedido: 2024-07-12)
- de Información Agroalimentaria y Pesquera, S. (2019). *Reporte de aptitud agroclimática de México del frijol oi 2019-2020*. Descargado de [https://www.gob.mx/cms/uploads/attachment/file/495087/Reporte\\_de\\_Aptitud\\_agroclim\\_tica\\_de\\_M\\_xico\\_del\\_frijol\\_OI\\_2019-2020.pdf](https://www.gob.mx/cms/uploads/attachment/file/495087/Reporte_de_Aptitud_agroclim_tica_de_M_xico_del_frijol_OI_2019-2020.pdf) (Accedido: 2024-09-25)
- Econexia. (s.f.). *¿qué papel juega el sector agrícola en la economía colombiana?* Descargado de <https://econexia.com/es/contenidos-articulo/agroindustria-y-alimentos> (Accedido: 2023-10-01)
- Gawande, V., Zschech, P., y Heinrich, K. (2023). Potencial de las tecnologías de agricultura de precisión para una agricultura ecológica. *International Journal of Plant Soil Science*, 35(19), 101–112.
- Géron, A. (2022). *Hands-on machine learning with scikit-learn, keras, and tensorflow* (3rd ed.). O'Reilly Media, Inc.

Ministerio de Agricultura Colombia, U. d. P. R. A. U. (2024). *La upra adelanta la zonificación del cultivo de fríjol en colombia*. (Bogotá)

Ospina Parra, C. E., Martínez Medrano, J. C., Contreras Valencia, K., y Tautiva Merchan, L. A. (2020). Análisis socioeconómico del cultivo de fríjol en cundinamarca (colombia), para la identificación de un sistema agroalimentario localizado (sial). *RIVAR (Santiago)*, 7(21), 13-32.

Padhiary, M., Saha, D., Kumar, R., Sethi, L. N., y Kumar, A. (2024). Enhancing precision agriculture: A comprehensive review of machine learning and ai vision applications in all-terrain vehicle for farm automation. *Smart Agricultural Technology*, 8.

Radoglou-Grammatikis, P., Sarigiannidis, P., Lagkas, T., y Moscholios, I. (2020). A compilation of uav applications for precision agriculture. *Computer Networks*, 172, 107148.

### **Apéndice A: Repositorio en GitHub**

Se ha establecido un repositorio en GitHub Repositorio en GitHub. con el propósito de facilitar el acceso académico y fomentar la colaboración en el proyecto. Este repositorio incluye todos los códigos, tales como los de mejora de la calidad del dataset, división de las imágenes, modelos de clasificación y los modelos de detección y conteo. Esto promueve el aprendizaje y alienta a otros investigadores y estudiantes a contribuir y mejorar el sistema, ampliando así su impacto en la comunidad académica.

### **Apéndice B: Videos del Cultivo**

Como parte del proceso de recolección del dataset para este proyecto, se capturaron videos del cultivo de frijol en diversas etapas de crecimiento. Estos videos son fundamentales para comprender el comportamiento del cultivo y se utilizaron como referencia para el procesamiento y análisis de las imágenes. Los videos están disponibles en el siguiente enlace de Google Drive: Videos del Cultivo. A través de estos materiales audiovisuales, se puede observar la evolución del cultivo y las condiciones ambientales en las que se llevó a cabo el experimento, proporcionando un contexto valioso para el desarrollo del sistema de supervisión del crecimiento del cultivo de frijol.

### **Apéndice C: Funcionamiento de la GUI**

En el siguiente enlace se encuentra disponible una demostración detallada sobre el funcionamiento de la Interfaz Gráfica de Usuario (GUI) desarrollada para la clasificación y detección de plantas del cultivo de frijol. A través de este recurso, se puede visualizar la interacción del usuario con la interfaz, desde la carga de imágenes hasta la detección y clasificación de las plantas dentro de las subimágenes generadas. El enlace proporciona acceso a un video explicativo y a los archivos de código de la aplicación: [Funcionamiento GUI](#).