

Monitoreo Activo con Drones del Estado de Plantas de Arándanos en Invernaderos

Andres Gomez, *Master Student*. Daniel Martinez, *Master Student* Luisa Martinez, *Master Student*, Enrique González, *Ph.D.*

Abstract— El cultivo de arándanos ha experimentado un aumento significativo en los últimos años, generando ingresos sustanciales para el sector agrícola. Sin embargo, el cuidado de estas plantas, que producen frutos dos años después de la germinación, implica considerar diversos factores. El monitoreo activo, es crucial en el primer año de desarrollo, se realiza comúnmente de forma manual, especialmente en invernaderos, donde los factores ambientales pueden ser controlados. A pesar de esto, las plantaciones de arándanos pueden experimentar síntomas de estrés, como manchas rojizas en las hojas y adelgazamiento de tallos. El estrés hídrico es una causa común de malestar en los cultivos, afectando tamaño, cantidad y calidad del fruto. La implementación de tecnologías como inteligencia artificial y robótica en la agricultura ha permitido la automatización del monitoreo activo, pero muchos agricultores aún no tienen acceso debido a costos elevados y dificultades de implementación. En este contexto, se propone una rutina de inspección autónoma mediante un UAV para la detección de estrés en plantas de arándanos. El trabajo se estructura en cuatro fases: análisis del estado del arte, recolección de datos para construir datasets, clasificación mediante redes neuronales convolucionales y control visual con marcadores ArUco y algoritmos de lógica difusa. Este enfoque busca ofrecer una solución eficiente y accesible para el monitoreo de la salud de las plantas en entornos controlados como invernaderos

Declaración de impacto — La utilización de vehículos aéreos no tripulados (UAV) se ha destacado en diversas áreas y aplicaciones. En particular, el empleo de estos dispositivos, combinado con inteligencia artificial, para el monitoreo y cuidado de cultivos ha experimentado un aumento significativo. A pesar de este avance, los elevados costos y las dificultades de acceso e implementación limitan la adopción generalizada de estas tecnologías en la agricultura. No obstante, a través de un algoritmo de lógica difusa para el control del dron y redes neuronales profundas encargadas de clasificar las imágenes capturadas por el mismo, se logra llevar a cabo un monitoreo activo dentro del invernadero. Esta implementación contribuye a reducir las pérdidas de producción de frutos en plantas de arándanos al identificar de manera temprana posibles plantas enfermas.

Index Terms— UAV, precision agriculture, artificial intelligence, classification, blueberry cultivation, CNN, active monitoring, transfer learning, YOLO

I. INTRODUCTION

La producción de arándanos en Colombia ha estado aumentando en los últimos años generando ingresos significativos para el sector agrícola. Sin embargo, el cuidado de estas plantas implica considerar diferentes factores ya que producen frutos dos años después de ser germinadas, siendo el primer año crucial para un desarrollo adecuado. El cultivo de arándanos demanda tiempo y requiere de un monitoreo activo que generalmente se realiza de forma manual, muchos agricultores cultivan en invernaderos, donde los factores ambientales como la temperatura, la luz, y la humedad pueden ser controlados, pero incluso bajo estas condiciones, las plantaciones de arándanos pueden experimentar síntomas de estrés que se evidencian por ejemplo en presencia de manchas con tonos rojizos en las hojas, marchitamiento, y adelgazamiento de los tallos. El estrés hídrico es una de las causas más comunes de malestar en los cultivos, impactando el tamaño, cantidad y la calidad del fruto.

Hoy en día, tecnologías como inteligencia artificial y robótica son implementadas en distintas industrias en las que se incluye la agricultura, abriendo el camino a herramientas que permiten por ejemplo automatizar tareas como el monitoreo activo que es fundamental para detectar problemas y enfermedades en etapas tempranas, y tomar medidas correctivas oportunamente, sin embargo, muchos cultivadores locales todavía no cuentan con este tipo de herramientas debido principalmente a su elevado costo y a la dificultad tanto en el acceso como su implementación, por esta razón se propone una rutina de inspección autónoma basada en control servo visual para un UAV (vehículo aéreo no tripulado) de bajo costo provisto únicamente con una cámara RGB, que permita realizar la detección de síntomas de estrés en plantas de arándanos. Para esto hemos encontrado un invernadero de arándanos, en donde se realizaron diferentes tomas de imágenes de plantas sanas y enfermas utilizando un dron DJI Tello, cuya elección se realizó teniendo en cuenta principalmente el espacio limitado del invernadero, buscando evitar causar daños en las plantas y las estructuras.

El siguiente trabajo está dividido en 4 fases: En la primera fase de análisis del estado del arte se identifican trabajos relacionados que contengan información relevante sobre algoritmos y técnicas utilizados para monitoreo activo basado en control

servo visual, y clasificación de imágenes enfocados en la detección de estrés en plantas. Además, se realiza un análisis y contextualización del problema lo que permite encontrar una ruta para comenzar a abordar una posible solución.

En seguida, se inicia la fase de recolección de datos enfocada en la construcción de datasets, y que consiste inicialmente en recolectar imágenes por medio de vuelos controlados por un operador para crear un conjunto de datos orientado a detección de plantas de arándanos, y posteriormente se realizan diferentes sesiones de captura estructurada y semi automatizada para recolectar imágenes y etiquetarlas con el fin de construir el conjunto de datos para entrenar un modelo clasificador de plantas enfermas y sanas, teniendo como base la opinión del administrador del invernadero para determinar el estado de cada una de las plantas capturada.

La fase de clasificación entre plantas sanas y con síntomas de estrés, se realiza mediante el entrenamiento y la implementación de un modelo de redes neuronales convolucionales para clasificación de imágenes, además, se aplican técnicas de procesamiento de imágenes buscando incrementar la confianza en la clasificación. Adicionalmente se implementa un algoritmo de inferencia basado en lógica difusa que calcula y asigna una confianza a cada imagen clasificada, a partir de su brillo, nitidez y certeza del clasificador a la hora de asignarle una etiqueta, las imágenes son capturadas desde distintas posiciones respecto a la planta. Por último, se genera un reporte por planta a partir de la etiqueta asignada por el modelo de clasificación y la confianza asignada por el sistema de lógica difusa a cada una de las imágenes.

La fase de control visual se basa en la propuesta de una rutina de inspección para obtener imágenes desde distintas posiciones de la planta a clasificar. En esta rutina, se hace uso de marcadores ArUco para la identificación e individualización de la planta objetivo. Se implementa un algoritmo de detección y seguimiento de objetos para posicionar correctamente el dron respecto a la planta y extraer la región de interés. Por último, si la confianza asignada por el sistema de lógica difusa no alcanza un determinado umbral, se toma una segunda imagen desde una distancia menor respecto a la planta y se guarda tanto la etiqueta como la confianza de la imagen que tenga un valor mayor para su posterior uso en la generación del reporte por planta.

Por último, en la fase de análisis de resultados, se compara realizando una simulación a partir de un banco de imágenes; el tiempo que toma ejecutar la rutina y el rendimiento de la clasificación en tres escenarios distintos: realizar acercamientos a la planta únicamente en las imágenes que no cumplen con el umbral de confianza, realizar acercamientos en todas las posiciones, y no realizar ningún acercamiento durante la rutina de inspección.

II. ANÁLISIS DEL ESTADO DEL ARTE

Para el desarrollo de un sistema de control visual es indispensable tener en cuenta los siguientes 3 temas claves en el análisis del estado del arte: el proceso de control del vehículo

no tripulado donde se analizan los distintos algoritmos que se pueden implementar para la detección del objetivo teniendo en cuenta la distancia del dispositivo UAV y su objetivo. Por otro lado, la detección de objetos en el que se identifican algoritmos y estrategias utilizadas en trabajos anteriores teniendo en cuenta los diferentes tamaños del objeto, iluminación y perspectiva. Por último, en la clasificación de imágenes se analiza cómo identificar cuáles son las plantas sanas y no sanas en un cultivo de arándanos y se observan los algoritmos de inteligencia artificial que se utilizaron en trabajos anteriores. A continuación, se explora cada uno de los temas claves.

A. Detección de Objetos

Satish Kumar [1] propone un modelo para detección en el contexto de reconocimiento en tiempo real de objetos comúnmente presentes en carreteras, destacando la implementación de la arquitectura SSD (Single Shot Detection) destinada a reconocer objetos en un solo paso, una característica importante debido a que se trabaja en tiempo real, y muchas arquitecturas trabajan con dos pasos para realizar la detección aumentando el tiempo necesario para realizar la inferencia. Se destaca también el uso de la técnica FPN (Feature Pyramid Networks) que consiste en utilizar ventanas de convolución en diferentes tamaños, formando una pirámide, con el fin de detectar objetos en diferentes formas y tamaños. La recolección de imágenes se realizó grabando videos desde cámaras ubicadas en vehículos mientras se conducía por carreteras con tráfico fluido; los videos son separados frames obteniendo 11.300 imágenes, y etiquetando 9 clases de diferentes objetos se construye el correspondiente dataset. El etiquetado de las imágenes se realizó utilizando una herramienta open source llamada LabelImg, que permite exportar el dataset en formato Pascal VOC. La configuración utilizada para el entrenamiento, 25000 pasos, con tamaño de batch 2, y una tasa de aprendizaje 0.00399, obteniendo como métricas accuracy 0.73, total loss 0.63, recall 0.78.

Sabina N [2], presenta un comparativo entre el rendimiento presentado por un modelo SSD MobileNet, y el presentado por un modelo YOLO (You Only Look Once), destacando que ambos modelos son clasificados como detectores de un solo paso, reduciendo el tiempo de inferencia lo que es importante cuando se requiere trabajar en tiempo real, y otras arquitecturas trabajan con dos pasos para lograr la detección. Las características más relevantes del modelo YOLO, son su velocidad y alta precisión que ofrece, lo que lo hace ideal para inferencias en tiempo real, este modelo cuenta con varias versiones que han venido optimizando procesos, y se han añadido distintas funcionalidades mejorando la precisión y reduciendo el tiempo necesario. La arquitectura básica de Yolo cuenta con 24 capas de convolución, 4 capas max-pooling, y 2 capas completamente conectadas al final. El modelo MobileNet SSD se destaca por implementar la arquitectura SSD (Single Shot Detection) que lo hace ideal para detección en tiempo real. Este modelo introduce técnicas de detección multi referencia y multi resolución que permite la detección en diferentes escalas y ajustando con precisión las cajas delimitadoras que ubican el objeto detectado, y resultan más acordes al tamaño y forma del objeto. Se resalta de este trabajo que, para ambos modelos,

Yolo y MobileNet SSD, se obtienen resultados aceptables al detectar diferentes objetos de diferentes tamaños, y variando aspectos como la iluminación, la perspectiva de la imagen, la complejidad del fondo de la imagen y la presencia de otros objetos diferentes en la misma imagen. Para propósitos de trabajo en tiempo real el modelo Yolo ofrece mayor precisión y mejor rendimiento, aunque utiliza mayor cantidad de recursos de cómputo. Por lo que cuando se cuenta con recursos limitados, o se requiere trabajar con sistemas embebidos, se recomienda utilizar el modelo SSD MovilNet.

B. Control Servo Visual de un UAV

Koppány Máthé [3] expone un sistema basado en control servo visual para un dron, en donde el objetivo es detectar y localizar un objeto, en este caso un semáforo, acercarse y volar alrededor de él usando una velocidad lateral constante y una manteniendo una distancia frontal de referencia previamente establecida. Se construye un controlador PID que usa como realimentación el error entre el centroide del objeto y el centroide de la cámara. La detección del objetivo se logra entrenando e implementando un modelo YoloV5, y se destaca el uso de la técnica data Augmentation Simple Scale Invariant para mejorar el dataset construido. El cálculo de la distancia respecto al objetivo se hace a partir de un tamaño de referencia tomado de una captura obtenida desde la distancia deseada. Es de destacar también la implementación de un algoritmo llamado filtro de Kalman que permite predecir el estado de un objeto a través del análisis de su movimiento en una serie de fotogramas, y así predecir el siguiente estado, esto permite mantener el foco sobre un objetivo específico y evitar los errores de detección.

Muaz Al Radi [4] postula un sistema en el que se extraen características como el centroide del objetivo (x , y), y el área del objetivo obtenidos entrenando un modelo de detección de objetos, en este caso con 50 epoch, se entrena un modelo YOLO en su versión 5. Utilizan un control PID para las velocidades yaw, Z (arriba abajo), y X (adelante y atrás). El proceso de inspección es: 1 Detectar el objetivo que es la estructura, 2 Alcanzarla dicha estructura, 3, comenzar a subir hasta llegar a la cima, Localizar la llamarada, Ubicar la llamarada en el centro de la cámara, tomar fotografías, volver, y aterrizar. Construyen el dataset para detección descargando y etiquetando imágenes de internet. Se usaron 1742 imágenes que fueron etiquetadas con la herramienta Computer Vision Annotation Tool (CVAT)

Miguel A. Olivares-Méndez [5] presenta un sistema que trabaja en paralelo con 2 controladores de lógica difusa para una plataforma UAV dotada con una cámara RGB, este algoritmo busca ubicar el centro del objetivo en el centro de la cámara ejecutando las velocidades necesarias para lograrlo. Se destaca la implementación de un algoritmo para mantener una referencia sobre el objetivo detectado, y así evitar el ruido debido a la presencia de otros objetos de la misma naturaleza en la misma imagen, el algoritmo implementado fue Lucas-Kanade Tracker. Se hace uso un controlador para cada eje de movimiento, (X) pitch, (Z) yaw, y (Y) arriba abajo, cada controlador utiliza funciones triangulares que toman como error una diferencia en píxeles entre el centro del objetivo y el centro de la cámara, el error máximo permitido para el movimiento

yaw es 160px, el máximo error permitido para el movimiento pitch es 120px.

Ertugrul Bayraktar [23] propone un marco de trabajo que emplea vehículos aéreos no tripulados (UAVs) de bajo costo para abordar los desafíos asociados con las condiciones impredecibles del entorno exterior y para establecer un esquema efectivo en la detección y conteo de plantas ornamentales. El proceso de detección inicia con la revisión de imágenes capturadas en el campo, y la secuencia de imágenes se somete a un modelo VGGNet ajustado para determinar si el UAV se encuentra sobre un paisaje cultivado o una zona urbana. Una vez que el sistema identifica la consulta como perteneciente a un área de cultivo, se implementa un filtro de color para retener únicamente tonalidades verdes, ya que las plantas de interés presentan este color. El autor detalla cómo se logran resultados exitosos a través de algoritmos de filtrado de color que incorporan relaciones geométricas, como la altura de la zona de interés, incluso en la presencia de árboles diversos en la escena. Además, se destaca que la segmentación de color en el espacio de vectores RGB proporciona resultados superiores, y se presenta una ecuación para el filtrado de colores no verdes en dicho espacio. En la construcción del conjunto de datos, las imágenes se escalan a una resolución de 300x300, dividiendo el conjunto en 500 imágenes para entrenamiento y 250 para pruebas. Se emplearon los algoritmos YoloV3 y RetinaNet, y se realizó una comparación entre ambos modelos, arrojando un accuracy del 79.85% para YoloV3 y 73.41% para RetinaNet.

C. Clasificación de Imágenes

El autor Eduardo A Holzapfel [6], llevó a cabo investigaciones sobre los efectos de la falta de agua en distintas etapas de desarrollo del arándano. Diferentes parámetros son utilizados para indicar los estados hídricos en una planta, entre ellos y el más común es el potencial del agua en la hoja. Los resultados revelaron que el estrés hídrico, incluso en niveles moderados tuvo un impacto significativo en el rendimiento de la producción, se observó una reducción tanto en el peso promedio del fruto como en su tamaño. También se observaron cambios en la planta como, cambio de color en sus hojas tomando tonos rojos, presencia de hojas marchitas, y reducción en el crecimiento de la planta. Las temporadas de sequía afectan directamente el potencial hídrico de las hojas y aunque se presentan diversos síntomas de estrés en la planta como la reducción del crecimiento, menos fotosíntesis, más crecimiento radicular (este es el proceso de multiplicación, elongación celular en la punta de la raíz y donde se forman nuevas células [15]) y presencia de tonalidades rojas en las hojas siendo este un síntoma más frecuente en el cultivo. También se evidencia en las investigaciones realizadas que durante 7 días sin agua la planta genera marchitez en sus hojas, en estas condiciones las plantas jóvenes tienden a secarse fácilmente y la planta puede ser más propensa a las afectaciones hídricas tras el comienzo del proceso de maduración del fruto.

Krishna Neupane [7] plantea la reducción de errores de la imagen realizada mediante el preprocesamiento de imágenes, por lo que se requiere crear un método que incluya:

segmentación de imágenes, conversión de colores y filtros y realizar correcciones de brillo o escalas de grises. Se puede utilizar diferentes tipos de software para la transformación de las imágenes, como el environment for visualizing images ENVI versión 4.7. Para tener una mejor etiquetación de la imagen se recomienda remover o cortar partes de la imagen innecesarias. Luego, se elige un modelo de inteligencia artificial para el procesamiento de la data de la imagen, por lo que se recomienda utilizar modelos como: árboles de decisión, K-means, Support Vector Machine (SVM), análisis de regresión y redes convolucionales. Se muestra una comparativa entre los modelos SMV que obtiene una precisión del 93% y el modelo de CNN con una precisión del 92.85%. Estos modelos fueron utilizados para identificar posibles enfermedades en un cultivo de plantas monitoreadas por medio de un dispositivo UAV.

A partir del análisis del estado del arte, y con el fin de proponer una estrategia para la inspección de plantas de arándanos, se considera utilizar un dispositivo pequeño UAV programable para la toma de imágenes. Asimismo, se contempla el uso de redes neuronales convolucionales y técnicas de preprocesamiento de imágenes para la detección y clasificación de las plantas

III. CONTEXTUALIZACIÓN

Para el desarrollo del sistema de control visual se eligió un invernadero ubicado en Guasca, Cundinamarca. Durante la evolución del sistema se busca distinguir entre una planta sana y una planta enferma teniendo en cuenta principalmente el color de las hojas. Dado que las plantas están dispuestas en filas, se explora la selección del dron contemplando el tamaño del invernadero, asegurándose de que el dron pueda cubrir eficientemente el camino entre las hileras. A continuación, se explora a detalle los rasgos de las plantas y la selección del dron con base a las características del invernadero.

A. Estrés en plantas de Arándanos

El estrés en las plantas de arándanos se puede dar por diversos factores como por ejemplo el tipo de acidez del suelo y la edad de la planta, sin embargo, un factor muy común que puede afectar la producción es el estrés hídrico [17]. una mala hidratación en las plantas genera sequedad en su raíz, en el tallo, y en las hojas, también reduce el crecimiento en la planta, disminución del peso y tamaño en frutos en su etapa productiva [17], además conlleva a que las hojas pierdan la tonalidad verde y generen un color rojo en su lugar. Para identificar las plantas sanas y enfermas presentes en el invernadero, se contó con la opinión y ayuda del administrador del invernadero, que ratificó los síntomas visuales utilizados para dar un veredicto.

1. Planta de arándanos sana

Una planta de arándanos sana como la que se presenta en Fig. 1. muestra vigorosidad, lo que significa tallos gruesos y fuertes, las hojas muestran tonos verdes intensos.



Fig. 1. Planta de arándanos sana

2. Planta de arándanos enferma

Una planta enferma por estrés como la presentada en Fig. 2. muestra falta de vigorosidad, tonos rojizos en sus hojas, presenta menor crecimiento, y sus hojas se ven marchitas. En este caso las enfermedades que puede tener una planta de arándanos no difieren en su tipo sino en las condiciones climáticas en las que se encuentra. Cuando una planta se expone a exceso de hidratación como por ejemplo temporadas de lluvias fuertes puede conllevar a enfermedades como la pudrición de la raíz [16], o por el contrario si no cuenta con suficientes recursos hídricos o nutrientes, sus tallos, hojas y raíz tienden a secarse, las hojas cambian de color a un tono rojo y su tallo es delgado lo cual dificulta su crecimiento y producción.



Fig. 2. Planta de arándanos enferma

Teniendo en cuenta que la característica visual que diferencia una planta sana a una enferma es la coloración rojiza en sus hojas, este trabajo se enfoca en detectar estas características visuales por medio de una rutina de inspección realizado por un dispositivo UAV de forma semi automática y así monitorear cuáles son las plantas enfermas que se encuentran en el invernadero.

B. Selección del dron

Para seleccionar la plataforma UAV (dron) con la cual trabajar, se tuvo en cuenta principalmente el espacio de maniobra limitado que se tendría en un invernadero, ya que los agricultores desean aprovechar lo mejor posible el espacio disponible. En este caso, el invernadero en el que se trabajó tiene una distancia entre filas de 1.5 metros y cada maceta está inmediatamente al lado de la siguiente formando así una fila (Fig. 3), además las macetas tienen una altura promedio de 0.5 metros. Se identificó también que la distancia segura mínima

respecto a la planta en la que se puede desplazar el dron estaría en un rango entre 30 y 50 centímetros.



Fig. 3. Invernadero en Guasca, Cundinamarca

Por lo anterior, el dron debe ser lo suficientemente pequeño y maniobrable para moverse fácilmente dentro del invernadero sin causar daño a las plantas o estructuras, de igual forma, debe tener características de vuelo estables y un control lo suficientemente preciso para garantizar la captura de datos de manera confiable y segura. Es importante resaltar que al trabajar dentro de un espacio cerrado no se requiere hacer frente a condiciones adversas del clima, por lo que resulta viable el uso de un dron de pequeñas dimensiones y de baja potencia.

El Dji tello está diseñado para interiores, provisto con una velocidad máxima de 8m/s y un alcance máximo de 30 metros, sus dimensiones son de 9.8 x 9.25 centímetros. Este dron está dotado con una cámara RGB de resolución 720 x 960 píxeles, y cuenta con un tiempo de vuelo aproximado de 13 minutos. La comunicación y programación del dron se hace a través de librería DJITelloPy, Open Source y disponible en GitHub [9], para ser usada con el lenguaje de programación Python, y que provee una interfaz programable para comunicación con el dron usando para ello un SDK provisto por el fabricante DJI.

C. Redes neuronales Profundas

Una red neuronal profunda está compuesta por redes neuronales artificiales con varias capas ocultas entre las capas de entrada y salida. Una red neuronal artificial recibe parámetros de entrada, luego procesa esta información con cálculos complejos y por último entrega esta información en la salida para resolver problemas de clasificación y reconoce patrones como caras, voz, frutas, animales [18][22] etc.

La arquitectura de una red neuronal profunda consta de varias capas ocultas con millones de nodos conectados entre sí de forma no lineal. Dichos nodos cuentan con unos pesos, estos pesos estimulan al siguiente nodo si el valor del peso es positivo, si es negativo lo suprime, siempre será relevante el nodo que tenga mayor peso. Así mismo las redes neuronales profundas al ser más complejas, requieren mayor entrenamiento y millones de datos, no miles como una red neuronal sencilla [19]. Existen diferentes tipos de redes neuronales profundas, estos tipos se pueden clasificar según cómo fluyen los datos desde el nodo de entrada y el nodo de salida. Algunos ejemplos son:

1. **Redes neuronales prealimentadas:** los datos van en una sola dirección, los nodos van conectados desde la entrada hasta la salida. Esta red realiza una retroalimentación para mejorar la predicción.
2. **Redes de retro propagación:** Los datos fluyen desde la entrada hasta la salida por un solo camino, sin

embargo, para encontrar cuál camino es el correcto para la salida. Para encontrar dicho camino la red genera bucles de retroalimentación, por lo que los nodos al adivinar cual es el camino asignan en el recorrido pesos a los caminos que sean más precisos y continúan al siguiente nodo, cuando el camino es incorrecto asignan un valor de peso bajo. Luego los nodos realizan una nueva predicción siguiendo todos los caminos de pesos altos, por lo que el ciclo se repite hasta que encuentre la trayectoria correcta hacia la salida.

3. **Redes neuronales convolucionales (CNN):** Las capas internas de las redes neuronales convolucionales llevan a cabo operaciones matemáticas particulares, como la síntesis o el filtrado, conocidas como convoluciones [19]. Son utilizadas para la clasificación de imágenes ya que extraen características relevantes de la imagen como los límites, la paleta de colores y la percepción de profundidad, estas características son fundamentales para realizar la predicción. Cada capa interna identifica y analiza dichas características por lo que la hace eficiente en el control de la alta dimensionalidad de las imágenes [19]. La arquitectura de las CNN se divide en tres secciones, en primer lugar la capa extrae información de los datos ingresados mediante filtros preconfigurados, luego los datos son agrupados divididos en sectores o regiones diferentes y por último las capas totalmente conectadas establecen trayectorias neuronales suplementarias entre las distintas capas, posibilitando que la red adquiera comprensión de relaciones complejas entre las características y realice predicciones de nivel superior [20] en la siguiente figura (Fig 4) se expresa un ejemplo de una CNN

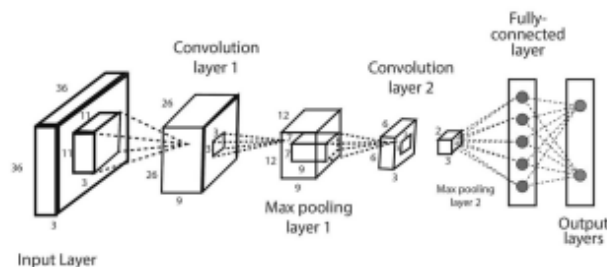


Fig. 4. Red neuronal convolucional con múltiples capas, tomado de [21]

Las redes neuronales profundas son complejas tanto como por su entrenamiento como por la cantidad de datos que requiere para mejorar su predicción, sin nombrar el tiempo y recursos de computadora que pueden requerir a la hora de entrenar la red neuronal. Existe hoy en día un concepto denominado aprendizaje por transferencia (transfer learning) lo cual nos permite acceder a redes neuronales pre-entrenadas y dichos modelos permiten el desarrollo eficaz de una solución completa para un problema en particular. Entonces el uso de transfer learning nos permite reutilizar conocimiento ya adquirido para resolver un determinado problema. En desarrollo del proyecto se utilizó transfer learning mediante Inception V3 y Yolo v8, ambos tienen redes convolucionales en su arquitectura, sin

embargo, en cuanto eficiencia Yolo V8 tuvo un mejor accuracy y en eficiencia de ejecución comparado con Inception V3.

IV. RECOLECCIÓN DE DATOS

La construcción del conjunto de datos es uno de los aspectos más importantes ya que es necesario para entrenar un algoritmo o una red neuronal que permita detectar y clasificar las plantas de arándanos.

La toma y recolección de imágenes se realizó teniendo en cuenta que al recorrer una fila del invernadero se tiene acceso únicamente una vista de 180° de la planta, ya que los 180° restantes son visibles del otro lado. Este proceso se dividió en dos fases:

En la primera, se recolectaron imágenes para crear un dataset genérico con el objetivo de reconocer y detectar plantas de arándanos. Con este dataset se busca entrenar una red neuronal que tenga la capacidad detectar plantas de arándanos y ubicar el bounding box (caja delimitadora) de cada planta presente en una imagen, para individualizarlas y así realizar una inspección de cada planta.

En la segunda fase se realizaron diferentes sesiones de toma de imágenes: en las primeras sesiones un operador pilotó el dron siguiendo una rutina para garantizar una recolección estructurada que permita detallar cada planta sana y enferma desde diferentes posiciones. En sesiones posteriores se incluye la captura de imágenes a diferentes distancias respecto a la planta, y se automatiza la rutina para que el dron realice la inspección de forma semi autónoma, utilizando un marcador ArUco [10] para identificar la planta a la que se aplicará la rutina de inspección. En Fig.5. se muestra una imagen de una planta objetivo junto a su marcador ArUco.



Fig. 5. Planta de arándanos con marcador ArUco

1. Dataset para detección de objetos

La toma de imágenes se realizó mediante vuelos controlados del dron capturando fotogramas que son enviados por el mismo, la primera toma fue el 11 de junio del 2023, y una segunda toma se realizó el 30 de julio del mismo año; se hace una recolección genérica de imágenes de plantas de arándanos para ser etiquetadas y así entrenar un modelo que realice detección. Para

lograrlo, se tomaron imágenes de las plantas siguiendo dos estrategias distintas:

- Realizar una toma inicial haciendo un barrido de cada fila del invernadero, para tener tomas rápidas de todas las plantas presentes en cada fila.
- Realizar tomas de cada una de las plantas, comenzando desde la parte superior de cada una de ellas hasta llegar a la maceta, recorriendo los 180° visibles de la planta.

El dataset construido para esta primera fase está conformado por un total de 2210 imágenes de diferentes plantas, sin discriminar entre plantas sanas y plantas enfermas. Una vez se realiza el etiquetado del dataset, éste se utiliza para entrenar una red neuronal para detección de plantas de arándanos.

2. Dataset para clasificación de estrés

En esta fase se recolectaron imágenes de forma estructurada teniendo en cuenta diferentes posiciones y distancias del dron respecto a una planta de interés. Cada planta es discriminada entre dos clases establecidas, SANA y ENFERMA, teniendo en cuenta principalmente la opinión del administrador del invernadero, guiado por los síntomas visuales que presenta la planta. El dataset resultante fue construido utilizando el recorte generado por el detector de plantas de arándano previamente entrenado, y cuenta con un total de 2534 imágenes, 1501 enfermas y 1033 sanas, que fueron capturadas aproximadamente a 30, 60 y 100 centímetros de distancia.

V. CLASIFICACIÓN DE IMÁGENES

Para determinar si una planta tiene presencia de estrés, se exploran diferentes opciones que incluye el procesamiento clásico de imágenes, y el entrenamiento e implementación de redes neuronales profundas enfocadas en la clasificación de imágenes.

A. Clasificación por color y forma de las hojas

Se realiza la aplicación de técnicas clásicas de procesamiento de imágenes para segmentar las hojas y las manchas rojas que evidencian la presencia de estrés hídrico. Para lograrlo se hace una conversión del espacio de color RGB al espacio de color LAB [14], éste espacio de color tiene en uno de sus ejes los colores verde y rojo en extremos diferentes, lo que favorece por un lado la segmentación de las hojas que presentan tonos verdes, y la segmentación las manchas que presentan tonos rojos.

Para esto, se logran identificar umbrales para cada uno de los canales correspondientes a los ejes del espacio de color LAB, creando un filtro para ser aplicado en cada imagen dando como resultado la segmentación de los colores que cumplen con dichos umbrales.

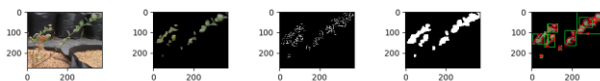


Fig. 6. Segmentación hojas en condiciones ideales

Al momento de aplicar los filtros construidos, se encuentra presencia de ruido, es decir segmentos de imagen que no corresponden a las hojas de la planta de interés. El filtro para tonos rojos presenta ruido debido a que el sustrato en el que está sembrada la planta está protegido con pajilla que tiene un color amarillo rojizo, por lo que en algunas ocasiones quedan incluidos pequeños fragmentos de pajilla en la segmentación, lo que genera falsos positivos.

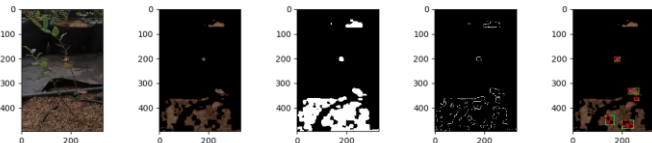


Fig. 7. Segmentación de manchas rojas condiciones normales

El filtro para segmentar las hojas presenta mayores problemas debido a que en la mayoría de los casos quedan incluidos fragmentos de las hojas de plantas ubicadas en la parte posterior de la planta de interés.

Para minimizar estos problemas, se buscó identificar características morfológicas de las hojas para obtener un resultado más limpio. Así, la imagen segmentada pasa por los siguientes procesos:

1. Filtro Gaussiano, que permite unificar los tonos cercanos para simplificar detalles.
2. Conversión a escala de Grises para trabajar con un solo canal.
3. Dilatación con 5 iteraciones, y Erosión con una ventana de 5x5. Que favorece la detección de bordes.
4. Filtro Canny para detectar los bordes predominantes.
5. Búsqueda de contornos cerrados que representarán las hojas y las manchas rojas.

Una vez se ubicados los contornos cerrados, se calculan las siguientes características morfológicas:

Rectangularidad: Determina qué tan rectangular es un contorno teniendo como medida un rango entre 0 y 1. Para eliminar el ruido obtenido en los tonos rojos con la pajilla antes mencionada, se establece un umbral mínimo de 0.5 para buscar contornos que tengan forma de cuadrado y forma rectangular. Para eliminar el ruido obtenido en las hojas se busca se establece un umbral mínimo de 0.6 para buscar formas predominantemente rectangulares.

Área: Se calcula y guarda el área de cada contorno presente en una imagen, para calcular la media y utilizarla como umbral mínimo para tener en cuenta únicamente los contornos igual o superior a la media calculada.

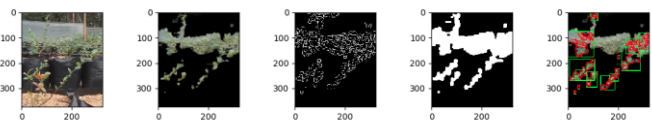


Fig. 8. Segmentación de hojas en condiciones difíciles

Como se muestra en las imágenes, aunque no se logra eliminar completamente el ruido, se obtienen resultados que se consideran aceptables debido a que en algunos casos se pierde

información relevante a la planta de interés, pero se compensa con el ruido obtenido.

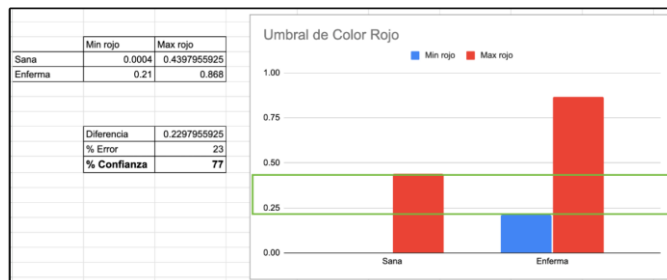


Fig. 9. Segmentación de hojas en condiciones difíciles

Una vez construidos los filtros, se ejecuta sobre *dataset*, se identifican los valores mínimos y máximos de porcentaje de rojo vs verde, identificando un umbral de 21% como posible presencia de estrés debido a que es el menor valor de rojo obtenido en las imágenes de plantas enfermas, como se muestra en la Fig 9. Teniendo en cuenta que el máximo valor de rojo encontrado en las imágenes de plantas sanas es de 44%, se identifica un rango de ambigüedad de 23% resaltado con el recuadro verde. Por esta razón se presume una confianza del 77% al momento de realizar una predicción.

B. Entrenamiento del Clasificador Inception V3

Este modelo fue entrenado con un total de 564 imágenes, se recortaron y etiquetaron las imágenes de forma manual, estas imágenes tienen un tamaño de 256 x 256, el modelo se entrenó con 50 epochs con una duración de 2h por ejecución, los recursos de la computadora se vieron afectados. El modelo tuvo un accuracy del 94.5%.

C. Entrenamiento del Clasificador YOLO

Se entrenó por 70 épocas un modelo Yolo de tamaño nano, versión 8, con 2534 imágenes recortadas por el detector previamente entrenado y etiquetadas manualmente como sana o enferma, para el entrenamiento, los recortes, escalados a un tamaño de 640 x 640 píxeles, fueron enviados al clasificador en lotes de 16. Este modelo logró un accuracy del 98.3% en el conjunto de imágenes de prueba y, al ejecutarse en la computadora utilizada para controlar el dron, demora aproximadamente 100 milisegundos en clasificar cada imagen. En Fig. 10. Se muestra la evolución del accuracy en el conjunto de validación durante el entrenamiento.

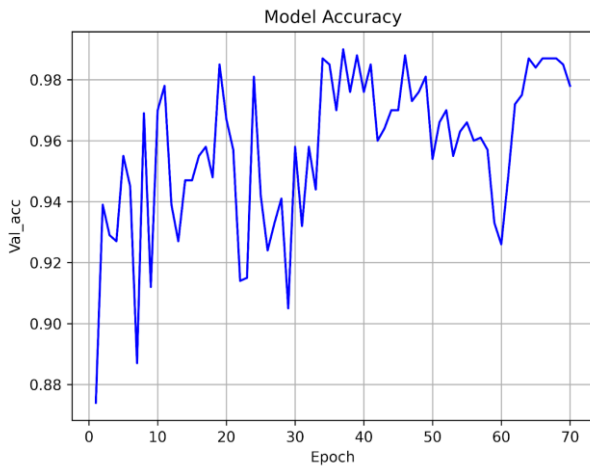


Fig. 10. Resultados del entrenamiento del modelo Yolo v8 para la clasificación de las fotografías de las plantas.

De los modelos y técnicas exploradas se eligió la implementación de un modelo Yolo V8 como motor de clasificación debido a la eficiencia ofrecida en cuanto a tiempo de ejecución y consumo de recursos de cómputo. Teniendo en cuenta que el modelo Inception V3 requiere de mayor tiempo para realizar una inferencia. Por otro lado, clasificación por segmentación de color ofrece una confianza sustantivamente menor que la ofrecida por los modelos de redes neuronales.

VI. DESARROLLO DE LA RUTINA DE INSPECCIÓN

En esta sección se explica a detalle el sistema de control visual diseñado para ejecutar una rutina de inspección sobre las plantas de arándanos, y consta de tres etapas fundamentales. La primera etapa consiste en la detección de la planta objetivo y el posicionamiento del dron respecto a ésta, con el fin de obtener imágenes de dicha planta desde distintas posiciones. En la segunda etapa, cada una de las imágenes recolectadas se clasifican como sana o enferma y a continuación se les asigna una confianza mediante un sistema de inferencia basado en lógica difusa. Finalmente, en la tercera etapa se genera un reporte de la planta, teniendo en cuenta el peso y clase asignada a cada una de las imágenes correspondientes a ella. En Fig. 11. se muestra a detalle los pasos ejecutados en cada una de las etapas de la rutina de inspección.

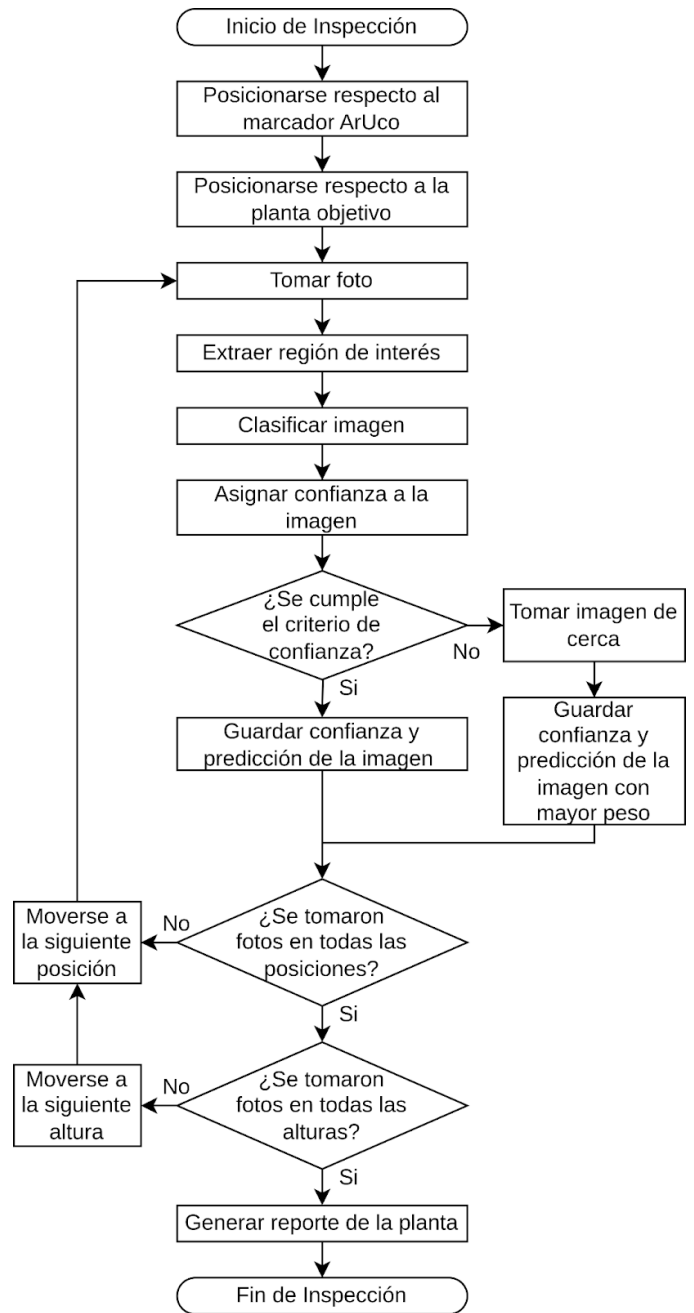


Fig. 11. Diagrama de proceso sistema de control visual

A. Descripción del Control del Dron

1. Detección de Plantas de Arándanos

Para ser capaces de posicionar correctamente el dron respecto a la planta, es necesario detectarla dentro de una imagen y determinar su ubicación, para lograrlo se utiliza el dataset construido con el fin de entrenar una red neuronal para detección de plantas de arándanos. Éste dataset cuenta con 2803 imágenes, siendo 1967 para entrenamiento, 676 para validación, y 160 para test. Dicho dataset se encuentra público y disponible en [11].

Teniendo en cuenta la información recolectada en el análisis del estado del arte sobre las técnicas usadas para este propósito, se tomó la decisión de probar modelos que han ofrecido buenos

resultados en contextos en donde se debe detectar en tiempo real.

2. TensorFlow - EfficientDet

Se realiza el entrenamiento de un modelo EfficientDet D2, que es una evolución de la arquitectura MobileNet perteneciendo ambos al framework Tensor Flow, en este caso fue entrenado con imágenes con un tamaño redimensionado a 768X768 px. El entrenamiento se realizó utilizando 5000 epochs, un tamaño de lote de 4. El resultado obtenido al calcular el Recall promedio de 0.79, y promedio de Precisión de 0.84 como se ve en Fig. 12. Ofreciendo un performance de 4 FPS ubicando con precisión las plantas de arándanos presentes en la imagen. Aunque en la práctica este modelo ofrece resultados aceptables que incluyen algunos falsos positivos que corresponden a otro tipo de planta presente en la imagen, su principal debilidad se encuentra al momento de iniciar la solución, ya que cargar dicho modelo a memoria para realizar inferencia requiere un tiempo de entre 1 y 2 minutos, lo que hace que la ejecución de la solución se vea entorpecida.

```

Accumulating evaluation results...
DONE (t=0.39s).
Average Precision (AP) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.777
Average Precision (AP) @[ IoU=0.50 | area= all | maxDets=100 ] = 0.986
Average Precision (AP) @[ IoU=0.75 | area= all | maxDets=100 ] = 0.888
Average Precision (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
Average Precision (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.570
Average Precision (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.794
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 1 ] = 0.306
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 10 ] = 0.824
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.829
Average Recall (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
Average Recall (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.681
Average Recall (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.845
INFO:tensorflow:Eval metrics at step 50000

```

Fig. 12. Métricas modelo EfficientDet

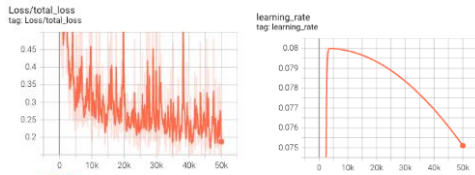


Fig. 13. Comportamiento EfficientDet - Error total y Tasa de aprendizaje

3. Ultralytics - Yolo

Se realiza el entrenamiento de un modelo Yolo de tamaño nano y en su versión 8, utilizando 100 epochs y un tamaño de lote de 16. El resultado obtenido al calcular un Recall de 0.97, y promedio de Precisión de 0.99, ofreciendo inferencia a una tasa de 19 FPS, aunque en la práctica la precisión obtenida no resulta tan buena como la ofrecida por el modelo EfficientDet, por lo que posteriormente se realizó un entrenamiento igualmente usando 100 epochs y tamaño de lote 16 de un modelo Small de la misma versión de Yolo, que reduce un poco el rendimiento de pero mejora la precisión de sus resultados, ejecutándose a una tasa de 10 FPS.

```

Model summary (fused): 168 layers, 3005843 parameters, 0 gradients, 8.1 GFLOPs
-----
Class      Images  Instances  Box(P)  R      mAP50  mAP50-95)
-----
all        559    1626      0.971   0.974  0.992  0.874

```

Fig. 14. Métricas modelo Yolo V8

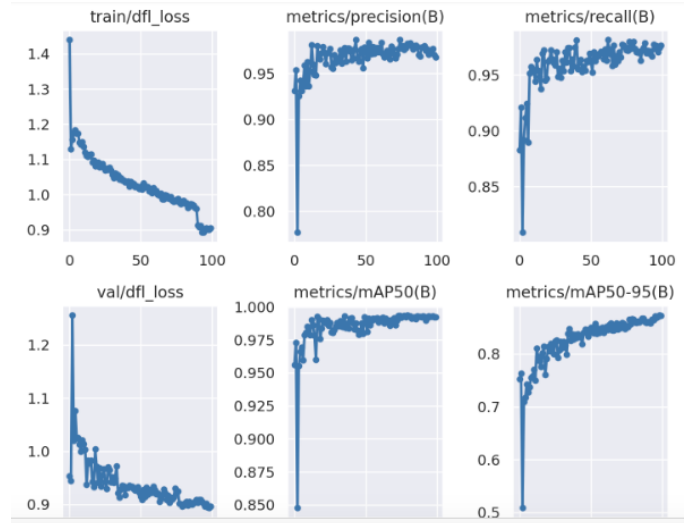


Fig. 15. Comportamiento Yolo V8 - Error total, Precisión, Recall

Se realizaron diferentes pruebas en el invernadero buscando determinar cuál de los modelos entrenados ofrece los mejores resultados en términos de precisión y performance, de tal forma que la cantidad de fotografías que se procesan por segundo sea suficiente para que el dron pueda realizar la rutina de inspección minimizando la posibilidad de errores. Obteniendo como resultado la decisión de utilizar el modelo Yolo ya que ofrece el mejor balance entre velocidad de precisión de inferencia.

4. Seguimiento de Plantas de Arándanos

Para lograr que el dron realice la rutina de inspección sobre una planta objetivo, es necesario individualizarla respecto a las otras plantas de arándanos presentes en la imagen, teniendo en cuenta que generalmente se encuentran más dos o más plantas en la misma imagen. Cuando el dron se encuentra en movimiento permanece transmitiendo fotografías que son analizadas para detectar las plantas de arándanos presentes en cada uno de ellos, además se realiza una detección del marcador ArUco para generar una asociación y así determinar cuál de las plantas detectadas tiene el marcador ubicado en su maceta lo que indica que esa planta es el objetivo de la inspección.

Teniendo en cuenta que el dron está en constante movimiento, cada fotografía es diferente al anterior por lo que la asociación de la planta con el marcador ArUco puede resultar compleja, además, en algunos pasos de la rutina, el marcador no estará presente en la imagen por lo que no podrá realizarse dicha asociación. Una de las referencias estudiadas en el estado del arte hace mención de la implementación de un algoritmo conocido como filtro de Kalman que permite predecir el estado de un objeto en el siguiente fotograma, además, se encontró un estudio en el que se propone un algoritmo conocido como SORT (Simple Online and Real Time Tracking) [12] que consiste en asociar las detecciones a lo largo de múltiples fotografías generando un rastreo consistente de cada objeto presente, utilizando el filtro Kalman para predecir la posición y velocidad de cada objeto en el siguiente fotograma asignando también un identificador único a cada objeto presente. Aunque este algoritmo tiene algunas limitaciones que se evidencian por ejemplo cuando existe un movimiento brusco por parte del dron, o el objetivo rastreado sale de la imagen, o es ocluido por

otro objeto, para dar solución a este problema se hace uso de la implementación de una evolución para este algoritmo [13] conocido como OC-SORT que permite mantener la referencia de los objetos incluso si éstos permanecen fuera de la imagen durante varios fotogramas.

B. Posicionamiento del Dron Respecto a la Planta

Con el fin de realizar una correcta inspección de las plantas de arándanos con el dron, todas las tareas relacionadas con el procesamiento de imágenes y la toma de decisiones se ejecutan de forma remota desde otro equipo. Esto es posible gracias a la facilidad de controlar el dron de manera remota a través del Software Development Kit (SDK) proporcionado por el fabricante utilizando Python. El SDK facilita el envío de comandos para controlar el vuelo del dron y la recepción de datos de sus sensores en tiempo real a través de una conexión UDP. En esta sección, se describe detalladamente el proceso de inspección de las plantas de arándanos mediante control servo-visual.

1. Posicionamiento del Dron Respecto al Marcador ArUco

La primera fase de la rutina de inspección consiste en detectar y posicionar el dron respecto al identificador de la planta a inspeccionar, para llevar a cabo esta tarea, se optó por implementar marcadores ArUco. Estos marcadores ArUco desempeñan un papel fundamental en la navegación del dron al proporcionar puntos de referencia visuales confiables de manera rápida y robusta, que ayudan en la estimación precisa de la posición del dron [10]. Además, dado que cada marcador tiene un identificador propio, es posible asignar un identificador a cada planta a inspeccionar. Esta última particularidad, simplifica la tarea de individualizar la planta sobre la cual se hará la rutina.

Una vez que se ha detectado con éxito el marcador ArUco, se inicia el proceso de posicionamiento del dron respecto al marcador. Para lograr esto, se utiliza un sistema compuesto por el mencionado detector de marcadores ArUco y un controlador basado en lógica difusa. Este controlador se encarga de ajustar las velocidades en los ejes X, Y y Z del dron, tomando como entradas la diferencia en píxeles entre la posición del centro del marcador ArUco y el centro de la imagen capturada por la cámara, así como la derivada de esta diferencia para controlar las velocidades en los ejes Y y Z.

Además, para el control del eje X, se utiliza como referencia el tamaño en píxeles del marcador. Dado que se conoce el tamaño real del marcador, esto proporciona una estimación precisa de la distancia entre el dron y el objeto de interés. Esta información es esencial para el proceso de posicionamiento y garantiza una buena estimación de la distancia entre el dron y el marcador ArUco.

2. Posicionamiento del Dron Respecto a la Planta Objetivo

Tan pronto como el dron está posicionado correctamente respecto al marcador ArUco, el siguiente paso consiste en

posicionar el dron respecto a la planta a inspeccionar usando el algoritmo de seguimiento de objetos descrito en el capítulo anterior. En esta fase, el dron se eleva 20 centímetros hacia arriba y luego se desplaza hacia la izquierda hasta que el recuadro correspondiente a la planta objetivo se encuentre a la derecha del todo de la imagen capturada. En esta posición se inicia el ciclo de captura de imágenes de la planta a inspeccionar desde distintas posiciones.

A. Ciclo de Captura y Evaluación de Imágenes desde diferentes Perspectivas

En esta fase de la rutina de inspección, de manera cíclica, mientras el dron se desplaza hacia la derecha a velocidad constante, se capturan imágenes de la planta a inspeccionar. Se extrae la región de interés (el recuadro de la planta a inspeccionar). Posteriormente, a esta región de interés se le asigna una etiqueta con el modelo de clasificación y una confianza, usando un sistema de inferencia basado en lógica difusa compuesto por los siguientes componentes:

1. Funciones de pertenencia de las variables de entrada

Este componente tiene como finalidad transformar los valores crisp de cada una de las variables de entrada, en el grado de pertenencia a los conjuntos difusos. Para la selección de las variables de entrada del sistema se realizó un análisis en un conjunto de imágenes de prueba que no fue usado para entrenar el clasificador con el fin de identificar qué variables propias de cada imagen podrían afectar la precisión del modelo de clasificación:

- **Certeza:** Una vez pasada la imagen por el modelo de clasificación, este, además de una etiqueta, asigna un valor de certeza el cual representa cuánto confía el modelo en su propia predicción. Dado que para esta aplicación se implementó un modelo de clasificación binaria, se tiene en cuenta únicamente la certeza asociada a la etiqueta asignada, la cual puede tomar valores en el rango de 0.5 a 1. En el histograma presentado en Fig. 1. se puede apreciar que a medida que aumenta la certeza, el porcentaje de acierto sobre el total de imágenes también aumenta. Por otra parte, en el histograma mostrado en Fig. 2. se presenta la distribución de los valores para la variable certeza en el conjunto de imágenes de prueba, en el que se puede apreciar que la mayoría de los valores para esta variable se concentran en los intervalos más cercanos a 1.

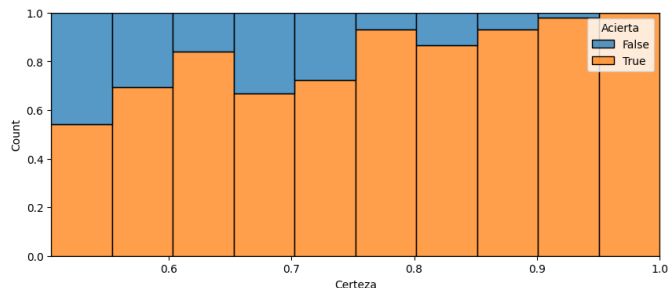


Fig. 1. Distribución de certeza respecto al porcentaje de acierto del modelo.

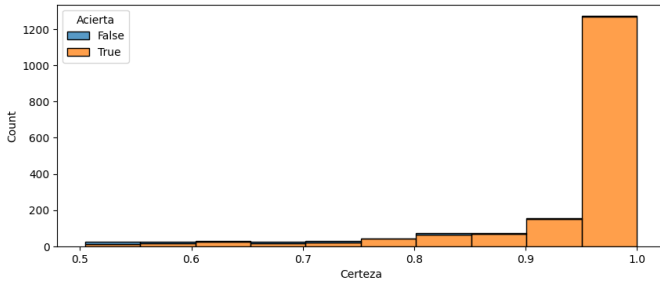


Fig. 2. Distribución de la variable certeza.

- Nitidez: Esta segunda variable evalúa la calidad visual de la imagen, específicamente la capacidad de percibir con claridad los bordes y detalles. En una imagen nítida, los contornos están bien definidos y los detalles son visibles, mientras que en una imagen menos nítida, estos pueden aparecer borrosos o con contornos difuminados. En el histograma presentado en Fig. 1. se destaca que a medida que aumenta la nitidez, también aumenta el porcentaje de acierto. Por otra parte, en el histograma mostrado en Fig. 2. se presenta la distribución de los valores para la variable nitidez en el conjunto de imágenes de prueba. En este caso, la mayoría de los valores se concentran en los intervalos más bajos.

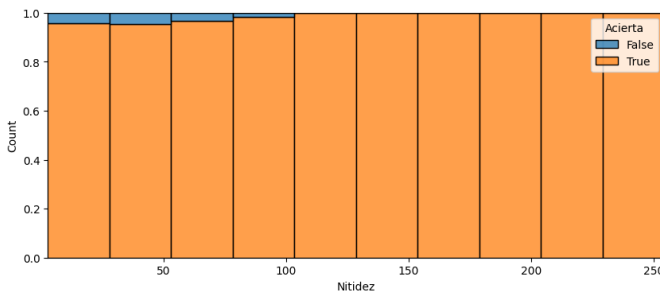


Fig. 1. Distribución de nitidez respecto al porcentaje de acierto del modelo.

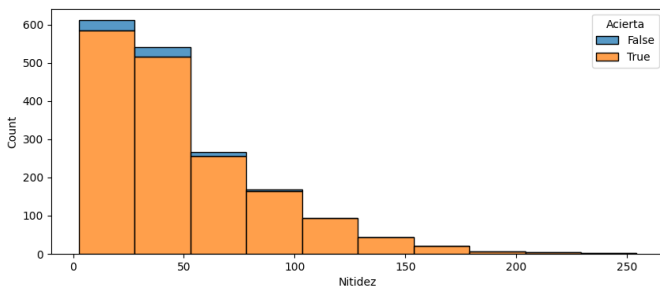


Fig. 2. Distribución de la variable nitidez.

- Brillo: Con esta última variable de entrada, se evalúa el nivel de luminosidad global presente en la imagen. Esta variable puede tomar valores entre 0 y 255. En el histograma presentado en Fig. 1. se muestra que a medida que aumenta el brillo, se reduce el porcentaje de acierto del modelo. Además, en el histograma mostrado en Fig. 2. se presenta la distribución de los valores para la variable brillo en el conjunto de imágenes de prueba, la cual, a diferencia

de las anteriores variables, no muestra sesgos significativos en su distribución.

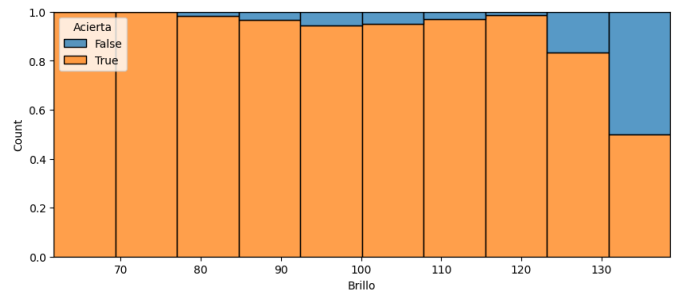


Fig. 1. Distribución de brillo respecto al porcentaje de acierto del modelo.

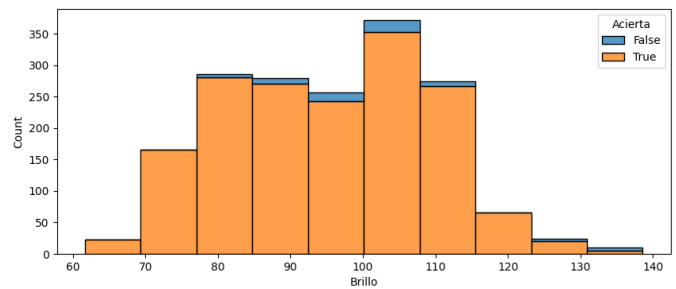


Fig. 2. Distribución de la variable brillo.

A partir del análisis de la distribución y comportamiento del porcentaje de acierto del modelo respecto a cada una de estas variables de entrada, se plantearon las funciones de pertenencia presentadas en las figuras 16, 17 y 18.

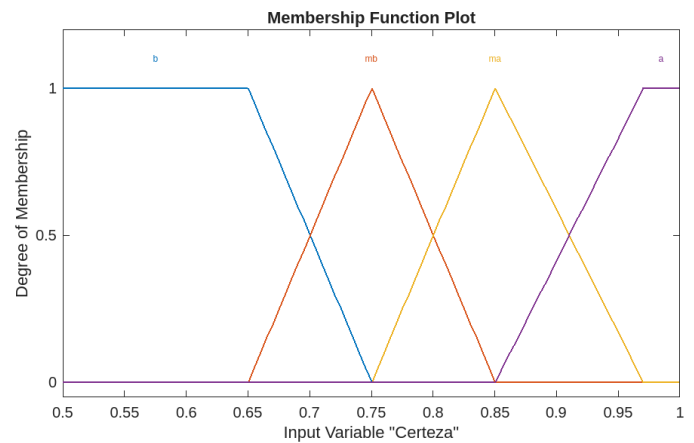


Fig. 16. Función de pertenencia de la variable certeza.

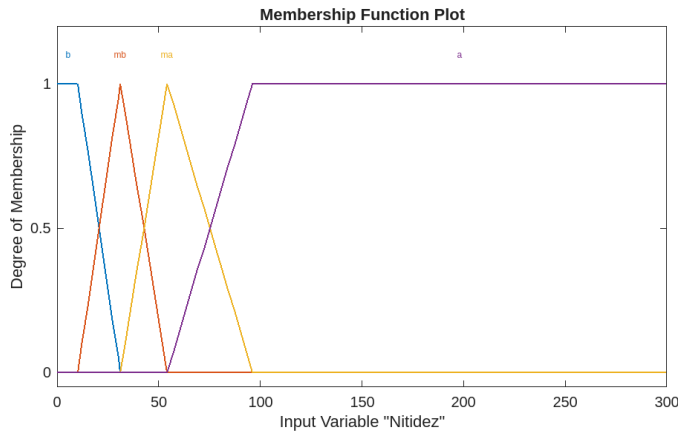


Fig. 17. Función de pertenencia de la variable nitidez.

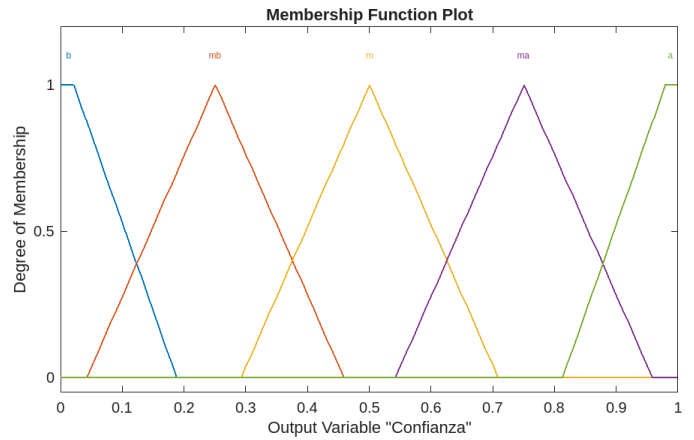


Fig. 19. Función de pertenencia de la variable de salida (confianza)

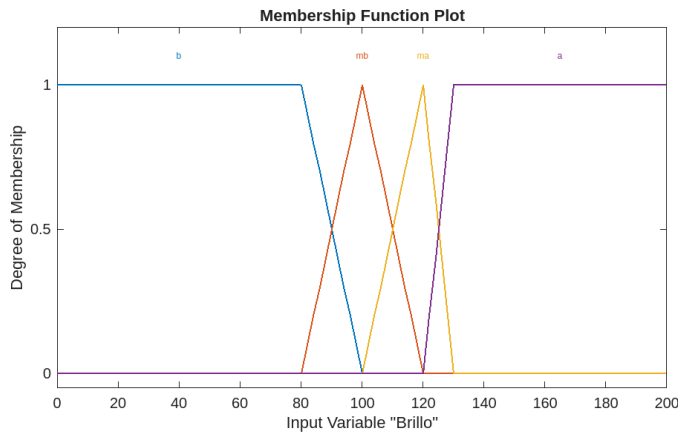


Fig. 18. Función de pertenencia de la variable brillo.

2. Reglas

Este componente establece la conexión entre las variables de entrada (certeza, nitidez y brillo) y la variable de salida del sistema (confianza). Para esta etapa se definió un total de 64 reglas teniendo en cuenta el porcentaje de acierto del clasificador en los conjuntos difusos.

3. Función de pertenencia de la variable de salida y sistema de defusificación

Con este último componente, se define el grado de pertenencia de la imagen a cada uno de los conjuntos difusos de salida para posteriormente transformarlo nuevamente en un valor crisp usando el sistema de defusificación que en este caso es una defusificación por centroide. En Fig. 19. Se muestra la función de pertenencia de la variable de salida.

Para verificar que el sistema de inferencia difuso asigna una mayor confianza a las imágenes con mayor probabilidad de acierto, se utilizó un conjunto de imágenes de prueba. En Fig. 20. Se puede evidenciar el correcto funcionamiento del sistema, ya que a medida que aumenta la confianza asignada, se reduce el porcentaje de fallo en la clasificación.

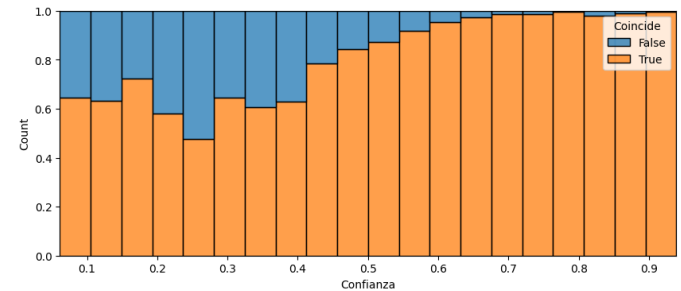


Fig. 20. Distribución de confianza respecto al porcentaje de acierto del modelo.

Continuando con el ciclo de captura de imágenes, una vez se ha asignado una confianza a la imagen, si esta es menor a un determinado umbral, el dron se acercará a la planta para tomar una imagen de cerca. A esta nueva imagen se le aplicará el mismo proceso de recorte, clasificación y asignación de confianza. En caso de que esta imagen cercana tenga una confianza superior, reemplazará a la imagen original. Este ciclo se repite para cada siguiente posición hasta que el recuadro correspondiente a la planta objetivo se encuentre a la izquierda del todo de la imagen capturada.

Al llegar a esta posición, el dron se eleva 5 centímetros más para capturar un segundo grupo de imágenes de la planta desde esta nueva altura. Finalmente, después de obtener las imágenes en la segunda altura, concluye el ciclo de captura de imágenes para pasar a la siguiente fase, en la que se genera un reporte final del estado de la planta.

B. Generación de Reporte del Estado de la Planta

Para esta última etapa, se genera un reporte sobre el estado de la planta mediante un sistema de votación ponderado. En este

sistema, el voto de cada una de las imágenes capturadas es la etiqueta que le fue asignada por el modelo de clasificación, y su peso es la confianza que le fue asignada por el sistema de inferencia difusa. El resultado de esta votación se resume en un reporte como el presentado en Fig. 21. en el que se le asigna una predicción a la planta en su totalidad, seguida por predicciones por altura y finalmente por posición.

```

Predicción Total: sana 0.96
-----
Reporte por Altura:
  Altura: 1: sana 0.96
  Altura: 2: sana 0.96
-----
Reporte por Posición:
  Altura: 1
    izquierda: sana 1.0
    centro: sana 0.84
    derecha: sana 1.0
  Altura: 2
    izquierda: sana 0.90
    centro: sana 0.96
    derecha: sana 1.0

```

Fig. 21. Ejemplo de reporte final de la planta

II. Resultados del Sistema Integrado

En este último apartado, enfocado en el análisis de resultados, se realiza un protocolo experimental en el que se busca determinar el efecto del valor de umbral definido en el capítulo anterior sobre el rendimiento del sistema integrado, el cual es calculado teniendo en cuenta el tiempo que le toma ejecutar la rutina y la precisión en la clasificación. Además, para esta experimentación se emplea un conjunto de datos compuesto por imágenes capturadas de acuerdo con la rutina de inspección en seis plantas diferentes (dos enfermas y cuatro sanas).

Como se describió anteriormente, Para llevar a cabo este experimento, se consideraron las siguientes variables independientes:

- Umbral: Representa el valor con el cual se compara la confianza asignada por el sistema de inferencia difusa. Los valores utilizados en la experimentación para esta variable fueron 0, 0.2, 0.4, 0.6, 0.8 y 1.
- Alfa (α): Este valor es utilizado para definir el peso de los dos factores que se tuvieron en cuenta para calcular el rendimiento del sistema (Precisión y tiempo de ejecución) utilizando la siguiente ecuación:

$$\text{Rendimiento} = \text{Precisión} \alpha + \text{TiempoDeEjecución} (1 - \alpha)$$

Es importante destacar que se realizó previamente una normalización Min-Max para ambos factores con el

objetivo de evitar que el rango de valores afecte su ponderación. El factor precisión toma valores en el rango de 0.81 a 1, mientras que el factor tiempo de ejecución, medido en segundos, se encuentra en el rango de 31.4152 a 467.8176.

Los valores utilizados en experimentación para esta variable fueron 0.3, 0.5 y 0.7 para comparar el rendimiento del sistema en tres escenarios: dándole un mayor peso al tiempo de ejecución, dándole el mismo peso a ambos factores y dándole un peso mayor a la precisión.

Además, como se comentó, la respuesta observada en este protocolo experimental fue el rendimiento del sistema integrado. Se implementó un análisis ANOVA para estudiar el efecto de las variables independientes en la respuesta observada.

A. Análisis gráfico.

En primer lugar, se generaron diagramas de caja con el fin de identificar posibles diferencias notables en el comportamiento del rendimiento según las variables de entrada. A partir de la representación gráfica presentada en Fig. 22. se puede intuir que existe una diferencia en el rendimiento alcanzado según el umbral. Por otra parte, para el caso de la variable alfa, no se encuentra una diferencia significativa entre los distintos valores seleccionados.

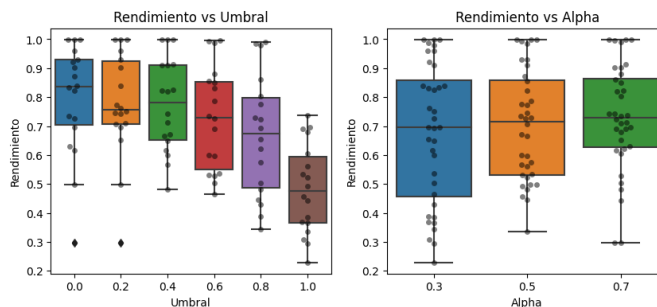


Fig. 22. Diagrama de caja del rendimiento en función de las variables independientes.

Además, al realizar un análisis de la interacción entre las variables de entrada, como se muestra en Fig. 23. se aprecia que para el valor de alfa en el que se le da una mayor prioridad a la precisión del modelo, representado por la línea azul, la media del valor de rendimiento más alta se obtiene con un valor de umbral de 0.6. Mientras que para los otros dos valores de alfa, el mayor rendimiento es obtenido para un valor de umbral de 0.0.

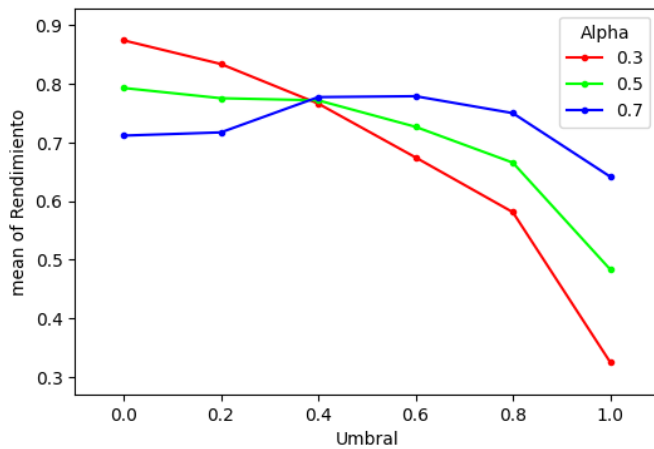


Fig. 23. Diagrama de interacción de las variables independientes.

B. Análisis ANOVA.

Por último, se puede apreciar los resultados obtenidos en el análisis de varianza sobre los datos en la Tabla I. A partir de estos resultados, se puede concluir que, para la variable Umbral, la presencia de un valor de p (p -unc) significativamente pequeño y un tamaño de efecto (np^2) grande indican la existencia de diferencias estadísticamente significativas entre al menos dos de los niveles. En contraste, no se observan diferencias significativas en los niveles de Alfa ni una interacción significativa entre Umbral y Alfa.

TABLA I
RESULTADOS DEL ANÁLISIS DE ANOVA

Fuente	SS	DF	MS	F	p -unc	Np^2
Umbral	1.2303	5	0.941	7.9675	0.000	0.3068
Alfa	0.518	2	0.979	0.8392	0.4354	0.0183
Umbral* Alfa	0.4883	10	0.937	1.5813	0.1250	0.1494

VII. CONCLUSIÓN

En el presente trabajo, se planteó una rutina de inspección autónoma basada en control servo visual para un Vehículo Aéreo No Tripulado (UAV) con el objetivo de realizar la detección de síntomas de estrés en plantas de arándanos en su etapa de crecimiento.

Para lograr esto, en una primera etapa se implementó un modelo de detección de objetos Yolo Small en su versión 8, junto con un algoritmo de rastreo de objetos denominado OC-SORT que permite posicionar correctamente del dron respecto a la planta objetivo y extraer la región de interés de las imágenes obtenidas por el dron (el recuadro de la planta a inspeccionar), este sistema probó funcionar de manera robusta en las condiciones para las que fue ajustado, es decir, dentro del invernadero en el que se realizaron las pruebas y a bajas velocidades.

Por otra parte, para la correcta clasificación de la región de interés extraída y determinar si pertenece a una planta sana o

enferma, fue implementado un modelo de clasificación usando la arquitectura Yolo v8, que fue escogido como la mejor opción de las posibilidades exploradas ya que mostró un mejor desempeño en términos de velocidad de procesamiento y métricas de evaluación de precisión.

Además, se diseñó un sistema de inferencia basado en lógica difusa encargado de asignar un valor de confianza a las imágenes clasificadas, el cual es posteriormente usado en un sistema de votación ponderado para generar un reporte final del estado de la planta, este último sistema compuesto permite generar un diagnóstico integral del estado de la planta a partir de las imágenes obtenidas desde distintas perspectivas, dándole más peso a las imágenes con mayor confianza.

En el presente trabajo se explora el camino para implementar una rutina de autónoma que permita la inspección activa de cultivos, este caso, se realiza la inspección con el fin de detectar signos de estrés plantas de arándanos, en la que se destaca el diseño un control servo visual basado en lógica difusa que con ayuda de un marcador ArUco, es capaz de posicionar y mover al dron respecto a la planta de interés.

Se exploraron alternativas de inspección utilizando diferentes estrategias de acercamiento, lo que permitió concluir que se puede usar una estrategia sin aproximaciones a la planta obteniendo resultados aceptables, si el objetivo es optimizar energía, o se podría optar por una estrategia de acercamiento exhaustivo que ofrece resultados con mayor confianza, pero utilizando una mayor cantidad de energía y tiempo de vuelo. Por otro lado, se destaca el enfoque de etiquetado por imagen y no por planta, en el que imágenes correspondientes a una planta pueden tener diferente etiqueta, esto debido a que los signos de estrés o enfermedad podrían ser evidentes desde algunas perspectivas y desde otras posiciones podría verse como una planta sana. Este enfoque que es complementado con un algoritmo de inferencia basado en lógica difusa que toma como variables la distancia respecto a la planta, el brillo y la nitidez de la imagen, y el valor de confianza entregado por el modelo clasificador permite obtener un veredicto riguroso sobre el estado de la planta.

Este algoritmo de inspección se propone como el primer paso para una solución en la que se realice una inspección autónoma de todo el cultivo y así reducir considerablemente el tiempo y esfuerzo necesario para mantener una política de monitoreo activo. Este algoritmo requiere únicamente de una cámara RGB como sensor y un marcador ArUco ubicado en la maceta de cada planta, dejando como trabajo futuro resolver la navegación autónoma del dron dentro del invernadero, lo que podría lograrse con la ayuda de un sensor por ejemplo basado en radar, o por ejemplo utilizando posicionamiento GPS si la plantación es lo suficientemente grande.

REFERENCES

1. S. Kumar, R. Kumar and Saad, "Real-Time Detection of Road-Based Objects using SSD MobileNet-v2 FPNlite with a new Benchmark Dataset," 2023 4th International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 2023, pp. 1-5, doi: 10.1109/iCoMET57998.2023.10099364.
2. Sabina N, Aneesa M. P, Haseena P. V, 2022, Object Detection using YOLO And Mobilenet SSD: A Comparative Study, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 11, Issue 06 (June 2022),
3. K. Máthé, L. Buşoniu, L. Barabás, C. -I. Iuga, L. Miclea and J. Braband, "Vision-based control of a quadrotor for an object inspection scenario," 2016 International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA, USA, 2016, pp. 849-857, doi: 10.1109/ICUAS.2016.7502522.
4. M. Al Radi, H. Karki, N. Werghe, S. Javed and J. Dias, "Vision-based Inspection of Flare Stacks Operation Using a Visual Servoing Controlled Autonomous Unmanned Aerial Vehicle (UAV)," IECON 2022 – 48th Annual Conference of the IEEE Industrial Electronics Society, Brussels, Belgium, 2022, pp. 1-6, doi: 10.1109/IECON49645.2022.9968467.
5. M. A. Olivares-Méndez, P. Campoy, C. Martínez and I. Mondragón, "A pan-tilt camera Fuzzy vision controller on an unmanned aerial vehicle," 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 2009, pp. 2879-2884, doi: 10.1109/IROS.2009.5354576.
6. Biblioteca Digital INIA - Biblioteca Digital INIA. Accedido el 15 de noviembre de 2023. [En línea]. Disponible: <https://biblioteca.inia.cl/bitstream/handle/20.500.14001/40752/Articulo?sequence=1>
7. Neupane, K., & Baysal-Gurel, F. (2021). Automatic Identification and Monitoring of Plant Diseases Using Unmanned Aerial Vehicles: A Review. Remote Sensing, 13(19), 3841. <https://doi.org/10.3390/rs13193841>
8. Undurraga, P., y Vargas, S. (eds.) 2013. Manual del arándano. Boletín INIA N° 263. 120 p. Instituto de Investigaciones Agropecuarias INIA, Centro Regional de Investigación Quilamapu, Chillán, Chile.
9. D. Fuentes Escoté. "DJITelloPy." GitHub. Accessed: Aug. 10, 2023. [Online]. Available: <https://github.com/damiafuentes/DJITelloPy#djitelopy>
10. Automatic generation and detection of highly reliable fiducial markers under occlusion, Pattern Recognition, Volume 47, Issue 6, 2014, Pages 2280-2292, ISSN 0031-3203, <https://doi.org/10.1016/j.patcog.2014.01.005>
11. D. Martínez, L. Martínez, and A. Gómez. "PlantaArandanos Object Detection Dataset and Pre-Trained Model by Monitoreoarandanos." Roboflow. Accessed: Nov. 8, 2023. [Online]. Available: <https://universe.roboflow.com/monitoreoarandanos-3mem1/plantaarandanos>
12. A. Bewley, Z. Ge, L. Ott, F. Ramos and B. Upcroft, "Simple online and realtime tracking," 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 2016, pp. 3464-3468, doi: 10.1109/ICIP.2016.7533003.
13. Cao, J. et al. (2023) 'Observation-centric sort: Rethinking sort for robust multi-object tracking', 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) [Preprint]. doi:10.1109/cvpr52729.2023.00934.
14. S. Babaheni, F. Charif and A. Taleb-Ahmed, "Study of the influence of RGB and Lab color spaces on the performance of multifocus image fusion techniques," 2022 4th International Conference on Pattern Analysis and Intelligent Systems (PAIS), Oum El Bouaghi, Algeria, 2022, pp. 1-6, doi: 10.1109/PAIS56586.2022.9946904.
15. "“La raíz es el cerebro de la planta” (Darwin). Manejos y productos para raíces". Redagícola. Accedido el 18 de noviembre de 2023. [En línea]. Disponible: <https://redagricola.com/la-raiz-es-el-cerebro-de-la-planta/#:~:text=El%20crecimiento%20radicular%20ocurre%20por,la%20punta%20de%20las%20raíces.>
16. Proplantas SA. Proplantas S.A. - Finca El Lago - Cultivo de Arándanos. (7 de noviembre de 2018). Accedido el 17 de noviembre de 2023. [Video en línea]. Disponible: <https://www.youtube.com/watch?v=JzZLiaPqQVA>
17. "Respuesta agronómica de arándano (*Vaccinium corymbosum* L.) al estrés osmótico", scielo, vol. 52, n.º 2, 2018, art. n.º 52: 231-239. 2018.
18. N. M. Rubio. "Redes neuronales profundas: qué son y cómo funcionan". Psicología y Mente. Accedido el 20 de noviembre de 2023. [En línea]. Disponible: <https://psicologiymente.com/cultura/redes-neuronales-profundas>
19. "¿Qué es una red neuronal? - Explicación de las redes neuronales artificiales - AWS". Amazon Web Services, Inc. Accedido el 20 de noviembre de 2023. [En línea]. Disponible: <https://aws.amazon.com/es/what-is/neural-network/#:~:text=Las%20redes%20neuronales%20profundas,%20o,entr e%20un%20nodo%20y%20otro>
20. "Redes neuronales y aprendizaje profundo: diferencia entre campos de la inteligencia artificial - AWS". Amazon Web Services, Inc. Accedido el 20 de noviembre de 2023. [En línea]. Disponible: [https://aws.amazon.com/es/compare/the-difference-between-deep-learning-and-neural-networks/#:~:text=Hay%20dos%20tipos%20principales%20de,redes%20neuronales%20recurrentes%20\(RNN\)](https://aws.amazon.com/es/compare/the-difference-between-deep-learning-and-neural-networks/#:~:text=Hay%20dos%20tipos%20principales%20de,redes%20neuronales%20recurrentes%20(RNN))
21. I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions", SN Comput. Sci., vol. 2, n.º 3, marzo de 2021. Accedido el 20 de noviembre de 2023. [En línea]. Disponible: <https://doi.org/10.1007/s42979-021-00592-x>.
22. "Redes neuronales profundas - Tipos y Características - Código Fuente". Código Fuente. Accedido el 20 de noviembre de 2023. [En línea]. Disponible: <https://www.codigofuente.org/redes-neuronales-profundas-tipos-caracteristicas/>
23. Bayraktar, E., Basarkan, M. E., & Celebi, N. (2020). A low-cost UAV framework towards ornamental plant detection and counting in the wild. ISPRS Journal of Photogrammetry and Remote Sensing, 167, 1–11. <https://doi.org/10.1016/j.isprsjprs.2020.06.012>