UNIVERSIDAD TÉCNICA DEL NORTE

Facultad de Ingeniería en Ciencias Aplicadas

Carrera de Software

DESARROLLO DE ALGORITMOS DE PROCESAMIENTO DE IMÁGENES AGRÍCOLAS OBTENIDAS POR DRONES PARA LA DETECCIÓN DE PLANTAS FALTANTES EN CULTIVOS DE MAÍZ.

Trabajo de grado previo a la obtención del título de Ingeniero en Software

Autor:

Ronald David Moreira Ramos

Directo:

Ing. Marco Pusdá MSc.

Ibarra - Ecuador 2022

UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presentetrabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	120528852-3		
APELLIDOS Y NOMBRES:	RONALD DAVID MOREIRA RAMOS		
DIRECCIÓN:	VALENCIA – LOS RÍOS, INTERSECCIÓN ENTRE LA CALLE SIMÓN BOLÍVAR Y JOSÉ MARÍA SALGUERO.		
EMAIL:	rmoreirar@utn.edu.ec bguj.moreira.ramos.ronald@gmail.com		
TELÉFONO FIJO:	052 948 938	**TELÉFONOMÓVIL:**	0959811175

DATOS DE LA OBRA	
TÍTULO:	"DESARROLLO DE ALGORITMOS DE PROCESAMIENTO DE IMÁGENES AGRÍCOLAS OBTENIDAS POR DRONES PARA LA DETECCIÓN DE PLANTAS FALTANTES EN CULTIVOS DE MAÍZ"
AUTOR:	RONALD DAVID MOREIRA RAMOS
FECHA:	01/11/2022
PROGRAMA:	■ PREGRADO ☐ POSGRADO

TÍTULO POR EL QUE OPTA:	INGENIERO EN SOFTWARE
ASESOR /DIRECTOR:	ING. MSC. MARCO PUSDÁ

2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, al 1 día del mes de noviembre del 2022

EL AUTOR:

Nombre: Ronald David Moreira Ramos

Cédula: 120528852-3

CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE GRADO

UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN DEL ASESOR

Certifico que la Tesis previa a la obtención del título de Ingeniero en Software con el tema: ""DESARROLLO DE ALGORITMOS DE PROCESAMIENTO DE IMÁGENES AGRÍCOLAS OBTENIDAS POR DRONES PARA LA DETECCIÓN DE PLANTAS FALTANTES EN CULTIVOS DE MAÍZ"" ha sido desarrollada y terminada en su totalidad por el Sr. Ronald David Moreira Ramos, con cédula de identidad Nro. 120528852-3 bajo mi supervisión para lo cual firmo en constancia.

0401200951
MARCO REMIGIO
PUSDA CHULDE

Firmado digitalmente por 0401200951 MARCO REMIGIO PUSDA CHULDE
Fecha: 2022.11.01 16:01:25 -05'00'

Msc. Marco Pusdá

DIRECTOR DE TESIS

DEDICATORIA

Dedico el presente trabajo a mi madre Lilian Ramos, mi padre Johnny Moreira y a mi hermano Bryan Moreira, los cual me impulsaron a seguir adelante y me acompañaron durante los momentos buenos y malos. Sobre todo, a mis padres quienes me enseñaron a no darse por vencido y a perseverar durante cualquier circunstancia, ya que gracias a ellos soy la persona de ahora.

También quiero dedicarles este trabajo a mis tíos y abuelos quienes también me apoyaron y quiero que sepan que todo esto es posible gracias a su apoyo incondicional.

AGRADECIMIENTO

Con la culminación de este trabajo de grado solo me queda decir una palabra: ¡Gracias!

Todo el trabajo realizado solo fue posible con el apoyo de mi familia, mis padres y mi hermano que siempre estuvieron al pendiente de mis estudios y me ayudaron a seguir adelante, ya que nada de esto pudiera haber sido posible sin ustedes.

También quiero agradecer a los maestros de la Universidad Técnica del Norte por la educación que me han proporcionado durante estos años, sobre todo a mi tutor MSc. Marco Pusdá por el apoyo y seguimiento de mi trabajo de titulación para poder culminarlo.

Adicionalmente agradezco a mis opositores el MSc. Iván García y al MSc. Cosme Ortega por su tiempo que le dedicaron a mi trabajo de grado para sacar las observaciones correspondientes, debido a que esto permitió mejorar el trabajo que se presente en este documento.

Gracias a todos ustedes y por supuesto a Dios, por todo su apoyo.

CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

La investigación realizada en el presenta artículo se basa en la detección de plantas faltantes dentro de los cultivos de maíz, debido a que muchos agricultores gastan gran cantidad de recursos y tiempo para poder realizar el monitoreo por medios tradicionales. En muchos países extranjeros ya han comenzado por utilizar la tecnología dentro del campo de la agricultura como, por ejemplo, el uso de los vehículos aéreos no tripulados (UAV) para la aplicación dentro de la agricultura de precisión. El objetivo de la investigación fue el desarrollo de algoritmos utilizando la técnica de visión por computador para detectar estas plantas faltantes, para lo cual se utilizó imágenes capturadas por drones con distancias de 5, 10 y 15 metros de altura con respecto al suelo. Para el desarrollo de los algoritmos se utilizó Matlab con la versión 2017 b. Se aplicó la metodología de cascada con la finalidad de cumplir con todos los entregables durante cada una de las fases. Además, se empleó uno de los capítulos de la guía de Swebok para validar los resultados. Los resultados obtenidos se hicieron en base a la precisión, sensibilidad y eficiencia de los algoritmos. Durante varias pruebas se determinó que el algoritmo óptimo para la detección de plantas faltantes dentro de los cultivos de maíz es el de 10 metros de altura.

Palabras clave: Agricultura de Precisión, Matlab, Visión por Computador, Vehículos Aéreos No Tripulados.

ABSTRACT

The research conducted in this article is based on the detection of missing plants within corn crops, because many farmers spend a lot of resources and time to perform monitoring by traditional means. Many foreign countries have already started to use technology in the field of agriculture, such as the use of unmanned aerial vehicles (UAVs) for precision agriculture applications. The objective of the research was to develop algorithms using the computer vision technique to detect these missing plants, using images captured by drones at distances of 5, 10 and 15 meters above the ground. For the development of the algorithms, Matlab version 2017 b was used. The cascade methodology was applied in order to comply with all the deliverables during each of the phases. In addition, one of the chapters of the Swebok guide was used to validate the results. The results obtained were based on the accuracy, sensitivity and efficiency of the algorithms. During several tests, it was determined that the optimal algorithm for the detection of missing plants within the corn crop is the 10-meter height algorithm.

Keywords: Precision Agriculture, Matlab, Computer Vision, Unmanned Aerial Vehicles.

INTRODUCCIÓN

Antecedentes

La agricultura tradicional se encuentra vulnerable a diversos fenómenos, ya sean climáticos como la corriente del niño, enfermedades, plagas, etc. Esto ha llevado a que se implementen nuevas tecnologías como alternativas para el monitoreo y control de los cultivos agrícolas durante sus diferentes etapas de desarrollo. (Meneses et al., 2015).

Existen varios aspectos negativos que los agricultores se enfrentan a diario, como la dificultad de recorrer todo el terreno, el cual se puede dar por la gran extensión del terreno o por las condiciones del mismo. Otra dificultad importante por resaltar en la agricultura son las platas faltantes en los cultivos, para lo cual se necesita recorrer todas las líneas de cultivo para identificar estos faltantes.

Todo lo mencionado anteriormente consume muchos recursos, tanto económico, recurso humano y tiempo. Adicionalmente, si no se identifican las plantas faltantes en los cultivos a tiempo, el agricultor no tiene la oportunidad de resembrar nuevas plantas, esto provoca que existan pérdidas económicas para los agricultores e incluso puede existir la posibilidad de que las pérdidas superen a las ganancias por la cosecha.

Situación Actual

El campo de la agricultura es fundamental para la supervivencia de los seres humanos, fundamentalmente ahora que la población mundial supera los 7 mil millones de personas por lo cual necesita proporcionar un aumento en la salida de productos con respecto al campo de la agricultura. Según Pederi & Cheporniuk (2015) se estima que la protección de los cultivos representa un 50% del rendimiento de la agricultura, lo que nos lleva a pensar que se necesita una gran cantidad de recursos como personal, dinero y tiempo. Además, se debe tener en cuenta que existen otros problemas como la dificultad que existe para recorrer el terreno debido a la ubicación de este, lo que conlleva a que no se pueda establecer un control continuo sobre los cultivos. Todos los problemas mencionados anteriormente afectan la eficiencia y el rendimiento de los cultivos, a la vez que se incrementa el tiempo para evaluar los cultivos ocupando métodos tradicionales.

Actualmente los cultivos de maíz se enfrentan a varios problemas como el factor climático, el cual afecta a la productividad del maíz, lo que provoca pérdidas para los agricultores. Según Baca (2016) el calentamiento global tiene un impacto en los cultivos el cual puede ser positivo o negativo dependiendo de la zona geográfica del cultivo por las elevadas de temperaturas. Estos problemas provocan pérdidas significativas. Sin embargo, en la actualidad muchos de los agricultores de países extranjeros comienzan a utilizar la tecnología para solventar estos problemas.

Por otra parte, el Ecuador ya está implementado la tecnología en la agricultura. Un caso de implementación es el que se utilizó en el "Programa de Cooperación Internacional Brasil-FAO" en donde la Universidad Técnica de Manabí (UTM) implementó vehículos aéreos no tripulados (drones) para la producción algodonera. (FAO, 2020)

Planteamiento del problema

La detección tardía o el tiempo excesivo que los agricultores emplean para identificar las plantas de los cultivos de maíz significa pérdida de recursos económicos, humano y tiempo. Todo esto afecta a la productividad del cultivo, debido a que el agricultor no puede monitorear constantemente los cultivos, lo cual no permite tomar las decisiones necesarias en el momento adecuado como la resiembra de las plantas faltantes en los cultivos de maíz. En la Fig. 1 se muestra el árbol de problemas.

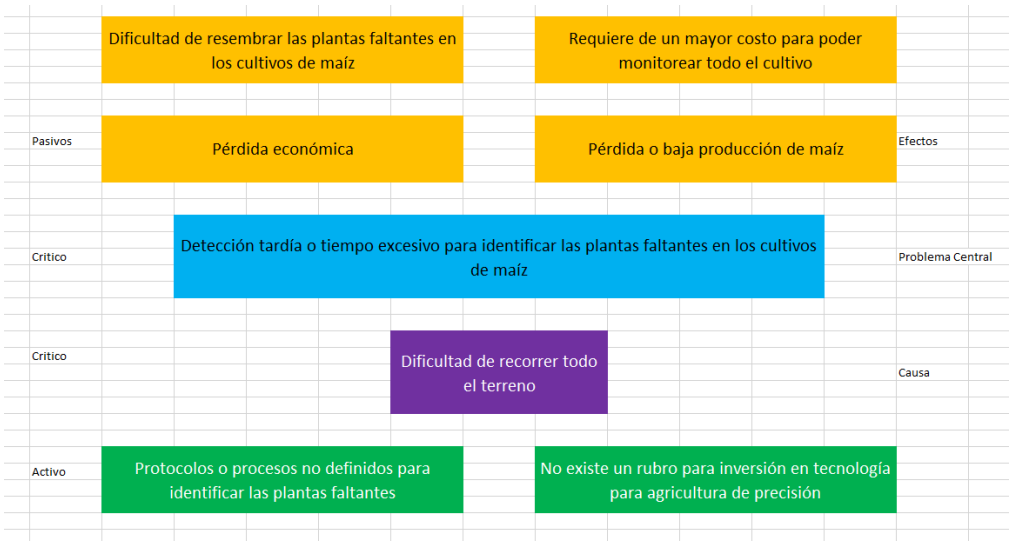


Fig. 1. Árbol de problemas.
Fuente: Propia

Objetivos

Objetivo General

Desarrollar algoritmos de procesamiento de imágenes agrícolas obtenidas por drones para la detección de plantas faltantes en cultivos de maíz.

Objetivos Específicos

- Documentar la situación actual de algoritmos y aplicaciones de procesamientos de imágenes agrícolas utilizando vehículos aéreos no tripulados (UAV) por sus siglas en inglés Unmanned Aerial Vehicle.
- Desarrollar el algoritmo para la identificación de plantas faltantes en los cultivos de maíz mediante imágenes adquiridas por UAV.
- Validar los resultados obtenidos en el presente proyecto mediante fundamentos de ingeniería perteneciente a la guía Swebok.

Alcance y metodología

Alcance

El alcance del presente proyecto consistió en realizar una aplicación de escritorio con interfaz gráfica encargada de recibir las imágenes obtenidas con UAV. El procesamiento de las imágenes no es en tiempo real (off-line), razón por la cual las imágenes capturadas por el UAV se almacenaron en la PC para realizar el análisis. Posteriormente se procedió a realizar el algoritmo correspondiente en Matlab para la detección de las plantas faltantes en los cultivos de maíz. Sin embargo, para futuras investigaciones, se pueden utilizar otros lenguajes de programación como Python. Una vez realizado el procesamiento de las imágenes se emplearon varias pruebas con distintas capturas de imágenes realizadas por UAV. Cada captura que se realizó tuvo una distancia diferente con respecto al suelo. Por ejemplo: La primera captura a 5 m de distancia, la segunda a 10 m, la tercera a 20 m, etc. Esto se realizó con la finalidad de verificar y validar cuáles son las mejores imágenes y posteriormente

determinar la distancia óptima del UAV con respecto al suelo para realizar la captura de la imagen en los cultivos de maíz.

El formato de las imágenes con el que se trabajó es (.jpg) y la resolución de las imágenes empleadas en el proyecto fueron de mínimo FHD de 1920*1080. El ángulo del UAV con respecto al suelo es de 90 grados, es decir se va a trabajó con imágenes cenitales, las cuáles se definen como paralelas al suelo. Adicionalmente se trabajó con una banda de espectro electromagnético visible.

Las imágenes capturadas durante el proyecto se centraron en los cultivos de maíz de la zona 1 del Ecuador (Esmeraldas, Carchi, Imbabura y Sucumbíos). En la Fig. 2 se muestra un gráfico acerca del alcance del proyecto.

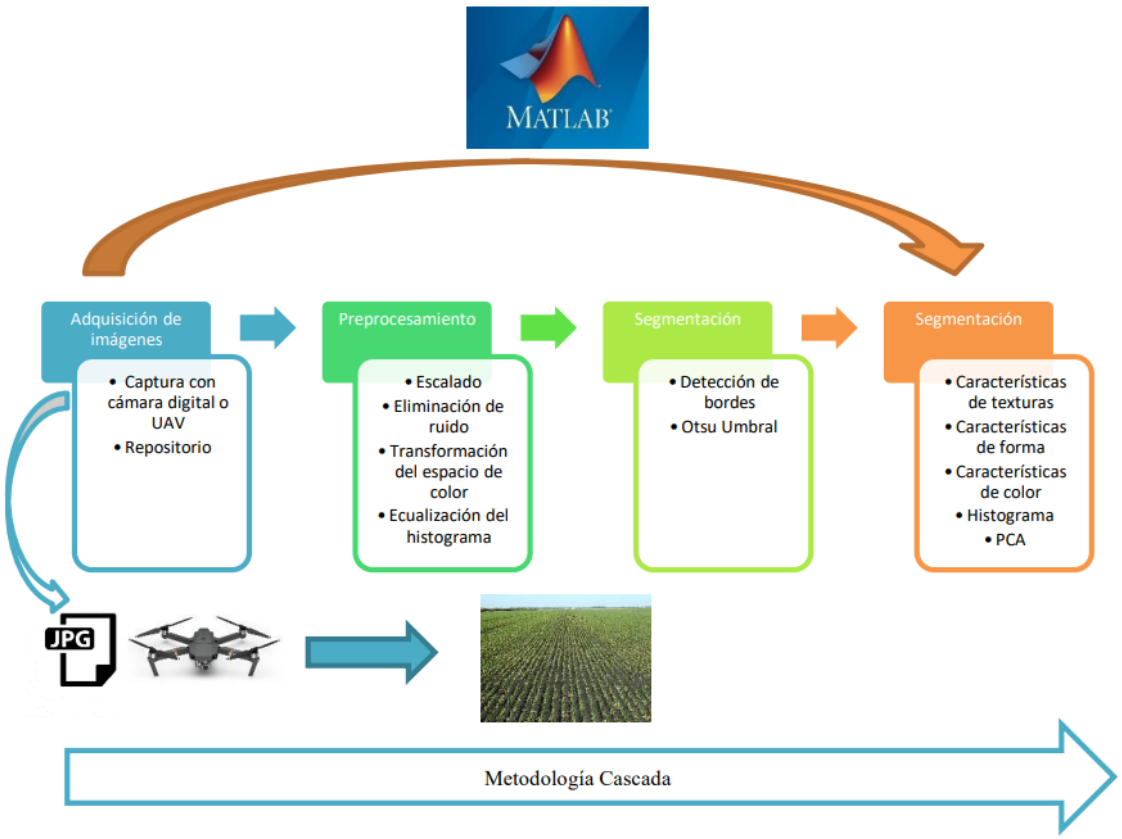


Fig. 2. Alcance del proyecto.
Fuente: Propia

Metodología

Para el desarrollo del software se empleó una metodología tradicional, específicamente la metodología de cascada debido a la sencillez de la arquitectura del proyecto al tratarse de programas unitarios. La metodología a utilizar en el proyecto se presenta en la Fig. 3.

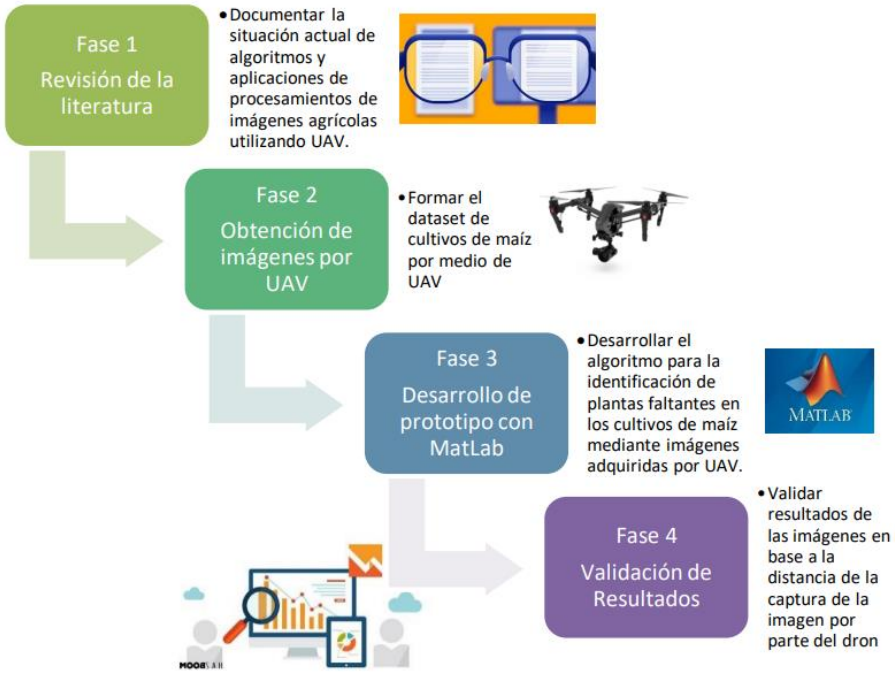


Fig. 3. Metodología.
Fuente: Propia

Justificación y Riesgos

Justificación

Actualmente, el uso de UAV en la agricultura se ha vuelto una tendencia debido a los múltiples usos que se les puede dar dentro de esta área. Por ejemplo, los UAV son empleados como medios para la fumigación de los cultivos, monitoreo y control, detección de enfermedades agrícolas, etc. Sin embargo, al ser una nueva tendencia tecnológica existen muchos sectores donde no se ha implementado. De igual manera, aún existen muchas aplicaciones en la que se puede implementar los UAV.

La presenta investigación se enfocó en detectar los faltantes dentro de los cultivos de maíz a una edad temprana por medio de la utilización de UAV debido a que actualmente los métodos tradicionales que se emplean en la agricultura consumen muchos recursos (humano, económico) y en especial el tiempo.

Además, permitió apoyar a una mejora en la productividad de los cultivos de maíz debido a que el agricultor por medio del UAV pudo identificar las plantas faltantes, lo que da la oportunidad de resembrar los cultivos de manera inmediata. De esta manera se disminuyen las pérdidas de rendimiento del cultivo al momento de cosechar el maíz.

Esta investigación apoya al (Objetivo de Desarrollo Sostenible) ODS 2: Hambre cero, ya que trata de mejorar la productividad de los cultivos de maíz. A la vez esto permitirá proveer de más alimento a la humanidad dándole la oportunidad de que todas las personas accedan a una buena alimentación tanto para el productor como para el consumidor apoyando a una de las metas de este objetivo.

Justificación Tecnológica

Esta investigación buscó proponer una alternativa ante la problemática del consumo de recurso humano, económico y tiempo por la implementación de métodos tradicionales en la detección de plantas faltantes en los cultivos de maíz debido a que actualmente se aplican métodos tradicionales para realizar esta tarea. Comúnmente una o varias personas recorren todo el terreno para detectar los faltantes en los cultivos de maíz a una edad temprana, pero esto conlleva a invertir tiempo y dinero para contratar al personal. De aquí surge la oportunidad de emplear los UAV para solventar esta problemática ya que ofrece muchos beneficios como:

- Tomar imágenes del terreno para la detección de los faltantes de los cultivos
- Permite recorrer todo el terreno aún en espacios de difícil acceso para las personas
- Permite realizar el análisis de los cultivos en un menor tiempo.

Riesgos

No se posee un dron: Para mitigar el riesgo, se ha solicitado que la carrera de software preste el dron para realizar las capturas de las imágenes.

No se tiene ningún lugar de cultivo de maíz para prácticas: Se puede conseguir cualquier lugar de la zona 1 del Ecuador para poder realizar la captura de imágenes. Como última instancia la UTN cuenta con cultivos de maíz dentro del algunos campus.

No se tiene las imágenes para procesar: Se tiene como respaldo un repositorio de imágenes de cultivos de maíz, gracias a investigaciones previas por parte del tutor.

Las imágenes tomadas no tienen buena resolución: Gracias al alcance delimitamos que las imágenes deben tener una resolución mínima de FHD (1920*1080)

Confinamiento por enfermedades pandémicas: En último caso se utilizaría las imágenes del repositorio del tutor para el algoritmo sin la necesidad de salir de casa.

En la TABLA 1 se presenta de manera resumida los riesgos. Además, se presenta un gráfico en la Fig. 4 acerca de la matriz de riesgos.

TABLA 1
RIESGOS

Código	Riesgo	Nivel
R1	No se posee un dron	ALTO
R2	No se tiene ningún lugar de cultivo de maíz para prácticas	ALTO
R3	No se tiene las imágenes para procesar	BAJO
R4	Las imágenes tomadas no tienen buena resolución	BAJO
R5	Confinamiento por enfermedades pandémicas	ALTO

Fuente: Propia

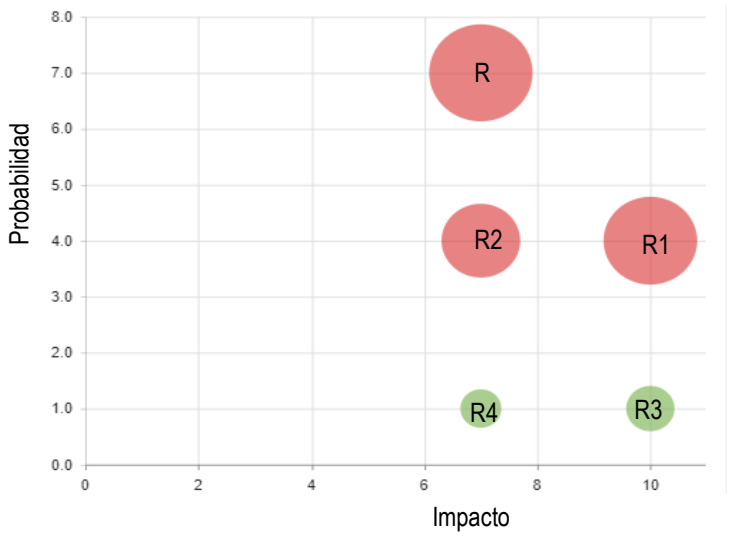


Fig. 4. Matriz de riesgos.
Fuente: Propia

CAPÍTULO 1

1. MARCO TEÓRICO

En este capítulo se abarca la teoría y definiciones importantes a tomar en cuenta para entender todo lo que engloba el proyecto, cuáles son los tipos de imágenes, en qué consiste la agricultura de precisión y las diferentes aplicaciones de los UAV en la agricultura.

1.1 Tipos de imágenes agrícolas

Antes de detallar los diferentes tipos de imágenes que existen, es necesario conocer el concepto acerca de qué es una imagen. Según Sindhu M (2009) una imagen es una señal en 2D (segunda dimensión) procesada por el sistema visual humano, estas señales se encuentran de forma analógica. Por este motivo para su procesamiento, almacenamiento y transición mediante aplicaciones informáticas se convierte esta de forma analógica a digital.

1.1.1 Imágenes digitales

Sindhu M (2009) menciona que una imagen digital es una matriz bidimensional de píxeles. Estas imágenes digitales también se las conoce como imagen de mapa de bits. Narciso (2018) menciona algunos ejemplos de este tipo de imágenes como: las fotografías que se obtienen a partir de una cámara digital o las que se generan con programas como Gimp, Paint, Photoshop, etc. Estas imágenes se forman por pequeños puntos a los que se denominan píxeles, los cuáles a la vez tienen un color determinado. La agrupación de estos píxeles es lo que forma una imagen. Además, el tamaño de una imagen puede ser variado ya que depende de la cantidad de píxeles que contiene, a esto se lo conoce como resolución, es decir mientras mayor cantidad de píxeles posea una imagen, mayor será la calidad del mismo. Las imágenes forman una parte significativa de los datos, particularmente en aplicaciones biomédicas, videoconferencia y en aplicaciones de teledetección. En la Fig. 5 se presenta un ejemplo de una imagen digital.

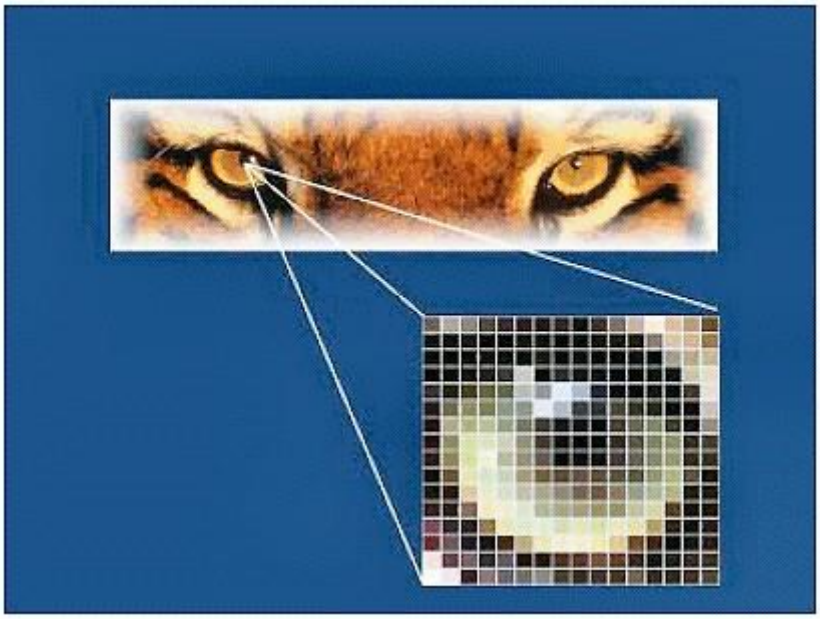


Fig. 5 Imagen digital
Fuente: (Miranda, 2009)

Formatos de imágenes digitales

Actualmente existen muchos formatos para las imágenes de mapas de bits como:

- JPG
- PNG
- GIF
- TIFF
- PSD
- BPM

JPG (Joint Photographic Experts Group)

JPG es el formato más utilizado para imágenes digitales. IONOS (2020) menciona que el formato JPG o JPEG hace referencia a la norma ISO/IEC 10918-1 de 1992. Sindhu M (2009) describe que JPG funciona analizando las imágenes y descartando información que probablemente el ojo note. Es importante destacar que el grado de compresión de JPG es ajustable, pero este proceso está ligado a una pérdida de calidad debido a que existen una pérdida de información de la imagen. IONOS (2020) menciona que el formato de comprensión JPG modifica la estructura de los píxeles de la imagen agrupando cada 8 x 8 píxeles en un bloque.

PNG (Portable Network Graphics)

IONOS (2020) describe que PNG es un formato de imagen que surge como alternativa del formato GIF (Graphic Interchange Format). Por lo general un archivo PNG puede estar entre un 10% y 20% más comprimido que un formato GIF. (Sindhu M, 2009). Este formato se destaca por ofrecer la profundidad de un color de 24 bits por píxel (16,7 millones de colores) con una canal alfa de 32 bits, además ofrece comprimir imágenes sin perdidas.

Según Sindhu M (2009) permite comprimir las imágenes teniendo en cuenta el tamaño de la imagen y la calidad de la imagen. Puede producir archivos más pequeños, pero a la vez permite más colores. PNG también permite la transparencia parcial gracias al canal alfa integrado.

GIF (Graphics Interchange Format)

Sindhu M (2009) menciona que el formato GIF es útil para imágenes que tiene menos de 256 colores. Esto es debido a que GIF solo funciona con imágenes con 8 bits por píxel, lo que significa 256 colores o menos, convirtiéndose en la mayor debilidad de este formato debido a que la mayoría de las imágenes poseen 24 bits por píxel. Es verdad que GIF es un formato de archivo sin pérdidas, pero solo para imágenes de menos de 256 colores, caso contrario puede perder hasta el 99.998% de los colores. Esto hace que se inadecuado para imágenes fotográficas.

Por otra parte, IONOS (2020) afirma que la ventaja de GIF es permitir hacer pequeñas animaciones debido a que este formato permite agrupar varias imágenes en un único archivo.

TIFF (Tagged Image File Format)

TIFF es un formato de gráfico utilizado para imágenes de alta resolución y para la transmisión de datos impresos. (IONOS, 2020). Se desarrolló en 1986 por Microsoft y es un formato que puede tener una comprensión sin pérdida y con pérdida. Sindhu M (2009) menciona que normalmente este formato guarde de 8 bits o 16 bits para cada canal de color (rojo, verde, azul) logrando una profundidad total de 24 a 48 bits. TIFF es utilizado casi siempre como un formato de almacenamiento de imágenes sin pérdidas que no utiliza comprensión alguna, esto provoca que sean archivos grandes, por este motivo no se utiliza TIFF como imágenes en la web y también porque la mayoría de los navegadores no mostrarán este tipo de imágenes.

PSD (Photoshop Document)

El formato PSD es ofrecido por Adobe para proyectos gráficos desarrollados en Photoshop. La información relativa se asegura en capas, canales o vectores permitiendo la edición de los mismos, es decir se pueden editar las capas que se hayan añadido, duplicado, desplazado, ocultado y eliminado. Estos formatos de imagen solo pueden ser abiertos por Adobe Photoshop y además permite guardar las capas y datos de la imagen sin pérdidas. (IONOS, 2020).

BPM

BMP (Windows Bitmap) se desarrolló en 1990 para sistemas operativos de Microsoft e IBM. Sindhu M (2009) menciona que este formato suele ser de gran tamaño debido a que no se encuentra comprimido. La ventaja que ofrece es su simplicidad, amplia aceptación y uso en el programa de Windows. IONOS (2020) afirma que el formato BPM tiene una profundidad de color de hasta 24 bits por píxel y que debido a su gran tamaño no es adecuado para utilizarlo dentro de página web.

1.1.2 Imágenes vectoriales

Alonso (2018) menciona que las imágenes vectoriales se basan en fórmulas matemáticas que se representan a través de figuras geométricas divididas en manchas de color y líneas, por este motivo no se divide en píxeles como en las imágenes de mapas de bits. De aquí surge la ventaja de que son independientes de la resolución, logrando mantener nitidez a pesar de que se amplíe la imagen. Otra ventaja de este tipo de imágenes es que se almacenan en archivos livianos ya que solo necesitan la información de las fórmulas matemáticas para generar la imagen.

Las imágenes vectoriales muestran una forma mediante fórmulas matemáticas denominados vectores. IONOS (2020) define al vector como comparable a una flecha el cual posee un origen, una dirección y una longitud.

A partir de estos conceptos podemos definir a la imagen vectorial como aquella imagen que se almacena y representa mediante trazos geométricos controlados por cálculos y fórmulas matemáticas tomando como referencia alguno puntos de la imagen para poder construir el resto de la misma. Esta imagen se forma por vectores los cuales son objetos que

se forman por una serie de puntos, líneas rectas y curvas. La Fig. 6 muestra la diferencia entre una imagen digital y una imagen vectorial.

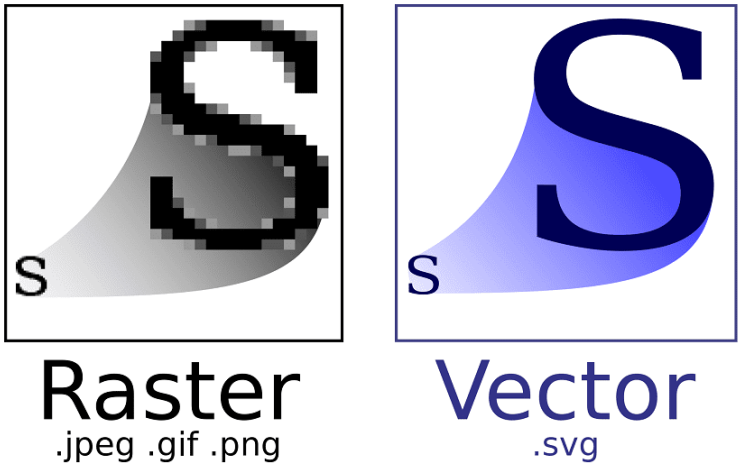


Fig. 6. Imagen vectorial
Fuente: (Neira, n.d.)

Formatos de Imágenes Vectoriales

Actualmente existen varios formatos de archivos que se han establecido a lo largo de los años. A continuación, en la TABLA 2 se describen los diferentes formatos de las imágenes vectoriales:

TABLA 2
FORMATOS DE IMÁGENES VECTORIALES

Formato vectorial	**Abreviación**	**Ámbito de uso**	**Explicación**
Adobe Illustrator	A1	Gráficos para impresión	Formato de intercambio orientado a vectores; guarda imágenes vectoriales editadas
Scalable Vector Graphics	SVG	Web	Formato de imagen escalable; se puede utilizar para animación y manipular con CSS
Encapsulated PostScript	EPS	Impresión	Basado en el lenguaje de programación PostScript

Portable Document Format	PDF	Impresión, web, archivo	Estándar mundial para compartir documentos electrónicos
Freehand Format	FH	Ilustraciones en color y sin color	Guarda ilustraciones con colores y rellenos
Corel Draw Format	CDR	Creación de imágenes para impresión y web	Formato de imagen sin comprimir con características como AI

Fuente: (IONOS, 2020)

1.2 Agricultura de precisión

Njoroge et al. (2018) menciona que según datos recopilados y analizados por la Organización de las Naciones Unidas para el año 2050 la población alcanzará los 9500 millones de personas. Otros autores como Vasudevan et al. (2016) también explica la misma situación y adiciona que se necesitará un 70% más de alimentos de lo que se produce actualmente para poder alimentar a esta cantidad de personas. Debido a los limitados recursos en el campo de la agricultura se requiere de la innovación para tener una agricultura sostenible y eficiente.

Un enfoque sostenible que se ha estado adoptando recientemente es la agricultura de precisión, la cual combina una serie de nuevas tecnologías dedicadas a la recopilación y transmisión de datos para un análisis efectivo y una toma de decisiones intuitiva. (Njoroge et al., 2018).

Por otra parte, Vasudevan et al. (2016) menciona que la práctica de la agricultura de precisión ofrece muchos beneficios, por ejemplo, un mejor uso y rendimiento del agua, los fertilizantes y pesticidas.

Hakkim et al. (2016) define a la agricultura de precisión como el enfoque en el que los insumos son utilizados, tomando en cuenta las cantidades precisas para obtener un mayor rendimiento en la agricultura en comparación con las técnicas tradicionales de cultivo que se emplean actualmente. Por tal motivo se puede considerar a la agricultura de información como un sistema integral diseñado para la optimizar la producción mediante el uso de información clave, gestión y tecnología, la cual se empleará con la finalidad de aumentar la

calidad del producto, mejorar la eficiencia de la producción y el uso de productos químicos sobre los cultivos, conservar la energía y proteger el medio ambiente.

La agricultura de precisión surge como respuesta a los desafíos futuros dentro del campo de la agricultura, ya que actualmente se puede evidenciar una disminución y degradación de los recursos naturales, estancamiento de los ingresos agrícolas, la falta de un enfoque ecorregional, la disminución y fragmentación a la posesión de tierras, las limitadas oportunidades en el sector agrícola y la variación climática global. Todos los problemas mencionados anteriormente se han convertido en las principales preocupaciones para el desarrollo futuro de la agricultura. (Hakkim et al., 2016).

Por este motivo el objetivo principal de la agricultura de precisión es mejorar y aumentar la eficacia de la producción al tiempo que se disminuyen los efectos negativos y la degradación ambiental causada por la sobreutilización de los recursos naturales como el agua, fertilizantes, pesticidas, insecticidas, semillas, entre otros. Actualmente ya existen herramientas y tecnologías que se emplean para la gestión y recopilación de datos como el uso de las WSN (Red de sensores inalámbricos), IoT (Internet de las cosas), algoritmos inteligentes, modelado meteorológico, dispositivos móviles y sistemas robóticos. (Njoroge et al., 2018).

Adicionalmente Vasudevan et al. (2016) menciona que otra herramienta poderosa es el uso de los UAV (vehículos aéreos no tripulados) y el uso de los UGV (vehículos terrestres no tripulados), ya que la idea principal es hacer que estos vehículos inspeccionen los cultivos periódicamente y esta información se utilice para establecer operaciones de indicación y controles adecuados.

Dentro de la agricultura de precisión, Njoroge et al. (2018) describe cuatro fases principales:

- La recopilación de los datos
- El análisis de los datos recopilados
- La gestión o la toma de decisiones
- La aplicación

La primera fase de recolección de datos se la puede realizar mediante las herramientas mencionadas anteriormente acerca de la recolección de datos. Posteriormente dentro del análisis de datos se involucra el uso de algoritmos que permita clasificar y procesar los datos para convertirlos en información valiosa que ayudarán a la fase tres correspondiente a la gestión y toma de decisiones. Finalmente, una vez analizado los datos y tomado la decisión

que se va a aplicar, viene la fase cuatro que consiste en la aplicación de la decisión que se tomó previamente.

1.2.1 Ejemplos de agricultura de precisión

Una vez definido la agricultura de precisión, el autor Njoroge et al. (2018) menciona algunas tecnologías que se están adoptando actualmente para monitorear y tomar decisiones precisas basadas en los datos recopilados mediante sensores. Como ejemplo tenemos el sistema de control inteligente basado en IA (Inteligencia Artificial) propuesto por Pahuja et al. (2013) para el desarrollo y síntesis de un controlador climático inteligente para la agricultura de precisión en invernaderos. Este sistema emplea ANN (Red Neuronal Artificial) para el control de la humedad y temperatura dentro de los invernaderos climatizados artificialmente.

Otro estudio relacionado con la agricultura de precisión fue realizado por Karim et al. (2014) en donde se empleó el análisis de imágenes hiperespectrales (imágenes que implican la recopilación y el procesamiento de información en todo el espectro electromagnético) con SVM (Máquinas de Vectores de Soporte) y ANN como herramientas para la detección y clasificación de malezas en campos de maíz con el objetivo de controlar inteligentemente la cantidad de nitrógeno aplicado para el manejo de malezas. De este estudio se obtuvo que la técnica SVM poseía tasas de error menores a la técnica de ANN, concluyendo que el SVM era la opción más adecuada para la detección y el control efectivo de las malas hierbas.

De esta manera se pueden mencionar otras investigaciones que se han venido adoptando en el campo de la agricultura de precisión. Por ejemplo, un estudio interesante se menciona en el artículo propuesto por Roldán Ortega et al. (2019) acera de la detección de enfermedades en el sector agrícola utilizando inteligencia artificial. El proceso consistía en obtener características de las hojas o frutos de las plantas que mediante técnicas de inteligencia artificial podrían determinar si una hoja de la planta presentaba signos de alguna enfermedad. Como conclusión de la investigación se llegó a determinar que sí es posible detectar enfermedades mediante la IA.

1.2.2 Herramientas y equipos

Sistema de posicionamiento global (GPS)

El GPS es un sistema de posicionamiento el cual consiste en una red de satélites que ayuda a los usuarios a registrar información de ubicación (latitud, longitud y altitud). El sistema permite a los agricultores identificar con precisión la ubicación del campo para que los insumos (semillas, fertilizantes, pesticidas, herbicidas y agua de riego) se pueden aplicar en los campos individuales en función de criterios de rendimiento y aplicaciones de insumos anteriores. (Hakkim et al., 2016).

Tecnologías de sensores

Los sensores permiten recopilar datos ya que pueden medir la temperatura, la humedad, la vegetación, el nivel de nutrientes, el vapor, el aire, etc. Todo esto permite que ya no sea necesario el análisis en un laboratorio para saber esta información. Gracias a estos datos obtenidos por la teledetección se pueden ubicar las condiciones de estrés, distinguir especies de cultivos, identificar plagas y malezas, monitorear la sequía, el suelo y las condiciones de las plantas. (Hakkim et al., 2016).

Sistema de información geográfica (GIS)

GIS es una especie de mapa computarizado cuya función es utilizar la estadística y métodos espaciales para analizar caracteres y geografía. Los mapas GIS computarizados son diferentes de los mapas convencionales ya que contienen varias capas de información. Una base de datos GIS agrícola puede proporcionar información sobre topografía archivada, tipos de suelo, drenaje superficial, drenaje subterráneo, pruebas de suelo, riego, tasas de aplicación de productos químicos y rendimiento de cultivos. Esto permite evaluar la gestión actual sobre el cultivo y las diferentes alternativas que se pueden dar modificando las capas de datos para producir un análisis del nuevo escenario. (Hakkim et al., 2016).

Muestreo del suelo en cuadrículas y aplicación de fertilizantes con tecnologías de tasa variable (VRT)

Las VRT son automáticas y se pueden aplicar en numerosas operaciones agrícolas. El muestreo del suelo en cuadrícula ocupa los mismos principios de un muestreo del suelo normal, pero con la diferencia que en el muestro en cuadrícula se aumenta la intensidad del muestreo. La finalidad de este muestreo es obtener un mapa detallado acerca de las necesidades de los nutrientes denominado mapa de aplicación. Este mapa es ingresado en la computadora de un esparcidor de fertilizante de dosis variable el cual se guía mediante el

GPS para fertilizar a los cultivos de acuerdo con la dosis necesaria que le provee el mapa de aplicación. (Hakkim et al., 2016).

Software

La aplicación de tecnologías de agricultura de precisión permitirá que se requiera el uso de software para llevar a cabo ciertas tareas como la interfaz del controlador de pantalla, el mapeo de capas de información en análisis e interpretación de datos previos y posteriores al procesamiento, la contabilidad de insumos agrícolas, entre otros. Otra aplicación donde se necesitará del software sería en la generación de mapas para obtener los detalles sobre el rendimiento del suelo de un área determinada. (Hakkim et al., 2016).

1.3 Aplicaciones agrícolas con imágenes con UAV

Ponce-Corona et al. (2019) define a los UAV (vehículos aéreos no tripulados) como aeronaves que vuelan automáticamente por medio de un sistema piloto automático el cual es monitoreado desde un centro de control en tierra. El ejército estadounidense, durante la década de los 50 fue el primero en utilizar los UAV en operaciones militares para tareas de reconocimiento, vigilancia y mapeo.

Por otra parte, los UAV también se aplican en el área de la agricultura. Entre las principales aplicaciones están la detección, identificación y conteo de cultivos, árboles, hierbas y coníferas. Estas aplicaciones han ayudado de manera significativa en la toma de decisiones por parte de los agricultores; también los UAV ayudan a mejorar la eficiencia y disminuir el tiempo para monitorear los diferentes cultivos. Por ejemplo, para el conteo de cultivos por métodos tradicionales (de manera manual) debido a la gran cantidad de tiempo que esto conlleva, los agricultores optan por contar una parte de la cosecha y tomar decisiones en base a la información que obtuvieron del análisis de esa parte de la cosecha, dando lugar a recuentos de plantas, árboles o cultivos poco frecuentes e inexactos. (Ponce-Corona et al., 2019).

Los UAV se pueden aprovechar para identificar acerca de las zonas de los cultivos que necesiten de una gestión. Esto da a los agricultores la capacidad de reaccionar a tiempo ante cualquier problema detectado. Además de los beneficios que ofrecen los UAV en los cultivos, también existen otras aplicaciones como monitoreo de salud y detección de enfermedades,

monitoreo de crecimiento y estimación de rendimiento, manejo y detección de malezas, entre otros. (Tsouros et al., 2019).

Honrado et al. (2017) también afirma que la llegada de los UAV ha comenzado a reducir los costos de las aplicaciones de detección remota para la agricultura de precisión, ya que por primera vez se ha podido recopilar datos con mayor regularidad dando como resultado modelos de productividad de cultivos más sólidos. La razón por la que los UAV están siendo cada vez más utilizados en el campo de la agricultura es porque ofrecen muchos más beneficios que otros métodos de detección. Por ejemplo, la insuficiencia de teledetección satelital para la agricultura de precisión se atribuye a la baja accesibilidad de las imágenes de alta resolución, esto se debe a que las imágenes multiespectrales de acceso abierto tienen resoluciones muy bajas. Por otra parte, utilizar imágenes satelitales multiespectrales de alta resolución conlleva un gasto mayor, puesto a que estas imágenes son muy caras. Otra desventaja de utilizar la teledetección satelital es que a menudo las imágenes se ven empañadas por la presencia de nubes. Generalmente una alternativa para estos casos era utilizar vuelos tripulados para detectar las imágenes, pero esto requería de tener una mayor altitud y de un sistema de cámaras especializado. A diferencia de esto los UAV utilizan cámaras disponibles en los mercados, las cuales pueden abordar las diferentes necesidades que se requieran para la captura de imágenes de los cultivos.

A pesar de que el uso de los UAV data de la década de los 90, su popularización se ha dado en los años recientes gracias a las aplicaciones relacionadas al monitoreo de cultivos o áreas forestales. Como se ha dicho anteriormente, esto se ha convertido en una solución atractiva debido al bajo costo que se requiere para obtener las imágenes para el monitoreo. Ponce-Corona et al. (2019) explica que además de las imágenes adquiridas por los UAV, también es importante considerar el software, ya que dichas imágenes necesitan ser evaluadas y procesadas por un software específico como Pix4DMapper, PhotoScan, EnsoMOSAIC, etc. Generalmente se usan estos programas para la generación de los ortomosaicos (ortofotos). Según Ponce-Corona et al. (2019), un mosaico corresponde a un conjunto de imágenes que presentan áreas de traslape entre sí, estas imágenes son combinadas en un salo imagen para ampliar el rango de visión de la escena.

Para extraer información de estas imágenes se utilizan diferentes técnicas relacionadas con la segmentación de imágenes (binarización, filtros, operadores morfológicos, entre otras), hasta técnicas de aprendizaje máquina (redes neuronales artificiales, máquinas de soporte vectorial, entre otras). (Ponce-Corona et al., 2019).

1.3.1 Tipos de aplicaciones de los UAV en la agricultura de precisión

Según Tsouros et al. (2019) las aplicaciones más comunes de los UAV en la agricultura de precisión son:

- Mapeo y manejo de malezas
- Monitoreo de crecimiento de la vegetación y estimación del rendimiento
- Monitoreo de la salud de la vegetación y detección de enfermedades
- Gestión del riego
- Fumigación de cultivos

Mapeo y manejo de malezas

Entre las aplicaciones más populares se encuentra el mapeo de malezas. Se considera malezas a las plantas no deseables que crecen en los cultivos agrícolas y que causan varios problemas debido a compiten con las plantas de los cultivos por los recursos del suelo como el agua, espacio y nutrientes. Esto provoca pérdidas de rendimiento y crecimiento en los cultivos. Para resolver esta problemática la opción más utilizada es el uso de herbicidas. Es importante que la aplicación del herbicida sobre las malezas del cultivo se realice de manera precisa, ya que el exceso del herbicida sobre los cultivos puede llegar a ser perjudicial para el medio ambiente y para el cultivo. Además, esto también representa un aumento en los costos para el mantenimiento del cultivo. Para esta problemática se utiliza el SSWM (Manejo de malezas específicos del sitio) para rociar el herbicida en sitios específicos donde se encuentre la maleza en lugar de rociarlo por todo el cultivo. Para lograr este objetivo es necesario tener un mapa preciso que muestre las áreas donde se necesita rociar el herbicida. Para generar este mapa se utilizan los UAV para recopilar imágenes y obtener datos de todo el campo generando un mapa preciso sobre las zonas necesarias para rociar el herbicida. (Tsouros et al., 2019).

Monitoreo de crecimiento de la vegetación y estimación del rendimiento

La falta de medios para monitorear el crecimiento del cultivo se considera uno de los principales obstáculos para aumentar la productividad y la calidad agrícola. A esto también se le suma la variabilidad de las condiciones climáticas, lo cual altera el clima poniendo en peligro el cultivo. Para solventar este problema se ha requerido la aplicación de los UAV para supervisar la vegetación y proporcionar estimaciones sobre el rendimiento. Gracias a la

recopilación periódica de información y a la visualización de los cultivos, los UAV brindan mayores oportunidades para monitorear el crecimiento de los cultivos y observar la variabilidad de los diferentes parámetros del campo. Estos parámetros pueden ser la biomasa y el nitrógeno de los cultivos. Una vez identificado el nivel de los parámetros en los cultivos se pueden tomar decisiones sobre la necesidad del fertilizante en el cultivo u otras acciones necesarias. (Tsouros et al., 2019).

Otras aplicaciones de los UAV sobre esta área puede ser la creación de mapas tridimensionales del cultivo y la medición de diversos parámetros como: la altura del cultivo, la distancia entre las hileras o las planta, entre otros.

Monitoreo de la salud de la vegetación y detección de enfermedades

Los UAV también son utilizados para monitorear la salud de la vegetación. La salud es uno de los factores más importante del cultivo que se necesita monitorear debido a que las enfermedades causan grandes pérdidas económicas a los cultivos, disminuyendo la productividad, el rendimiento y la calidad de la cosecha. Por esta razón el monitoreo en esta parea es importante, ya que se considera vital detectar las enfermedades a tiempo para evitar problemas de propagación de las mismas. Tradicionalmente esta tarea la realizaba algunas personas expertas en esta área, las cuales necesitan ir directamente al campo para monitorear el cultivo. Esto significaba consumir recursos tanto económicos y de tiempo, lo que conlleva a que no se pueda realizar un monitoreo continuo sobre el cultivo. Otro método común para tratar las enfermedades del cultivo es la aplicación de pesticidas en los productos en fechas determinadas. Esta solución puede conllevar a un alto costo para el agricultor y también a la contaminación del de fuentes subterráneas de agua. Por tal razón, la agricultura de precisión es la opción más viable debido a que se lleva un control de enfermedades en sitios específicos. Gracias a esta técnica es factible detectar la enfermades del cultivo. Una vez más se emplean los UAV para la recolección de datos mediante las imágenes de los cultivos para identificar los cambios en la biomasa vegetal y su salud. Según Tsouros et al. (2019) los UAV pueden aplicar en dos fases: La primera fase consiste en el monitoreo de las enfermedades durante sus primeras etapas para detectar la enfermedad antes de que aparezca indicios visuales. La segunda etapa se la realiza durante el tratamiento de la infección, donde los agricultores usan los UAV para la fumigación dirigida.

Gestión del riego

Tsouros et al. (2019) menciona que actualmente el 70% de agua que se consume a nivel mundial se utiliza para el riego de cultivos, lo que pone en evidencia la necesidad de técnicas de riego de precisión. Las técnicas de precisión ayudan a gestionar de mejora manera el uso de este recurso aplicando de manera efectiva en los lugares correctos, en el momento adecuado y en la cantidad correcta. Lograr identificar el área donde se necesita del riego ayudará a los agricultores a horrar tiempo y recursos hídricos logrando una mayor productividad y calidad de los cultivos. Para lograr esto es necesario la aplicación de UAV que utilicen sensores adecuados que permiten identificar las zonas que necesiten más agua.

Fumigación de cultivos

Aunque es poco común utilizar los UAV en la fumigación de los cultivos, esta aplicación existe. Por lo general se emplean equipos diseñados para esta tarea como las mochilas pulverizadoras manuales a presión de aire y alimentados con batería. Sin embargo, el uso de estos equipos puede causar grandes pérdidas de pesticidas. Otra desventaja del uso de este equipo es la necesidad de que un operador esté presente, así como también la cantidad de tiempo que se necesita para fumigar todo el campo. De esta manera el uso de los UAV resulta ser más útiles que los equipos tradicionales, debido a la menor exposición del operador y la capacidad mejorada de fumigar el cultivo con precisión. Otra ventaja es que los UAV pueden seguir la morfología del suelo manteniendo una altura constante. Todo esto permite lograr un mejor resultado en la fumigación de los cultivos, en menor tiempo y agotando menos el recurso de los pesticidas. Además de las aplicaciones de los UAV en la agricultura mencionados anteriormente, Tsouros et al. (2019) también menciona que existen otras aplicaciones como: análisis del suelo, selección de genotipos de algodón, detección de mamíferos y evaluación de conductividad eléctrica del suelo.

1.4 Lenguajes de programación para procesamiento de imágenes agrícolas

1.4.1 Procesamiento de imágenes

Antes de hablar de los lenguajes de programación implementados en el área de la agricultura, es necesario definir acerca del procesamiento de imágenes.

Según Jimenez Lopez et al. (2016) el procesamiento de imágenes es un ciencia de manipulación de imágenes en la cual se utilizan computadoras para realizar procedimientos

específicos de acuerdo a las aplicaciones y requisitos de los usuarios. Estos requisitos pueden ser: filtrado, corte, segmentación, comprensión y reconocimiento. Esta área ha captado la atención de varios investigadores para e desarrollo y mejoras de algoritmos relacionados con la robótica, comunicaciones, teledetección, biomedicina, navegación y metrología óptica, entre otras.

Por otra parte, Tyagi (2018) menciona que el procesamiento de imágenes hace referencia a las técnicas para el procesamiento de una imagen, obtener una imagen mejorada o extraer información útil de ella para tomar decisiones en base al resultado obtenido del procesamiento. Dentro de la literatura se definen tres niveles de operaciones de procesamiento de imágenes:

- **Procesamiento de imágenes de bajo nivel:** Consiste en operaciones primitivas sobre la imagen como mejora de contraste, ruido, etc. En este nivel tanto la entrada como la salida son imágenes.
- **Procesamiento de imágenes de medio nivel:** Incluye a las operaciones que implica la extracción de atributos como bordes, contornos, regiones, etc.
- **Procesamiento de imágenes de alto nivel:** Esta categoría implica operaciones complejas de procesamiento de imágenes relacionadas al análisis y la interpretación de los contenidos de una escena para la toma de decisiones.

Operaciones típicas de procesamiento de imágenes

El procesamiento de imágenes involucra una serie de algoritmos y técnicas. Según Tyagi (2018) entre las operaciones más típicas se encuentran las siguientes:

- **Binarización:** Consiste en la conversión de una imagen a color a una imagen en escala de grises binaria con el objetivo de simplificar y acelerar el procesamiento. Esta conversión a una imagen binaria con dos niveles de gris (blanco y negro) se conoce como binarización.
- **Suavizado:** Técnica que se utiliza para difuminar o suavizar los detalles de los objetos de una imagen.
- **Afilado:** Técnica que se utiliza para mejorar los bordes y los detalles finos de los objetos en una imagen con la finalidad de aumentar su visualización. A esta técnica también se la denomina como “nitidez”.
- **Eliminación de ruido y desenfoque:** Andes de procesamiento es importante reducir la cantidad de ruido en las imágenes mediante filtros de eliminación de ruido. A veces

se puede utilizar la técnica de eliminación de imágenes según el tipo de ruido o desenfoque de la imagen.

- **Extracción de bordes:** Esta técnica se utiliza para encontrar objetos antes de analizar el contenido de la imagen.
- **Segmentación:** Consiste en el proceso de dividir una imagen en varias partes para el reconocimiento y clasificación de los objetos. La segmentación es un paso previo al procesamiento.

1.4.2 Lenguajes de programación

Los lenguajes de programación utilizados en la agricultura son muy variados. Esto se debe a que los algoritmos se pueden desarrollar en todo tipo de lenguajes. La diferencia radica en que unos lenguajes son más utilizados que otros.

Si nos centramos en los lenguajes de programación aplicados en la agricultura, la literatura revela algunos lenguajes de programación que se han utilizados en diversos proyectos. Los lenguajes más utilizados son los siguientes:

- Python
- JavaScript
- Matlab (Lenguaje M)
- PHP

Python

Según Nolasco Valenzuela (2018) Python es un lenguaje de programación creado por Guido Van Rosum, el cual trabaja actualmente para Dropbox. Este lenguaje de programación posee una sintaxis muy limpia y legible. Además, el lenguaje tiene tipado dinámico, es decir que una variable puede poseer varios tipos de datos, esto hace que sea un lenguaje fácil de aprender. Python es un lenguaje interpretado, por lo cual no necesita compilar el código fuente para poder ejecutarlo.

El lenguaje con el que está escrito Pyhon es “C”, razón por la cual se puede extender en C o C++. Una de las características más importantes de este lenguaje es que es multiparadigma: Programación estructurada, programación orientada a objetos y programación funcional. (Nolasco Valenzuela, 2018)**.**

Actualmente para el desarrollo web con Python se puede utilizar los frameworks: Flask y Django. Nolasco Valenzuela (2018) menciona a Dropbox e Instagram como ejemplos de desarrollo web con Python.

Por otra parte, en el campo de Data Science y Machine Learning tenemos: Scikit-Learn, Pandas y TensorFlow. Python también posee la ventaja de ser multiplataforma por lo que pude desplegarse en Windows, Linux, Mac OS, Solaris, etc.

En lo que respecta Python en el uso de procesamiento de imágenes en el campo de la agricultura, la literatura muestra que es uno de los lenguajes más utilizados para esta área.

Existen muchos proyectos que aplican procesamiento de imágenes con Python para la agricultura. A continuación, se muestran algunos proyectos desarrollados en Python.

- Sistema de detección de enfermedades en las plantas de tomates por medio del procesamiento de imágenes por (Adhikari et al., 2018).
- Detección de vegetación usando imágenes tomadas desde vehículos aéreos no tripulados: Una revisión sistemática por (Ponce-Corona et al., 2019).

JavaScript

Para (Mohedano, 2013) JavaScript es un lenguaje de programación utilizado principalmente para la creación de páginas Web que puedan interactuar con el usuario, es decir que por medio de JavaScript se puede agregar cierto dinamismo a las páginas Web debido a que añade procesos de respuesta a las acciones del usuario. Es importante destacar que estos procesos se ejecutan en la máquina del cliente ya que hace uso de los navegadores. (Mohedano, 2013) también menciona que JavaScript es un lenguaje interpretado, lo que significa que las instrucciones son analizadas en secuencia por medio del intérprete que posee el navegador Web.

Por otra parte, al igual que el lenguaje Python, JavaScript también es utilizado que el área de la agricultura. Aunque el porcentaje de aplicación en esta área es menor comparado con Python, la literatura muestra que si existen proyectos desarrollados con este lenguaje en el área de la agricultura. A continuación, se muestran algunos ejemplos:

- Diseño de Sistema de Agricultura Verde Utilizando Internet de las Cosas y Técnicas de procesamiento de imágenes por (Tran et al., 2018)
- Una aplicación basadga en la web para la agricultura: “Sistema de agricultura inteligente” por F. M. (2020)

Matlab

(J. & Rodriguez, 2020) menciona que Matlab es un programa destinado para el uso de cálculos numéricos con vectores y matrices además de utilizar números escalares. Una de sus características principales es la amplia variedad de gráficos que puede proveer de dos y tres dimensiones. Matlab tambiggggén tiene un lenguaje de programación propio el cual se denomina lenguaje M. En resumidas palabras, se podría decir que Matlab es un programa para el cálculo técnico y científico. Matlab también posee varias librerías especializadas (toolboxes) y un código básico.

Este programa también puede ser utilizado en el campo de la agricultura por medio del procesamiento de imágenes. Gracias a esto es posible desarrollar el proyecto presente de "Desarrollo de algoritmos de procesamiento de imágenes agrícolas obtenidas por drones para la detección de plantas faltantes en cultivos de maíz" por medio de este programa.

Otros ejemplos en donde se aplica Matlab en la agricultura son los siguientes:

- Un estudio del procesamiento de imágenes en la agricultura por Prakash et al. (2017).
- Detección y clasificación de enfermedades en las hojas de las plantas por el procesamiento de imágenes usando MATLAB por Prakash et al. (2017)

PHP

Según Pavon Puertas & Llarena Borges (2015) PHP es un lenguaje de programación de nivel alto el cuál es interpretado por el servidor, es decir que las páginas se encuentran alojadas en el servidor al contrario de otros lenguajes que se ejecutan en el propio navegador. Este lenguaje es empleados en sitios importantes como Facebook, WordPregggss o Wikipedia.

Unas de las ventajas de este lenguaje es que es gratuito, por lo cual se puede utilizar sin coste alguno. Un ejemplo aplicado en la agricultura es el siguiente:

- Diseño de un sistema de riego agrícola controlado a distancia por Alvarez (2015).

CAPÍTULO 2

2. DESARROLLO

2.1 Requisitos funcionales y no funcionales

Según Sommerville (2005) un requerimiento o requisito se visualiza como una declaración abstracta de alto nivel de un servicio que debe proveer el sistema. Si tomamos este concepto, se puede relacionar al requerimiento como una necesidad por parte del cliente. Es importante definir las necesidades de forma abstracta para poder establecer una solución. Para (Sommerville, 2005) las definiciones del requerimientos pueden ser dados tanto por el usuario como por el sistema. Los requerimientos del usuario son las declaraciones en lenguaje natural y en diagramas de aquellos servicios que el sistema debe proveer bajo las condiciones de operación establecidas. Por otra parte, los requerimientos del sistema establecen en detalle las restricciones y servicios del sistema. Este documento de requerimientos del sistema debe ser preciso debido a que sirve como un contrato entre el desarrollador del software y el comprador del sistema.

Tipos de Requisitos

(Sommerville, 2005) estable dos tipos de requisitos:

- **Requisitos funcionales:** Son las declaraciones de los servicios que proveerá el sistema. Dentro de la funcionalidad del sistema estos requisitos deben ser completados y ser consistentes. La consistencia significa que no pueden tener definiciones contradictorias.
- **Requisitos no funcionales:** Se refiere a las restricciones de los servicios o funciones ofrecidos por el sistema. Estas restricciones se pueden referir al tiempo sobre el proceso de desarrollo, estándares, etc.

Con respecto a los algoritmos desarrollados en este documento se encontraron los siguientes requisitos que se muestran en la TABLA 3:

TABLA 3
REQUISITOS DEL SISTEMA

Código	Historia de Usuario	Prioridad
RF-001	Formato de imagen	Alta

RF-002	Señalar la planta faltante dentro del cultivo	Alta
RF-003	Aplicación de escritorio	Medio
RF-004	Dimensiones de la imagen	Medio
RF-005	Ejecución de algoritmo con diferentes distancias	Alta
RNF-ARQ-001	Requisitos de software	Alta

Fuente: Propia

2.1.1 Requisitos funcionales

La TABLA 4 muestra el primer requisito funcional:

TABLA 4
REQUISITO FUNCIONAL 1: FORMATO DE IMAGEN

Código	**Nombre**	**Prioridad**
RF-001	Formato de imagen	Alta
Descripción	La aplicación de escritorio debe recibir imágenes con el formato jpg.	
Proceso	Al ingresar a la aplicación el sistema tiene que validar el formato de imagen ingresado. Solo debe permitir el ingreso de imágenes con el formato jpg.	
Nota		

Fuente: Propia

El segundo requisito funcional incluye la funcionalidad principal del sistema la cual se muestra en la TABLA 5:

TABLA 5
REQUISITO FUNCIONAL 2: SEÑALAR LA PLANTA FALTANTE DENTRO DEL CULTIVO

Código	**Nombre**	**Prioridad**
RF-002	Señalar la planta faltante dentro del cultivo	Alta
Descripción	Como usuario se requiere que el sistema sea capaz de detectar las plantas faltantes en los cultivos de maíz.	

Proceso	Los usuarios al insertar la imagen dentro del aplicativo obtendrán como resultado una imagen en formato jpg donde se señalará la planta faltan dentro de las hileras de los cultivos de maíz mediante un bounding box.
Nota	

Fuente: Propia

El siguiente requisito funcional especifica la arquitectura que va a tener el aplicativo. Esto se muestra en la TABLA 6:

TABLA 6
REQUISITO FUNCIONAL 3: APLICACIÓN DE ESCRITORIO

Código	**Nombre**	**Prioridad**
RF-003	Aplicación de escritorio.	Medio
Descripción	Como usuario se requiere que el algoritmo se despliegue mediante una aplicación de escritorio.	
Proceso	Se desarrollará con Matlab una aplicación de escritorio para permitir el ingreso de las imágenes a analizar con el algoritmo.	
Nota		

Fuente: Propia

En la TABLA 7 se especifican los requisitos con respecto a la imagen que se va a ingresar al algoritmo:

TABLA 7
REQUISITO FUNCIONAL 4: DIMENSIONES DE LA IMAGEN

Código	**Nombre**	**Prioridad**
RF-004	Dimensiones de la imagen.	Medio
Descripción	Como usuario se requiere que el sistema valide que las imágenes ingresadas sean de mínimo 1920x1080.	
Proceso	Se implementará una validación para que las imágenes ingresadas tengan la resolución mínima especificada en el requisito.	

Nota

Fuente: Propia

El siguiente requisito funcional se refiere a los diferentes casos de prueba que se tiene. Este requisito se especifica en la TABLA 8.

TABLA 8
REQUISITO FUNCIONAL 5: EJECUCIÓN DE ALGORITMO CON DIFERENTES DISTANCIAS

Código	**Nombre**	**Prioridad**
RF-005	Ejecución de algoritmo con diferentes distancias	Media
Descripción	La aplicación de escritorio debe poder analizar las imágenes de los cultivos de maíz con las distancias de 5m, 10m y 15m con respecto al suelo.	
Proceso	La aplicación tendrá la funcionalidad de poder analizar las imágenes con diferentes distancias. Para esto el aplicativo tendrá una opción para elegir la distancia con la que se desea analizar. Dependiendo de la distancia se escogerá el algoritmo correspondiente.	

Nota

Fuente: Propia

2.1.2 Requisitos no funcionales

El requisito no funcional se muestra en la TABLA 9:

TABLA 9
REQUISITO NO FUNCIONAL 1: REQUISITOS DE SOFTWARE

Código	**Nombre**	**Prioridad**
RNF-ARQ-001	Requisitos de software	Alta
Descripción	Herramientas de desarrollo	
Proceso	Toda la aplicación se desplegará en el entorno de Matlab. Para esto se utilizará el algoritmo para la detección de malezas el cual se adaptará para la detección de cultivos faltantes.	

Nota

Fuente: Propia

2.2 Diseño del algoritmo

Para el diseño del algoritmo se utilizó la disciplina de visión por computador. Según David et al. (n.d.) el método de visión por computador o también conocido como visión artificial es una de las ramas de la inteligencia artificial clásica. Esta se constituye por algunas fases que proveen al ordenador la capacidad de percibir y comprender una imagen tratando de imitar el proceso de percepción que realizan los seres humanos. Estas fases van a depender de la aplicación que se vaya variando. A continuación, en la Fig. 7 se muestra las fases o etapas que se encuentran en la mayoría de las aplicaciones que utilizan visión por computador:

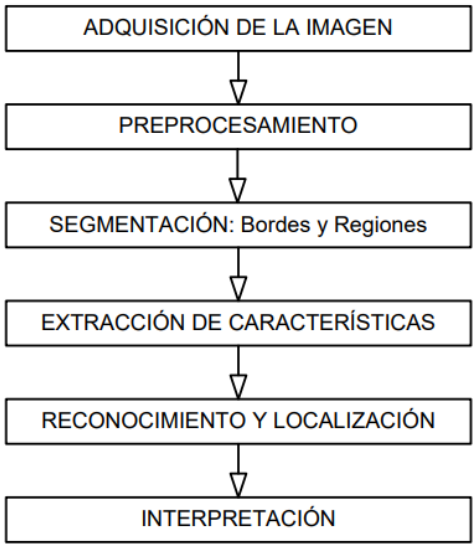


Fig. 7. Etapas de un sistema de visión artificial
Fuente: (Molleda Meré, 2008)

Adicionalmente se describirá cada una de las etapas según (Molleda Meré, 2008) y cómo se relaciona con el algoritmo desarrollado en este documento:

- **Adquisición de la imagen:** Esta etapa consiste en la captura de una proyección en dos dimensiones de la luz reflejada por los objetos, es decir la captura de la imagen. Para la captura de la imagen se utilizó un dron que tomará las imágenes de los cultivos de maíz a una cierta distancia. Las distancias con las que se tomará las imágenes son de 5m, 10m y 15m de altura con respecto al suelo.

- **Preprocesamiento:** Se realiza tareas de eliminación de ruido y/o realce de la imagen. Dentro del algoritmo se extraen los valores RGB (red, green y blue) para obtener una imagen en escala de grises.
- **Segmentación:** Permite separar los diferentes elementos de la escena. Para esto se hace la imagen binaria por medio del umbral de Otsu. De esta manera resaltamos las plantas y malezas que se encuentran en la imagen descartando todos los demás objetos que se encuentran en la imagen. También se aplica una función dilatar para resaltar los objetos de interés. Una vez realizado este proceso se utiliza la transformada de Hough, este método nos permite reconocer líneas dentro de la imagen por medio de patrones. En este caso las líneas que se identifican son las hileras de cultivos de maíz dentro de las imágenes.
- **Extracción de características:** Se obtiene una representación formal de los objetos segmentados en la etapa anterior. En la parte de segmentación ya se detecta las hileras de los cultivos de maíz. Posteriormente se descarta todo lo que se encuentra entre las hileras de cultivos para obtener en la imagen binarizada solo las hileras de los cultivos. De esta manera se separa las plantas de la maleza y también se descartan los objetos que no sean de interés.
- **Reconocimiento y localización:** En esta etapa se procesa la información obtenida anteriormente, luego se realiza una interpretación y se aplica una acción de acuerdo con lo analizado. Dentro del algoritmo se realiza la localización de las plantas de cultivo. Una vez obtenidas las coordenadas de las plantas por cada hilera del cultivo se procede a calcular la distancia entre las plantas. Estas distancias permiten verificar cuando una planta falta dentro de la hilera del cultivo, debido a que si la distancia que se obtienen entre dos plantas de la hilera es mayor a la distancia promedio de las plantas en el cultivo significaba que existen un faltante dentro del mismo.
- **Interpretación:** A partir de la información anterior y del conocimiento acerca del entorno se interpreta la escena. Cabe recalcar que el algoritmo solo llega hasta la fase de reconocimiento y localización de las plantas faltantes de maíz. En este caso no existe una interpretación por parte del algoritmo.

Una vez explicado cómo funciona el algoritmo por cada una de las etapas de visión por computador, adicionalmente se presentará un diagrama de flujo del algoritmo en la Fig. 8. Un diagrama de flujo es una técnica de representación de algoritmos.

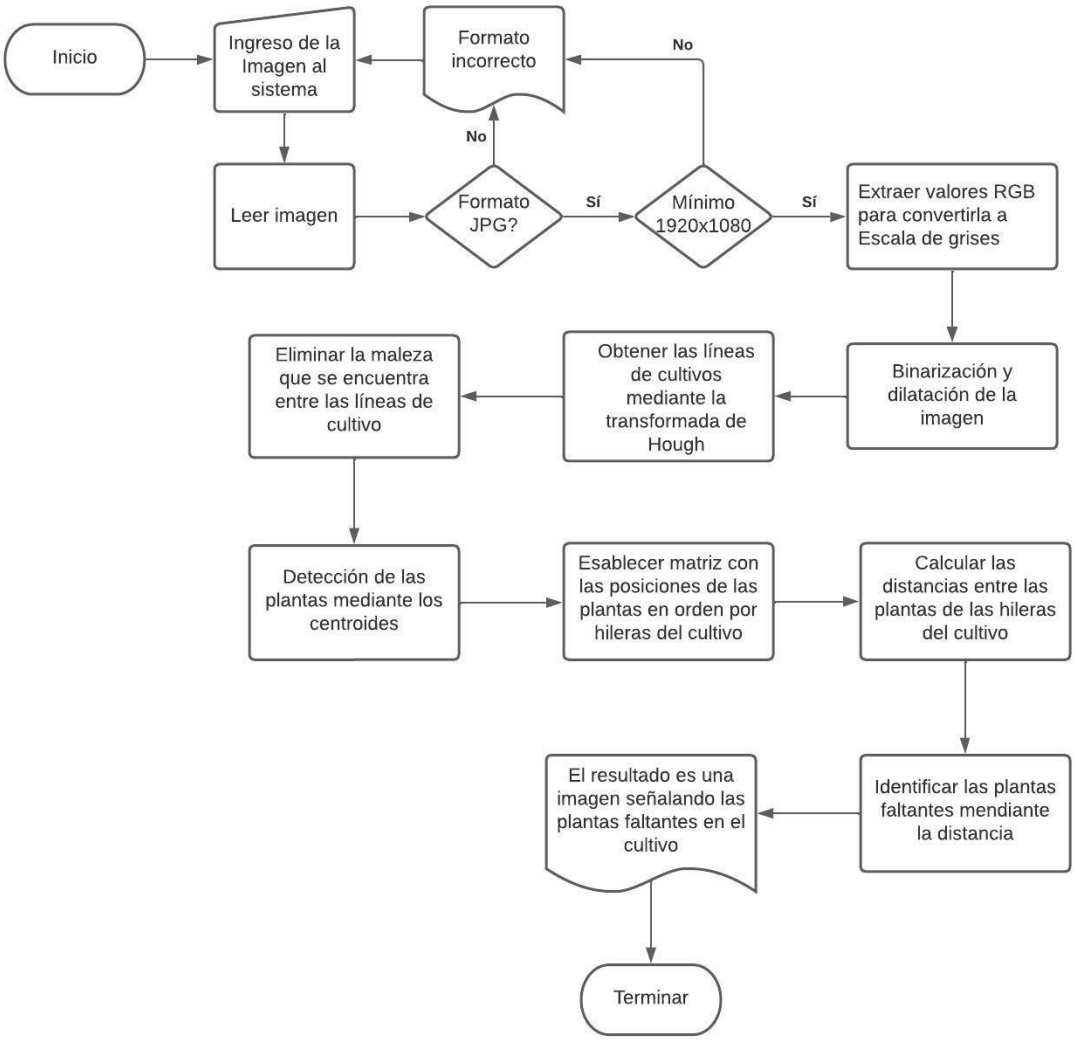


Fig. 8. Diagrama de flujo del algoritmo
Fuente: Propia

Además del diagrama de flujo del algoritmo, dentro del diseño también es importante la creación de los mockups sobre la interfaz de la aplicación. En las Fig. 9, Fig. 10 y Fig. 11 se muestra el diseño de la interfaz de la aplicación.

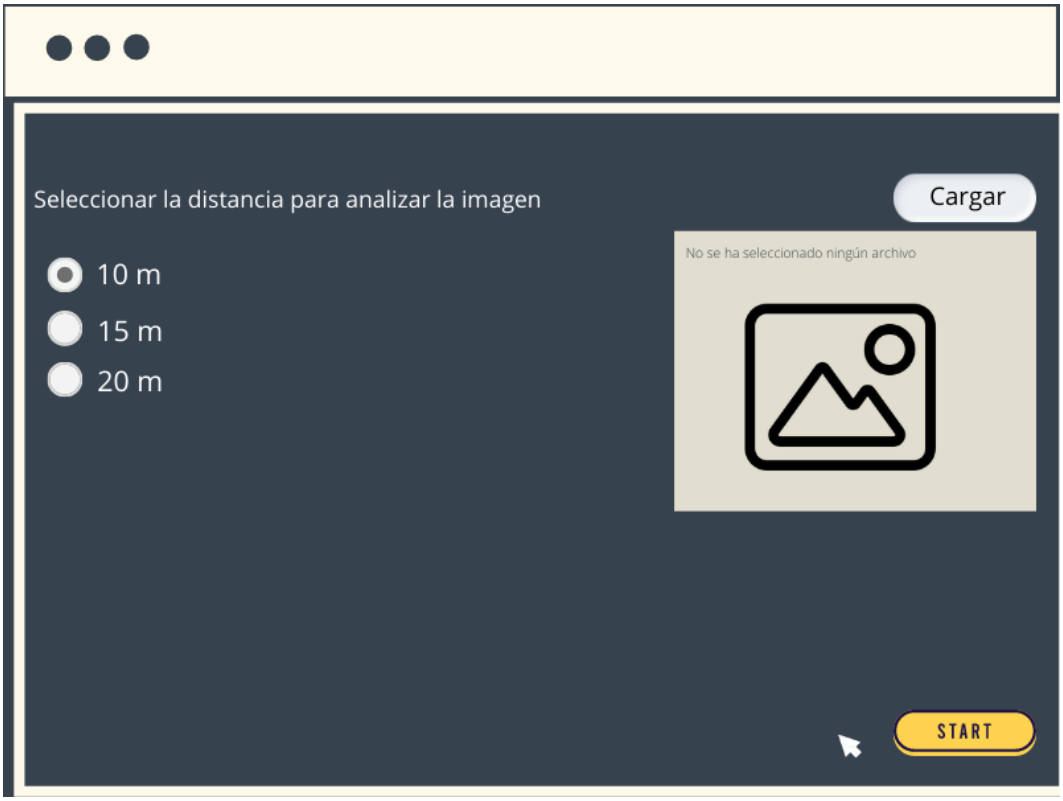


Fig. 9. Pantalla principal del aplicativo
Fuente: Propia

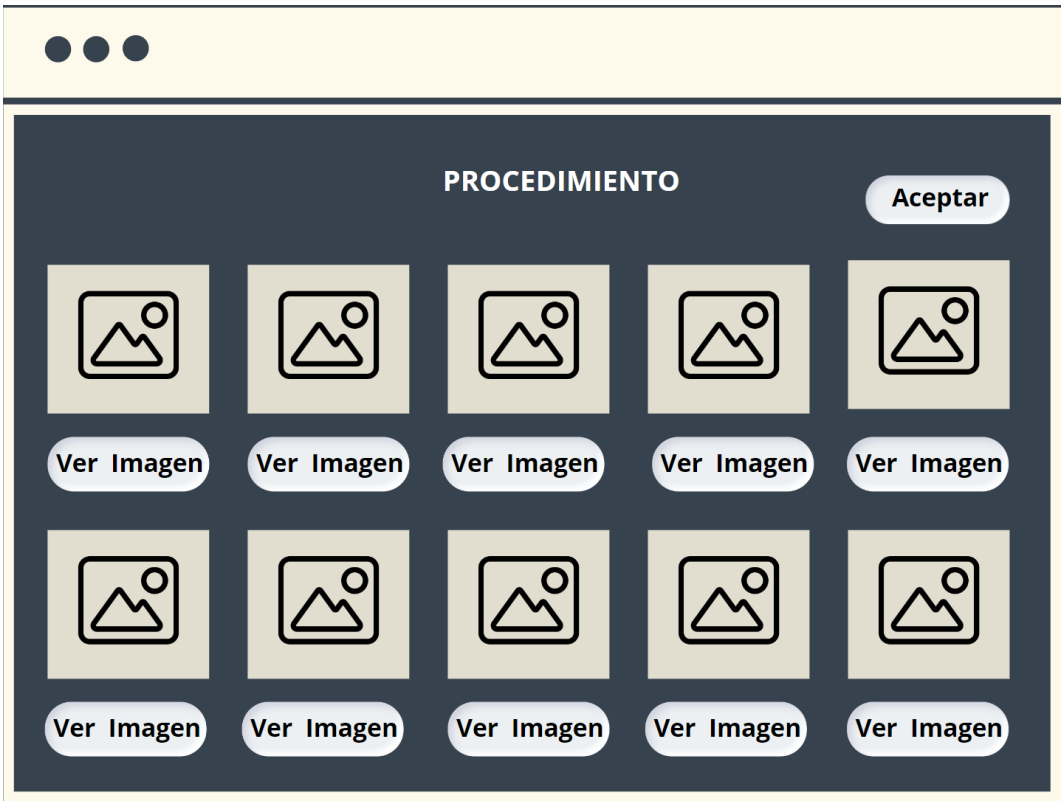


Fig. 10. Pantalla del procedimiento del aplicativo
Fuente: Propia

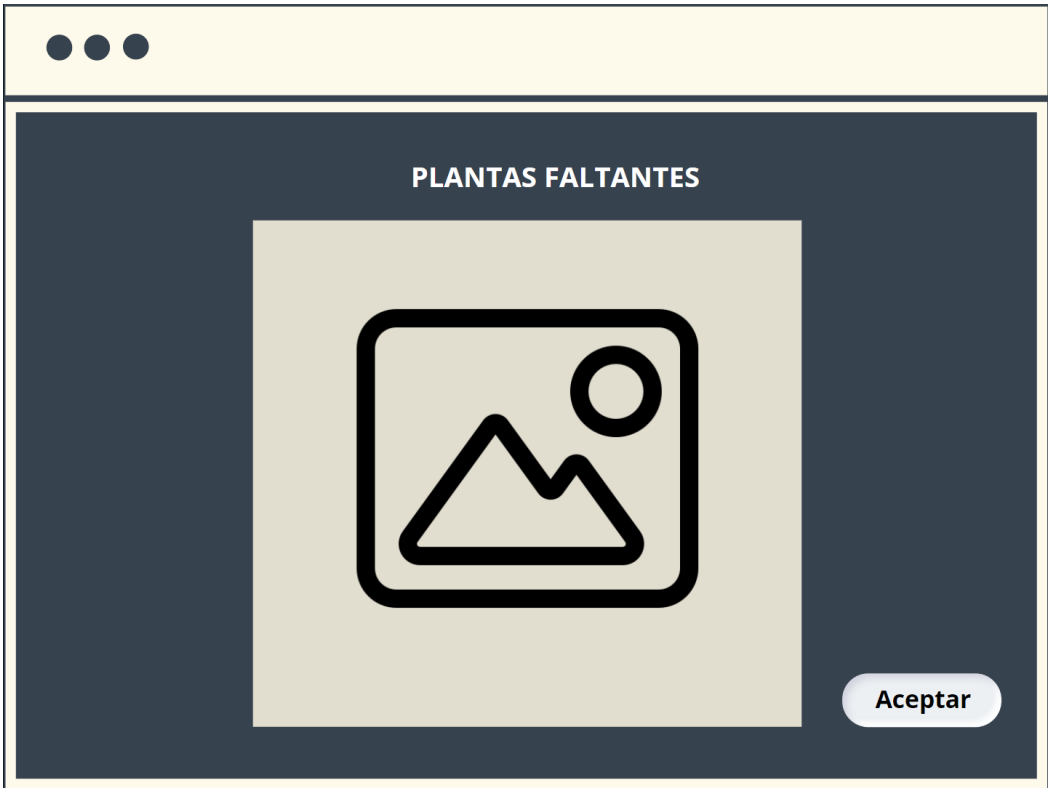


Fig. 11. Pantalla para la visualización de la imagen completa
Fuente: Propia

2.3 Implementación del algoritmo

En este apartado se explica a detalle el algoritmo que se desarrolló en Matlab para poder detectar las plantas faltantes en los cultivos de maíz y adicionalmente se dará a conocer las pantallas de la aplicación. A continuación, se describen las partes por la que el algoritmo se encuentra formado:

- Lectura de la imagen a analizar.
- Conversión de la imagen a escala de grises.
- Binarización y dilatación de la imagen.
- Detección de las líneas de cultivo mediante la transformada de Hough.
- Eliminación de la maleza entre las líneas de cultivo.
- Detección de las plantas
- Clasificación de las plantas detectadas por líneas de cultivos.
- Calcular las distancias entre las plantas de las hileras del cultivo.
- Identificar las plantas faltantes mediante las distancias.

Debido a que se utiliza una interfaz gráfica, el algoritmo propuesto se lo implementa por medio de una función, la cual se detalla a continuación:

```
function [Y,Ori,Exg,Bin, Dil,H1,Hou,Mal,
SinMal,Plan,PlanDetec,PlanDetec2] =
deteccion_faltantes_10m(Ruta)
```

La función mencionada anteriormente devuelve doce atributos los cuáles se describen de la siguiente manera:

- **Y:** Imagen que indica las plantas faltantes dentro del cultivo.
- **Ori:** Devuelve la imagen original, es decir la imagen que se ingresó para el análisis del algoritmo.
- **Exg:** Contiene la imagen en escala de grises.
- **Bin:** Es la imagen binarizada.
- **Dil:** Imagen dilatada aplicando la figura morfológica de disco y la función dilatar imopen.
- **H1:** Muestra los puntos de Hough en función de theta **(θ)** y rho **(ρ).**
- **Hou:** Devuelve la imagen dilatada con las líneas de cultivo identificadas mediante la transformada de Hough.
- **Mal:** Imagen que contiene la maleza detectada.
- **SinMal:** Imagen que contiene las plantas del cultivo sin la maleza detectada.
- **Plan:** Imagen que muestra solo los objetos que cumplan una determinada área.
- **PlanDetec:** Contiene las plantas identificas mediante el bounding box.
- **PlanDetec2:** Imagen que muestra las plantas detectadas por el bounding box y además identifica los centroides de cada planta por cada línea de cultivo.

Adicionalmente, la función necesita un parámetro denominado **Ruta,** la cual contiene la ruta de la imagen que se va a analizar con el algoritmo.

Es importante destacar que primero se explicará el algoritmo que detecta las plantas faltantes. Luego de esto se procederá a relatar acerca de las deferentes pantallas que tiene la aplicación.

Lectura de la imagen a analizar

Se procede a leer la imagen que se quiere analizar con el algoritmo. Para esto se declara una variable **path** a la cual se le pasa la dirección de la imagen. Seguidamente la variable **tStart** inicia un cronómetro para medir el tiempo de ejecución del algoritmo. Mediante el

comando **imread** se procede con la lectura de la imagen. En la Fig. 12 se muestra cómo se observa en pantalla la imagen que se envió a analizar por medio del algoritmo.

```
%Leer imagen
path=Ruta;
tStart = tic;
img = imread(path);
Ori=img;
```

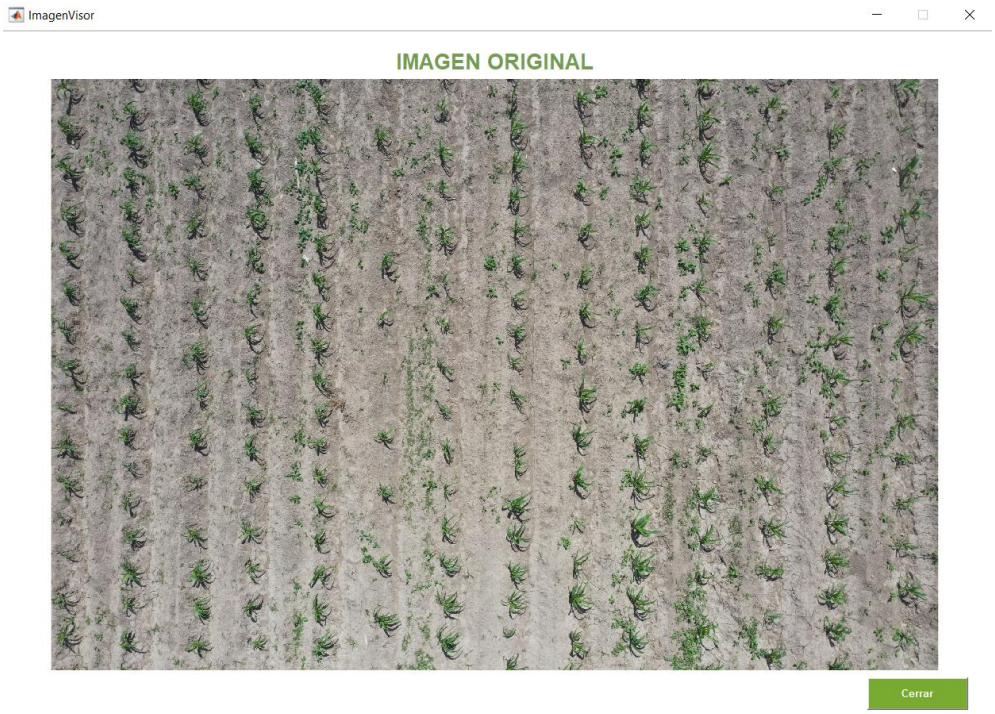


Fig. 12. Imagen original
Fuente: Propia

Conversión de la imagen a escala de grises

Las siguientes líneas de código se utiliza para extraer los valores RGB de la imagen, de tal manera que se pasó de una imagen RGB a una imagen en escala de grises. Para ello se separa la imagen por cada color que conforma RGB, es decir se tuvo la extracción de los colores rojo, verde y azul por separado. Luego se procedió a unir las tres variables en una sola variable **exg** para obtener la imagen en escala de grises. En la Fig. 13 se muestra la imagen en escala de grises.

```
%Extraer valores RGB
R = img(:,:,1);
G = img(:,:,2);
B = img(:,:,3);
exg = 2*(G)-(R)-(B);
Exg=exg;
```

Fig. 13. Imagen extrayendo RGB
Fuente: Propia

Binarización y dilatación de la imagen

Antes de explicar el código es importante conocer el algoritmo o método de Otsu, ya que es el método que se ocupa para la binarización.

Algoritmo de Otsu: Herranz & Aguirre (2020) explican que es un procedimiento el cual se basa en los cambios de similitud entre el fondo y el objeto. Este método utiliza técnicas estadísticas para resolver el problema, específicamente utiliza la varianza, la cual es una medida para la dispersión de valores, para este caso se mediría la dispersión de los niveles de gris. La dispersión es la distancia de los valores con respecto a un valor medio. Mediante la fórmula que se muestra a continuación se calcula el umbral para distinguir el fondo y el objeto:

$$\sigma_w^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t)$$

Una vez explicado el método de Otsu se procede a explicar el código. Primero se calculó el umbral de la imagen mediante **graythresh** el cual obtiene el umbral de una imagen en escala de grises mediante el método de Otsu. Luego se procedió a binarizar la imagen mediante **im2bw,** esto lo convierte en una imagen binaria basándose en el umbral. Los resultados se muestran en la Fig. 16.

```
%Aplicación umbral
umbral = graythresh(exg);
%Binarización de imagen
I = im2bw(exg, umbral);
Bin=I;
```

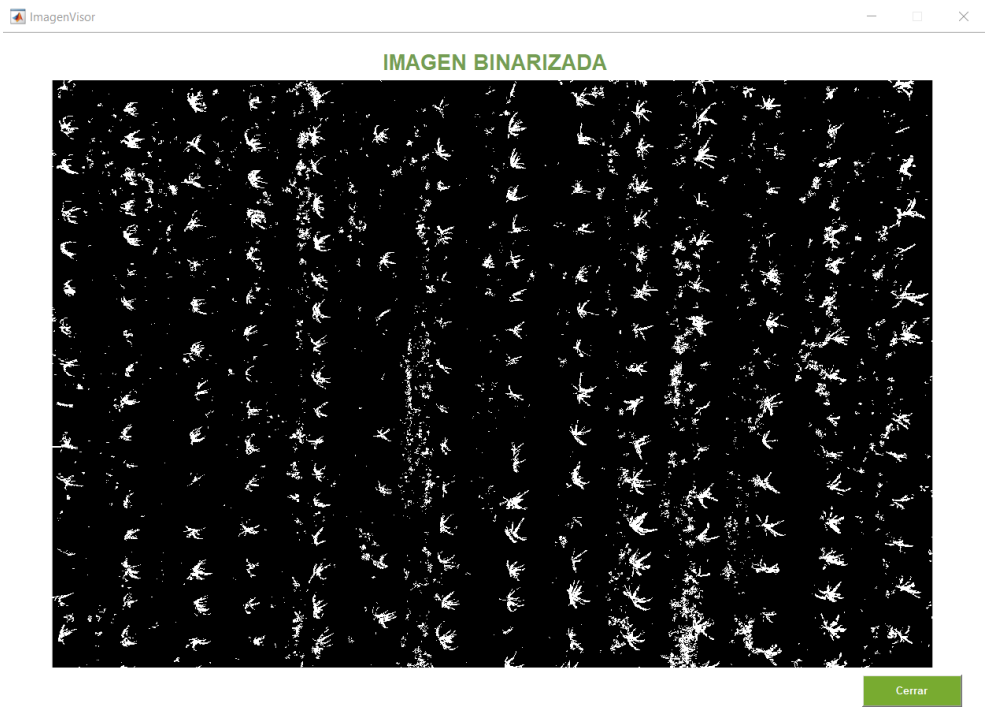


Fig. 14. Imagen Binarizada
Fuente: Propia

Cuando se obtiene la imagen binarizada, se procede a dilatar la imagen. Primero se explica lo que es un elemento estructurante morfológico y en qué consiste las operaciones morfológicas.

Elemento estructurante morfológico: En Matlab un elemento estructurante morfológico se representa mediante el objeto **strel**, el cuál es esencial para realizar una dilatación o erosión. Existen elementos de disco, línea, cuadrado y esfera. Por lo tanto, un elemento estructurante plano es un entorno que asume valores binarios, donde los píxeles que se incluyen dentro del elemento morfológico tienen un valor de true, por el contrario, los elementos que no se incluyen tienen un valor de false. Para esto se utiliza un píxel central denominado origen para identificar los píxeles de la imagen que se está procesando. (MathWorks, n.d.).

En la Fig. 15 se muestra una imagen que detalla la definición descrita anteriormente:

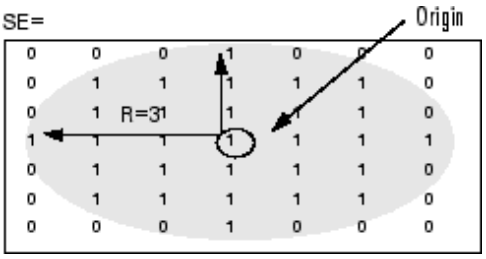


Fig. 15. Elemento estructurante morfológico de disco
Fuente: (MathWorks, n.d.)

Operaciones Morfológicas: Las operaciones morfológicas sirven para procesar las imágenes basándose en una forma geométrica. En una operación morfológica, cada píxel de la imagen se ajusta en función del valor de otros píxeles de su entorno. Con la selección de la forma y el tamaño del entorno adecuado, una operación morfológica puede ser más sensible a cierto tipo de imágenes. En Matlab existen numerosas operaciones morfológicas como imrode, imdilate, imopen, imclose, imtophat, imfill, etc. (MathWorks, n.d.).

El siguiente código muestra la creación de un elemento morfológico, en este caso es el elemento estructurante de disco con un radio de 4 píxeles. Luego se hace una apertura morfológica utilizando **imopen,** para esto se pasa la imagen binarizada con el elemento estructurando creado anteriormente. La apertura morfológica consiste en una erosión seguida de una dilatación utilizando el mismo elemento estructurante para ambas operaciones. Adicionalmente se muestra el resultado en una nueva ventana. La Fig. 21 presenta el resultado de la imagen dilatada.

```
%Valores para dilatación de imagen
se = strel('disk',4);
%Aplicación de la función dilatar
BW = imopen(I,se);
Dil=BW;
```

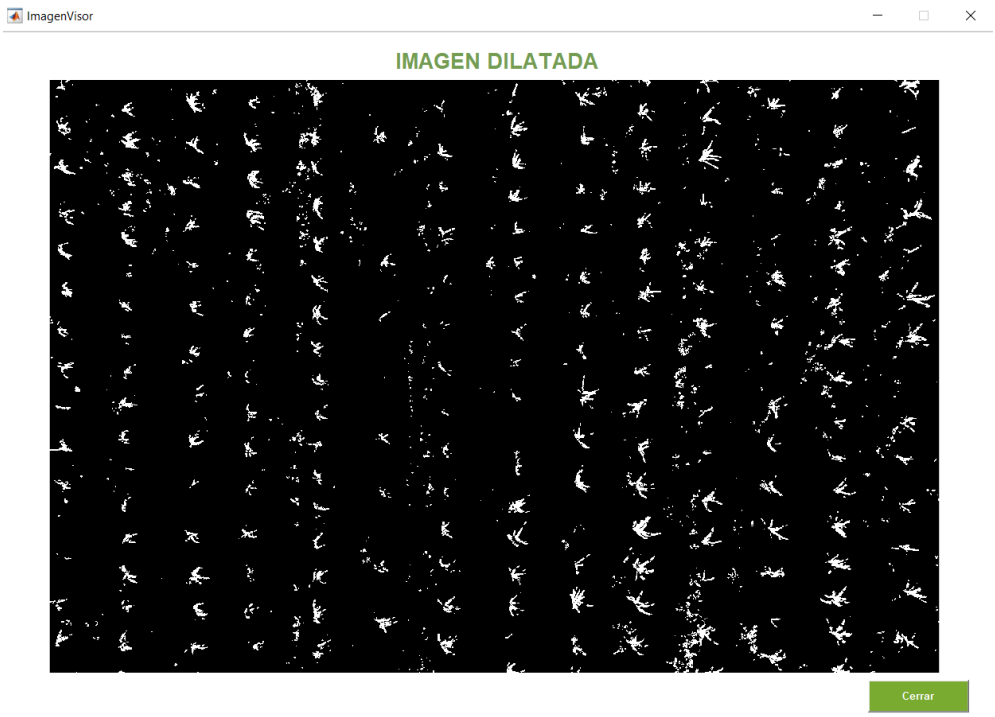


Fig. 16. Imagen dilatada con imopen
Fuente: Propia

Detección de las líneas de cultivo mediante la transformada de Hough

En esta sección del código se ocupa la transformada de Hough, la cual ayuda a detectar las hileras de cultivo de maíz dentro de la imagen.

Transformada de Hough: La transformada de Hough sirve para detectar rectas, por este motivo cómo surge la fórmula que ocupa Hough para la detección de las rectas partiendo desde la representación de la recta cuya fórmula es la siguiente:

$$y = m\mathrm{x} + \mathrm{b}$$

En donde m es la pendiente y se representa como delta de y sobre delta de x. Esto se puede definir como **b** que es la intersección con el eje **(y)** y **c** que es la intersección con el eje **(x)** como se muestra en la Fig. 17. Adicionalmente se pone el signo **(-)** debido a que la pendiente es negativa:

$$m = \frac{\Delta y}{\Delta x} = -\frac{b}{c}$$

Dentro de la Fig. 17 también se encuentran otro tipo de variables como rho **(ρ)** la cual representa la distancia mínima del origen a la recta, por medio de la perpendicular de la recta. Esta perpendicular forma un ángulo con respecto al eje (x), él cual se representa mediante la variable theta **(θ).**

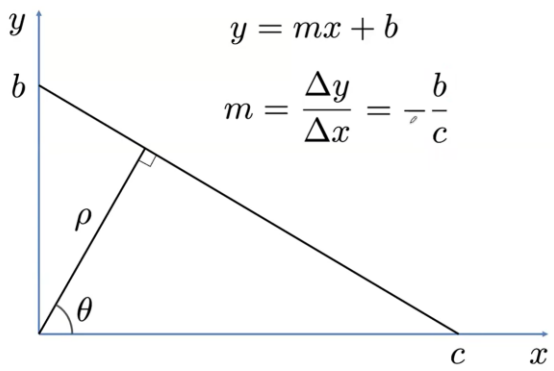


Fig. 17. Representación de la recta
Fuente: (Mery, 2020)

Mediante la trigonometría sacamos las representaciones de **(b)** y **(c)** con respecto al ángulo **(θ).** De esta manera tenemos lo siguiente:

$$b = \frac{\rho}{sin\theta} \qquad c = \frac{\rho}{cos\theta}$$

$$m = -\frac{b}{c} = -\frac{cos\theta}{sin\theta}$$

Una vez obtenido los valores de la pendiente (m), (b) y (c), se procede a reemplazar en la ecuación de la recta:

$$y = m\mathrm{x} + \mathrm{b}$$

$$y = -\frac{cos\theta}{sin\theta}\mathrm{x} + \frac{\rho}{sin\theta}$$

De esta manera obtenemos la ecuación de la recta que se ocupa para la transformada de Hough.

$$\rho = xcos\theta + ysin\theta$$

Ahora que se explicó el origen de la fórmula de la recta para la transformada de Hough, se procede a explicar cómo funciona Hough.

En la Fig. 18 se muestra dos gráficas. La primera gráfica representa un punto (x1, y1) en el plano cartesiano. La otra gráfica representa al mismo punto, pero en función de **(ρ)** y **(θ).** Es importante destacar que cada punto que conforma a la sigmoide de la segunda gráfica (**ρ** en función de **θ**) representa una recta que pasa por el punto (x1, y1). De tal manera que en Fig. 18 el punto que se señala en la sigmoide representa a la recta que se muestra en la primera gráfica en función de (y) y (x).

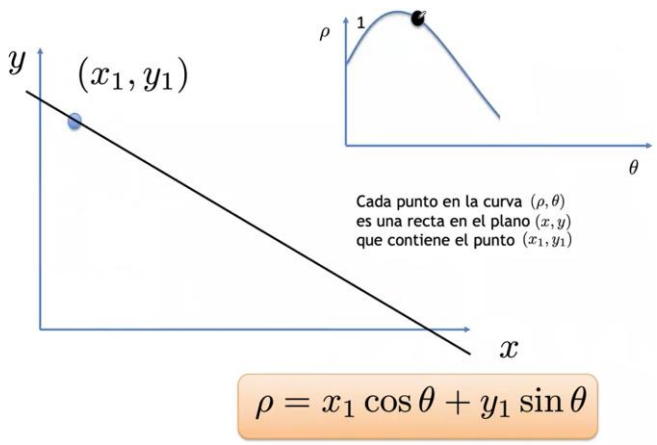


Fig. 18. Representación de las rectas de un punto por Rho en función de Theta
Fuente: (Mery, 2020)

Con esta aclaración, en la Fig. 19 se muestra en la primera gráfica tres puntos. En la segunda gráfica se muestran las tres sigmoides que representa a cada una de las rectas que pasan por los puntos. La intersección de las sigmoides que se muestra en la segunda gráfica nos indica que existe una recta que pasa por los tres puntos.

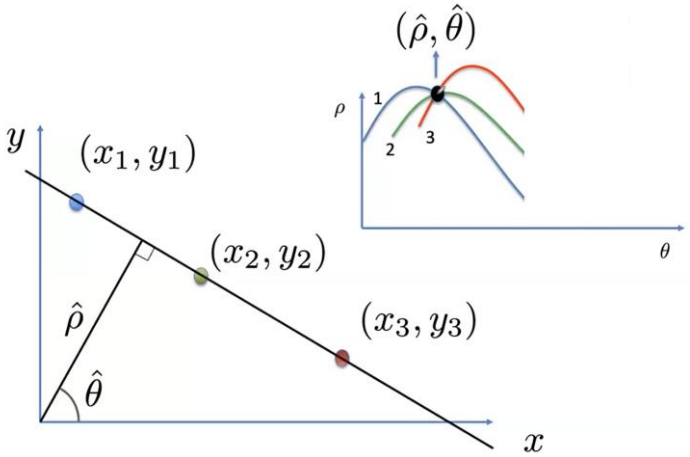


Fig. 19. Intersección de los puntos
Fuente: (Mery, 2020)

Ahora se procede a explicar el código en Matlab. [H,theta,rho] = hough(BW) calcula la transformada de Hough estándar (SHT) de la imagen binaria BW. La función Hough está diseñada para detectar rectas. La función utiliza la representación paramétrica de una recta: rho = x*cos(theta) + y*sin(theta). Esta función devuelve rho, la distancia desde el origen hasta la recta a lo largo de un vector perpendicular a la recta, y theta, el ángulo en grados entre el eje x y este vector. La función también devuelve la SHT, H, que es una matriz del espacio de parámetros cuyas filas y columnas corresponden a los valores rho y theta respectivamente. (MathWorks, n.d.). Los puntos obtenidos de las rectas se guardan en la variable **lines.** En este caso para la función de Hough se emplea un rango para el ángulo, que va desde -4 grados a 4 grados, en intervalos de 0.1 grados. Adicionalmente, en la Fig. 20 se muestra los

puntos de Hough, estos puntos se sacan mediante la función **houghpeaks**, en donde se indica que saque 50 puntos. Por otra parte, para sacar las líneas de cultivo mediante los puntos de Hough, se utiliza la función **houghlines,** en donde se especifica que solo saque líneas con una longitud mínima de 1800 píxeles. Adicionalmente se guarda la imagen donde se muestran los puntos de Hough para poder utilizarla en otra pantalla del programa.

```
%Obtener valores de hough
[H,theta,rho] = hough(BW,'Theta',-4:0.1:4);
H1 = figure, imshow(H,[],'XData',theta,'YData',rho,...
        'InitialMagnification','fit');
xlabel('\theta'), ylabel('\rho');
axis on, axis normal, hold on;
colormap(hot)
%Establecer houghpeaks
values = size(H);
nh2 = values(:,2);
if mod(nh2,2) == 0
    nh2 = values(:,2) - 3;
else
    nh2 = values(:,2) - 2;
end
P =
houghpeaks(H,50,'threshold',ceil(0.15*max(H(:))),'NHoodSize'
,[411 nh2]);
x = theta(P(:,2));
y = rho(P(:,1));
plot(x,y,'s','color','red');
lines =
houghlines(BW,theta,rho,P,'FillGap',1000,'MinLength',1800);
saveas(H1,'C:\Users\Ronald\Documents\Universidad\TESIS\DESAR
ROLLO\imagen1.jpg');
H1 =
imread('C:\Users\Ronald\Documents\Universidad\TESIS\DESARROL
LO\imagen1.jpg');
```

Una vez obtenidas los puntos de las líneas de Hough se procede a ordenarlas de izquierda a derecha por medio del método de ordenamiento de burbuja.

```
%Ordenar Estructura lines mediante el método burbuja
permutation = true;
iteracion = 0;
while(permutation == true)
    permutation = false;
    iteracion = iteracion + 1;
    for k = 1:(length(lines)-iteracion)
       if(lines(k).point1(1) > lines(k+1).point1(1))
          permutation = true;
          temp=lines(k);
```

```
            lines(k)=lines(k+1);
            lines(k+1)=temp;
        end
    end
end
```

Mediante dos ciclos **while** se sacan las líneas montadas, es decir, las líneas que se sobreponen encima de la otra, además se sacan las líneas que se encuentran muy cerca de los bordes de la imagen. De esta manera se disminuye la cantidad de líneas que no son relevantes para el análisis de las plantas faltantes.

```
%Eliminar líneas montadas y líneas cercas de los bordes
k=1;
a1 = size(BW);
while (k <=length(lines))
    j=1;
    while (j <=length(lines))
        if(lines(k).point1(1)~=lines(j).point1(1))
            if((abs(lines(k).point1(1)-
lines(j).point1(1))<200) || (lines(j).point1(1) < 70) ||
((a1(1,2) - lines(j).point1(1)) < 70))
                lines(j)=[];
                    if(j<k)
                        k=k-1;
                        j=j-1;
                    else
                        j=j-1;
                    end
            end
        end
        j=j+1;
    end
    k=k+1;
end
```

Luego se emplean dos ciclos while adicionales para poder sacar las líneas que se cruzan mediante un promedio entre los dos puntos de la recta.

```
%Sacar líneas que se cruzan por medio del promedio
k=1;
while (k <=length(lines))
    j=1;
    while (j <=length(lines))
        if(((lines(k).point1(1)-lines(j).point1(1))>=0 &&
(lines(k).point2(1)-lines(j).point2(1))<0) ||
((lines(k).point1(1)-lines(j).point1(1))<0 &&
(lines(k).point2(1)-lines(j).point2(1))>=0))
```

```
            puntox1 = round((lines(k).point1(1) +
lines(j).point1(1))/2);
            puntox2 = round((lines(k).point2(1) +
lines(j).point2(1))/2);
            lines(k).point1(1)=puntox1;
            lines(k).point2(1)=puntox2;
            lines(j)=[];
            if(j<k)
               k=k-1;
               j=j-1;
            else
               j=j-1;
            end
         end
         j=j+1;
     end
     k=k+1;
end
Hou = figure, imshow(BW),hold on
```

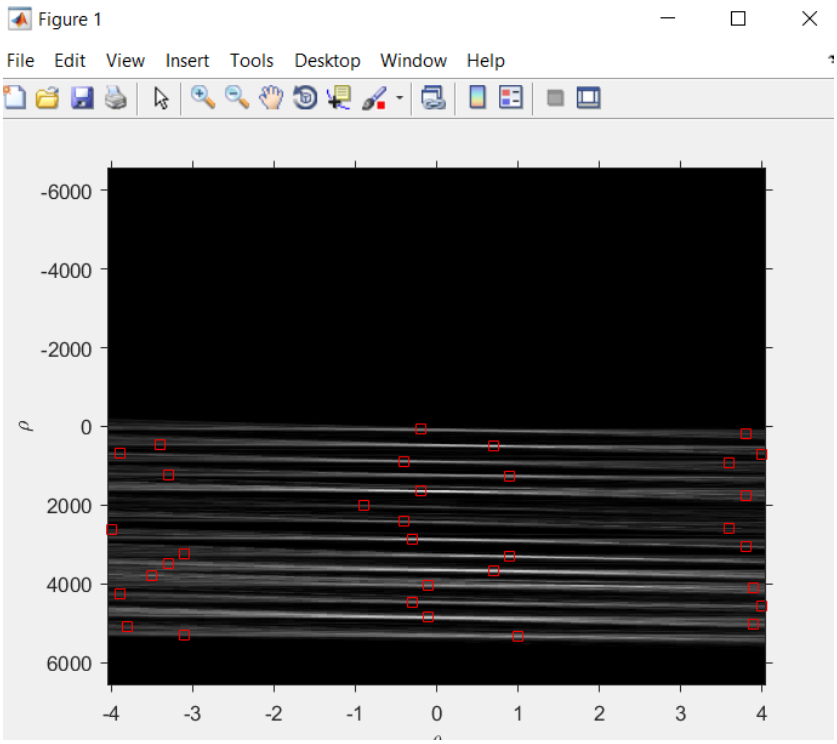


Fig. 20. Puntos de Hough
Fuente: Propia

Una vez obtenidos las líneas de Hough se crean dos rectas más (izquierda y derecha) a partir de la recta detecta por Hough. Para esto se copia la imagen binarizada anteriormente en una variable. Luego se crea dos arreglos para guardar los puntos de las líneas izquierda y derecha que se dan a partir de la de Hough. Para sacar la recta derecha simplemente sumamos 70 píxeles a los puntos de la recta de Hough. Por el contrario, para la línea izquierda se restan 70 píxeles a la recta de Hough. Una vez realizado este paso se procede a eliminar

todo lo que se encuentra entre la línea izquierda y derecha para obtener una imagen que tenga la maleza que se encuentran entra las líneas de cultivo. Esto se realiza para eliminar todas las líneas de cultivo de la imagen dejando solo la maleza. Para esto se asigna el valor de 0 a todo lo que se encuentre dentro de las líneas de cultivo, lo cual representa todo lo que se encuentra en la línea izquierda (azul) y derecha (verde). En la Fig. 21 se muestra el resultado de las líneas de Hough detectadas junto con las rectas izquierda y derecha, la cual se guarda en una carpeta específica para poder utilizarla en otra pantalla del sistema.

```
% Copia de la imagen
copy = BW;
%Guardar lineas izquierda y derecha
linLeft = [];
linRight = [];
for k = 1:length(lines)
   %Valores de xy de cada linea
   xy = [lines(k).point1; lines(k).point2];
   %Tamaño de la imagen
   a = size(BW);
   %Obtener línea derecha
   p1 = (xy(:,1));
   num1 = p1(1) + 70;
   num2 = p1(2) + 70;
   p1 = [num1 num2];
   p2 = (xy(:,2));
   num1 = p2(1);
   num2 = p2(2);
   p2 = [1 a(1)];
   linRight = cat (1, linRight,[[p1(1) p2(1)], [p1(2)
p2(2)]]);

   %Obtener línea izquierda
   p3 = (xy(:,1));
   num1 = p3(1) - 70;
   num2 = p3(2) - 70;
   p3 = [num1 num2];
   p4 = (xy(:,2));
   num1 = p4(1);
   num2 = p4(2);
   p4 = [1 a(1)];
   linLeft = cat (1, linLeft, [[p3(1) p4(1)], [p3(2)
p4(2)]]);

   %Establecer valores de 0 a la imagen
   array = [p1(:,1),p1(:,2),p3(:,1),p3(:,2)];
   for n1 = 1:a(1)
    for n2 = min(array):max(array)
        val = max(array) - min(array);
        if(n2 > 0 && val < 400 && n2 <= a(2))
```

```
                copy(n1,n2) = 0;
            end
        end
    end

    hold on
    plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','red');
    plot(p1,p2,'LineWidth',2,'Color','green');
    plot(p3,p4,'LineWidth',2,'Color','blue');
end
saveas(Hou,'C:\Users\Ronald\Documents\Universidad\TESIS\DESARROLLO\imagen2.jpg');
hold off
Hou =
imread('C:\Users\Ronald\Documents\Universidad\TESIS\DESARROLLO\imagen2.jpg');
```

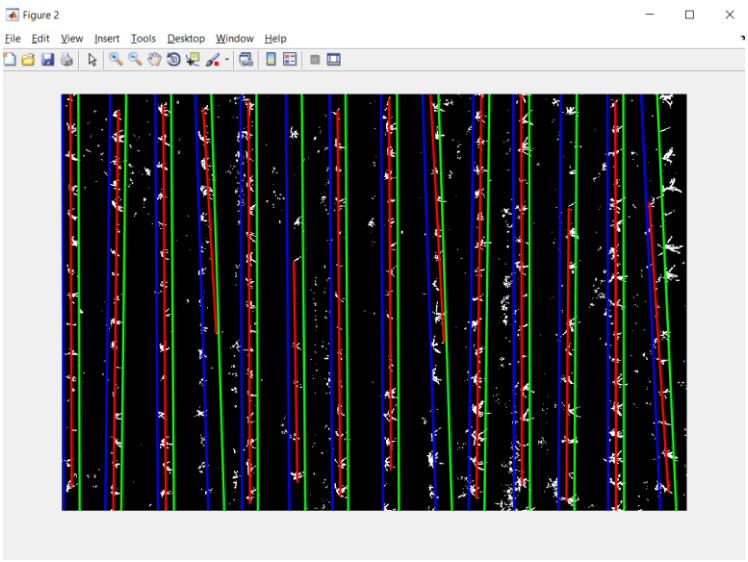


Fig. 21. Imagen con las líneas de Hough
Fuente: Propia

Eliminación de la maleza entre las líneas de cultivo

Gracias al código anterior se pudo detectar la maleza entre las líneas de cultivo. Para eliminar la maleza detectada, se procede a realizar una resta entre la imagen original binarizada y la imagen que se detectó la maleza, en este caso la imagen donde se detectó la maleza se encuentra en la variable **copy**. En Matlab la imagen binarizada se representa mediante una matriz de ceros y unos, donde uno es blanco y cero es negro. De esta manera se obtiene solo las líneas del cultivo eliminando el resto como se muestra en la Fig. 22. Adicionalmente se emplea el comando **bwareopen,** con el objetivo de eliminar objetos cuya área sea menor a los 50 píxeles, disminuyendo el ruido de la imagen.

```
%Elimino la maleza y solo dejo las plantas en la imagen binarizada;
ImagenPlants = BW - copy;
```

```
BW2 = bwareaopen(ImagenPlants, 50);
SinMal = BW2;
```

Fig. 22. Imagen sin Maleza
Fuente: Propia

Detección de las plantas

Para comenzar a realizar la detección de plantas se construye un elemento morfológico de disco de radio de 10 píxeles. Luego se aplica la dilatación de la imagen sin maleza para resaltar las figuras de las plantas en la imagen. Adicionalmente se hace un filtro tomar solo las áreas de interés que tengan entre 500 y 50000 píxeles de área. El resultado de la implementación de este código se muestra en Fig. 23.

```
%Detección de plantas
se2 = strel('disk',10);
final = imdilate(BW2,se2);
final = bwareafilt(final,[500 50000]);
hold on
Plan = final;
```

Fig. 23. Plantas Dilatadas
Fuente: Propia

Una vez que se han dilatado las plantas se procede a identificar las regiones de interés mediante **regionprops**, lo cual permite medir las propiedades de las diferentes regiones. En este caso las regiones son las plantas. La propiedad que mide primero es el área de cada región detectada. Las áreas detectadas se almacenan en el arreglo **areas**. Adicionalmente se declaran los siguientes arreglos: pos (guarda los puntos del boundingBox), cent (guarda los centroides de las regiones) y áreas (almacena las áreas de las regiones). Adicionalmente se crea una nueva figura de BW la cual representa a la imagen binarizada original, donde se sacará todas las plantas dejando únicamente la maleza.

```
%Identificar objetos
s=regionprops(final);
pos = [];
cent = [];
areas= [];
for k = 1 : length(s)
     thisBD = s(k).Area;
     areas = cat(1,areas, [thisBD(1)]);
end

%Sacar Maleza
Mal = figure,imshow(BW)
hold on
th = 0:pi/50:2*pi;
```

Mediante la función **mean** de Matlab se obtiene el promedio el cual es almacenado en la variable **promArea**. Se utilizó un ciclo para recorrer todas las regiones que se han identificado en la imagen binaria. A continuación, se inserta una condición para sacar el boundingBox de

las regiones cuya área sea mayor o igual al promedio de áreas previamente obtenido menos 2500. Con este número se obtuvo los mejores resultados para eliminar las regiones que no son de nuestro interés, ya que las regiones con menor área por lo general son las malezas que se encuentra entre las plantas del cultivo. Luego se obtiene el centroide (centro geométrico de una región) que se almacena en la variable **cent** y los cuatro puntos del boundingBox que se almacena en la variable **pos** de las regiones que cumplen la condición. Adicionalmente, se aprovecha el ciclo para sacar todas las plantas detectadas y dejar solo la maleza, para lo cual se utiliza las coordenadas del centro y se utiliza un radio de 60 para dibujar una especie de círculo de color negro en la imagen BW, la cual se creó una nueva figura en la variable Mal en el código anterior. De esta manera, cuando se dibuja el círculo negro sobre las plantas detectadas deja solo la maleza que se detectó en la imagen original. También se procede a guardar la imagen de la maleza detectada que se muestra en la Fig. 24 en una ruta específica para poder utilizarla en otras pantallas del programa.

```
%Sacar boundingBox de las áreas requeridas.
promArea= mean(areas);
for k = 1 : length(s)
  thisBD = s(k).Area;
  if (thisBD >= (promArea-2500))
      thisBB = s(k).BoundingBox;
      thisBC = s(k).Centroid;
      cent = cat(1,cent, [thisBC(1),thisBC(2)]);
      pos =
cat(1,pos,[thisBB(1),thisBB(2),thisBB(3),thisBB(4)]);
      %Esta parte corresponde a eliminar las plantas
      detectadas para dejar la maleza
      x_circle = 60 * cos(th) + thisBC(1);
      y_circle = 60 * sin(th) + thisBC(2);
      fill(x_circle, y_circle, 'k')

  end
end
saveas(Mal,'C:\Users\Ronald\Documents\Universidad\TESIS\D
ESARROLLO\imagen3.jpg');
Mal =
imread('C:\Users\Ronald\Documents\Universidad\TESIS\DESAR
ROLLO\imagen3.jpg');
hold off
```

Fig. 24. Imagen de maleza detectada por el sistema
Fuente: Propia

Con los puntos del boundingBox almacenados en la variable **pos**, se procede a dibujar el boundingBox de las regiones detectas mediante **insertObjectAnnotation** en la imagen original. Adicionalmente se especifica las propiedades del objeto que se va a insertar, en este caso la figura a insertar es un rectángulo que se forma mediante las coordenadas de la variable **pos**. Posteriormente se duplica la imagen en donde se detectó las plantas en una nueva variable Plantas. Esto se lo realiza con la finalidad de obtener más imágenes del procedimiento del algoritmo debido a que más adelante se insertarán más objetos a la imagen de la variable RGB. Por último, se muestran en ventanas distintas las imágenes obtenidas. En la Fig. 25 se muestra el resultado del código explicado.

```
%Imprimir imagen con plantas detectadas
RGB =
insertObjectAnnotation(img,'rectangle',pos,'Plantas',...

'FontSize',18,'Color','red','LineWidth',9,'TextBoxOpacity
',0.9);
Plantas = RGB;
PlanDetec = Plantas;
figure,imshow(Plantas)
PlanDetec2 = figure,imshow(RGB);
hold on
```

Fig. 25. Plantas Detectadas
Fuente: Propia

De igual manera en la Fig. 26 se muestra con más detalle cómo se identifica cada planta en la imagen:

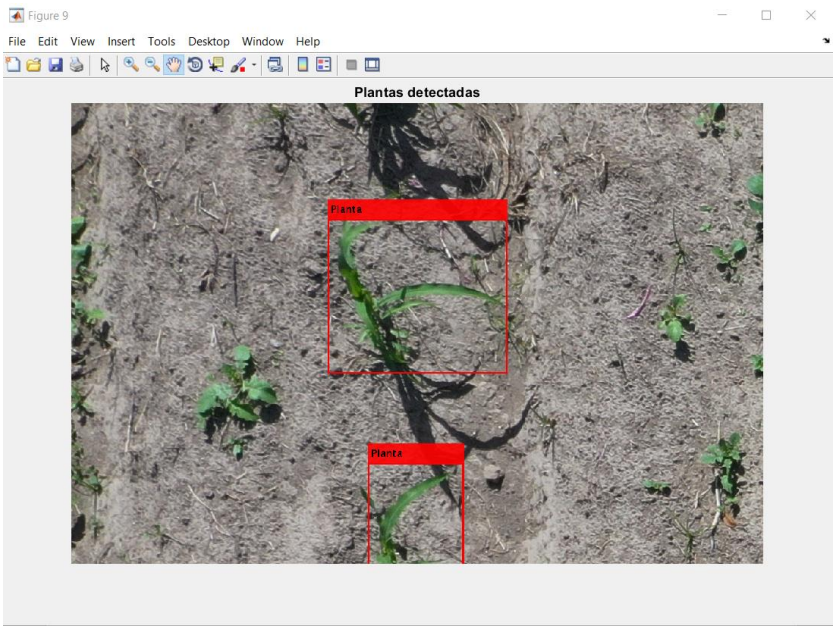


Fig. 26. Planta Detectada (Zoom)
Fuente: Propia

Clasificación de las plantas detectadas por líneas de cultivos

El siguiente código obtiene el número de objetos que se encuentra en el arreglo **cent**, el cual almacena los centroides de las regiones, lo cual representa las plantas identificadas.

```
%Obtener el número de objetos de la varible cent
```

```
centroSize= size(cent);
```

Se procede a ordenar las líneas izquierdas representadas por el color azul en la Fig. 21 (orden de izquierda a derecha) que se obtenían a partir de la transformada de Hough.

```
%Ordeno Fila izquierda de menor a mayor
[~, s] = sort(linLeft(:, 1));
linLeft = linLeft(s, :);
```

De igual forma se procede a ordenar las líneas derechas representadas por el color verde en la Fig. 21 (orden de izquierda a derecha) que se obtenían a partir de la transformada de Hough.

```
%Ordeno Fila derecha de menor a mayor
[~, s] = sort(linRight(:, 1));
linRight = linRight(s, :);
```

El siguiente código tiene la finalidad de ordenar los centroides de las plantas de menor a mayor con respecto al eje Y (eje vertical). Todos estos ordenamientos se los realizan con la finalidad de poder clasificar a las plantas por hileras, ya que hasta el momento se había identificado las plantas, pero no se encontraban clasificadas por cada una de las hileras del cultivo.

```
%Ordeno los centros de plantas de menor a mayor en Y
[~, s] = sort(cent(:, 2));
cent = cent(s, :);
centAux=cent;
```

Se procede a realizar la clasificación de las plantas detectadas por cada una de las hileras de cultivos. Para esto se declaró el arreglo de celdas **plants**, el cual va a contener a las plantas ordenadas. También se declara un contador **contRow** para poder ubicar las plantas dentro del arreglo. Para la clasificación se hace uso de dos ciclos. El primer ciclo consiste en recorrer cada una de las hileras de cultivos, por lo cual se utiliza la longitud del arreglo **lines** que se obtuvo anteriormente con la transformada de Hough. El segundo ciclo es usado para recorrer cada una de las plantas detectadas. Dentro del segundo ciclo se utiliza una condición que permite seleccionar a las plantas (centroides de las regiones) que se encuentre dentro de la primera hilera de cultivo. Para ello se establece que el centroide tiene que estar entre la línea izquierda y la línea derecha pertenecientes a la hilera de cultivo del ciclo.

Posteriormente, viene otra condición para clasificar las plantas en hileras pares e impares, para lo cual se utiliza el módulo 2. Si la planta se encuentra dentro de una hilera par, a la imagen RGB se le añade el centroide con el símbolo (*) de color azul. Por el contrario, si la hilera es impar a la imagen RGB se le añade el centroide con el símbolo (*) de color verde.

Esto se lo realiza con la finalidad de verificar que las plantas se hayan clasificado correctamente. El resultado de la clasificación de las plantas por hileras se muestra en la Fig. 27. Adicionalmente, se manda a imprimir el número de plantas detectadas en consola.

```
%Arreglo de celdas que guarda las posiciones de las
plantas por hileras
plants = {};
contRow= 0;
for k = 1:length(lines)
    for c=1:centroSize(1)
        if(((cent(c,1)>linLeft(k,1)) ||
((cent(c,1)>linLeft(k,3)))) &&
((cent(c,1)<linRight(k,1))||(cent(c,1)<linRight(k,3))))
            if(mod((k),2)==0)
                plot(cent(c,1),cent(c,2),'b*');
                contRow = contRow+1;
                plants{contRow,k} = [cent(c,1) cent(c,2)];
                cent(c,:)=0;
                %puntos = cat(1, puntos, [cent,k]);
            else
                plot(cent(c,1),cent(c,2),'g*');
                contRow = contRow+1;
                plants{contRow,k} = [cent(c,1) cent(c,2)];
                cent(c,:)=0;
            end
        end
    end
    contRow= 0;
end
imp = ['Plantas Detectadas: ',num2str(centroSize(1))];
disp(imp);
```

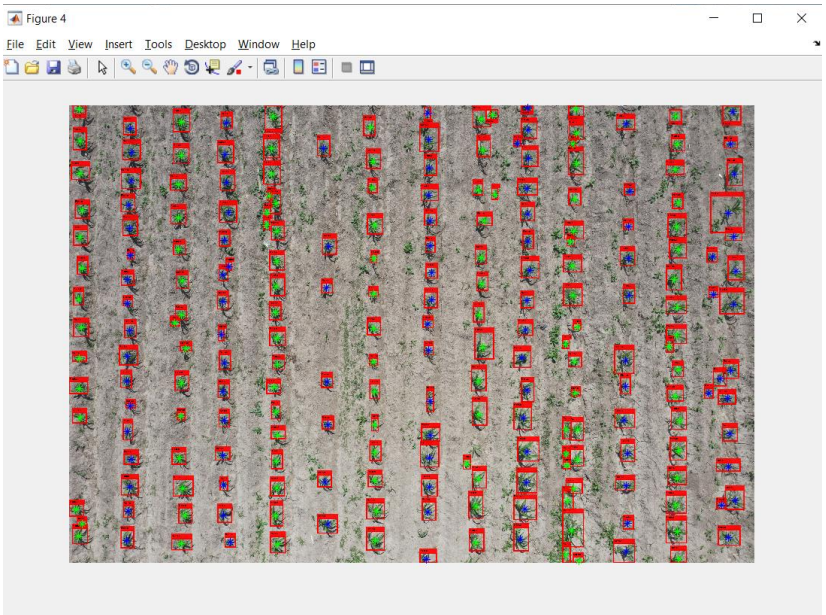


Fig. 27. Plantas clasificadas por hileras de cultivo
Fuente: Propia

Calcular las distancias entre las plantas de las hileras del cultivo

Es importante recordar que para la identificación de plantas faltantes dentro del cultivo de maíz se lo realiza mediante el cálculo de distancias que tienen entre las plantas de la misma hilera del cultivo. Por esta razón es importante identificar la distancia entre las plantas. Para ello se crea un arreglo de las mismas dimensiones que tiene el arreglo de celdas **Plants**, con la excepción de que el arreglo **distance** va a tener una fila adicional**.** Luego se establecen dos ciclos, uno para recorrer las columnas del arreglo Plants**,** y el otro ciclo se utiliza para recorrer las filas del arreglo Plants. Posteriormente se establece una condición para verificar si la fila aumentada en una posición (fila siguiente a la actual) se encuentra dentro de las dimensiones del arreglo Plants y si el arreglo ubicado en la fila aumentada en uno tiene algún objeto, es decir es diferente de 0. Si la condición se cumple, se procede a realizar la resta entre la planta de la siguiente fila con la planta de la fila actual con la finalidad de tener la distancia entre ambas. Una vez calculada la distancia, la misma se va almacenando en el arreglo **distance.** En caso de que no se cumpla la condición inicial se rompe el segundo ciclo para continuar con la siguiente columna. Adicionalmente se guarda la imagen de la Fig. 27 para poder utilizarla en otra pantalla del sistema.

```
%Identificar la distancia entre las plantas
sizePlants = size(plants);
distance=zeros(sizePlants(1,1), sizePlants(1,2));
for k = 1: sizePlants(1,2)
   for j = 1: sizePlants(1,1)
       if((j+1)<=sizePlants(1,1) &&
numel(plants{j+1,k})~=0)
            distance(j,k)=plants{j+1,k}(2)-
plants{j,k}(2);
       else
           break;
       end
   end
end
saveas(PlanDetec2,'C:\Users\Ronald\Documents\Universidad\
TESIS\DESARROLLO\imagen3.jpg');
PlanDetec2 =
imread('C:\Users\Ronald\Documents\Universidad\TESIS\DESAR
ROLLO\imagen4.jpg');
hold off
```

Una vez obtenido la distancia entre las plantas se procede a sacar la media de las distancias para obtener un promedio general. Para esto se rellena el arreglo **distance** con NaN en caso de que alguna distancia sea 0, esto se lo realiza con la finalidad de poder utilizar la operación **nanmean,** la cual sirve para sacar el promedio por cada una de las columnas

discriminando los valores NaN o nulos. Luego se procede a sacar un promedio general que se almacena en la variable **PromAll** (el parámetro omitnan se utiliza para omitir valores NaN).

```
%Promedio de distancias
distance(distance==0) = NaN;
Prom = nanmean(distance);
PromAll = mean(Prom,'omitnan');
```

Identificar las plantas faltantes mediante las distancias

Finalmente se procede a identificar las plantas faltantes mediante el promedio general de distancias obtenido anteriormente. Para ello se utiliza las dimensiones del arreglo **distance**, se instancia un arreglo **missPlants** el cual va a contener las coordenadas del boundingBox que se va a crear cuando exista una planta faltante. Para realizar esto se utilizan dos ciclos, el primero para recorrer las columnas del arreglo **distance**, por el contrario, el segundo ciclo itera las filas del arreglo **distance**. Luego se estable una condición para saber si el valor de la posición actual del arreglo **distance** no sea nulo. Si no es nulo pasa a la segunda condición, la cual se ocupa para verificar si se encuentra en el primer elemento de la hilera de cultivo. En caso de que sea la primera planta, se calcula la distancia entre el borde de la imagen superior hasta la primera planta detectada, esa distancia se almacena en la variable **FirstDistance,** la cual se utiliza para comparar con el promedio de las distancias para verificar si existe una planta faltante entre el borde superior de la imagen hasta la primera planta. Adicionalmente se establece las coordenadas del boundingBox si hubiera una planta faltante. En caso de que no sea la primera planta, se compara si el promedio general de distancia aumentado en 60 (valor que se obtuvo mediante algunas pruebas) es menor que la distancia actual. Esta condición se la realiza con la finalidad de saber si existe una distancia mayor al promedio, ya que si esta condición se cumple quiere decir que existe una planta faltante ya que se obtuvo una distancia de mayor longitud al promedio. Posteriormente se establece un boundingBox con ayuda de las coordenadas de las plantas de las que se sacó la distancia. En el **else** de la condición para saber si el valor de la distancia es nulo, se estable otra condición para comparar la última distancia de la hilera entre la última planta y el borde inferior de la imagen. Esta condición se establece cuando se encuentre un valor nulo dentro del arreglo de distancias ya que indica que se encuentra en la última fila de la columna. Igualmente, como se hizo con la primera distancia, se almacena en la variable **LastDistance** la distancia y se compara con la distancia promedio para identificar si existe una planta faltante. Adicionalmente se imprime en consola el número de plantas faltantes.

```
%Identificar las plantas faltantes mediante distancias
sizeDistance = size(distance);
missPlants = [];
contPf=0;
```

```
for k = 1: sizeDistance(1,2)
   for j = 1: sizeDistance(1,1)
       if (isnan(distance(j,k))~=1)
           if(j==1)
             FirstDistance = plants{j,k}(2)-0;
             if(PromAll+70 < FirstDistance)
               aux = FirstDistance/(PromAll+70);
               if(aux >= 2)
                   PosY = 1;
                   for n = 1:aux

missPlants=cat(1,missPlants,[plants{j,k}(1)-75, PosY ,
200 , PromAll]);
                       PosY = PosY + PromAll + 50;
                       contPf=contPf+1;
                   end
               else

missPlants=cat(1,missPlants,[plants{j,k}(1)-75, 1, 200 ,
FirstDistance - 150]);
                 contPf=contPf+1;
               end
             end
           end
           if((PromAll+70 < distance(j,k)))
               aux = distance(j,k)/(PromAll+70);
               if(aux >= 2)
                   PosY = plants{j,k}(2)+100;
                   for n = 1:aux

missPlants=cat(1,missPlants,[plants{j,k}(1)-75, PosY ,
200 , PromAll]);
                       PosY = PosY + PromAll + 50;
                       contPf=contPf+1;
                   end
               else

missPlants=cat(1,missPlants,[plants{j,k}(1)-75,
plants{j,k}(2)+100, 200 , distance(j,k)-150]);
                   contPf=contPf+1;
               end
           end
       else
           if(j~=1)
               LastDistance = a(1,1) - plants{j,k}(2);
               if(PromAll+70 < LastDistance)
                   aux = LastDistance/(PromAll+70);
                   if(aux >= 2)
                       PosY = plants{j,k}(2)+150;
                       for n = 1:aux
```

```
missPlants=cat(1,missPlants,[plants{j,k}(1)-75, PosY ,
200 , PromAll]);
                            PosY = PosY + PromAll + 50;
                            contPf=contPf+1;
                        end
                    else

missPlants=cat(1,missPlants,[plants{j,k}(1)-75,
plants{j,k}(2)+150, 200 , a(1,1)]);
                            contPf=contPf+1;
                    end
                end
            end
            break;
        end
    end
end
imp = ['Plantas Faltantes: ',num2str(contPf)];
disp(imp);
Y=img;
PlanDetec2=RGB;
```

Como último paso del algoritmo se inserta el boundingBox en la imagen original para identificar las plantas faltantes entre el cultivo de maíz. La imagen se manda a mostrar en una nueva ventana y se añade un título. Por último, en la variable **tElapsed** se almacena el tiempo en segundos de la duración de ejecución del algoritmo. En la Fig. 28 se muestra el resultado final del algoritmo.

```
%Imprimir imagen con plantas faltantes detectadas
if(~isempty(missPlants))
    MP =
insertObjectAnnotation(img,'rectangle',missPlants,'Planta
Faltante',...

'FontSize',18,'Color','green','LineWidth',9,'TextBoxOpaci
ty',0.9);
    figure,imshow(MP);
    title('Plantas faltantes');
    Y=MP;
end
hold off
tElapsed = toc(tStart); % en segundos
disp 'Tiempo total (segundos): ', tElapsed
end
```

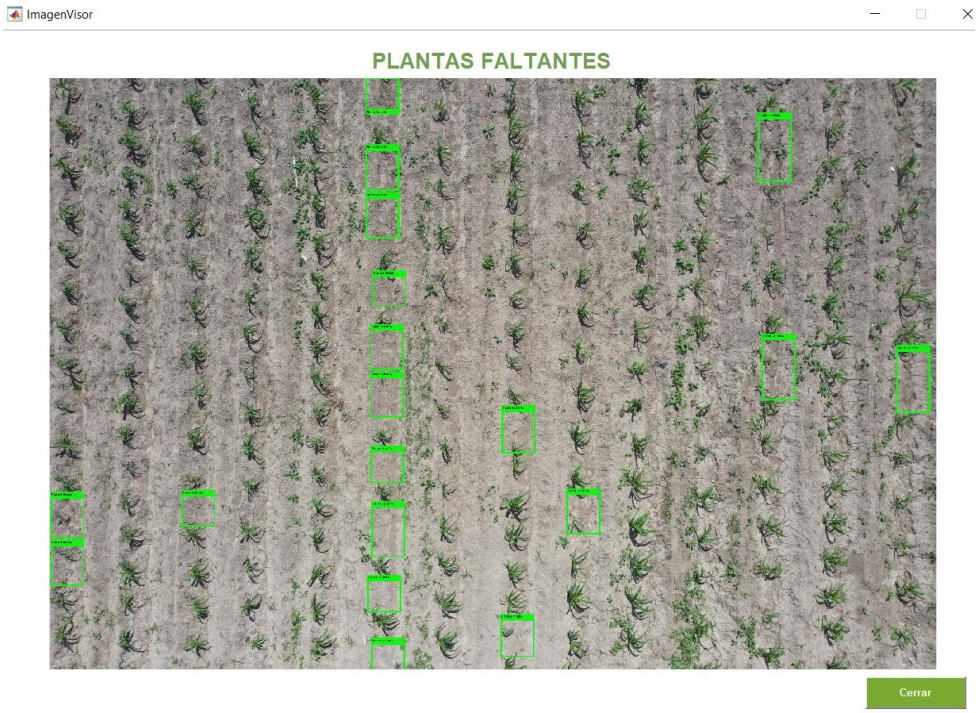


Fig. 28. Plantas faltantes
Fuente: Propia

Ahora que se ha explicado en detalle el algoritmo para detectar las plantas faltantes dentro de los cultivos de maíz, se procederá a explicar cómo se encuentra conformada la interfaz del aplicativo.

Interfaz de la Aplicación

La aplicación contine tres pantallas, las cuáles son:

- Pantalla de carga de imagen y ejecución del algoritmo
- Pantalla del procedimiento donde muestra el grid de las imágenes que se obtuvo con el algoritmo.
- Pantalla de visualización de imagen en pantalla completa.

En la primera pantalla que aparece al ejecutar la aplicación, se tiene que escoger la altura a la que se tomó la imagen con el dron. Cabe recalcar que el algoritmo funciona para 5m, 10m y 15m de altura. Por cada altura se ejecuta un algoritmo diferente, por esta razón se debe seleccionar la altura correspondiente a la que se capturó la imagen con el dron. Cuando seleccione la altura el siguiente paso es cargar la imagen a procesar. Se debe dar clic en el botón **Cargar Imagen** donde se abrirá su explorador de archivos para que seleccione la imagen. En la Fig. 29 se muestra la pantalla de inicio de la aplicación.

Fig. 29. Pantalla de inicio
Fuente: Propia

Cuando se realice la carga de la imagen se podrá observar una previsualización de la imagen cargada en la aplicación como se muestra en la Fig. 30.

Fig. 30. Pantalla cargada la imagen
Fuente: Propia

Luego de ejecutar la aplicación, aparecerá una pantalla mostrando del procedimiento de la imagen, donde se podrá tener un control sobre las distintas etapas del procedimiento como se muestra en la Fig. 31.

Fig. 31. Pantalla de procesamiento
Fuente: Propia

Para ver una imagen en pantalla completa, debe dar clic en el botón **Ver Imagen** de la pantalla de Procedimiento. Luego se desplegará la pantalla de visualización de imagen como se muestra Fig. 32.

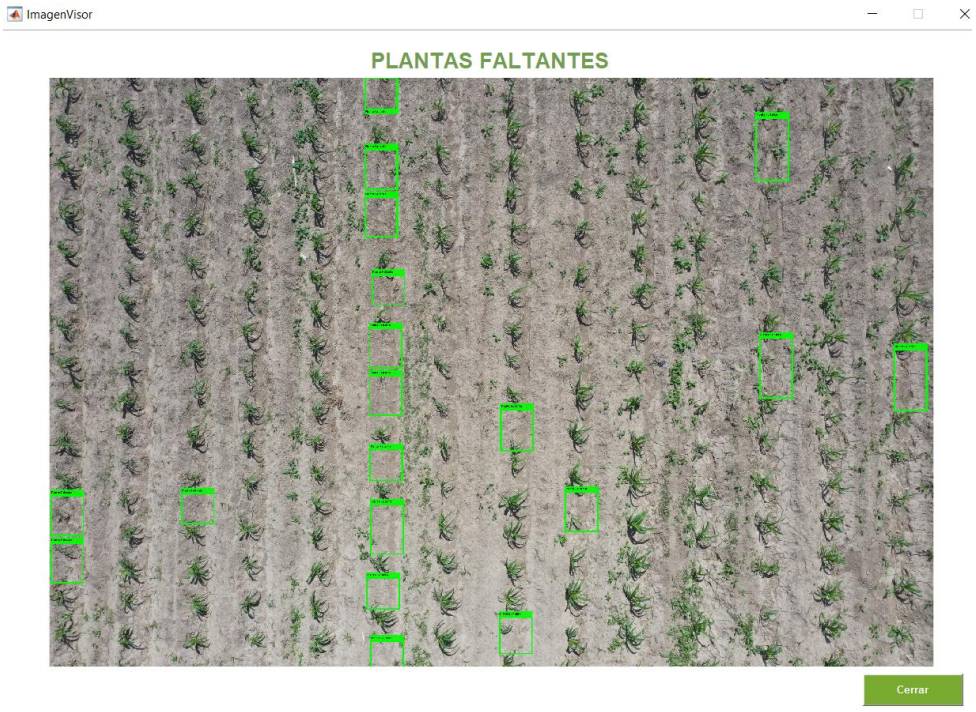


Fig. 32. Pantalla de visualización de imagen
Fuente: Propia

2.4 Pruebas con imágenes a distintas alturas

Las imágenes con las que se probó el algoritmo se clasifican en imágenes de 5, 10 y 15 metros de altura con respecto al suelo. Adicionalmente las plantas con las que se hizo las

pruebas son de aproximadamente 4 semanas. Es importante destacar que se probó el algoritmo con 5 imágenes por cada altura.

2.4.1 Prueba con imágenes a 5m de altura

Para las pruebas con imágenes de 5m de altura, se utilizaron 5 imágenes, las cuales se describen a continuación:

Imagen 1

En la Fig. 33 se muestran las líneas de cultivo detectadas por el sistema para la imagen 1 por medio de la transformada de Hough:

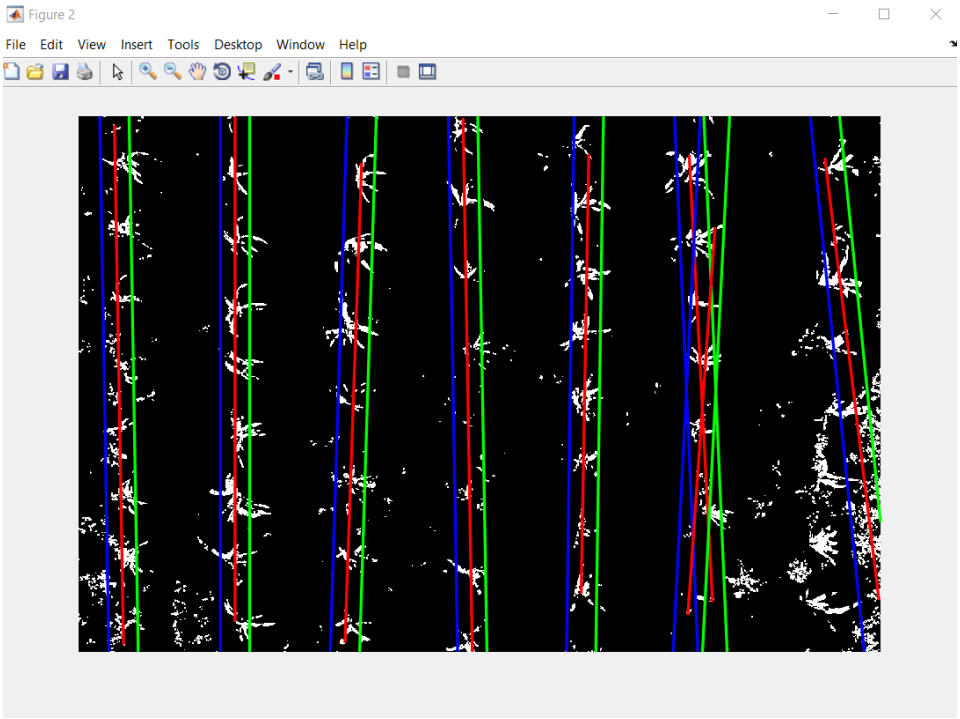


Fig. 33. Líneas de cultivo detectadas en la imagen 1 (5m)
Fuente: Propia

En la Fig. 34 se muestra las plantas de maíz detectadas por el sistema:

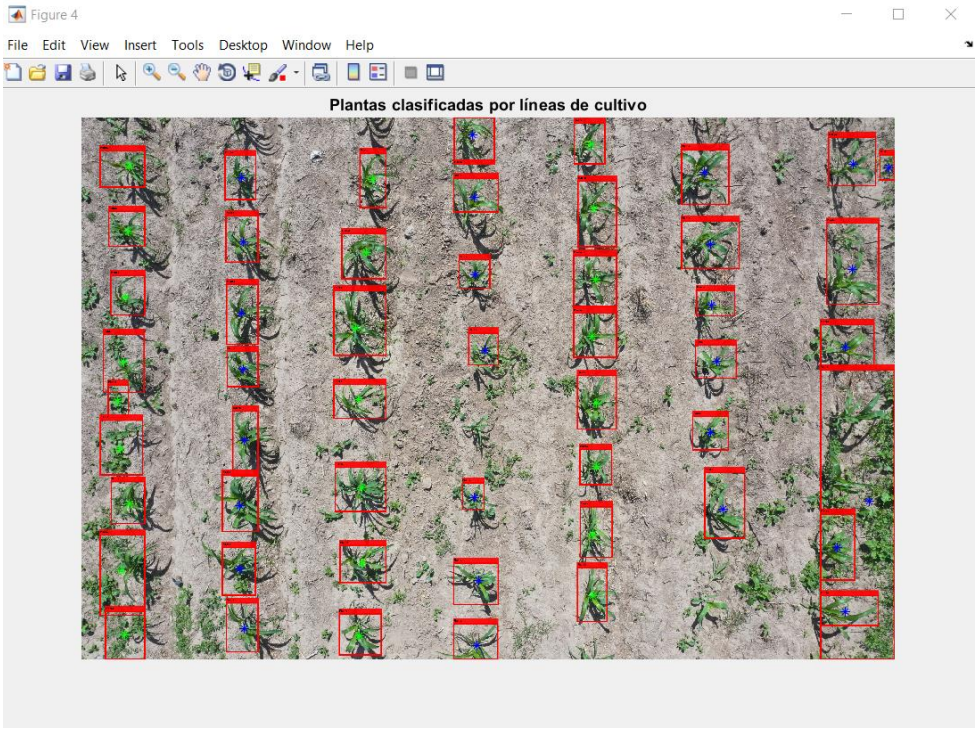


Fig. 34. Plantas detectadas en la imagen 1 (5m)
Fuente: Propia

El resultado de la imagen 1 se muestra en la Fig. 35 donde se muestra las plantas faltantes detectadas por el sistema:

Fig. 35. Plantas faltantes detectadas en la imagen 1 (5m)
Fuente: Propia

Para las imágenes 2, 3, 4 y 5 se siguió el mismo proceso que la imagen 1. En la TABLA 10 muestra los resultados que se obtuvieron del sistema con el algoritmo para las 5 imágenes correspondientes a la altura de 5m y en la TABLA 11 se muestran los resultados detectados de forma manual:

TABLA 10
RESULTADOS DE LAS IMÁGENES CON 5M DE ALTURA (SISTEMA)

Descripción	Líneas de cultivo detectadas	Plantas detectadas	Plantas faltantes detectadas
Imagen 1	7	52	3
Imagen 2	7	54	0
Imagen 3	7	56	3
Imagen 4	6	44	2
Imagen 5	8	63	0

Fuente: Propia

TABLA 11
RESULTADOS DE LAS IMÁGENES CON 5M DE ALTURA (MANUAL)

Descripción	Líneas de cultivo detectadas	Plantas detectadas	Plantas faltantes detectadas
Imagen 1	7	53	0
Imagen 2	7	53	0
Imagen 3	7	51	3
Imagen 4	6	44	2
Imagen 5	7	54	0

Fuente: Propia

2.4.2 Prueba con imágenes a 10m de altura

Para las pruebas con imágenes de 10m de altura, se utilizaron 5 imágenes:

Imagen 1

En la Fig. 36 se muestran las líneas de cultivo detectadas por el sistema para la imagen 1 por medio de la transformada de Hough:

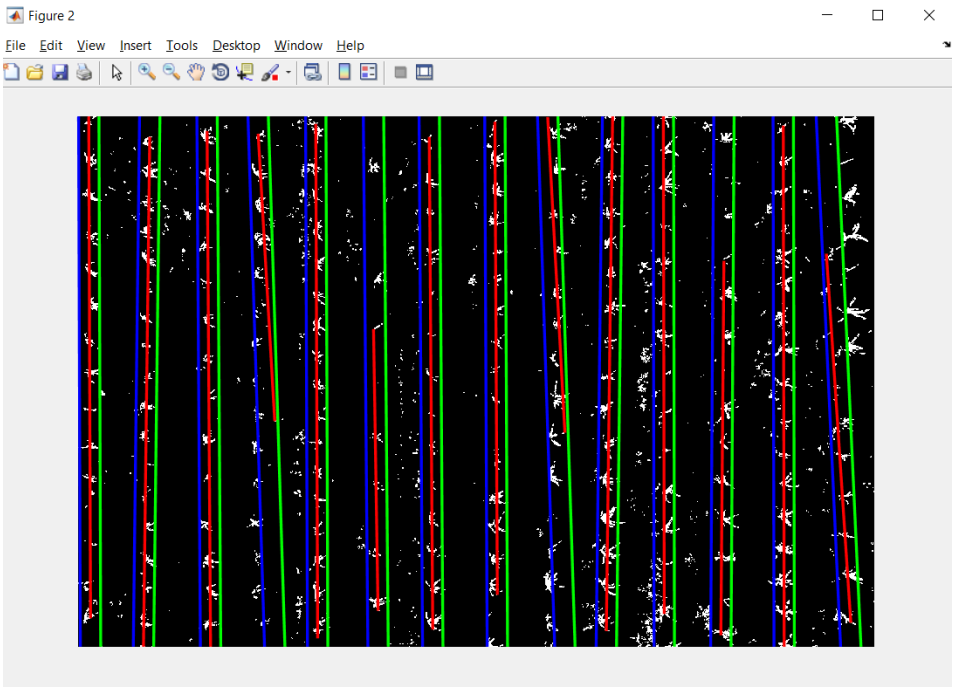


Fig. 36. Líneas de cultivo detectadas en la imagen 1 (10m)
Fuente: Propia

En la Fig. 37 se muestra las plantas de maíz detectadas por el sistema:

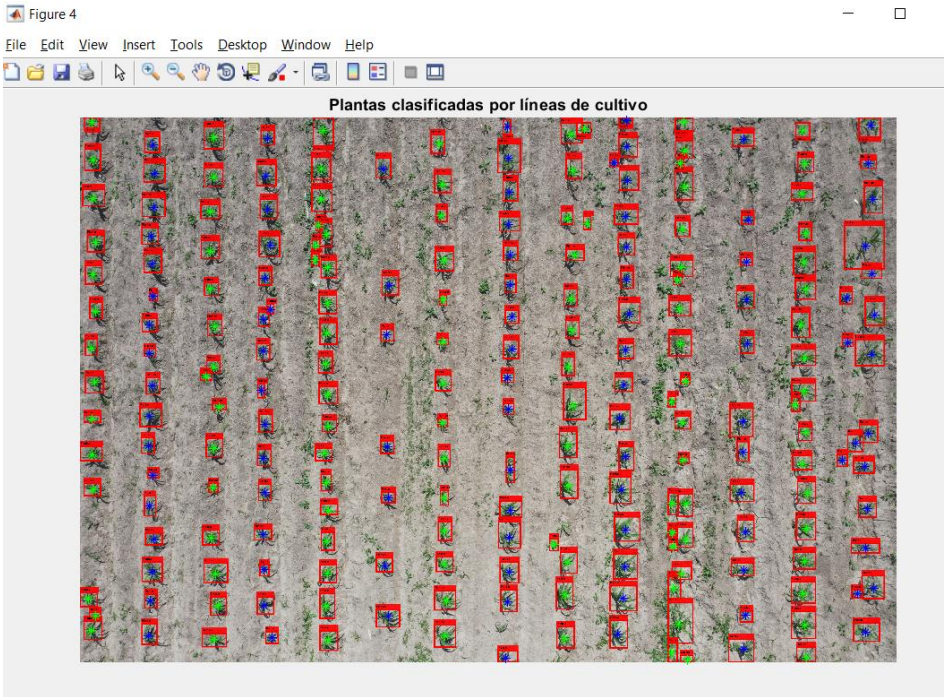


Fig. 37. Plantas detectadas en la imagen 1 (10m)
Fuente: Propia

El resultado de la imagen 1 se muestra en la Fig. 38 donde se muestra las plantas faltantes detectadas por el sistema:

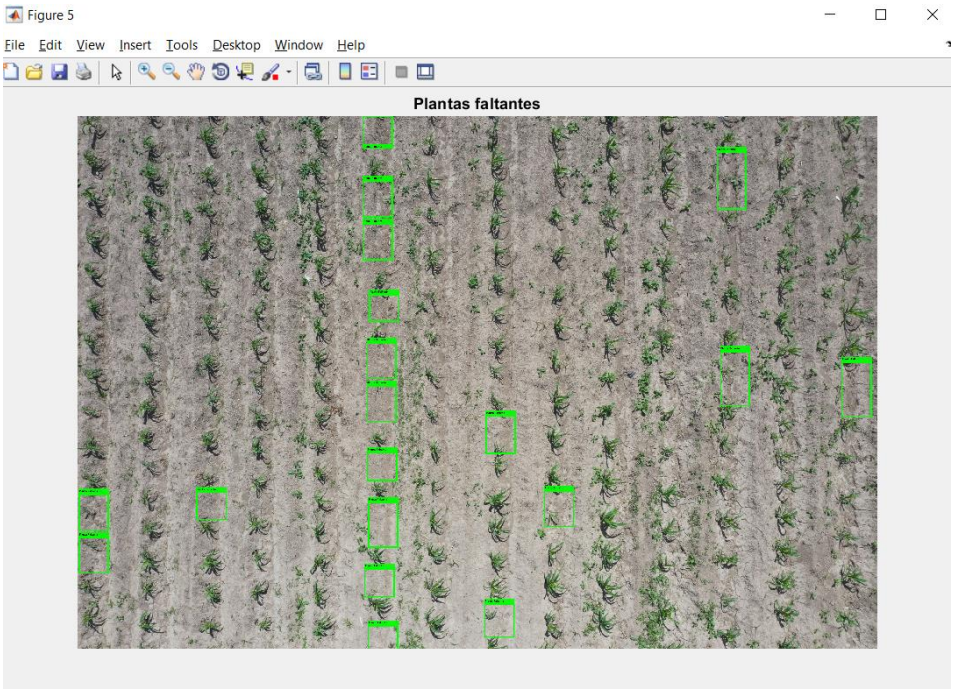


Fig. 38. Plantas faltantes detectadas en la imagen 1 (10m)
Fuente: Propia

Para las imágenes 2, 3, 4 y 5 se siguió el mismo proceso que la imagen 1. En la TABLA 12 se muestran los resultados que se obtuvieron del sistema para las 5 imágenes correspondientes a la altura de 10m y en la TABLA 13 se muestran los resultados detectados de manera manual:

TABLA 12
RESULTADOS DE LAS IMÁGENES CON 10M DE ALTURA (SISTEMA)

Descripción	Líneas de cultivo detectadas	Plantas detectadas	Plantas faltantes detectadas
Imagen 1	14	223	19
Imagen 2	14	319	44
Imagen 3	15	285	33
Imagen 4	14	308	37
Imagen 5	12	210	28

Fuente: Propia

TABLA 13
RESULTADOS DE LAS IMÁGENES CON 10M DE ALTURA (MANUAL)

Descripción	Líneas de cultivo detectadas	Plantas detectadas	Plantas faltantes detectadas

Imagen 1	14	197	17
Imagen 2	14	213	15
Imagen 3	12	185	19
Imagen 4	12	181	24
Imagen 5	13	186	19

Fuente: Propia

2.4.3 Prueba con imágenes a 15m de altura

Para las pruebas con imágenes de 15m de altura, se utilizaron 5 imágenes:

Imagen 1

En la Fig. 39 se muestran las líneas de cultivo detectadas por el sistema para la imagen 1 por medio de la transformada de Hough:

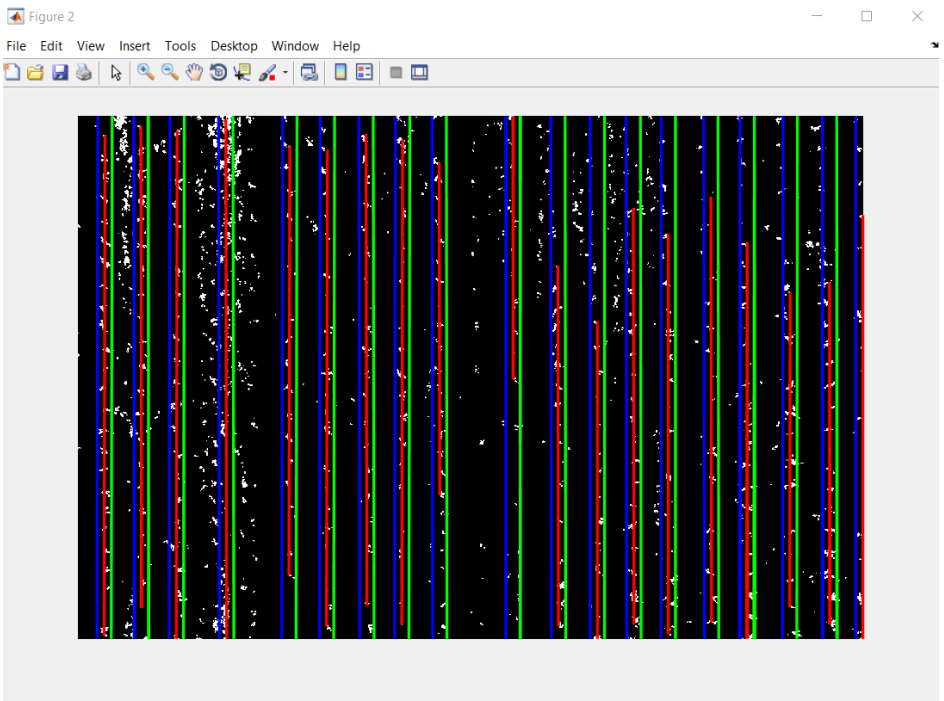


Fig. 39. Líneas de cultivo detectadas en la imagen 1 (15m)
Fuente: Propia

En la Fig. 40 se muestra las plantas de maíz detectadas por el sistema:

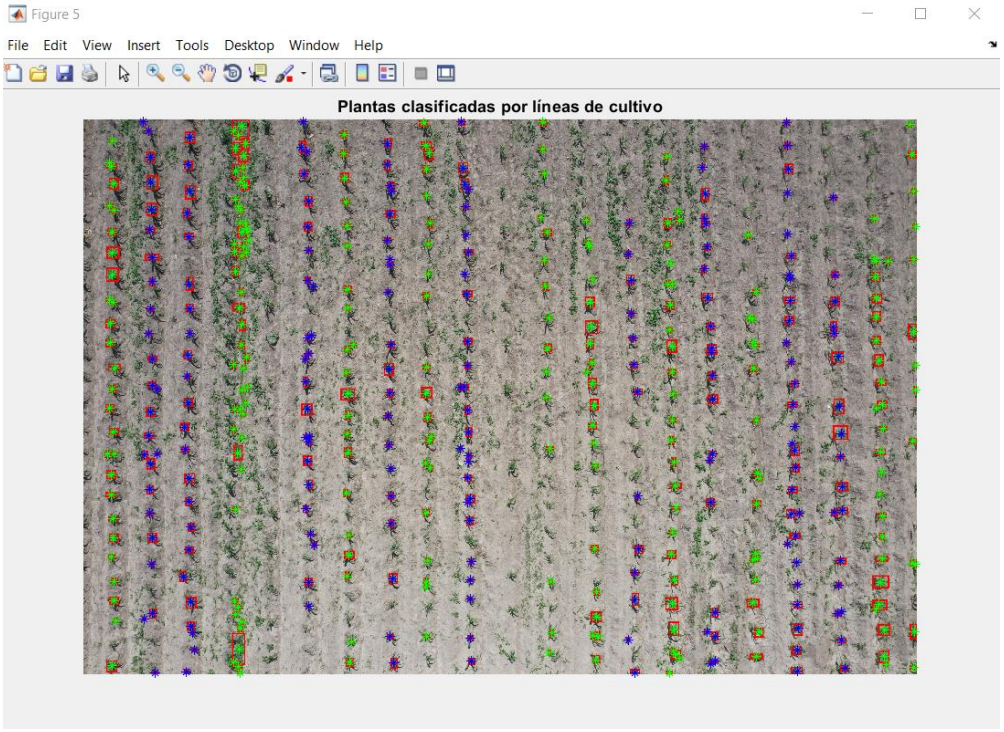


Fig. 40. Plantas detectadas en la imagen 1 (15m)
Fuente: Propia

El resultado de la imagen 1 se muestra en la Fig. 41 donde se muestra las plantas faltantes detectadas por el sistema:

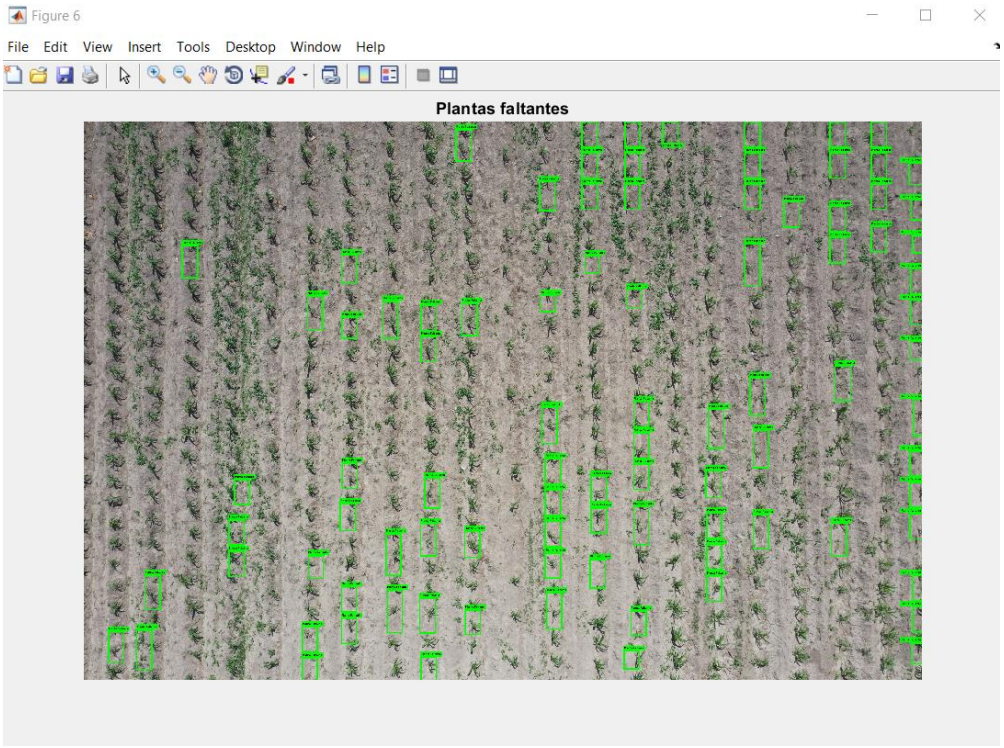


Fig. 41. Plantas faltantes detectadas en la imagen 1 (15m)
Fuente: Propia

Para las imágenes 2, 3, 4 y 5 se siguió el mismo proceso que la imagen 1. En la TABLA 14 se muestran los resultados que se obtuvieron por el sistema para las 5 imágenes correspondientes a la altura de 15m y en la TABLA 15 se muestran los resultados que se obtuvieron de forma manual:

TABLA 14
RESULTADOS DE LAS IMÁGENES CON 15M DE ALTURA (SISTEMA)

Descripción	Líneas de cultivo detectadas	Plantas detectadas	Plantas faltantes detectadas
Imagen 1	18	425	87
Imagen 2	21	482	89
Imagen 3	20	554	111
Imagen 4	18	425	87
Imagen 5	21	482	89

Fuente: Propia

TABLA 15
RESULTADOS DE LAS IMÁGENES CON 15M DE ALTURA (MANUAL)

Descripción	Líneas de cultivo detectadas	Plantas detectadas	Plantas faltantes detectadas
Imagen 1	20	426	39
Imagen 2	21	447	46
Imagen 3	21	436	51
Imagen 4	20	429	39
Imagen 5	21	456	50

Fuente: Propia

CAPÍTULO 3

3. RESULTADOS

3.1 Definición de métricas de evaluación

Para analizar la precisión y sensibilidad del algoritmo para imágenes de 5, 10 y 15 metros de altura se ha realizados dos análisis. El primer análisis se lo realizó de manera manual, es decir, se clasificó a las plantas faltantes de manera manual. El segundo análisis proviene por parte del algoritmo desarrollado. Esto se lo realizó con el objetivo de poder comparar el análisis real con el análisis del algoritmo y comparar los resultados. De esta manera se obtiene las siguientes variables:

- Verdaderos Positivos (VP): El valor real es positivo y la prueba predijo también que era positivo.
- Verdaderos Negativos (VN): El valor real es negativo y la prueba predijo también que el resultado era negativo.
- Falsos Negativos (FN): El valor real es positivo, y la prueba predijo el resultado como negativo.
- Falsos Positivos (FP): El valor real es negativo, y la prueba predijo el resultado como positivo.

3.1.1 Precisión

La precisión dentro de la técnica de segmentación se refiere al grado en el que los resultados de la segmentación coinciden con la segmentación verdadera. De esta manera la precisión, dará una idea de la sobresegmentación de la segmentación obtenida del algoritmo, respecto de la segmentación verdadera o también conocida como **ground truth**, tomando como valor máximo el 1, en caso de que no existan falsos positivos (sobre-segmentadas), y el valor mínimo de 0 cuando existan numerosas zonas de falsos positivos. (Panduro Cívico, 2010).

Según Panduro Cívico (2010), para obtener la precisión se emplea la siguiente fórmula:

- VP = Verdaderos Positivos
- FP = Falsos Positivos

$$precisión = \frac{VP}{VP + FP}$$

Precisión para imágenes de 5m de altura

Para obtener la precisión del algoritmo primero se realizó el conteo real de las plantas como se indica en la Fig. 42. También es importante destacar que las plantas muertas o marchitas se las ha considerado como plantas faltantes.

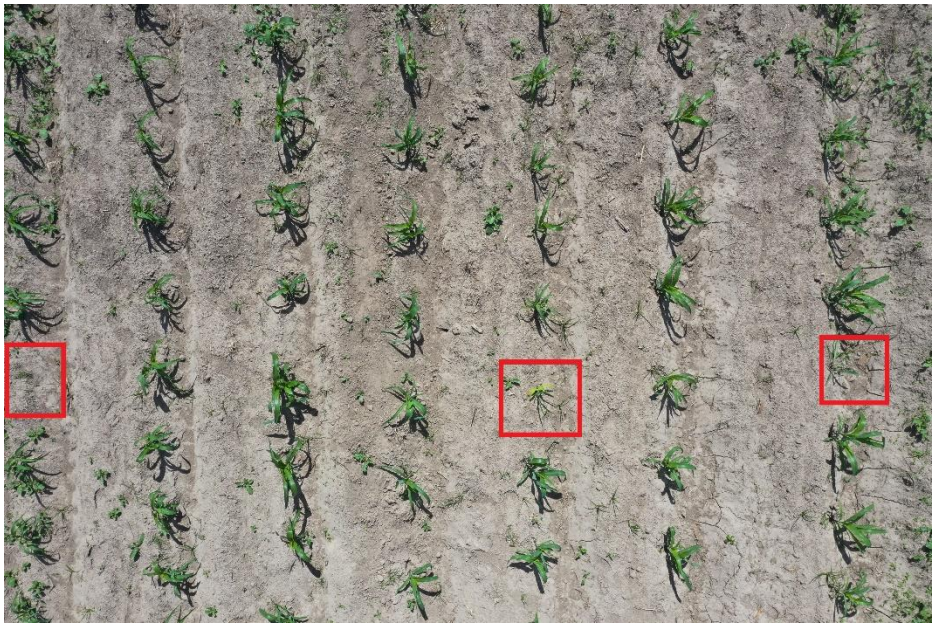


Fig. 42. Análisis manual de la imagen de 5m de altura
Fuente: Propia

Luego se procede a comparar la misma imagen, pero haciendo el análisis con el algoritmo desarrollado. El resultado se muestra en la Fig. 43:

Fig. 43. Análisis por parte del algoritmo de la imagen de 5m de altura
Fuente: Propia

Una vez realizado estos pasos, se procedió a sacar las variables requeridas para aplicar la fórmula de la precisión. En este caso los VP son aquellas plantas faltantes que se identifican en la Fig. 42 y coinciden con las plantas faltantes que se identifican en la Fig. 43. Por otra parte, los FP, son aquellas plantas faltantes identificadas en la Fig. 43 pero que en la Fig. 42 no se identificaron. Tomando en cuenta estas definiciones tenemos los siguientes resultados para la imagen 3:

$$VP = 3$$

$$FP = 0$$

$$precisión = \frac{VP}{VP + FP}$$

$$precisión = \frac{3}{3 + 0}$$

$$precisión = 1$$

Para este ejemplo se pudo observar que se obtuvo la precisión máxima debido a que el análisis manual con el análisis del algoritmo coincidía. Es importante destacar que la precisión para imágenes de 5m de altura se calculó de 5 imágenes diferentes, para las cuales se realizó el mismo proceso mencionado anteriormente.

A continuación, en la TABLA 16 se expone los resultados de la precisión de las 5 imágenes analizadas:

TABLA 16
RESULTADOS DE LA PRECISIÓN PARA IMÁGENES DE 5M DE ALTURA

Descripción	Verdaderos Positivos	Falsos Positivos	Precisión	Porcentaje
Imagen 1	0	3	0	0%
Imagen 2	0 no se identificó	0 no se identificó	1	100%
Imagen 3	3	0	1	100%
Imagen 4	2	0	1	100%
Imagen 5	0 no se identificó	0 no se identificó	1	100%
Precisión general para la altura de 5m			0.8	80%

Fuente: Propia

Precisión para imágenes de 10m de altura

Para obtener la precisión de las imágenes de 10m de altura se hizo el mismo procedimiento realizado con las imágenes de 5m de altura. Por ejemplo:

En la Fig. 44 se muestra el análisis manual de la imagen:

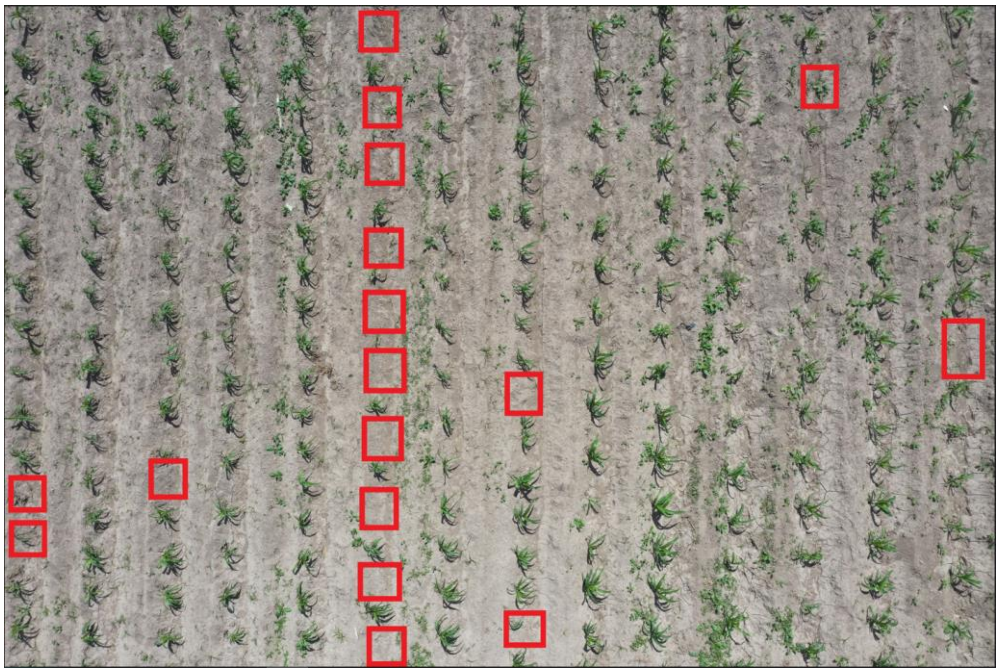


Fig. 44. Análisis manual de la imagen de 10m de altura
Fuente: Propia

Luego, en la Fig. 45 se muestra el análisis por parte del algoritmo:

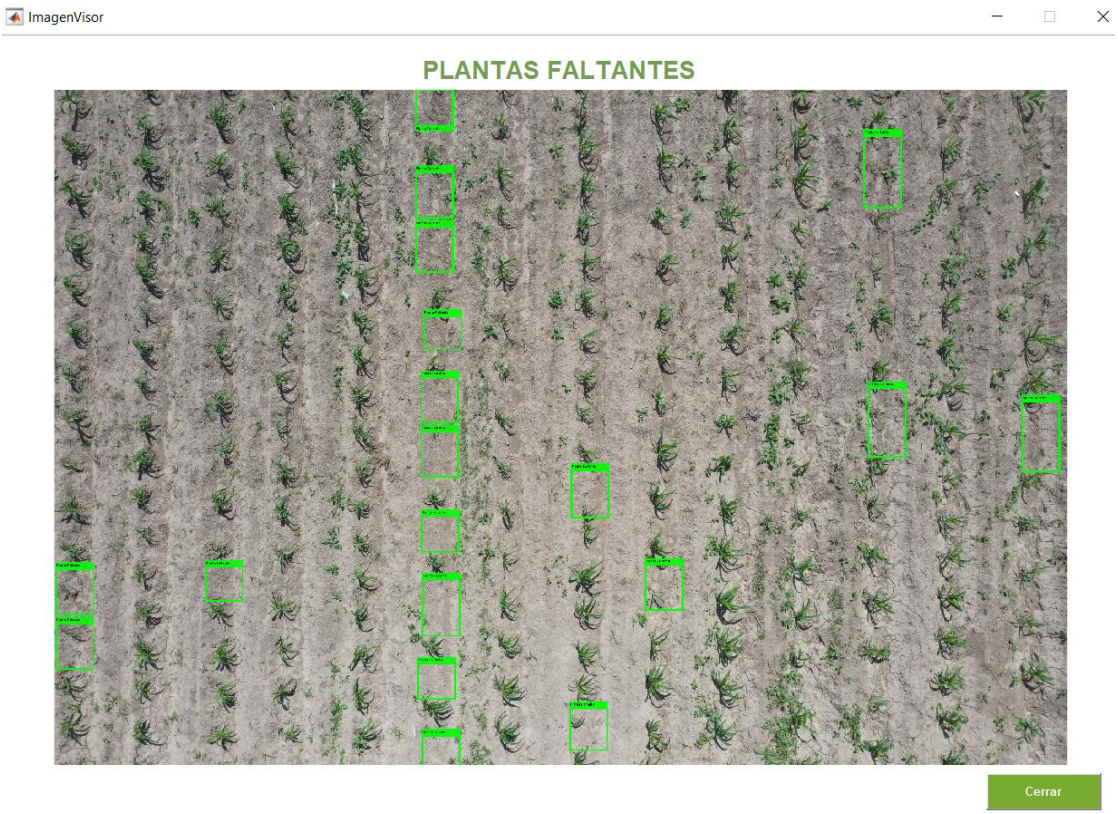


Fig. 45. Análisis por parte del algoritmo de la imagen de 10m de altura
Fuente: Propia

Posteriormente, se procede a emplear la fórmula de la precisión:

$$VP = 17$$

$$FP = 9$$

$$precisión = \frac{VP}{VP + FP}$$

$$precisión = \frac{17}{17 + 2}$$

$$precisión = 0.67$$

Para las otras imágenes de 10m de altura se realizó el mismo procedimiento. A continuación, en la TABLA 17 se expone los resultados de la precisión de las 5 imágenes analizadas:

TABLA 17
RESULTADOS DE LA PRECISIÓN PARA IMÁGENES DE 10M DE ALTURA

Descripción	Verdaderos Positivos	Falsos Positivos	Precisión	Porcentaje
Imagen 1	17	2	0.89	89%

Imagen 2	11	33	0.25	25%
Imagen 3	19	14	0.58	58%
Imagen 4	22	15	0.59	59%
Imagen 5	8	18	0.31	31%
Precisión general para la altura de 10m			0.52	52%

Fuente: Propia

Precisión para imágenes de 15m de altura

Para las imágenes de 15m de altura se hizo el mismo procedimiento realizado con las imágenes de 5 y 10 metros de altura. Por ejemplo:

En la Fig. 46 se muestra el análisis manual de la imagen:

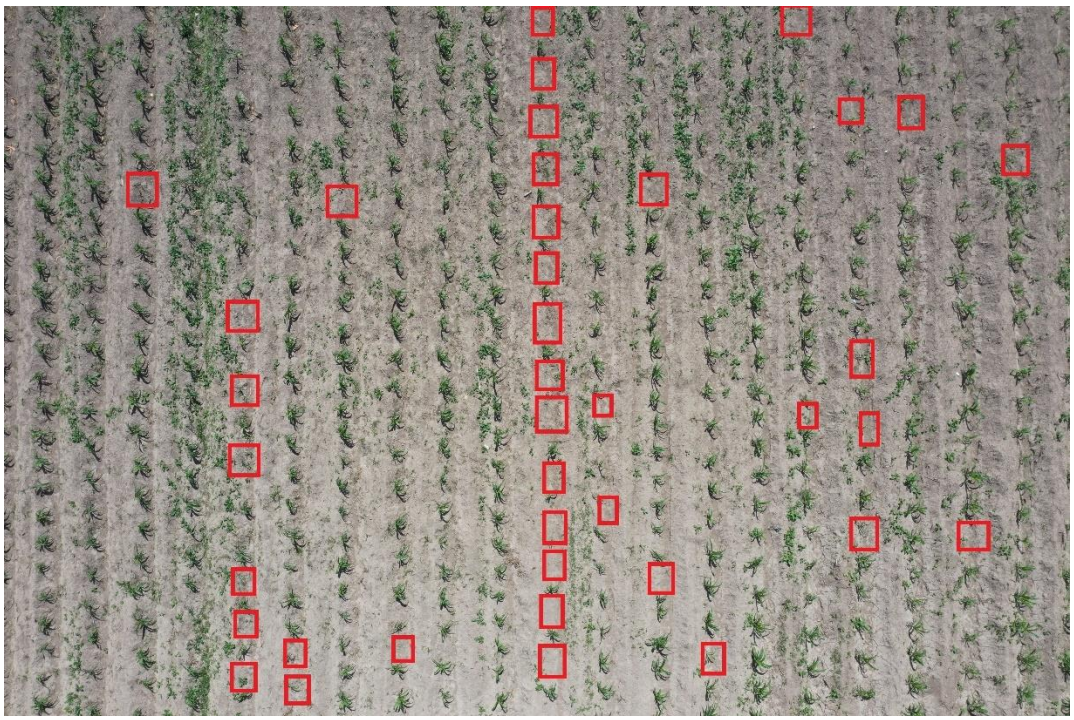


Fig. 46. Análisis manual de la imagen de 15m de altura
Fuente: Propia

Luego, en la Fig. 47 se muestra el análisis por parte del algoritmo:

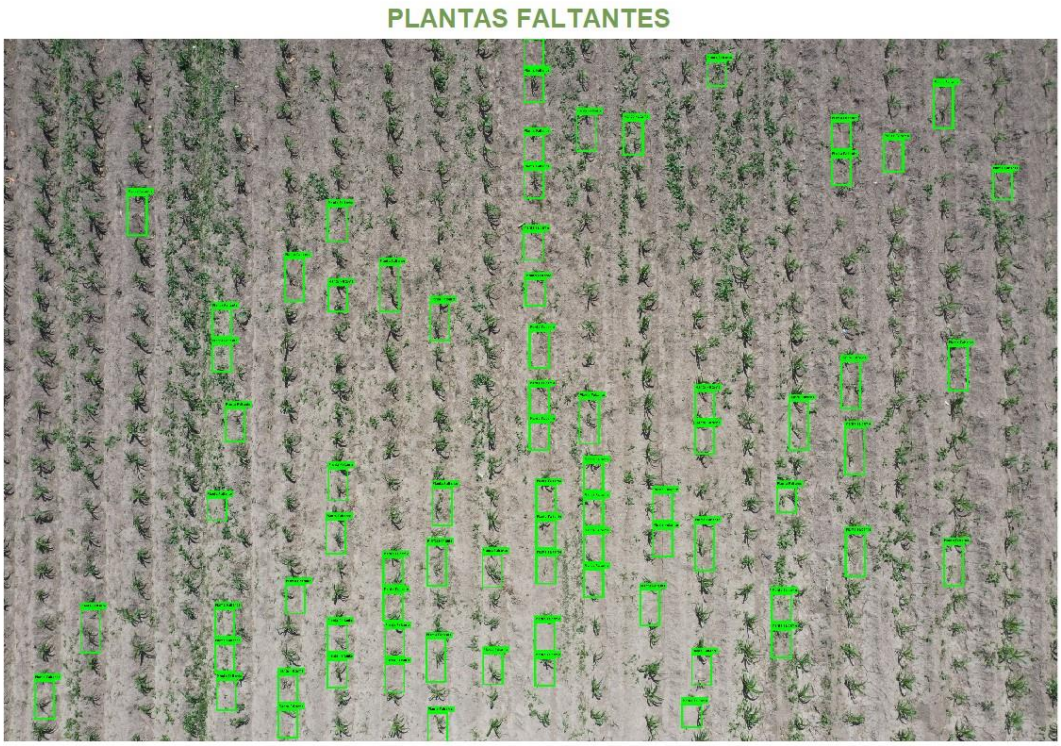


Fig. 47. Análisis por parte del algoritmo de la imagen de 15m de altura
Fuente: Propia

Posteriormente, se procede a emplear la fórmula de la precisión:

$$VP = 37$$

$$FP = 9$$

$$precisión = \frac{VP}{VP + FP}$$

$$precisión = \frac{37}{37 + 39}$$

$$precisión = 0.49$$

Para las otras imágenes de 15m de altura se realizó el mismo procedimiento. A continuación, en la TABLA 18 se expone los resultados de la precisión de las 5 imágenes analizadas:

TABLA 18
RESULTADOS DE LA PRECISIÓN PARA IMÁGENES DE 15M DE ALTURA

Descripción	Verdaderos Positivos	Falsos Positivos	Precisión	Porcentaje
Imagen 1	37	39	0.49	49%
Imagen 2	42	35	0.55	55%
Imagen 3	47	100	0.32	32%

Imagen 4	36	75	0.32	32%
Imagen 5	44	32	0.58	58%
Precisión general para la altura de 15m			0.45	45%

Fuente: Propia

3.1.2 Sensibilidad

La sensibilidad muestra la subsegmentación que existe en la imagen con respecto a la segmentación verdadera, tomando el valor máximo de 1 cuando no hay zonas de falsos negativos (sub-segmentadas) y tomando el valor mínimo de 0 cuando existe numerosas zonas de falsos negativos.

- VP = Verdaderos Positivos
- FN = Falsos Negativos

$$sensibilidad = \frac{VP}{VP + FN}$$

Sensibilidad para imágenes de 5m de altura

Para obtener la sensibilidad del algoritmo primero se realizó el conteo real de las plantas como se indica en la Fig. 48. También es importante destacar que las plantas muertas o marchitas se las ha considerado como plantas faltantes.

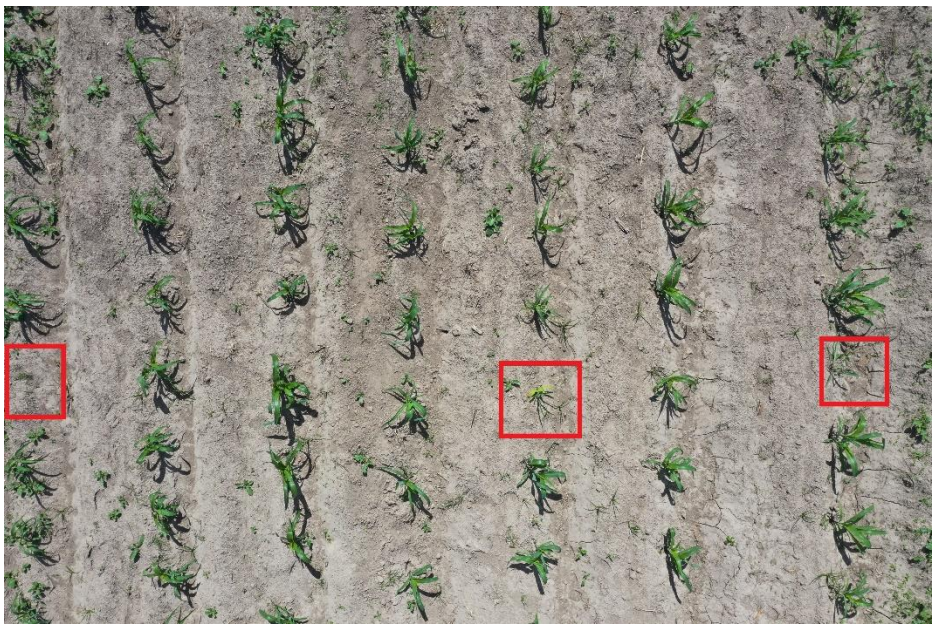


Fig. 48. Análisis manual de la imagen de 5m de altura
Fuente: Propia

Luego se procede a comparar la misma imagen, pero haciendo el análisis con el algoritmo desarrollado. El resultado se muestra en la Fig. 49:

Fig. 49. Análisis por parte del algoritmo de la imagen de 5m de altura
Fuente: Propia

Una vez realizado estos pasos, se procedió a sacar las variables requeridas para aplicar la fórmula de la sensibilidad. En este caso los VP son aquellas plantas faltantes que se identifican en la Fig. 48 y coinciden con las plantas faltantes que se identifican en la Fig. 49. Por otra parte, los FN, son aquellas plantas faltantes identificadas en la Fig. 48 pero que en la Fig. 49 no se identificaron. Tomando en cuenta estas definiciones tenemos los siguientes resultados:

$$VP = 3$$

$$FN = 0$$

$$sensibilidad = \frac{VP}{VP + FN}$$

$$sensibilidad = \frac{3}{3 + 0}$$

$$sensibilidad = 1$$

Para este ejemplo se pudo observar que se obtuvo la sensibilidad máxima debido a que el análisis manual con el análisis del algoritmo coincidía. Es importante destacar que la precisión para imágenes de 5m de altura se calculó de 5 imágenes diferentes, para las cuales se realizó el mismo proceso mencionado anteriormente.

A continuación, en la TABLA 22 se expone los resultados de la sensibilidad de las 5 imágenes analizadas:

TABLA 19
RESULTADOS DE SENSIBILIDAD PARA IMÁGENES DE 5M DE ALTURA

Descripción	Verdaderos Positivos	Falsos Negativos	Sensibilidad	Porcentaje
Imagen 1	0	0 (mala detec.)	0	0%
Imagen 2	0	0	1	100%
Imagen 3	3	0	1	100%
Imagen 4	2	0	1	100%
Imagen 5	0	0	1	100%
Sensibilidad general para la altura de 5m			0.8	80%

Fuente: Propia

Sensibilidad para imágenes de 10m de altura

Para obtener la sensibilidad de las imágenes de 10m de altura se hizo el mismo procedimiento realizado con las imágenes de 5m de altura. Por ejemplo:

En la Fig. 50 se muestra el análisis manual de la imagen:

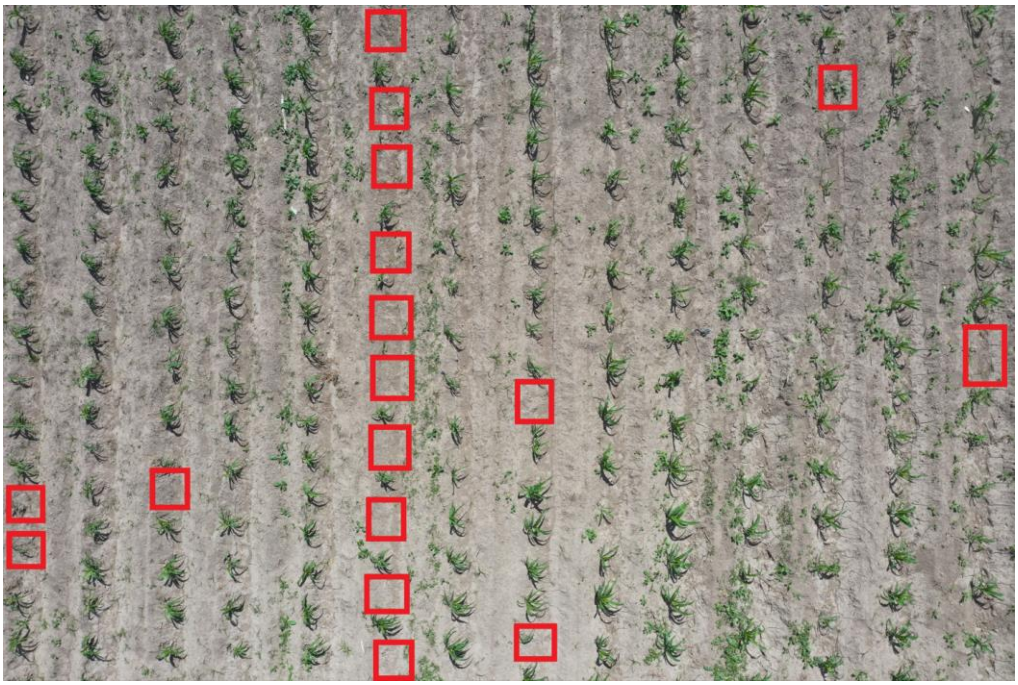


Fig. 50. Análisis manual de la imagen de 10m de altura
Fuente: Propia

Luego, en la Fig. 51 se muestra el análisis por parte del algoritmo:

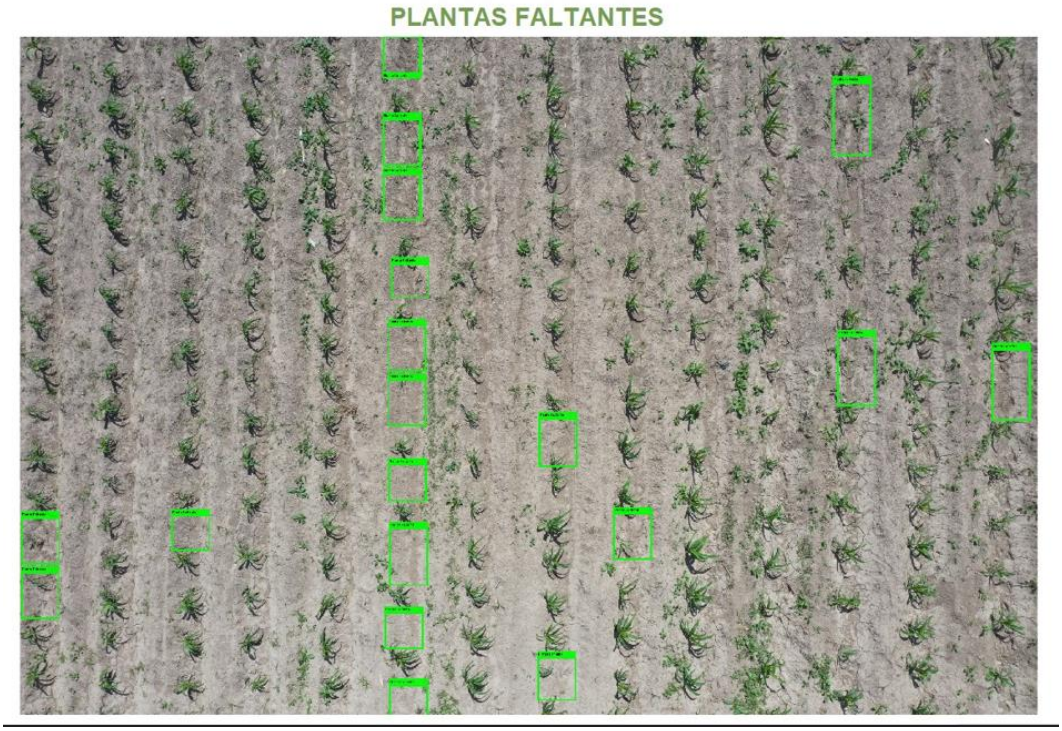


Fig. 51. Análisis por parte del algoritmo de la imagen de 10m de altura
Fuente: Propia

Posteriormente, se procede a emplear la fórmula de la sensibilidad:

$$VP = 17$$

$$FN = 3$$

$$sesibilidad = \frac{VP}{VP + FN}$$

$$sesibilidad = \frac{18}{18 + 3}$$

$$sesibilidad = 0.86$$

Para las otras imágenes de 10m de altura se realizó el mismo procedimiento. A continuación, en la TABLA 20 se expone los resultados de la sensibilidad de las 5 imágenes analizadas:

TABLA 20
RESULTADOS DE SENSIBILIDAD PARA IMÁGENES DE 10M DE ALTURA

Descripción	Verdaderos Positivos	Falsos Negativos	Sensibilidad	Porcentaje
Imagen 1	17	17	1	100%
Imagen 2	11	5	0.69	69%
Imagen 3	19	19	1	100%

Imagen 4	22	2	0.92	92%
Imagen 5	8	11	0.42	42%
Sensibilidad general para la altura de 10m			0.81	81%

Fuente: Propia

Sensibilidad para imágenes de 15m de altura

Para las imágenes de 15m de altura se hizo el mismo procedimiento realizado con las imágenes de 5 y 10 metros de altura. Por ejemplo:

En la Fig. *52.* Análisis manual de la imagen de 15m de altura Fuente: PropiaFig. 52 se muestra el análisis manual de la imagen:

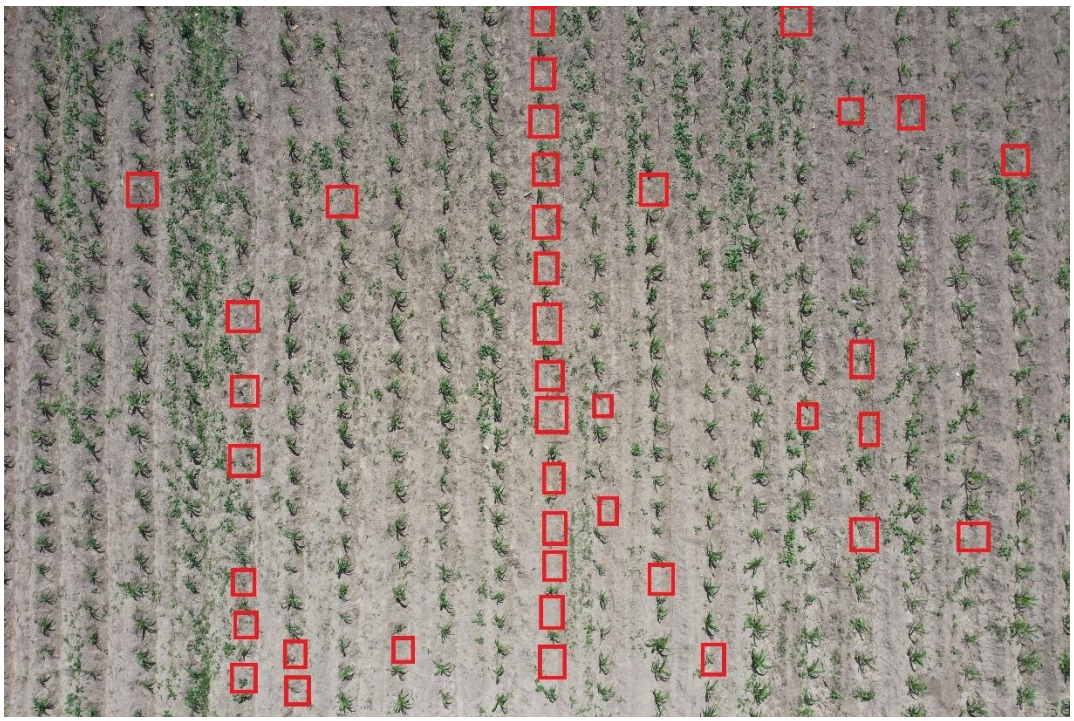


Fig. 52. Análisis manual de la imagen de 15m de altura
Fuente: Propia

Luego, en la Fig. 53 se muestra el análisis por parte del algoritmo:

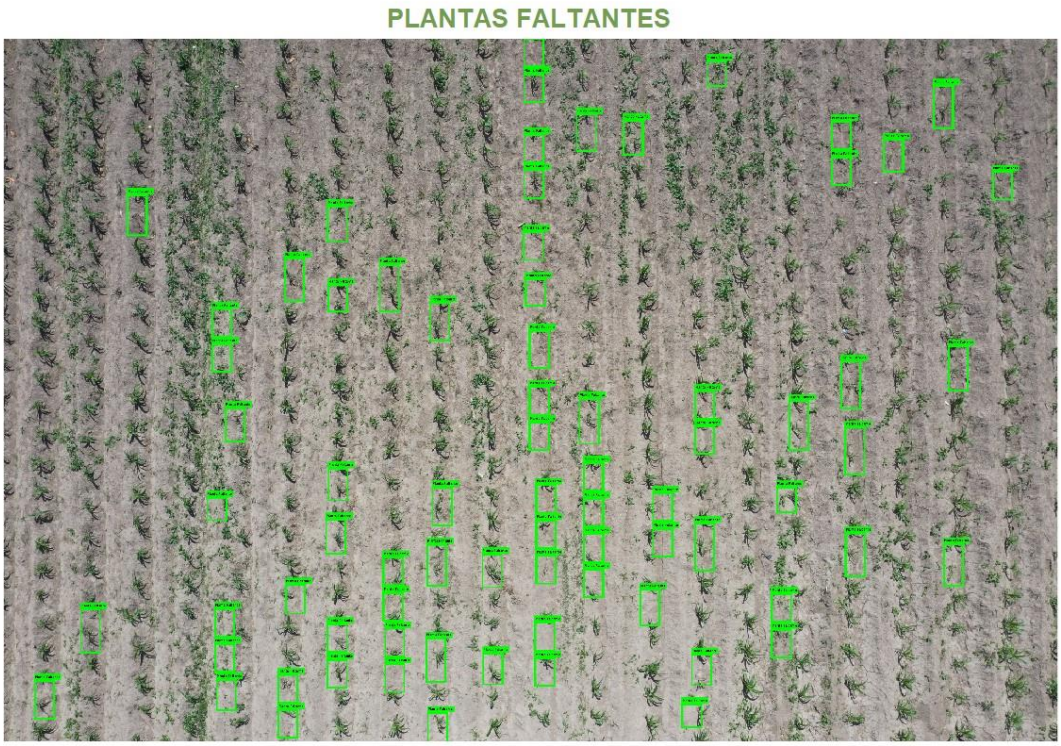


Fig. 53. Análisis por parte del algoritmo de la imagen de 15m de altura
Fuente: Propia

Posteriormente, se procede a emplear la fórmula de la sensibilidad:

$$VP = 37$$

$$FN = 2$$

$$sensibilidad = \frac{VP}{VP + FN}$$

$$sensibilidad = \frac{37}{37 + 2}$$

$$sensibilidad = 0.95$$

Para las otras imágenes de 15m de altura se realizó el mismo procedimiento. A continuación, en la TABLA 21 se expone los resultados de la sensibilidad de las 5 imágenes analizadas:

TABLA 21
RESULTADOS DE SENSIBILIDAD PARA IMÁGENES DE 15M DE ALTURA

Descripción	Verdaderos Positivos	Falsos Negativos	Sensibilidad	Porcentaje
Imagen 1	37	2	0.95	95%
Imagen 2	42	4	0.91	91%
Imagen 3	47	4	0.92	92%

Imagen 4	36	3	0.95	95%
Imagen 5	44	6	0.88	88%
Sensibilidad general para la altura de 15m			0.92	92%

Fuente: Propia

Eficiencia

Para medir la eficiencia del algoritmo es frecuente que se lo mida a través del tiempo, específicamente se evalúa el tiempo de ejecución del algoritmo. (Panduro Cívico, 2010). En la TABLA 22 se muestra los diferentes tiempos de ejecución del algoritmo para las tres distancias diferentes (5m, 10m y 15m de altura):

TABLA 22
TIEMPOS DE EJECUCIÓN DEL ALGORITMO

Descripción	**Imagen 1**	**Imagen 2**	**Imagen 3**	**Imagen 4**	**Imagen 5**	**Promedio**
Algoritmo de 5m de altura	11.6946	10.5739	9.9637	8.9583	10.7203	10.3822
Algoritmo de 10m de altura	15.8006	20.5607	17.8070	18.4161	16.2106	17.7590
Algoritmo de 15m de altura	14.4511	13.9140	14.8040	14.4166	13.9157	14.3003

Fuente: Propia

Las características del sistema donde se llevó a cabo la evaluación de los algoritmos y la medición de tiempos de ejecución se muestran en la TABLA 23:

TABLA 23
CARACTERÍSTICAS DEL COMPUTADOR

CPU	Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz, 2208Mhz
RAM	12 GB
Sistema Operativo	Microsoft Windows 10 Home Single Language

Fuente: Propia

3.2 Fundamentos de ingeniería en Swebok

Swebok es una guía que describe conocimiento que existe dentro de la ingeniería de software, por esta razón se ha utilizado el capítulo 15 de la guía correspondiente a lis fundamentos de la ingeniería para dar una mayor validez al documento desarrollado. Entre los fundamentos de la ingeniería de Swebok se emplea los conceptos relacionados al diseño de ingeniería y conceptos relacionados al modelado, simulación y creación de prototipos.

Según Abran et al., (2001), dentro del diseño de ingeniería se encuentran algunos pasos involucrados, los cuáles son los siguientes:

a) Definir el problema: Esto se lo realizó durante la fase del planteamiento del problema en donde se plateó como problema central “La detección tardía o tiempo excesivo para identificar las plantas faltantes en los cultivos de maíz”.
b) Recopilar la información pertinente: Dentro del documento se encuentra el capítulo 1 correspondiente al marco teórico; en este apartado se recopila toda la información pertinente para abordar el problema y poder proponer una solución.
c) Generar múltiples soluciones: Para resolver la problemática planteada se tuvo en cuenta múltiples soluciones como desarrollar una red neuronal para la detección de las plantas faltantes, así como también resolver la problemática mediante la técnica de visión por computado.
d) Analizar y seleccionar una solución: Debido a que este proyecto parte de una línea de investigación de otros proyectos se optó por desarrollar la solución utilizando la técnica de visión por computador para proceder a detectar las plantas faltantes mediante distancias.
e) Implementar solución: La implementación se realizó con las imágenes de prueba que se mostró anteriormente para poder obtener la precisión y sensibilidad del algoritmo.

Otro punto a destacar de la guía de Swebok es la “Medición”, ya que saber medir y qué método de medición utilizar es fundamental en la ingeniería. (Abran et al., 2001). Dentro de la medición se utilizan las escalas, las cuales pueden ser escala nominal, ordinal, de intervalo y de proporción. Para el análisis de resultados que se realizó por medio de las métricas de precisión, sensibilidad y eficiencia se utilizó una escala de intervalo, dentro de la cual se sacó la media o promedio de cada una de las métricas de evaluación del algoritmo.

3.3 Análisis e interpretación de resultados

De las pruebas realizadas para obtener la precisión y sensibilidad con imágenes de 5, 10 y 15 metros de altura, se ha demostrado que el que tiene la mejor precisión es el algoritmo de 5 m de altura con un 80%, seguido del de 10 m de altura con una precisión del 52% y por

último se tiene el algoritmo de 15 m de altura con la precisión de 45%. Los resultados se presentan en la TABLA 24 junto a la gráfica que se encuentran en la Fig. 54.

TABLA 24
PRECISIÓN DE LOS ALGORITMOS

Precisión de los algoritmos		
5 m	0,8	80%
10 m	0,52	52%
15 m	0,45	45%

Fuente: Propia

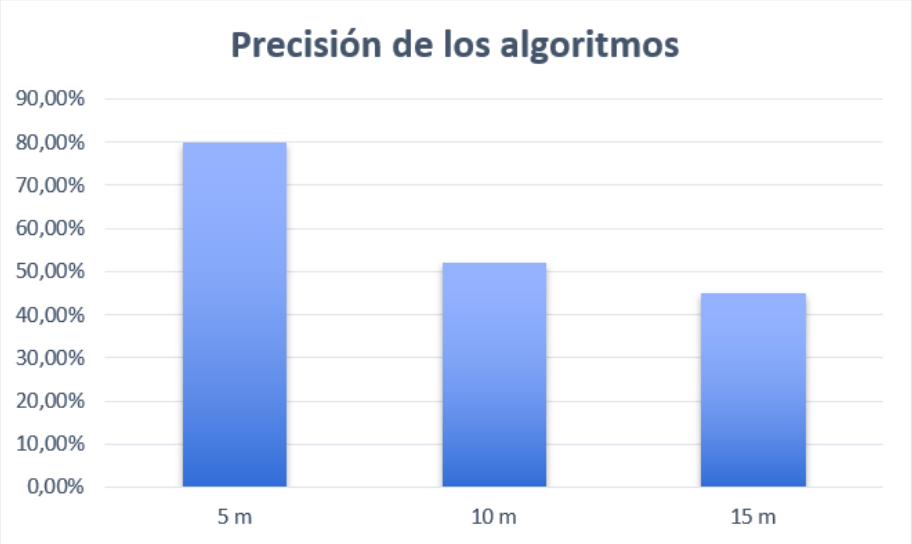


Fig. 54. Gráfica de la precisión de los algoritmos
Fuente: Propia

Con respecto a la sensibilidad de los algoritmos, el de mayor sensibilidad es el algoritmo de 15 m de altura con un 92%, seguido del algoritmo de 10 m de altura con 81% y finalmente se tiene al algoritmo de 5 m de altura con 80%. Los resultados se presentan en la TABLA 25 junto a la gráfica que se encuentran en la Fig. 55.

TABLA 25
SENSIBILIDAD DE LOS ALGORITMOS

Sensibilidad de los algoritmos		
5 m	0,8	80%
10 m	0,81	81%
15 m	0,92	92%

Fuente: Propia

Fig. 55. Gráfica de la sensibilidad de los algoritmos
Fuente: Propia

La baja precisión de los algoritmos de 10 y 15 metros de altura se da por una serie de factores como:

- La cantidad de maleza que se encuentran en las imágenes.
- El descarte de algunas plantas debido a que posee un tamaño demasiado pequeño por lo cual el algoritmo lo reconoce como maleza.
- Imágenes donde las líneas de cultivo no se encuentran por toda la imagen, es decir que existe una parte donde no se encuentran líneas de cultivo.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

Gracias a las diferentes aplicaciones de los UAV en la agricultura como la detección, identificación y conteo de cultivos, árboles, hierbas y coníferas; los agricultores pueden obtener un mejor rendimiento de sus cultivos, ya que permite agilizar el proceso de monitoreo del cultivo, permitiendo mejorar en la toma de decisiones, dando como resultado una mejora en el rendimiento de sus cultivos, disminuyendo en recursos, tiempo y personal, ya que los UAV pueden abarcar áreas difícil de monitorear para un ser humano en menor tiempo. Actualmente los UAV son utilizados para el mapeo de malezas, monitoreo del crecimiento de la vegetación, monitoreo de la salud de la vegetación y detección de enfermedades, en gestiones de riego y fumigación de los cultivos.

Por otra parte, dentro de los algoritmos desarrollados en este documento para la detección de plantas faltantes con imágenes de 5, 10 y 15 metros de altura, se concluyó que el óptimo fue el algoritmo de 10 metros de altura. Esto se debe en gran parte a que las plantas se pueden detectar de una mejor manera en comparación con el algoritmo de 15 metros, ya que en este caso es muy difícil discriminar la maleza de las plantas. Pese a que el algoritmo de 5 metros de altura tuvo una mejor precisión, no es una sorpresa su limitado rango de observación de las plantas, debido a que las imágenes contienen una menor área del cultivo a comparación de las otras.

En cuanto a la aplicación de la guía Swebok, se aplicó varios conceptos que se encuentran dentro del capítulo 15 “Fundamentos de Ingeniería”. Los conceptos utilizados se relacionan con la medición y el diseño de la ingeniería sobre todo, ya que fue la base para poder resolver la problemática planteada en este documento.

Recomendaciones

Gran parte de la precisión del análisis por parte del algoritmo se debe a las imágenes capturadas por el dron, por este motivo se debe tomar las siguientes recomendaciones a la hora de capturar una imagen:

- Utilizar imágenes con una resolución mínima de 1920*1080.
- La imagen debe capturar las líneas de cultivo en forma vertical.
- Los cultivos para analizar con el algoritmo deben ser cultivos de maíz con un tiempo de duración de 4 semanas desde que se cultivó el maíz.
- Las imágenes tomadas con el dron deben contener la mínima cantidad posible de maleza, con el objetivo de tener un análisis más preciso por parte del algoritmo.

REFERENCIAS Y BIBLIOGRAFÍA

(Alonso, 2018; García & Flego, 2005; Hakkim et al., 2016; Honrado et al., 2017; IONOS, 2020; Narciso, 2018; Njoroge et al., 2018; Ponce-Corona et al., 2019; Sindhu M, 2009; Tsouros et al., 2019)Abran, A., Moore, J. W., Dupuis, R., Dupuis, R., & Tripp, L. L. (2001). *Guide to the software engineering body of knowledge (swebok)*. http://www.mendeley.com/research/guide-software-engineering-body-knowledge-swebok/

Adhikari, S., Unit, D., Shrestha, B., & Baiju, B. (2018). *Tomato Plant Diseases Detection System*. *I*(September 2018), 81–86.

Alonso, L. (2018, January 30). *Qué es una imagen vectorial: características y diferencias con los mapas de bits*. Marketing4Ecommerce. https://marketing4ecommerce.net/que-es-una-imagen-vectorial-y-como-reconocerla/

Alvarez, J. (2015). *Diseño de un sistema de riego agrícola controlado a distancia*. 92.

Baca, L. (2016). *La producción de maíz amarillo en el Ecuador y su relación con la soberanía alimentaria*. 84. http://repositorio.puce.edu.ec/bitstream/handle/22000/12652/La produccion de maíz amarillo en el Ecuador y su relacion con la soberania alimentaria - Luis Al.pdf?sequence=1

F. M., J. M. S. (2020). A Web Based Application for Agriculture : “Smart Farming System.” *International Journal of Emerging Trends in Engineering Research*, *8*(6), 2309–2320. https://doi.org/10.30534/ijeter/2020/18862020

FAO. (2020). *Iniciativa pionera en Ecuador utiliza drones para identificar problemas y buscar soluciones para la producción eficiente de algodón | Programa de Cooperación Internacional Brasil-FAO | Organización de las Naciones Unidas para la Alimentación y la Agricult*. https://www.fao.org/in-action/programa-brasil-fao/noticias/ver/es/c/1257554/

García, E., & Flego, F. (2005). Agricultura de Precisión. *Ciencia y Tecnología*, *8*, 99–116. http://www.palermo.edu/ingenieria/downloads/pdfwebc&T8/8CyT12.pdf

Hakkim, V., Joseph, E., Gokul, A., & Mufeedha, K. (2016). Precision Farming: The Future of Indian Agriculture. *Journal of Applied Biology and Biotechnology*, *4*(06), 068–072. https://doi.org/10.7324/jabb.2016.40609

Herranz, A., & Aguirre, J. (2020). *(85) Segmentación: Algoritmo de Otsu - YouTube*. Youtube. https://www.youtube.com/watch?v=aVbK2oMTjqA

Honrado, J. L. E., Solpico, D. B., Favila, C. M., Tongson, E., Tangonan, G. L., & Libatique, N.

J. C. (2017). UAV imaging with low-cost multispectral imaging system for precision agriculture applications. *GHTC 2017 - IEEE Global Humanitarian Technology Conference, Proceedings*, *2017-Janua*, 1–7. https://doi.org/10.1109/GHTC.2017.8239328

IONOS. (2020, May 29). *Formatos de imagen | ¿Cuáles son los formatos de imagen más importantes? - IONOS*. Digital Guide IONOS. https://www.ionos.es/digitalguide/paginas-web/diseno-web/cuales-son-los-formatos-de-imagen-mas-importantes/

J., G., & Rodriguez, J. (2020). *MATLAB: guia de aprendizaje*. Jorge Sarmiento Editor - Universitas. https://elibro.net/es/lc/utnorte/titulos/181970

Jimenez Lopez, A. F., Prieto Pelayo, M. C., & Ramirez Forero, A. (2016). Teaching Image Processing in Engineering Using Python. *Revista Iberoamericana de Tecnologias Del Aprendizaje*, *11*(3), 129–136. https://doi.org/10.1109/RITA.2016.2589479

MathWorks. (n.d.). *Transformada de Hough - MATLAB hough - MathWorks España*. Retrieved June 5, 2022, from https://es.mathworks.com/help/images/ref/hough.html?searchHighlight=hough&s_tid=srchtitle_hough_1

Meneses, V. A. B., Téllez, J. M., & Velasquez, D. F. A. (2015). Uso De Drones Para El Analisis De Imágenes Multiespectrales En Agricultura De Precisión. *@limentech, Ciencia y Tecnología Alimentaria*, *13*(1), 28–40. https://doi.org/10.24054/16927125.v1.n1.2015.1647

Mery, D. (2020). *(73) 27 Procesamiento de Imágenes: Transformada de Hough, Watershed, Templates, Movimiento, - YouTube*. https://www.youtube.com/watch?v=V94QX4aCglo

Miranda, M. (2009). *La imagen digital*. http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S0016-35032009000200016

Mohedano, J. (2013). *Iniciacion a javascript*. Ministerio de Educacion y Formacion Profesional de Espana. https://elibro.net/es/lc/utnorte/titulos/49349

Molleda Meré, J. (2008). *De La Forma 3D De Productos Laminados*.

Narciso, T. F. (2018, April). *¿SABÍAS QUE EXISTEN DOS TIPOS DE IMÁGENES DIGITALES?* Aureavisura. http://aureavisurarevista.fad.unam.mx/?p=1929

Neira, J. (n.d.). *Aprende a utilizar los gráficos vectoriales | Creativos Online*. Retrieved May 15, 2022, from https://www.creativosonline.org/uso-graficos-vectoriales.html

Njoroge, B. M., Fei, T. K., & Thiruchelvam, V. (2018). A research review of precision farming

techniques and technology. *Journal of Applied Technology and Innovation*, *2*(1), 22–30. https://jati.sites.apiit.edu.my/files/2018/07/2018_Issue1_Paper4.pdf

Nolasco Valenzuela, J. S. (2018). *Python: aplicaciones practicas*. RA-MA Editorial. https://elibro.net/es/lc/utnorte/titulos/106523

Panduro Cívico, J. (2010). *Métricas de evaluación para algoritmos de segmentación.pdf*. *10*, 44–56. http://bibing.us.es/proyectos/abreproy/11863/fichero/PFC%252FCapitulo+V.pdf

Pavon Puertas, J., & Llarena Borges, E. (2015). *Creacion de un sitio web con PHP y MySQL (5a. ed.)*. RA-MA Editorial. https://elibro.net/es/lc/utnorte/titulos/106491

Pederi, Y. A., & Cheporniuk, H. S. (2015). Unmanned Aerial Vehicles and new technological methods of monitoring and crop protection in precision agriculture. *2015 IEEE 3rd International Conference Actual Problems of Unmanned Aerial Vehicles Developments, APUAVD 2015 - Proceedings*, 298–301. https://doi.org/10.1109/APUAVD.2015.7346625

Ponce-Corona, E., Sanchez, M. G., Fajardo-Delgado, D., Castro, W., De-La-Torre, M., & Avila-George, H. (2019). Detection of vegetation using unmanned aerial vehicles images: A systematic review. *2019 8th International Conference on Software Process Improvement, CIMPS 2019 - Applications in Software Engineering*, *February*. https://doi.org/10.1109/CIMPS49236.2019.9082434

Prakash, K., Saravanamoorthi, P., Sathishkumar, R., & Parimala, M. (2017). A Study of Image Processing in Agriculture. *Int. J. Advanced Networking and Applications*, *June*, 3311–3315.

Sindhu M, R. R. (2009). Images and Its Compression Techniques - A Review. *International Journal of Recent Trends in Engineering*, *2*(4), 71–75.

Sommerville, I. (2005). Requerimientos Del Software. *Universidad Veracruzana*, 109–110. https://www.uv.mx/personal/fcastaneda/files/2015/08/F_Capitulo_5_Requerimientos_del_software.pdf

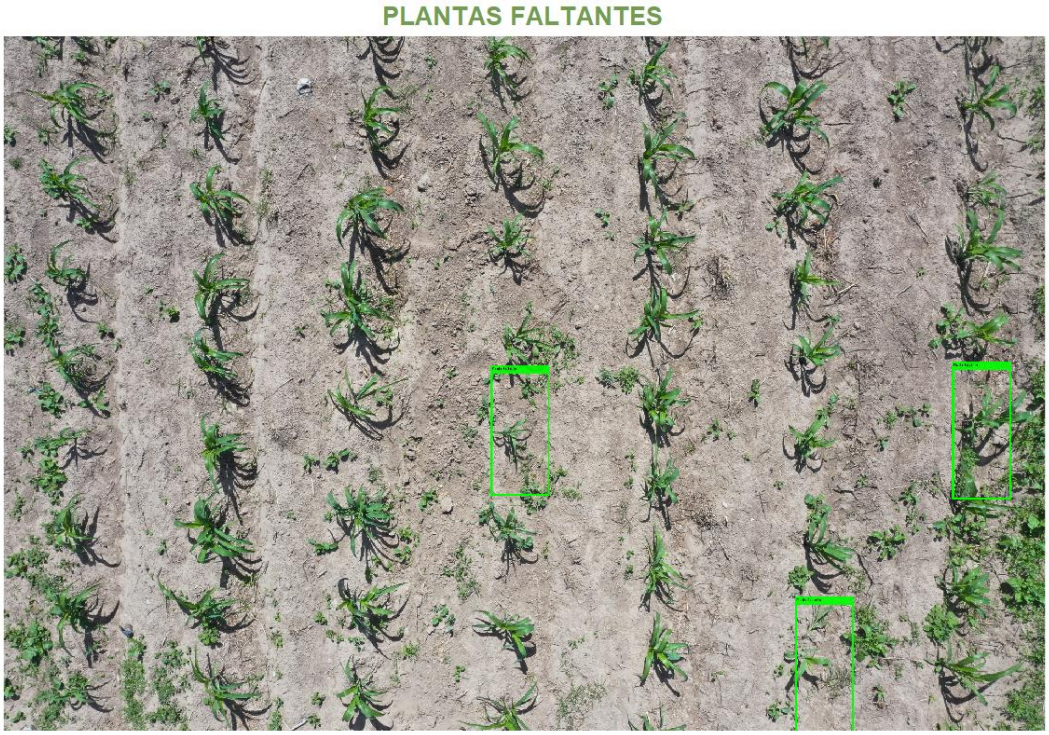
Tran, H. A. M., Ngo, H. Q. T., Nguyen, T. P., & Nguyen, H. (2018). Design of Green Agriculture System Using Internet of Things and Image Processing Techniques. *Proceedings 2018 4th International Conference on Green Technology and Sustainable Development, GTSD 2018*, 28–32. https://doi.org/10.1109/GTSD.2018.8595663

Tsouros, D. C., Bibi, S., & Sarigiannidis, P. G. (2019). A review on UAV-based applications for precision agriculture. *Information (Switzerland)*, *10*(11). https://doi.org/10.3390/info10110349

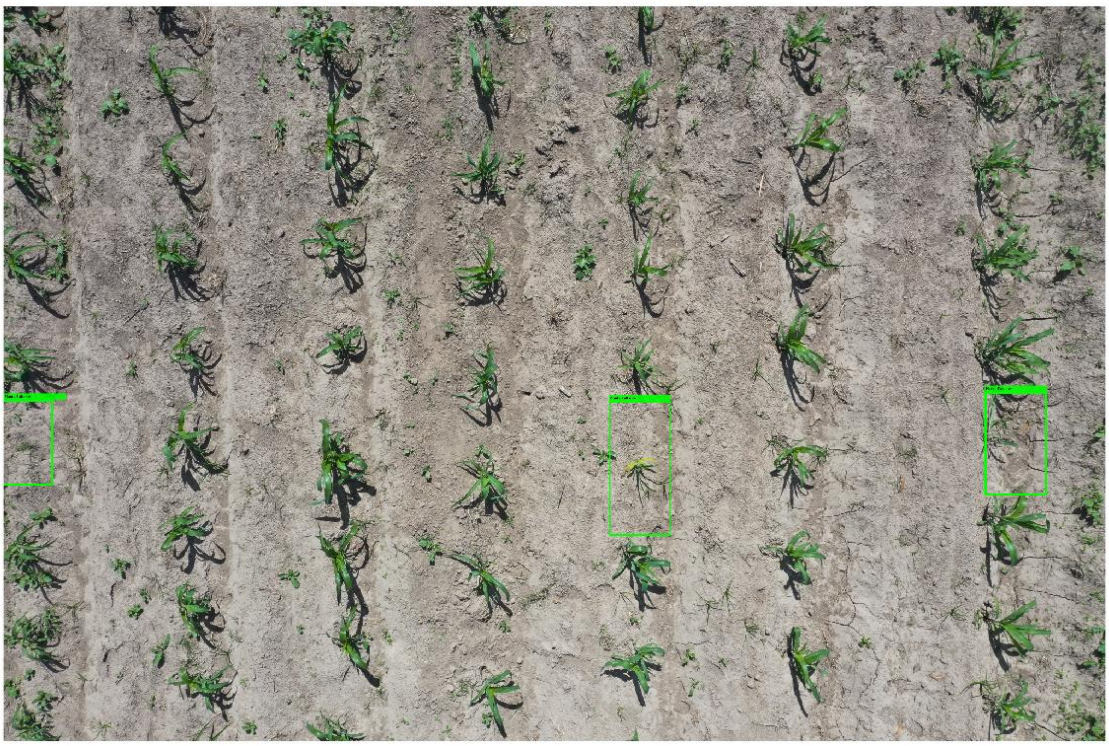
Tyagi, V. (2018). Understanding Digital Image Processing. *Understanding Digital Image Processing*, *September*. https://doi.org/10.1201/9781315123905

ANEXOS

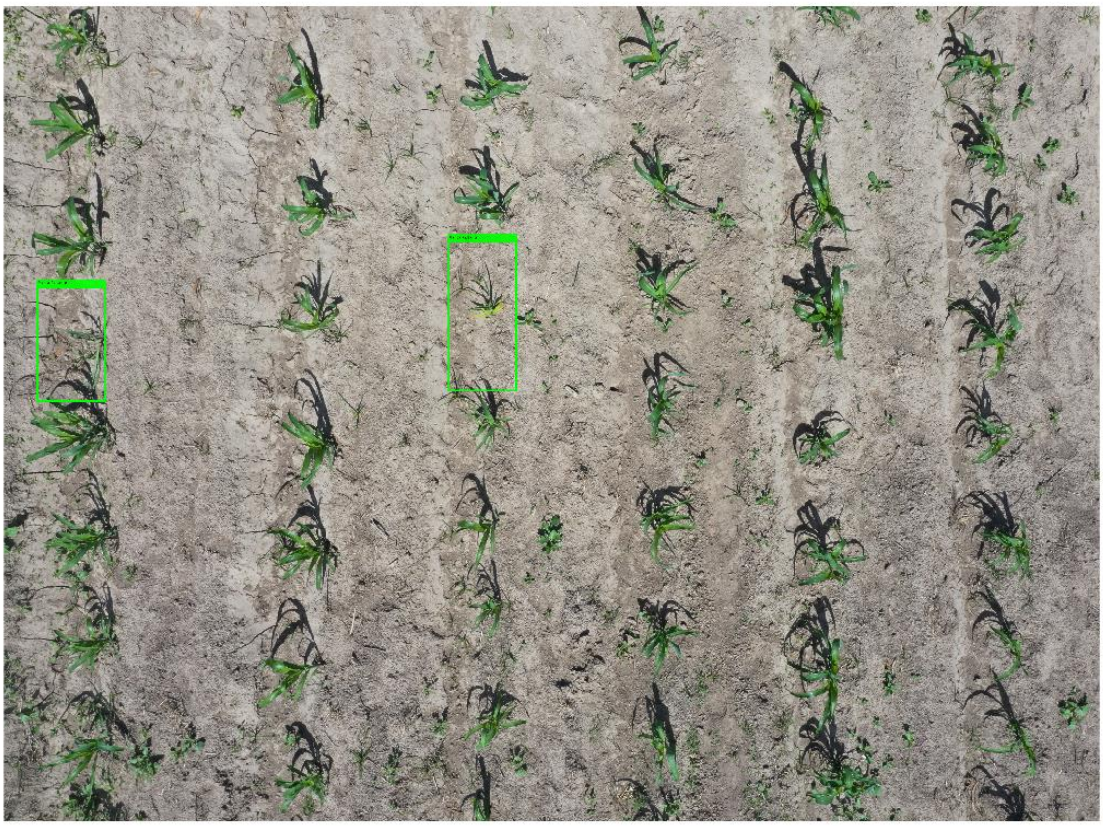
Resultados con el Algoritmo de 5 metros de altura:



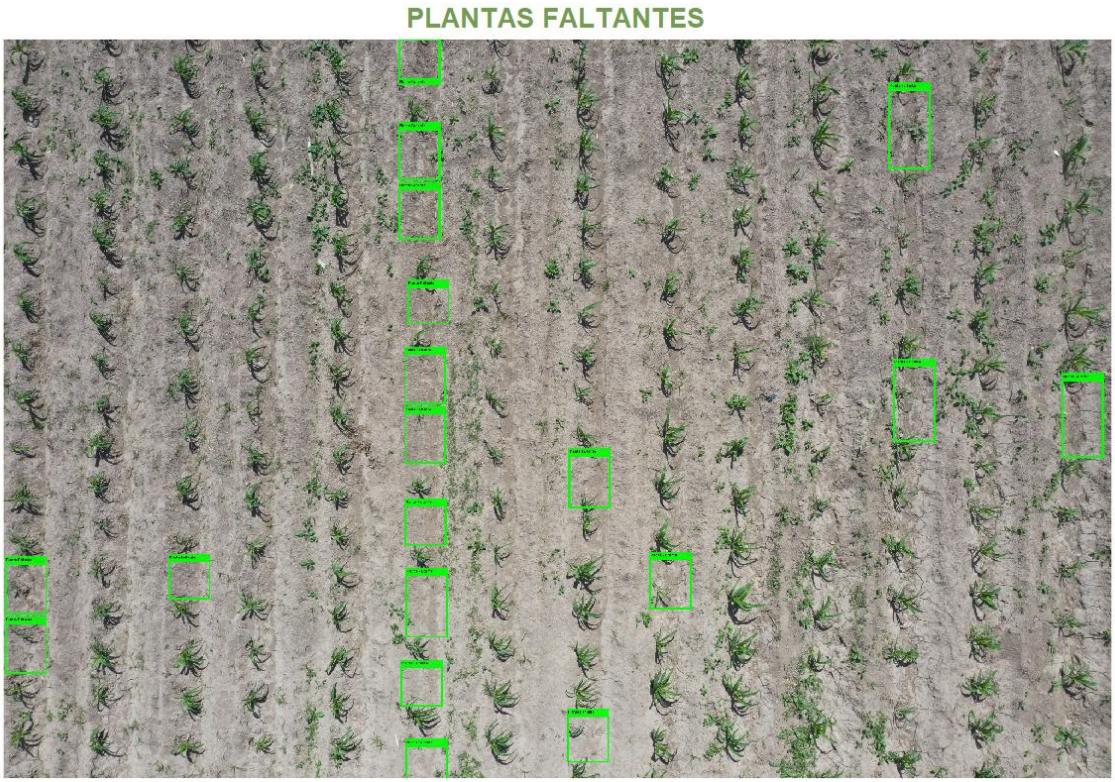
PLANTAS FALTANTES



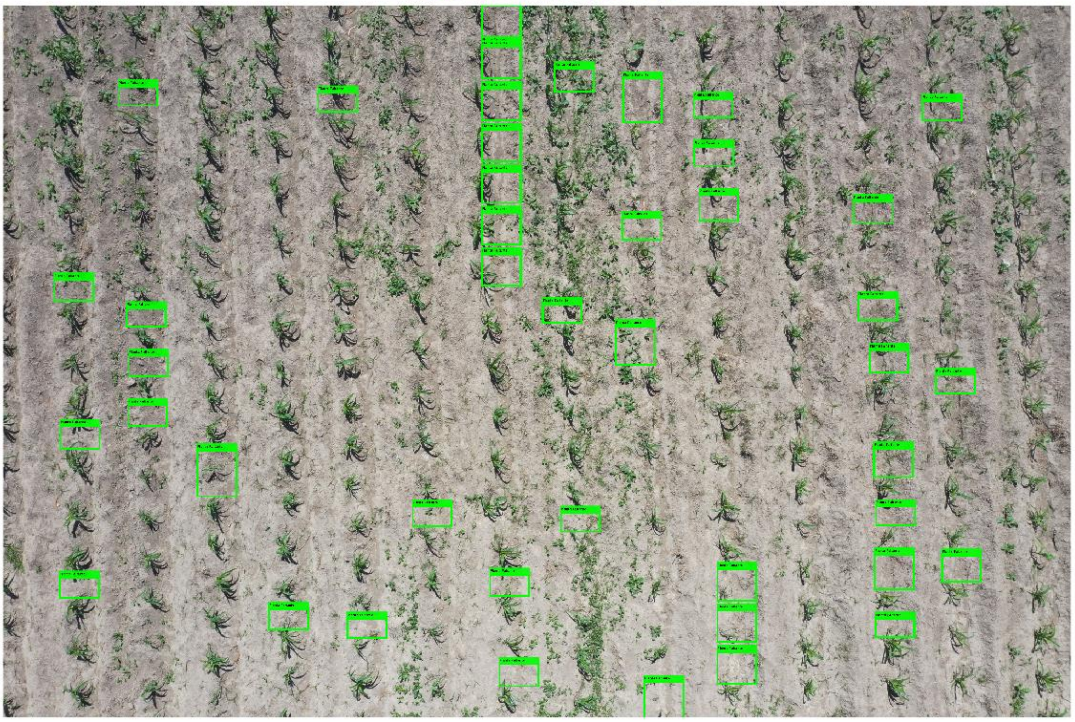
PLANTAS FALTANTES



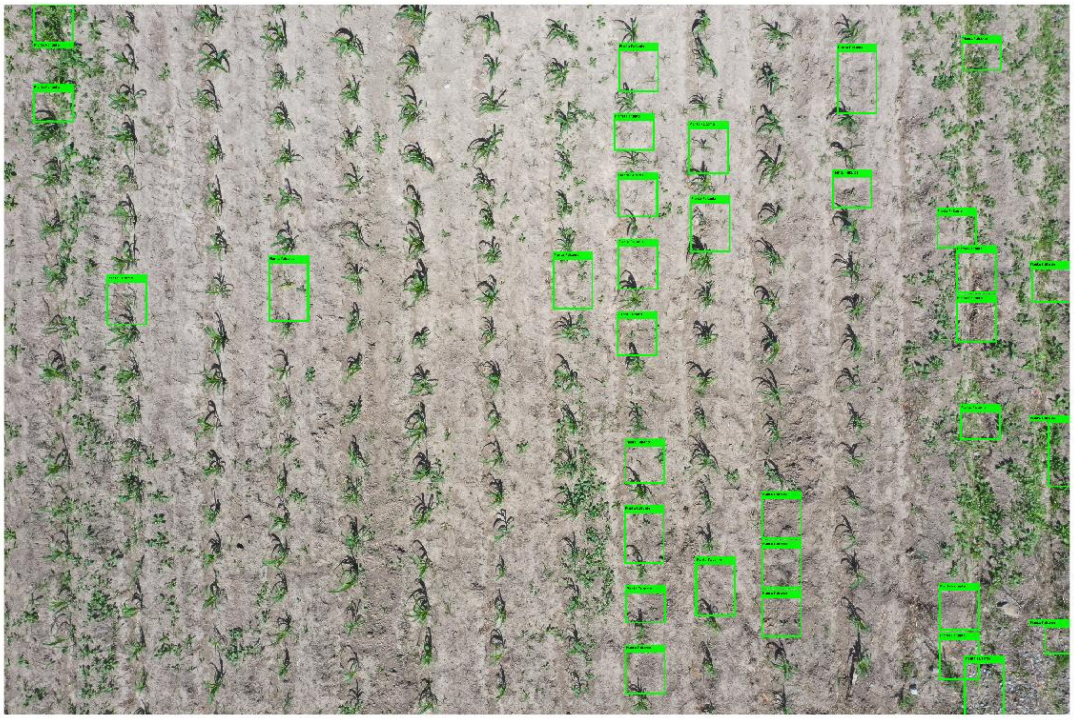
Resultados con el Algoritmo de 10 metros de altura:



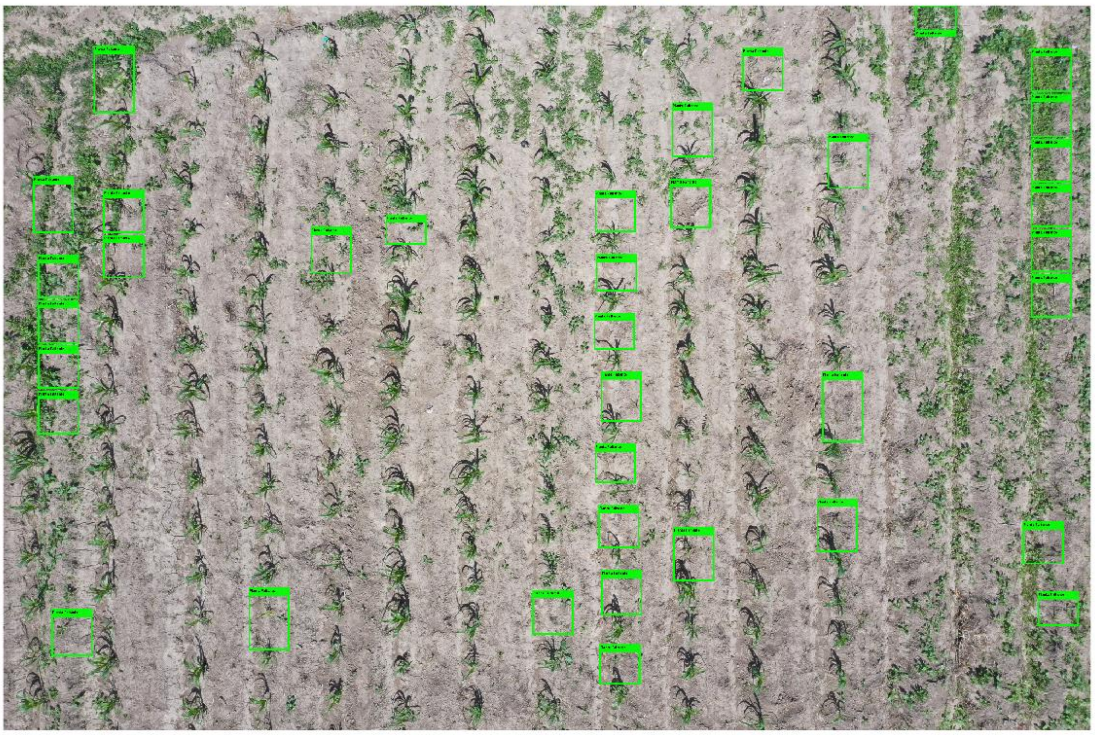
PLANTAS FALTANTES



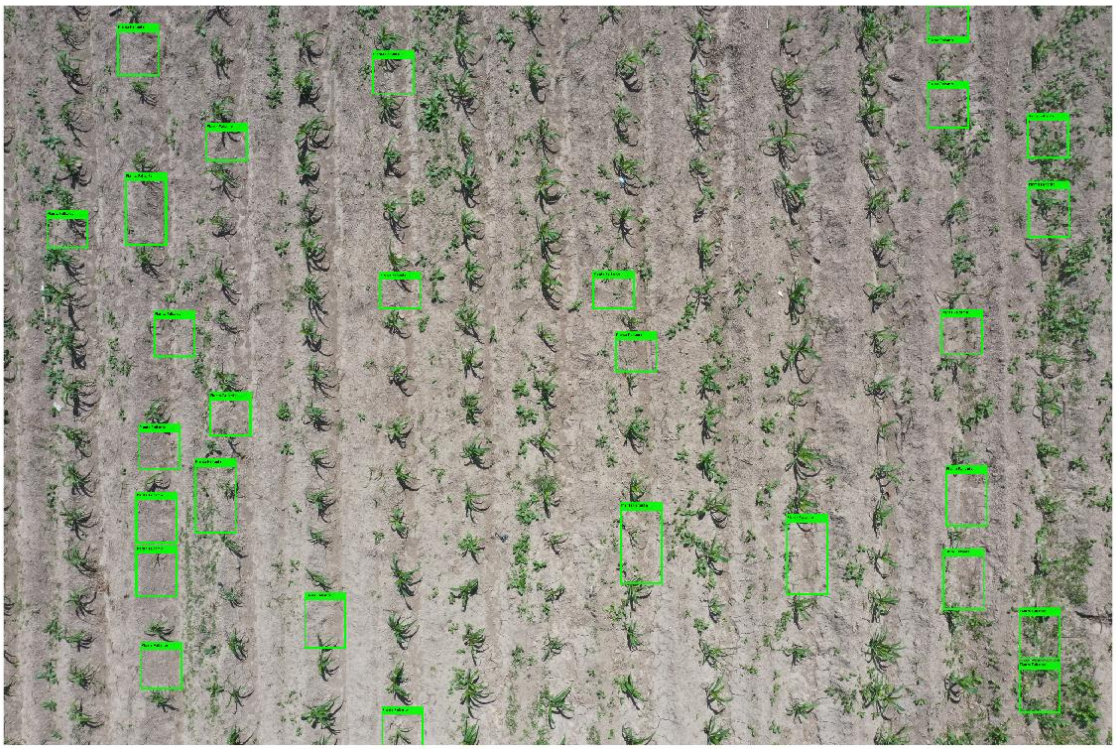
PLANTAS FALTANTES



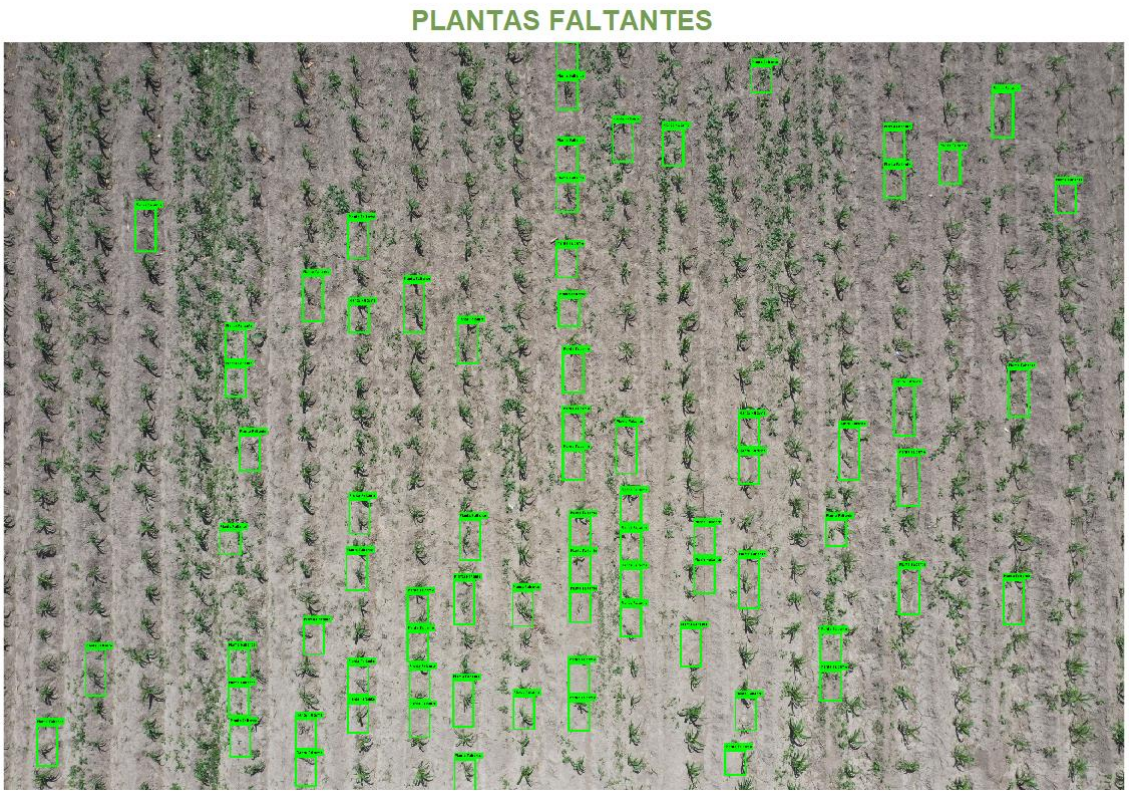
PLANTAS FALTANTES

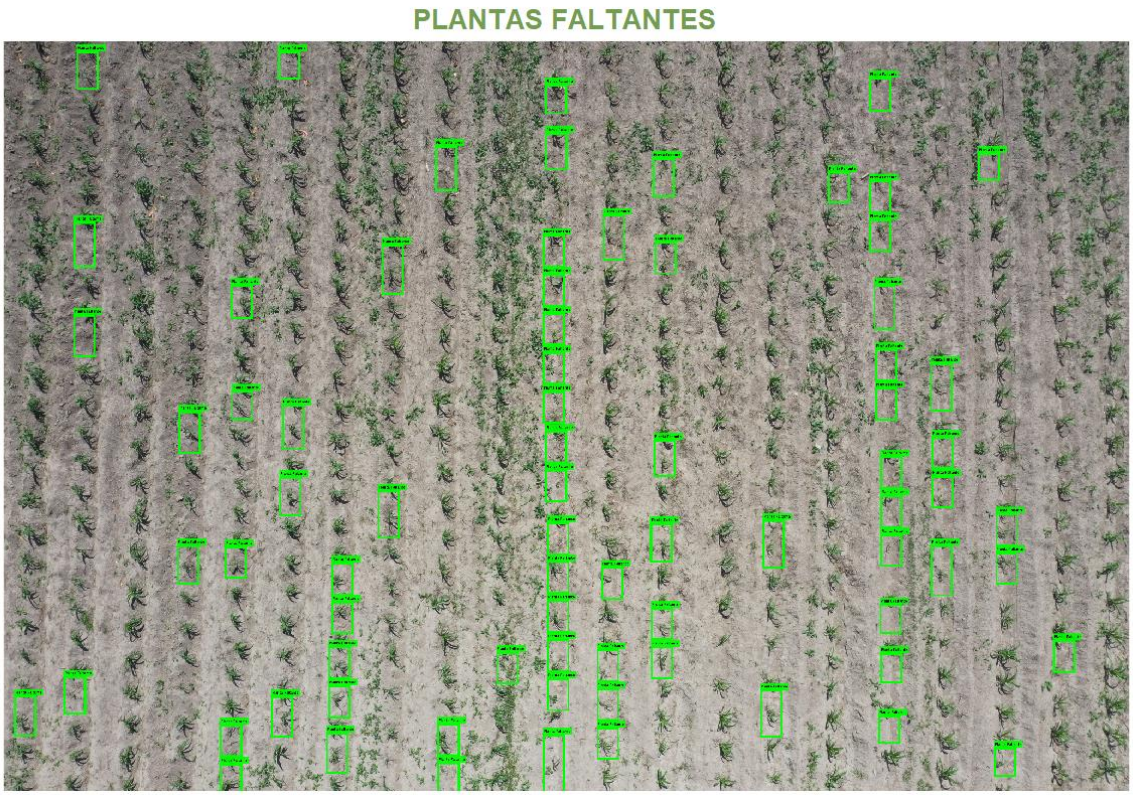


PLANTAS FALTANTES

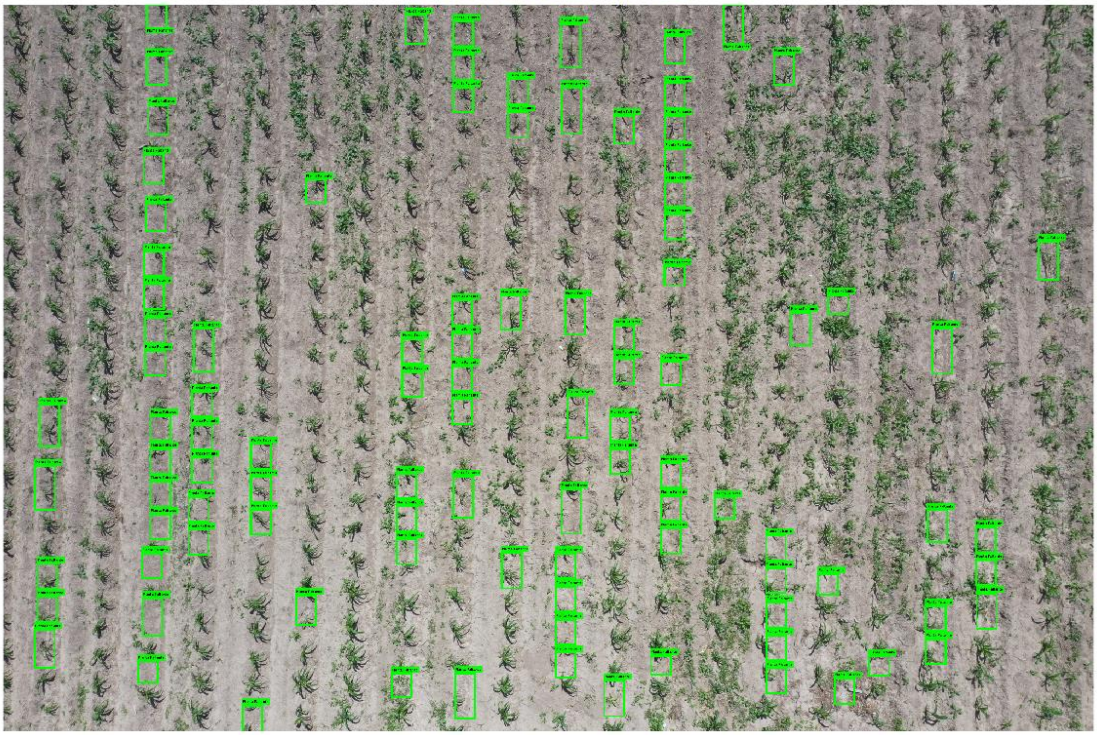


Resultados con el Algoritmo de 15 metros de altura:

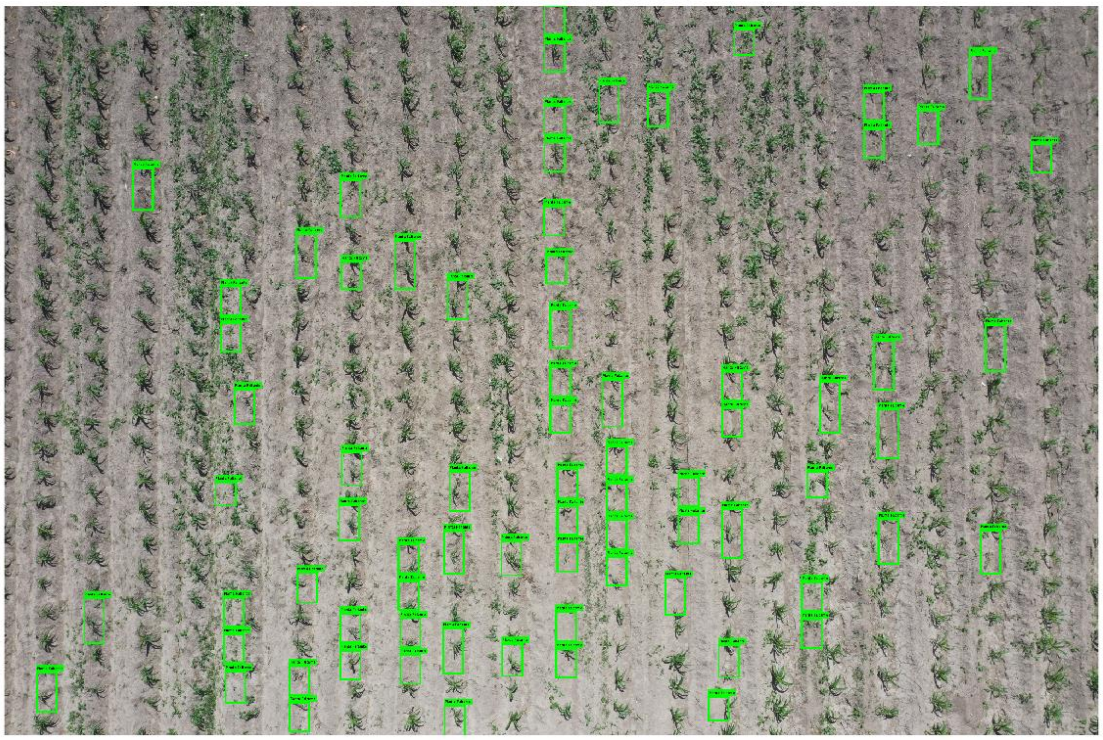




PLANTAS FALTANTES



PLANTAS FALTANTES



PLANTAS FALTANTES