

**Algoritmo de detección de mosca blanca por medio de inteligencia artificial en las hojas de
plátano**

Elaborado por

Viviana Yineth Sierra Guzmán

Asesor trabajo de grado

Paola Andrea Mateus Abaunza

Ingeniera electrónica

Magister en Ingeniería Electrónica

Universidad Nacional Abierta y a Distancia

Escuela de Ciencias Básicas Tecnología e Ingeniería

Programa de Ingeniería Electrónica

Bogotá D.C

2021

Resumen

En el presente artículo se propone un algoritmo de detección de mosca blanca en la planta de plátano mediante inteligencia artificial, con el fin de que el monitoreo e identificación se hagan de forma oportuna y que la producción satisfaga las necesidades del mercado. Para desarrollar la solución fue utilizado el software Matlab en el cual mediante instrucciones de código se realizó la implementación de técnicas de procesamiento de imágenes tales como cambios de modelo de color, umbralización, binarización, filtros de tipo Sobel y por rango y aplicación de operaciones morfológicas. Dicho algoritmo cuenta con un clasificador de tipo supervisado específicamente una máquina de soporte vectorial que fue entrenada para la detección y clasificación de las hojas sanas y con plaga de mosca blanca.

Palabras Clave: Algoritmo, inteligencia artificial, procesamiento de imágenes, clasificador de tipo supervisado.

Tabla de contenidos

Resumen.....	2
Introducción	8
Pregunta de investigación	9
Planteamiento del problema.....	10
Justificación	12
Marco referencial	13
Marco conceptual:.....	13
Estado del arte:.....	18
Objetivos	23
Objetivo general	23
Objetivos específicos	23
Metodología	24
Diseño	24
Resultados	43
Imágenes procesadas.....	43
Análisis del modelo.....	54
Conclusiones.....	60
Referencias.....	61

Lista de tablas

Tabla 1. Resultados análisis del modelo clasificador.	55
Tabla 2. Matriz de confusión del modelo.	56
Tabla 3. Parámetros obtenidos del modelo clasificador.	58

Lista de figuras

Figura 1. Árbol causa efecto – Monitoreo e identificación ineficiente de mosca blanca	11
Figura 2. Detección de las regiones enfermas de la planta	18
Figura 3. Identificación de cuatro tipos de enfermedades en los cultivos de arroz	19
Figura 4. Resultados identificación de enfermedades usando procesamiento de imágenes	20
Figura 5. Resultados detección de broca en frutos de café.	21
Figura 6. Diagrama de flujo general del algoritmo de detección de mosca blanca	24
Figura 7. Diagrama de flujo de procesamiento 1 de las imágenes.....	25
Figura 8. Diagrama de flujo procesamiento de imágenes RGB.....	25
Figura 9. Diagrama de flujo del procesamiento completo de la imagen.	26
Figura 10. Diagrama de flujo obtención de características	27
Figura 11. Diagrama de flujo obtención de características.....	28
Figura 12. Lectura imagen RGB, escala de grises, umbralización y binarización	29
Figura 13. Resultados muestra de imagen RGB y escala de grises	29
Figura 14. Resultado de binarización inicial.....	30
Figura 15. Filtro de rango, Sobel y dilatación	31
Figura 16. Resultado imagen binarizada a partir de la imagen de rango local	31
Figura 17. Resultado aplicación de técnicas Sobel y dilatación	32
Figura 18. Aplicación de compuertas lógicas.	33
Figura 19. Resultado aplicación de compuertas.	33
Figura 20. Operaciones morfológicas de llenado de orificios y dilatación a la imagen resultante.	34

Figura 21. Descomposición de capas y resta capa azul	34
Figura 22. Resultado resta de pixeles azules	35
Figura 23. Umbralización y binarización imagen restada.	35
Figura 24. Resultados umbralización, binarización y negación de la imagen restada.....	36
Figura 25. Aplicación de dilatación, llenado de orificios y filtro de eliminación de pixeles a la imagen restada	36
Figura 26. Resultado unión de procesamientos.	37
Figura 27. Conversión de tipo de dato y multiplicación por capas.....	37
Figura 28. Determinación de vector de valores máximos.....	38
Figura 29. Determinación de vector de valores mínimos	39
Figura 30. Cálculo de características de análisis	40
Figura 31. Excel de entrenamiento de la máquina de soporte vectorial	40
Figura 32. Entrenamiento – Maquina de soporte vectorial.....	41
Figura 33. Excel de prueba para la máquina de soporte vectorial	42
Figura 34. Resultados imagen Mosca_Blanca_4 – Procesamiento 1	43
Figura 35. Resultados imagen Mosca_Blanca_4 – Procesamiento 2 e imagen final procesada... 44	
Figura 36. Resultado de clasificación de la imagen Mosca_Blanca_4.....	45
Figura 37. Resultados imagen Mosca_Blanca_9 – Procesamiento 1.	46
Figura 38. Resultados imagen Mosca_Blanca_9 – Procesamiento 2 e imagen final procesada... 47	
Figura 39. Resultado de clasificación de la imagen Mosca_Blanca_9.....	48
Figura 40. Resultados imagen Hoja_Sana_7 – Procesamiento 1.....	49
Figura 41. Resultados imagen Hoja_Sana_7 – Procesamiento 2 e imagen final procesada.....	50
Figura 42. Resultado de clasificación de la imagen Hoja_Sana_7	51

Figura 43. Resultados imagen Hoja_Sana_9 – Procesamiento 1.....	52
Figura 44. Resultados imagen Hoja_Sana_9 – Procesamiento 2 e imagen final procesada.....	53
Figura 45. Resultado de clasificación de la imagen Hoja_Sana_9.....	54
Figura 46. Curva ROC del algoritmo.....	59

Introducción

El cultivo de plátano es uno de los más sembrados en Colombia y tiene como finalidad abastecer los mercados internacionales y también el mercado nacional. Dicha actividad económica genera empleos en más de ocho departamentos del país. Para que se dé una cosecha de calidad y se pueda proveer el plátano en las cantidades requeridas por los consumidores es necesario que las plantas se encuentren libres de plagas.

La plaga conocida como mosca blanca ha impactado de forma negativa las plantaciones a tal punto de provocar pérdidas del 50% de la producción de plátano, lo que genera un alto impacto negativo para el sector agrícola. Es por esta razón que se propone una técnica de monitoreo de los cultivos mediante un algoritmo de inteligencia artificial.

Mediante diversas técnicas se buscará identificar la plaga mosca blanca generando un algoritmo que procese las imágenes y determine cuando se encuentran afectadas o no por la plaga.

La construcción del algoritmo requiere de etapas de procesamiento de las imágenes hasta llegar a una segmentación adecuada que entregue la información de presencia de la plaga en la hoja o que no identifique elementos en el caso de una hoja sana. Para complementar el procesamiento y permitir la clasificación es necesario hacer uso del Machine Learning de inteligencia artificial, que facultará al algoritmo para analizar los datos resultantes y entregar el resultado correcto para cada imagen.

Con el algoritmo se tecnificará el proceso de supervisión que se realiza en las plantaciones de plátano, haciendo el proceso más efectivo en la identificación de la plaga lo que conllevará a tomar acciones correctivas inmediatas, de esta forma evitar afectaciones en la producción de tan importante alimento para nuestro país y el mercado extranjero.

Pregunta de investigación

¿Cuál es el algoritmo de inteligencia artificial que permite identificar la presencia de mosca blanca en los cultivos de plátano?

Planteamiento del problema

En las zonas productoras de plátano se presenta de forma cotidiana ataques a las plantas por parte de plagas. Las plagas son organismos animales o vegetales que destruyen y atacan los cultivos (Forero, 2019).

Para el caso específico del cultivo de plátano se encuentran en la actualidad una gran variedad de especies, siendo uno de los más importantes la mosca blanca que ha afectado miles de hectáreas en cultivos de los departamentos de Quindío, Risaralda y Meta. La mosca blanca ha afectado los cultivos de forma crítica, provocando pérdidas de hasta 50% de la producción. El riesgo que genera esta plaga en los cultivos de plátano se debe a su rápida reproducción y fácil dispersión en determinadas condiciones ambientales, generando pérdidas económicas importantes. (DANE, 2016, p. 1)

La mosca blanca succiona la sabia de la planta, limitando la conducción de alimentos de los tallos y las hojas, dicha condición se refleja con un color amarillento en las hojas, que se secan o se caen, afectando la calidad del racimo que requiere como mínimo ocho hojas funcionales.

El instituto colombiano agropecuario, ICA, ha determinado que para atacar dicha plaga se requiere un monitoreo constante de las plantaciones y también de especies arvenses, para evidenciar la presencia de la plaga “la mosca blanca”. Teniendo plena identificación de esta plaga se pueden efectuar acciones con insecticidas. Posteriormente la eliminación de las partes de la hoja afectadas, buscando mantener la cantidad mínima de funcionalidad de la planta de plátano. Finalmente compostar esas hojas con productos químicos o con exposición directa al sol. (DANE, 2016, p. 14).

Dado que se requiere dar solución a la identificación de mosca blanca en los cultivos de plátano, se propone una técnica de monitoreo basada en un algoritmo de inteligencia artificial, que permita evidenciar la presencia de la plaga por medio de procesamiento de imágenes y de esta forma identificarla antes que haya alterado el proceso natural de la plantación.

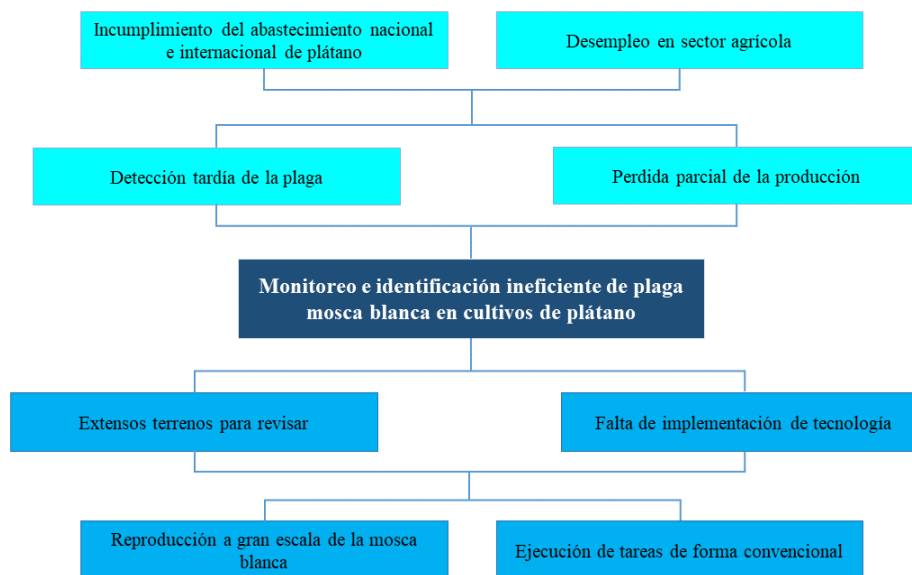


Figura 1. Árbol causa efecto – Monitoreo e identificación ineficiente de mosca blanca. (Autor).

Justificación

La presencia de la mosca blanca impacta de forma negativa la obtención de una cosecha de calidad, afectando de forma directa el abastecimiento nacional e internacional del plátano y por lo tanto perjudica el empleo del sector.

Teniendo en cuenta las pérdidas productivas que se han registrado en Colombia a raíz de la presencia de la plaga “mosca blanca”, es necesario adoptar medidas preventivas y de monitoreo permanente, para evitar la propagación de la plaga. Por esta razón, desde la rama de la ingeniería electrónica es posible aportar una solución basada principalmente en procesamiento digital de imágenes, diseñando e implementando un control de plagas mediante técnicas de tratamiento de imágenes e inteligencia artificial, que permitirá la detección de la mosca blanca en las hojas de plátano.

Una identificación mediante medios tecnológicos permitirá realizar inspecciones de forma rápida, sin importar las dimensiones de los cultivos y por ende podrán realizarse acciones correctivas en las plantas que estén afectadas, lo que ayudara a optimizar el proceso productivo y evitar las pérdidas económicas.

A partir de dicha implementación podrá tecnificarse más los procesos agrícolas que en la actualidad siguen basados en prácticas tradicionales que con ayuda de los avances tecnológicos podrán obtener mejores resultados y así ser más competitivos en los diferentes mercados.

Marco referencial

Marco conceptual:

Cultivo de Plátano en Colombia:

El plátano es el cultivo frutal más sembrado en Colombia y hace parte del consumo cotidiano de los colombianos en sus diversas variedades. En el país existen cuatro zonas principales de cultivo, las cuales son Urabá y Noreste de Antioquia (Choco, Córdoba), Centro (Caldas, Quindío, Risaralda, Tolima.), Llanos orientales (Arauca, Meta y Casanare.) y Sur del Cauca (Cauca y Valle del Cauca), siendo el departamento de Antioquia el principal productor con 61.000 hectáreas. (Minagricultura, 2018, p. 3 - 4)

De acuerdo a la información publicada por Minagricultura (2018) “alrededor del 91% del plátano producido es destinado a la exportación” (p.6).

Bélgica, compra casi 600.000 toneladas por año, seguido de Estados Unidos e Italia con casi 300.000 toneladas adquiridas, cada uno. China es un potencial comprador del plátano para Colombia, aunque este mercado también es competido con Ecuador. Colombia ocupa el 4° lugar en producción, rendimiento y área sembrada de plátano a nivel mundial, después de Uganda, Camerún y Ghana. (López, 2019)

Mosca Blanca

La mosca blanca se reproduce con rapidez en climas templados y húmedos. Su tamaño se encuentra entre 1 y 3mm. Es un insecto que una vez ha aparecido es muy difícil de controlar, debido a que su ciclo de vida está entre los 10 y los 30 días. Durante estos pueden llegar a reproducirse en varias ocasiones poniendo entre 80 y 300 huevos cada vez. (CERTIS , 2017).

Principalmente se puede detectar su presencia mirando el envés de las hojas, pues es la zona más porosa de la planta y donde mejor acceso tienen a la savia, aunque en ocasiones se

localiza en los tallos. La mosca blanca ataca a las plantas a través de su aparato bucal succionador con el que se alimenta de la savia de estas. (CERTIS , 2017).

Su alimentación desencadena la muerte del tejido en las hojas afectando la calidad y el peso de los racimos de plátano. (Simmonds, 2016).

Procesamiento de Imágenes

Una imagen es una matriz de píxeles, su análisis permite extraer información de la imagen. A nivel computacional las imágenes son representadas de forma numérica. (Centeno, 2019).

Para describir la composición de una imagen a nivel de sus píxeles 6Letras (2016) menciona que estos están definidos por modelos de color, los cuales están compuestos por una galería o gama de colores. La diferencia entre los modelos está basada en cuántos canales usan para cada color y su profundidad. Los canales almacenan la información del color y la profundidad es descrita por la cantidad de bits de cada píxel.

El espacio de color RGB, se encuentra formado por tres colores lumínicos que son el rojo, verde y azul. Usualmente, se utiliza en formatos digitales. El espacio de color CMYK, compuesto por los colores cian, magenta, amarillo y negro, habitualmente es utilizado en los medios impresos. El espacio de color LAB, está conformado por tres canales que son la Luminosidad, a y b, y es aquel que se asemeja más a la percepción del ojo del ser humano. El espacio de color HSB (Hue – Saturation – Brightness) muestra un valor de color por medio de la combinación entre el tono la saturación y el brillo. (6Letras, 2016).

Segmentación de Imágenes

La segmentación de imágenes consiste en dividir una imagen en partes o regiones, basadas en las características de los píxeles, como color o textura. Por ejemplo, una búsqueda por regiones halla discontinuidades abruptas evidenciando los bordes de objetos en la imagen. (MathWorks, 2021).

Técnicas de Segmentación

Binarización. Es una técnica que realiza una conversión de los valores de una imagen a color (RGB) a escalas de grises en blanco y negro con valores de 0 y 1. Proporciona contornos más nítidos y claros de diversos elementos de una imagen. La extracción de características mejora el aprendizaje de los modelos de inteligencia artificial. (Global, 2019).

Global (2019) afirma que en el proceso de Binarización, se selecciona un valor umbral definido por el análisis del histograma de la imagen de escala de grises, donde todos los valores por encima se clasificarán como píxeles blancos y los restantes como píxeles negros.

Detección de Bordes. En una imagen siempre existe un borde entre dos regiones adyacentes con valores de píxeles diferentes de la escala de grises. Los bordes son considerados discontinuidades locales en la imagen.

Sobel. Utiliza dos matrices para detección de bordes, una matriz para detección en eje horizontal y otra para detección en el eje vertical. La detección de bordes funciona convolucionando estas matrices sobre la imagen. (Sharma, 2019).

Canny. Este método utiliza dos umbrales con el fin de detectar bordes débiles y fuertes, debido al uso de los dos umbrales evita engaños de ruido. Calcula el degradado mediante la derivada de un filtro Gaussiano (MathWorks, 2021).

Inteligencia Artificial – Machine Learning

El Machine Learning es un aprendizaje basado en el análisis de datos, que requiere de un entrenamiento para su efectividad, usando algoritmos. Gracias a dicho entrenamiento la maquina creara un modelo de salida, teniendo en cuenta los datos de la entrada. (Hurwitz. J & Kirsch D., 2018)

Categorías

Aprendizaje Supervisado.

Aprendizaje basado en un conjunto de datos ingresados que han sido clasificados, generando resultados de salida esperados. Los algoritmos utilizan ejemplos pre – procesados, que son evaluados mediante datos de prueba. Su finalidad es hallar patrones que puedan ser aplicados en procesos. Un ejemplo podría ser la identificación de características específicas en una cantidad de imágenes determinadas. (Hurwitz. J & Kirsch D., 2018).

Aprendizaje no Supervisado

Maneja una cantidad masiva de datos que no han sido clasificados, por lo tanto, se realiza un análisis repetitivo que mediante algoritmos clasifica la información buscando una relación entre ellos. Un ejemplo de este tipo de aprendizaje es la clasificación de correos electrónicos no deseados. (Hurwitz. J & Kirsch D., 2018).

Aprendizaje Reforzado

No posee clasificación de datos. El algoritmo permite recibir retroalimentación lo cual conduce a una mejor decisión en la salida, este aprendizaje estaba basado en la prueba y error, de esta forma determina los resultados exitosos que conllevan a la efectividad de la toma de decisiones del proceso analítico. (Hurwitz. J & Kirsch D., 2018).

Técnicas de Machine Learning

Máquinas de Soporte Vectorial

Son un conjunto de algoritmos clasificadores que se utilizan en diversas aplicaciones tales como detección de personas u objetos, clasificación de contenido web, correos electrónicos, entre otros. Construyen un hiperplano en un espacio multidimensional para separar las clases, dicho hiperplano es construido de forma iterativa, lo que minimiza el error. Su objetivo es encontrar un hiperplano que divida de la mejor manera el conjunto de datos en clases.

Vectores de soporte. Puntos de datos más cercanos al hiperplano. Dichos puntos son de alta importancia para la construcción del clasificador

Margen. Es la distancia que se tiene entre los vectores de soporte y que separa las clases. (MathWorks, 2021).

Regresión Lineal

Es una técnica paramétrica es decir que antes de revisar los datos, es posible conocer cuántos parámetros o coeficientes se van a requerir el análisis y clasificación. En el caso que se esté evaluando una sola variable, x , la línea requerirá 2 parámetros.

El aprendizaje mediante regresión lineal consiste en encontrar cuáles son los mejores parámetros (coeficientes) para los datos que se tengan. Los mejores coeficientes serán los que minimicen alguna medida de error. (Martinez, 2020).

Estado del arte:

A continuación, se describirán investigaciones previas realizadas a nivel local nacional e internacional en cuanto al análisis y procesamiento de imágenes en plantas de cultivos.

Internacional

Francis, Sahaya y Anoop (2018) realizaron la investigación *Identification of leaf diseases in pepper plants using soft computing techniques*, el cual surgió a partir de la necesidad de realizar un monitoreo más efectivo de las plantaciones debido a que su ejecución de forma tradicional no permitía acciones a tiempo que contrarrestaran los hallazgos. La metodología utilizada fue el diseño de un algoritmo en Matlab, en el cual inicialmente se estableció una umbralización en la cual se le asignó un valor de 0 a los pixeles de color verde. Posteriormente se realizó una extracción y descarte de los pixeles verdes; con los pixeles resultantes mediante el modelo de color HSV se le asignó el valor 1 a la zona de la enfermedad. Como resultados se identificó presencia de enfermedad en la planta de pimienta, cuyo análisis de datos fue ejecutado mediante redes neuronales



Figura 2. Detección de las regiones enfermas de la planta. (Francis, Sahaya, & Anoop, 2016).

Kitpo y Inoue (2018) en su artículo *Early rice disease detection and position mapping system using drone and iot architecture*, buscó mejorar la revisión de cultivos de arroz que disminuyera las pérdidas ocasionadas por las plagas y enfermedades. La metodología utilizada

fue el diseño de un algoritmo de detección en Matlab, en el cual se transformó la imagen del espacio de color RGB a HSV. Con la nueva imagen generada se generó un histograma que permitió establecer los valores de píxeles a aislar en colores grises, blancos y rojo.

Posteriormente fue necesario calcular la intensidad para determinar una diferencia entre el eje menor y mayor de la imagen que conduciría a la identificación. Como resultados se obtuvo la identificación de enfermedades en diversas plantas de arroz, dichos resultados fueron evaluados bajo una máquina de soporte vectorial.

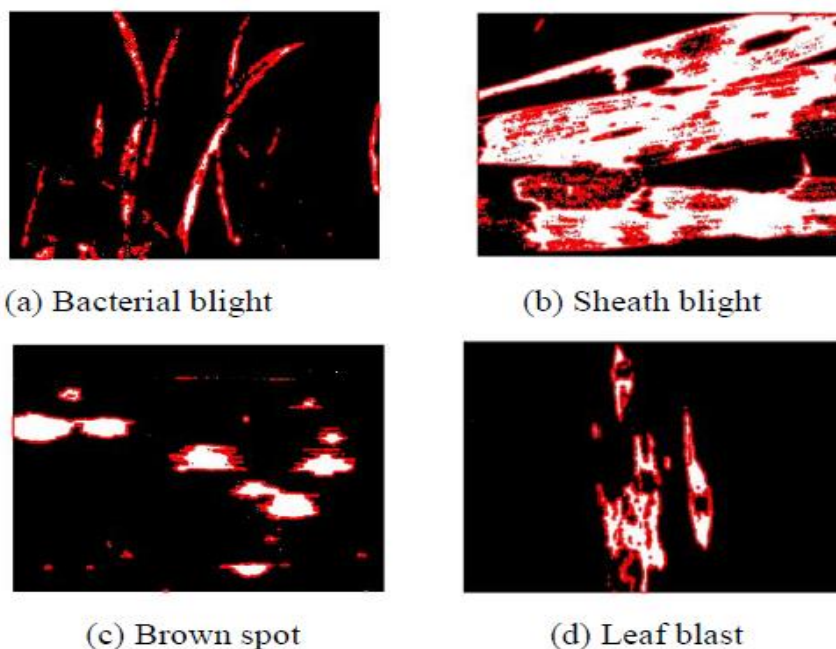


Figura 3. Identificación de cuatro tipos de enfermedades en los cultivos de arroz. (Kitpo & Inoue, 2018).

Devaraj, Rathan, Jaahnavi e Indira (2019) propusieron en su publicación *identification of plant disease using image processing technique*, un algoritmo en Matlab que permitiera encontrar e identificar enfermedades en cultivos y así evitar la afectación del sector agrícola en cuanto a las pérdidas que se generan producto de los ataques de plagas y enfermedades. La

metodología estuvo basada en cargar una imagen, pre – procesarla cambiando su tamaño, segmentarla filtrando el ruido, mejorando su contraste y realizando una conversión a escala de grises. Como resultado se extrajo la zona enferma y el análisis fue efectuado mediante bosques aleatorios.

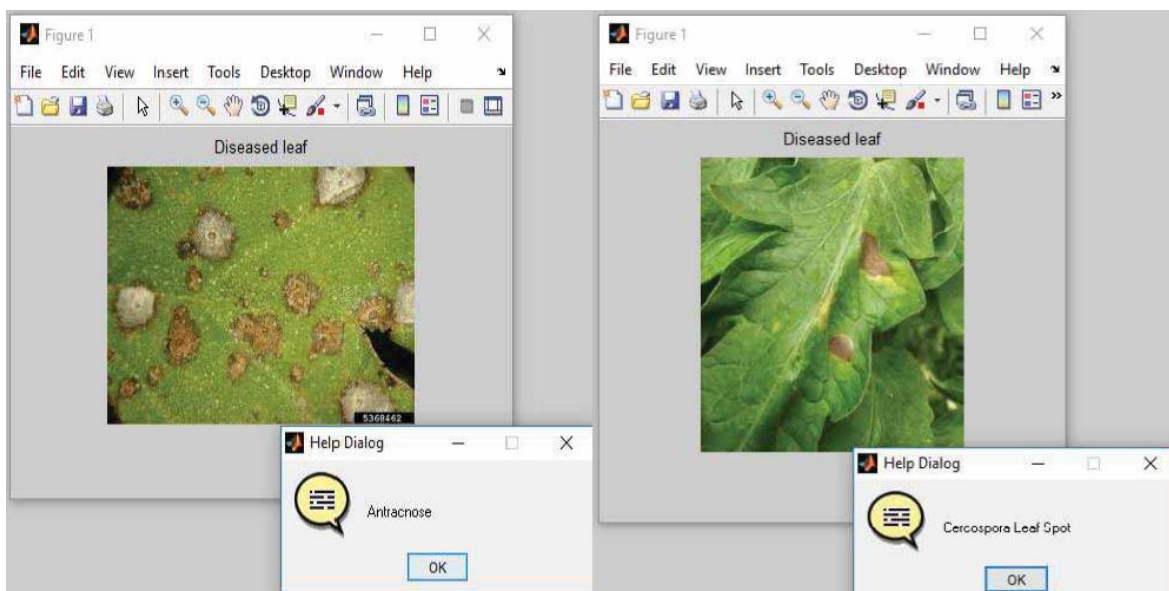


Figura 4. Resultados identificación de enfermedades usando procesamiento de imágenes.

(Devaraj, Rathan, Indira, & Jaahnavi, 2019).

Nacional

Herrera, Medina, Beleño y Gualdrón (2016), describieron en su artículo *Diseño de un sistema automático de selección de frutos de café mediante técnicas de visión artificial*, un algoritmo desarrollado en Python discriminado en dos etapas una de selección de frutos de café y en la segunda una identificación de broca en el fruto de café. La metodología desarrollada para la detección de la plaga en el fruto consistió en la conversión de la imagen a escala de grises y posterior binarización para determinar un umbral, a partir de allí los valores que estuvieran superiores serían tomados como parte de la plaga. Los resultados obtenidos fueron la

identificación de broca en el fruto de café mediante la extracción de píxeles oscuros de la imagen, los resultados fueron clasificados y evaluados mediante regla del mínimo error de Bayes.

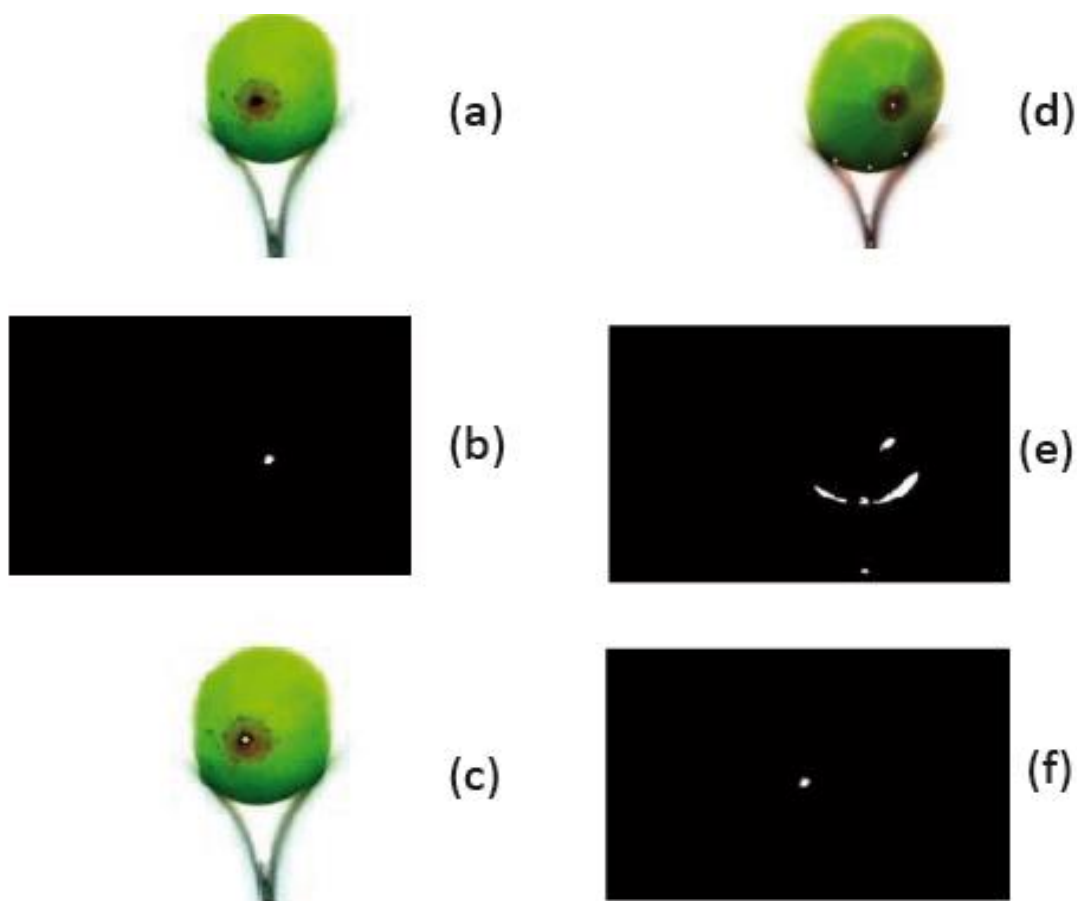


Figura 5. Resultados detección de broca en frutos de café. (Herrera, Medina, Beleño, & Gualdron, 2016).

Local

Calderón y Hurtado (2019) realizaron la investigación Machine Learning en la detección de enfermedades en plantas, en la cual se plantea la importancia del sector agrícola del país y la importancia de la identificación temprana de plagas y enfermedades para prevenir pérdidas de las cosechas. La metodología usada consistió en una investigación de las principales técnicas de

algoritmos utilizados para la detección de enfermedades como lo son redes neuronales convolucionales, bosque aleatorio, aprendizaje de reglas de asociación máquinas de vectores de soporte. Los resultados determinaron que el algoritmo de bosque aleatorio permite la mejor identificación a partir de esta información establece pasos como la extracción de características mediante el histograma de un gradiente orientado (HOG, las pruebas arrojaron una efectividad de 93% en una base de datos de 80.000 imágenes.

Objetivos

Objetivo general

Diseñar un algoritmo de inteligencia artificial para la detección de la mosca blanca en las hojas de plátano por medio de procesamiento de imágenes.

Objetivos específicos

Examinar del estado del arte sobre inteligencia artificial y procesamiento de imágenes para la detección de plagas en cultivos.

Proponer una base de datos que contenga imágenes de la hoja de plátano con y sin presencia de la mosca blanca.

Seleccionar las técnicas apropiadas para el procesamiento de imágenes que permita clasificar entre hojas que contengan la mosca blanca y hojas sanas.

Determinar qué características se requieren en el clasificador para la detección de la mosca blanca en las hojas de plátano.

Metodología

Diseño

El algoritmo desarrollado en Matlab efectuará un análisis de las características de la imagen de hoja de plátano en entornos controlados mediante unas etapas de procesamiento, como se evidencia en la figura 6.

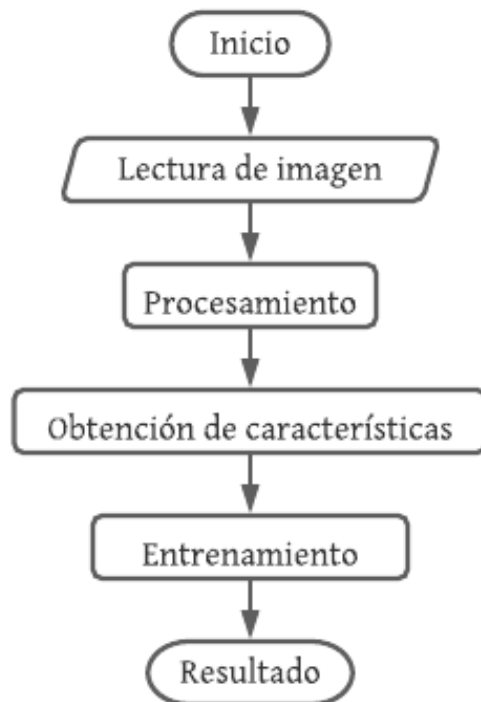


Figura 6. Diagrama de flujo general del algoritmo de detección de mosca blanca. (Autor).

El procesamiento está compuesto por aplicación de técnicas en la imagen en escala de grises (Figura 7) y en la imagen en espacio de color RGB (Figura 8).

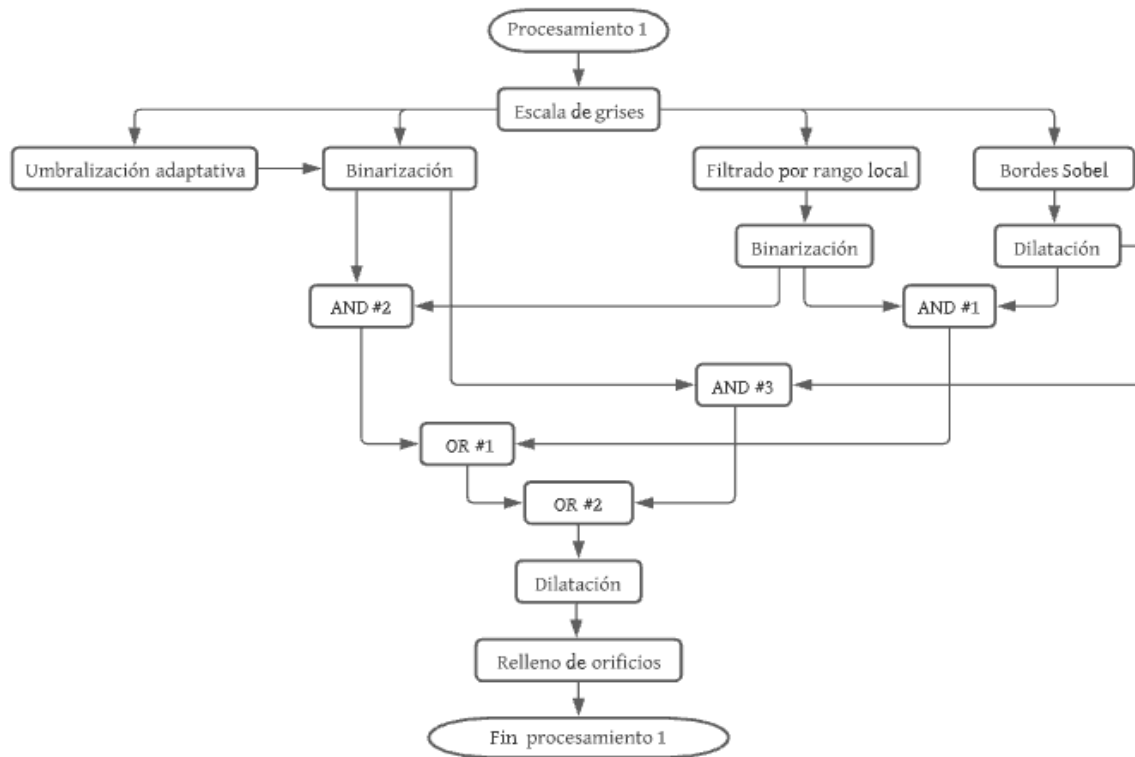


Figura 7. Diagrama de flujo de procesamiento 1 de las imágenes. (Autor).

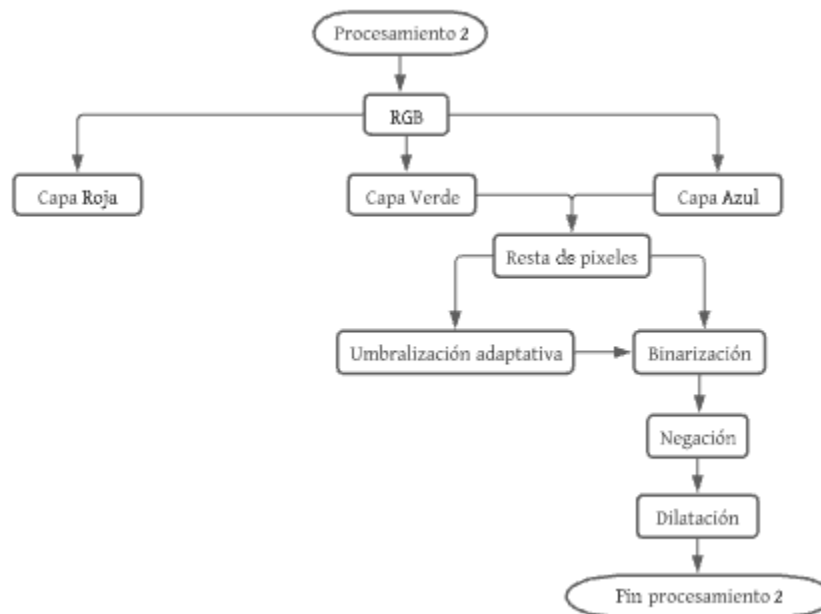


Figura 8. Diagrama de flujo procesamiento de imágenes RGB. (Autor).

Los procesamientos se unificarán con el fin de obtener la imagen final para realizar el estudio de las características como se muestra en la figura 9.

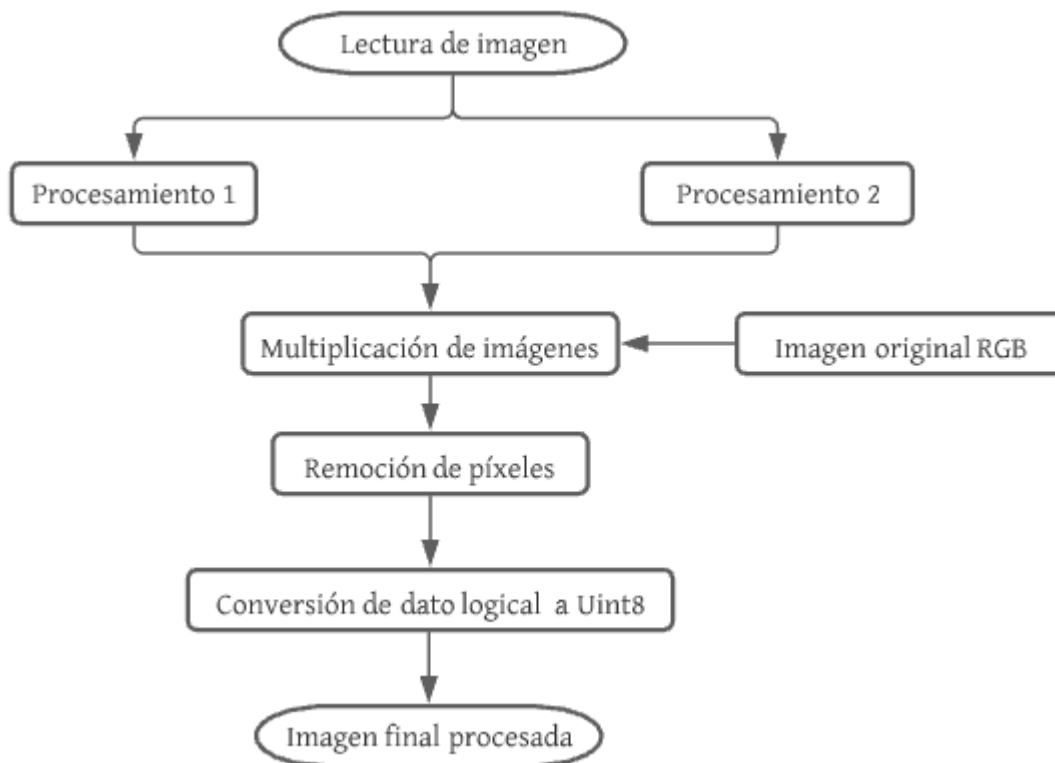


Figura 9. Diagrama de flujo del procesamiento completo de la imagen. (Autor).

Completando el procesamiento de la imagen se realiza la creación de un vector que por medio de sus valores máximos y mínimos permitirá realizar la extracción de las características que, mediante el análisis, determinarán la condición de la planta respecto a la presencia de la plaga mosca blanca. Dicho proceso es mostrado en la figura 10.

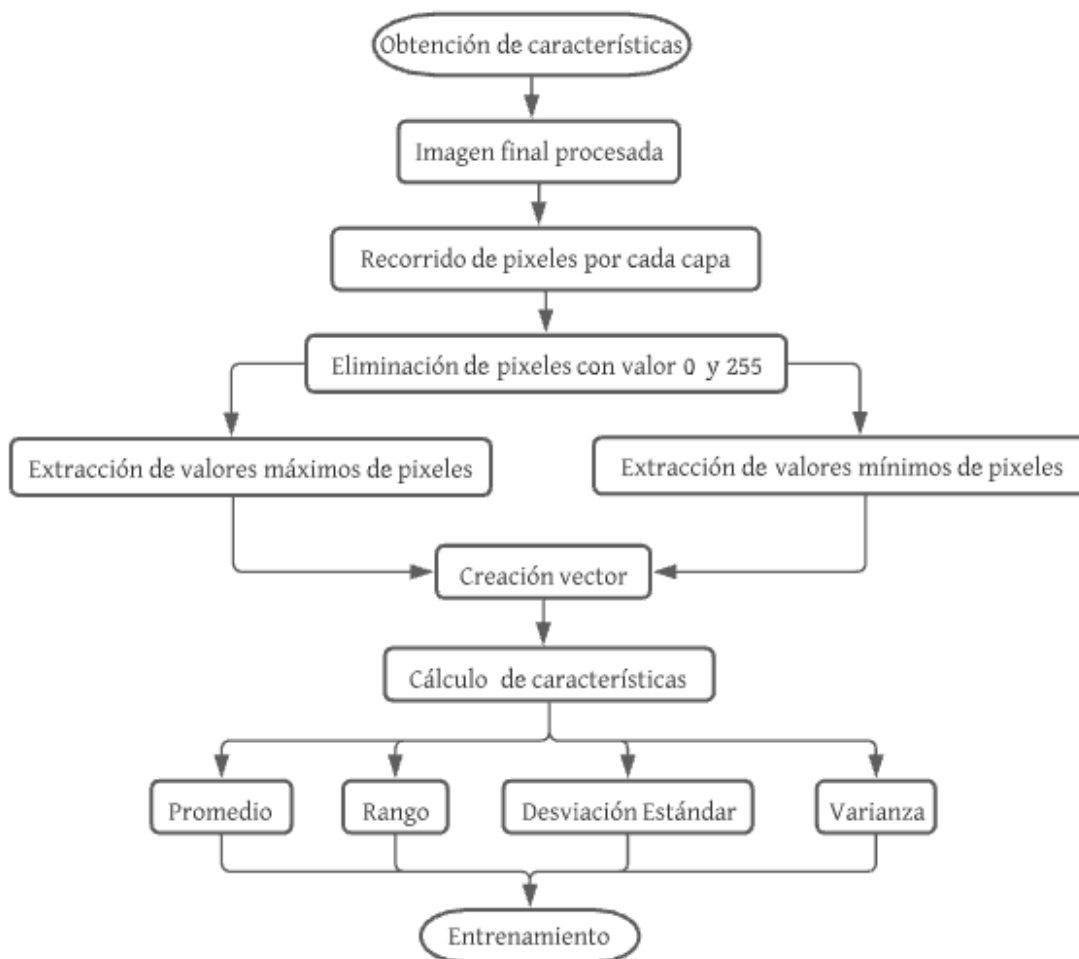


Figura 10. Diagrama de flujo obtención de características. (Autor).

Las características resultantes darán la información requerida para el entrenamiento de tipo supervisado del algoritmo como se muestra en la figura 11, completando los pasos de identificación de las plantas sanas de plátano y aquellas que contienen plaga mosca Blanca.



Figura 11. Diagrama de flujo obtención de características. (Autor).

En la primera etapa (Figura 12), el programa leerá la imagen en espacio de color RGB para posteriormente convertirla en escala de grises (Figura 13). Luego se realizará una técnica de umbralización adaptable de la imagen a escala de grises que contendrá valores decimales entre 0 y 1, la cual proporciona un mejor umbral para la binarización inicial de la imagen (Figura 14).

```

D:\Ingeniería Electrónica\Opción de grado\Proyecto de investigación Mosca blanca\Scripts Matlab\Detection_algorithm.m
EDITOR PUBLISH VIEW
1 -   clc
2 -   close all
3 -   clear all
4
5 -   %% Hoja plátano
6 -   % _____
7
8 -   %% Lectura de imagen
9
10
11 -  RGB_Hp = imread('Mosca_Blanca_1.jpg');
12 -  subplot (1,2,1)
13 -  imshow(RGB_Hp)
14 -  title 'Imagen RGB'
15
16 -  % _____
17
18 -  %% Imagen en escala de grises
19
20 -  Gris_Hp = rgb2gray(RGB_Hp);
21 -  subplot (1,2,2)
22 -  imshow(Gris_Hp)
23 -  title 'Imagen en escala de grises'
24 -  % _____
25
26 -  %% Umbralización Adaptativa
27
28 -  Aj_Local_Hp = adaptthresh(Gris_Hp, 0.4);
29 -  Bi_Local_Hp = imbinarize (Gris_Hp,Aj_Local_Hp);
30 -  figure
31 -  subplot (1,3,1)
32 -  imshow(Bi_Local_Hp)
33 -  title 'Binarización Inicial'
34
35 -  % _____

```

Figura 12. Lectura imagen RGB, escala de grises, umbralización y binarización. (Autor).

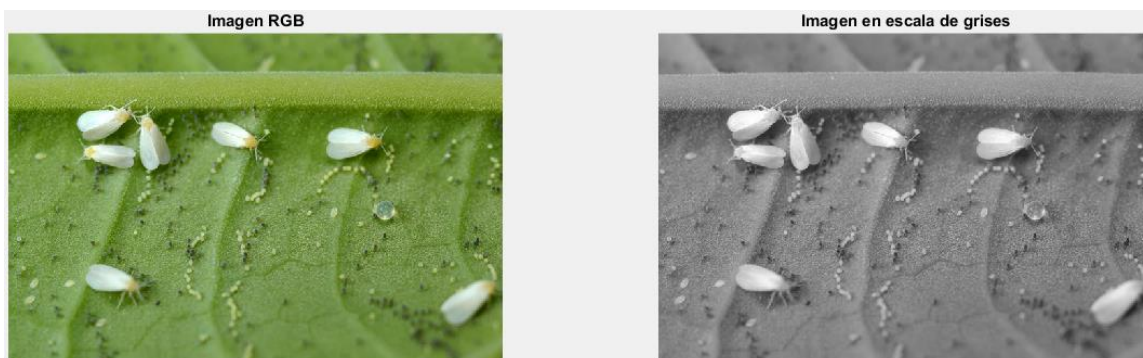


Figura 13. Resultados muestra de imagen RGB y escala de grises. (Autor).



Figura 14. Resultado de binarización inicial. (Autor).

En la segunda etapa (Figura15), es necesario remover el ruido que muestra la imagen binarizada para que exponga la información de la plaga y no demás elementos del entorno, para lograr dicha condición se incorpora un filtro por rango que a partir de la imagen a escala de grises exhibe los contornos de la imagen seguida de una binarización como se evidencia en la Figura 16 y una técnica de técnica de Sobel para la detección de bordes, lo que permite identificar las texturas adicionales presentes en la hoja de plátano.

La imagen Sobel contiene poca información de la plaga razón por la cual se adiciona la operación morfológica de dilatación, que permite reconstruir parte de la información de la imagen. Para su ejecución se requiere la asignación de un elemento estructurante; dicho elemento varía en cuanto a su forma y de acuerdo a las pruebas realizadas el elemento tipo disco añade pixeles de forma propicia a los bordes que han sido encontrados, complementando la información de identificación de la plaga mosca blanca (Figura 17).

```

D:\Ingeniería Electrónica\Opción de grado\Proyecto de investigación Mosca blanca\Scripts Matlab\Detection_algorithm.m
EDITOR PUBLISH VIEW
37 %Rango local de imagen
38
39 - RangoLocal_Hp = rangefilt(Gris_Hp);
40
41 - Bi_RangoLocal_Hp = imbinarize(RangoLocal_Hp, 0.2);
42 - subplot(1,3,2)
43 - imshow(Bi_RangoLocal_Hp)
44 - title 'Binarización imagen de rango local'
45 - %
46 - _____
47 % % Deteccion de bordes
48
49 % % Sobel
50
51 - Sobel_Hp = edge(Gris_Hp,'sobel');
52
53 %Operaciones Morfológicas
54
55
56 % Características elementos estructurantes
57
58 - Elemento1_Hp = strel('disk',5);
59 - Dilatacion_Sobel_Hp = imdilate(Sobel_Hp,Elemento1_Hp);
60 - subplot(1,3,3)
61 - imshow(Dilatacion_Sobel_Hp)
62 - title 'Sobel + Dilatación'

```

Figura 15. Filtro de rango, Sobel y dilatación. (Autor).

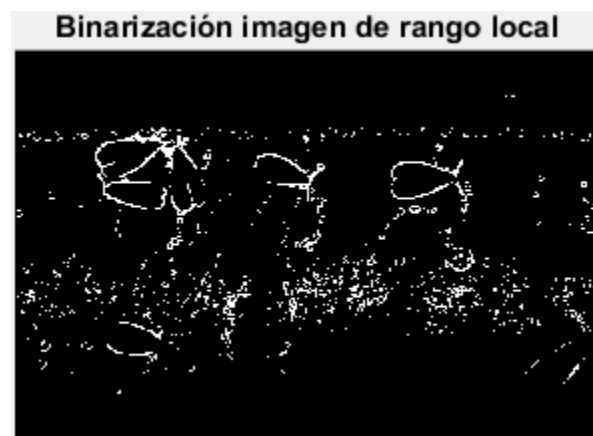


Figura 16. Resultado imagen binarizada a partir de la imagen de rango local. (Autor).



Figura 17. Resultado aplicación de técnicas Sobel y dilatación. (Autor).

Las imágenes binarizadas en las instrucciones anteriores aún necesitan ajustes para la correcta identificación de la mosca blanca lo que implica la adición de compuertas al algoritmo como lo muestra la Figura 18.

La imagen binarizada inicial, la imagen binarizada por rango local y la imagen con técnicas de Sobel y dilatación se evaluarán mediante compuertas AND para dar origen a una nueva imagen que contendrá los elementos de valor 1 que coincidan en las tres imágenes, los demás píxeles serán descartados. Posteriormente se implementan compuertas OR dando origen a una imagen que mostrara aquellos píxeles de valor 1 que se encuentren en cualquiera de las imágenes. Dicha implementación de compuertas se expone en la Figura 19.

Con el fin de completar la información entregada en la imagen de la plaga mosca blanca es adicionada una técnica de llenado de orificios (Figura 20).


```

D:\Ingeniería Electrónica\Opción de grado\Proyecto de investigación Mosca blanca\Scripts Matlab\Detection_algorithm.m
EDITOR PUBLISH VIEW
67
68 % % Compuertas
69
70 % % AND
71
72 - AND1_Hp = and(Bi_RangoLocal_Hp, Dilatacion_Sobel_Hp);
73 - AND2_Hp = and(Bi_RangoLocal_Hp, Bi_Local_Hp);
74 - AND3_Hp = and(Dilatacion_Sobel_Hp, Bi_Local_Hp);
75
76
77 - figure
78 - subplot (2,3,1)
79 - imshow(AND1_Hp)
80 - title 'AND #1'
81
82 - subplot (2,3,2)
83 - imshow(AND2_Hp)
84 - title 'AND #2'
85
86 - subplot (2,3,3)
87 - imshow(AND3_Hp)
88 - title 'AND #3'
89
90
91 % % OR
92 - OR1_Hp = or(AND1_Hp, AND2_Hp);
93 - OR2_Hp = or(OR1_Hp, AND3_Hp);
94 - subplot (2,3,4)
95 - imshow(OR2_Hp)
96 - title 'Combinación de Compuertas AND - OR'

```

Figura 18. Aplicación de compuertas lógicas. (Autor).

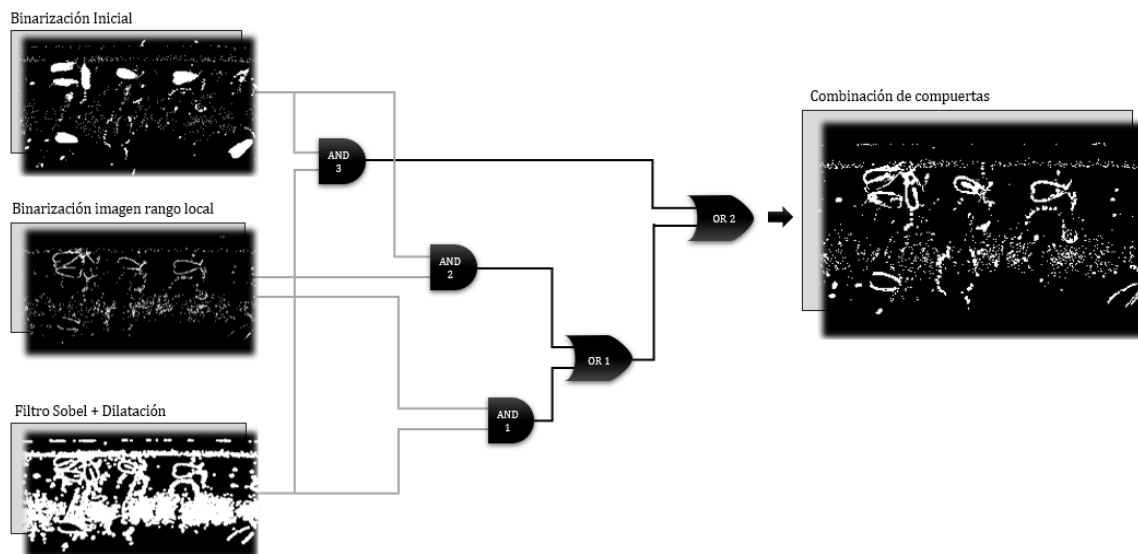
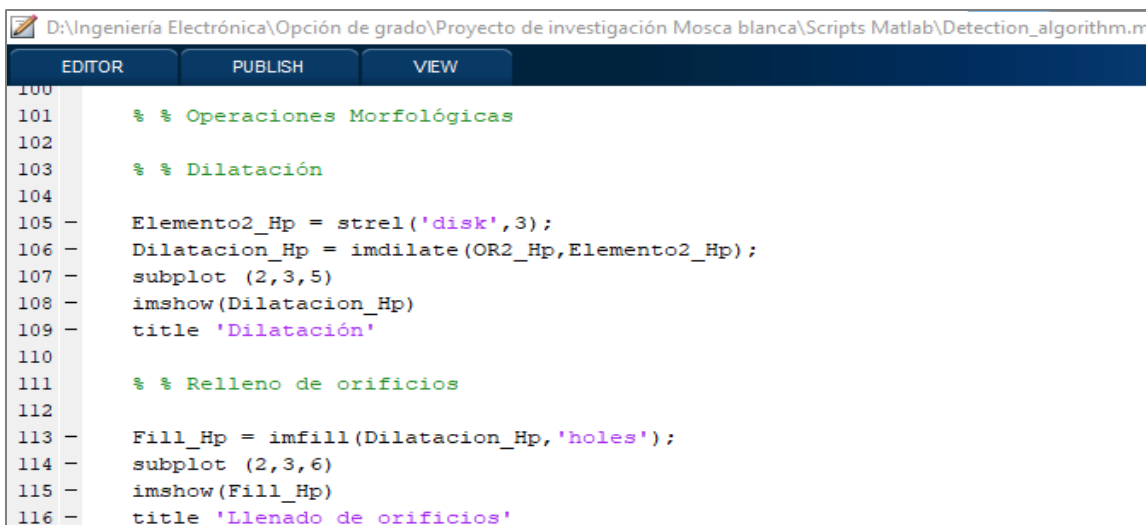


Figura 19. Resultado aplicación de compuertas. (Autor).



```

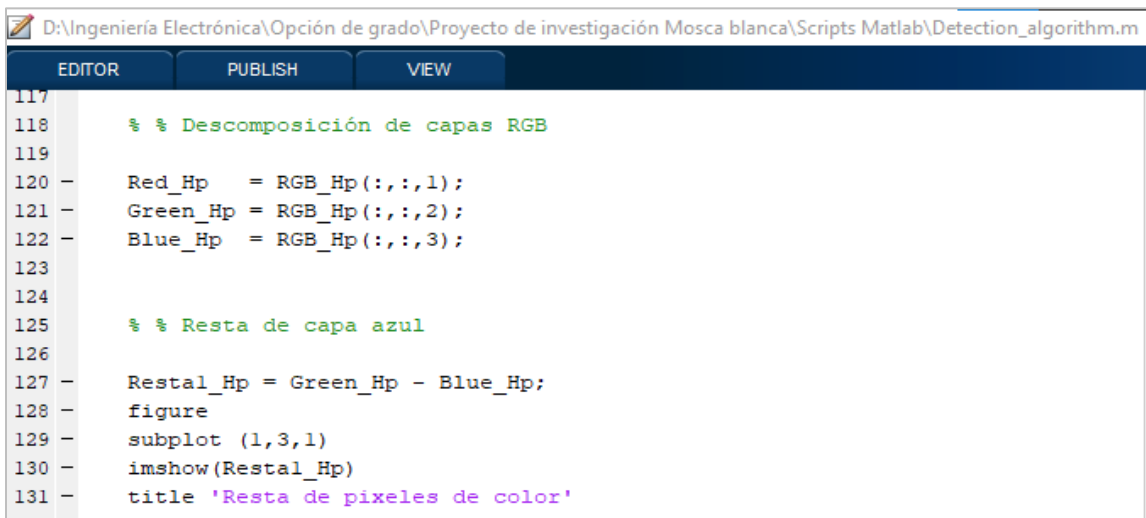
D:\Ingeniería Electrónica\Opción de grado\Proyecto de investigación Mosca blanca\Scripts Matlab\Detection_algorithm.m
EDITOR PUBLISH VIEW
100
101     %% Operaciones Morfológicas
102
103     %% Dilatación
104
105     Elemento2_Hp = strel('disk',3);
106     Dilatacion_Hp = imdilate(OR2_Hp,Elemento2_Hp);
107     subplot (2,3,5)
108     imshow(Dilatacion_Hp)
109     title 'Dilatación'
110
111     %% Relleno de orificios
112
113     Fill_Hp = imfill(Dilatacion_Hp,'holes');
114     subplot (2,3,6)
115     imshow(Fill_Hp)
116     title 'Llenado de orificios'

```

Figura 20. Operaciones morfológicas de llenado de orificios y dilatación a la imagen resultante.

(Autor).

En la tercera etapa, Figura 21, se descompondrá la imagen RGB en sus tres capas con el objetivo de realizar una resta de píxeles. Después de efectuar las restas entre capas, la resta de los píxeles azules fue la que permitió la eliminación del fondo con mayor entrega de información de la plaga, como lo expone la Figura 22.



```

D:\Ingeniería Electrónica\Opción de grado\Proyecto de investigación Mosca blanca\Scripts Matlab\Detection_algorithm.m
EDITOR PUBLISH VIEW
117
118     %% Descomposición de capas RGB
119
120     Red_Hp = RGB_Hp(:,:,1);
121     Green_Hp = RGB_Hp(:,:,2);
122     Blue_Hp = RGB_Hp(:,:,3);
123
124
125     %% Resta de capa azul
126
127     Restal_Hp = Green_Hp - Blue_Hp;
128     figure
129     subplot (1,3,1)
130     imshow(Restal_Hp)
131     title 'Resta de píxeles de color'

```

Figura 21. Descomposición de capas y resta capa azul. (Autor).

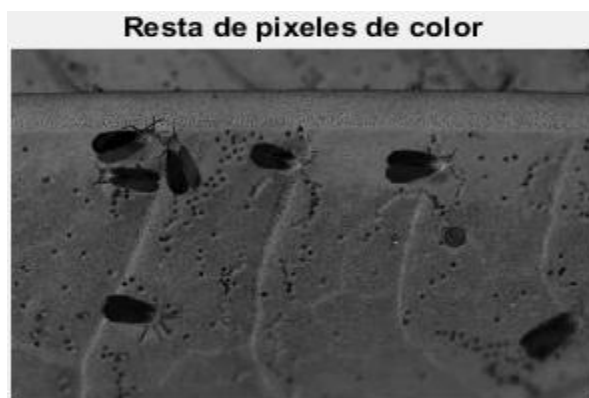


Figura 22. Resultado resta de pixeles azules. (Autor)

A la imagen restada se le realizará una umbralización adaptativa y posteriormente se binarizará (Figura 23), dicha imagen resultante ha tomado los valores correspondientes a la plaga como 0 y el fondo como 1, por esta razón es necesario negar la imagen para que mantenga los parámetros de segmentación de igual forma que con las imágenes procesadas (Figura 24). La operación morfológica de dilatación es aplicada a la imagen binarizada con un elemento estructurante de tipo disco de dos pixeles para complementar la información obtenida en la etapa de binarización.

```

D:\Ingeniería Electrónica\Opción de grado\Proyecto de investigación Mosca blanca\Scripts Matlab\Detection_algorithm.m
EDITOR PUBLISH VIEW
134 %% Umbralización imagen restada
135
136 - Aj_2_Local_Hp = adaptthresh(Restal_Hp, 0.8);
137 - Bi_2_Local_Hp = imbinarize (Restal_Hp, Aj_2_Local_Hp);
138
139 - Neg_Bi_2_Local_Hp = ~Bi_2_Local_Hp;
140 - subplot (1,3,2)
141 - imshow(Neg_Bi_2_Local_Hp)
142 - title 'Umbralización local (Imagen restada)'
143
144
145 %% Dilatación imagen negada
146
147 - Elemento3_Hp = strel('disk',2);
148 - Dilatacion_Neg_Bi_2_Local_Hp = imdilate(Neg_Bi_2_Local_Hp,Elemento3_Hp);

```

Figura 23. Umbralización y binarización imagen restada. (Autor).

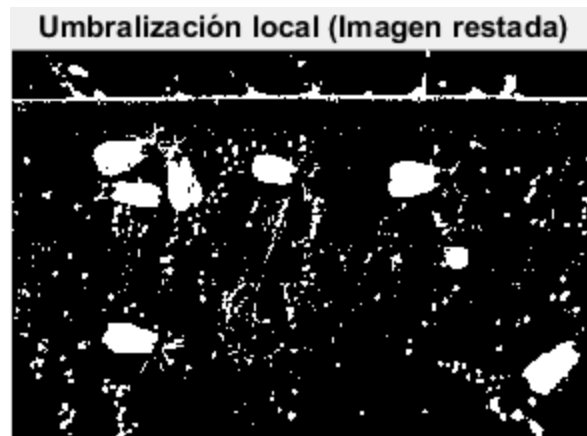


Figura 24. Resultados umbralización, binarización y negación de la imagen restada. (Autor).

Finalmente se realiza una unión de procesamientos (Figura 25), con la imagen resultante de la implementación de las compuertas y la imagen binarizada de la resta de píxeles azules (Después de la aplicación de las operaciones morfológicas), mediante una multiplicación donde solo se conservarán los píxeles con valor 1, como se evidencia en la Figura 26. Para la eliminación de ruido de la imagen se ha incorporado un filtro de remoción de píxeles.

```

D:\Ingeniería Electrónica\Opción de grado\Proyecto de investigación Mosca blanca\Scripts Matlab\Detection_algorithm.m
EDITOR PUBLISH VIEW
151 % % Multiplicación imagen con relleno de orificios y negada
152
153 - Fill_Dil_Hp = Fill_Hp .* Dilatacion_Neg_Bi_2_Local_Hp;
154 - subplot (1,3,3)
155 - imshow(Fill_Dil_Hp)
156 - title 'Unión de procesamientos'
157
158 - Filtro_Hp = bwareaopen(Fill_Dil_Hp,80);

```

Figura 25. Aplicación de dilatación, llenado de orificios y filtro de eliminación de píxeles a la imagen restada. (Autor).



Figura 26. Resultado unión de procesamientos. (Autor).

En la cuarta etapa será necesaria una conversión del tipo de dato como muestra la Figura 27, para efectuar una multiplicación de píxeles de las diferentes capas de la imagen RGB dando origen a la imagen a la cual se le extraerá los valores mínimos y máximos de los ocho bits de los valores de la matriz de la imagen mediante un ciclo condicional “for”.

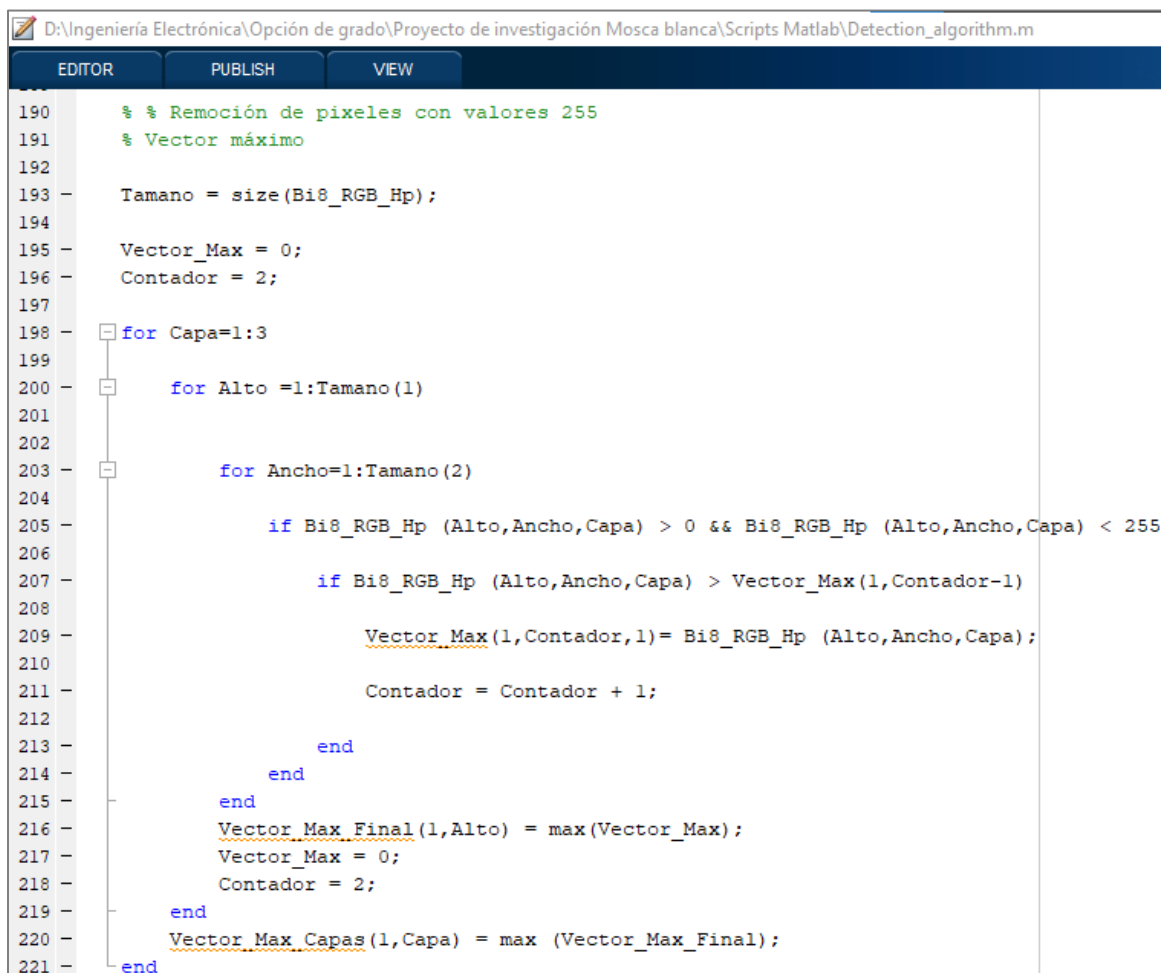
```

Editor - D:\Ingeniería Electrónica\Opción de grado\Proyecto de investigación Mosca blanca\Scripts Matlab\Detection_algorithm.m
EDITOR PUBLISH VIEW
Detection_algorithm.m x +
162 % % Conversión de imagen binarizada a uint8
163
164 - Bi_8 = uint8(Filtro_Hp);
165
166 % % Multiplicación de imagen binarizada y capa R
167
168 - Bi8_RGB_Hp(:, :, 1) = immultiply(Bi_8, Red_Hp);
169
170
171 % % Multiplicación de imagen binarizada y capa G
172
173 - Bi8_RGB_Hp(:, :, 2) = immultiply(Bi_8, Green_Hp);
174
175
176 % % Multiplicación de imagen binarizada y capa B
177
178 - Bi8_RGB_Hp(:, :, 3) = immultiply(Bi_8, Blue_Hp);
179 - figure
180 - imshow(Bi8_RGB_Hp)
181 - title 'Segmentación final'

```

Figura 27. Conversión de tipo de dato y multiplicación por capas. (Autor).

Mediante el ciclo “for” se hará un recorrido por la matriz de pixeles donde inicialmente se tomará el valor máximo por cada fila de la matriz de cada capa y posteriormente se tomará el valor máximo encontrado en toda la matriz, es decir que se tendrán tres valores máximos como resultantes, uno por la capa roja, uno por la capa verde y uno por la capa azul (Figura 28). Este mismo proceso se efectúa para los valores mínimos (Figura 29). Con el fin de realizar un correcto análisis de los datos se excluyeron los valores de pixeles 1 y 255.

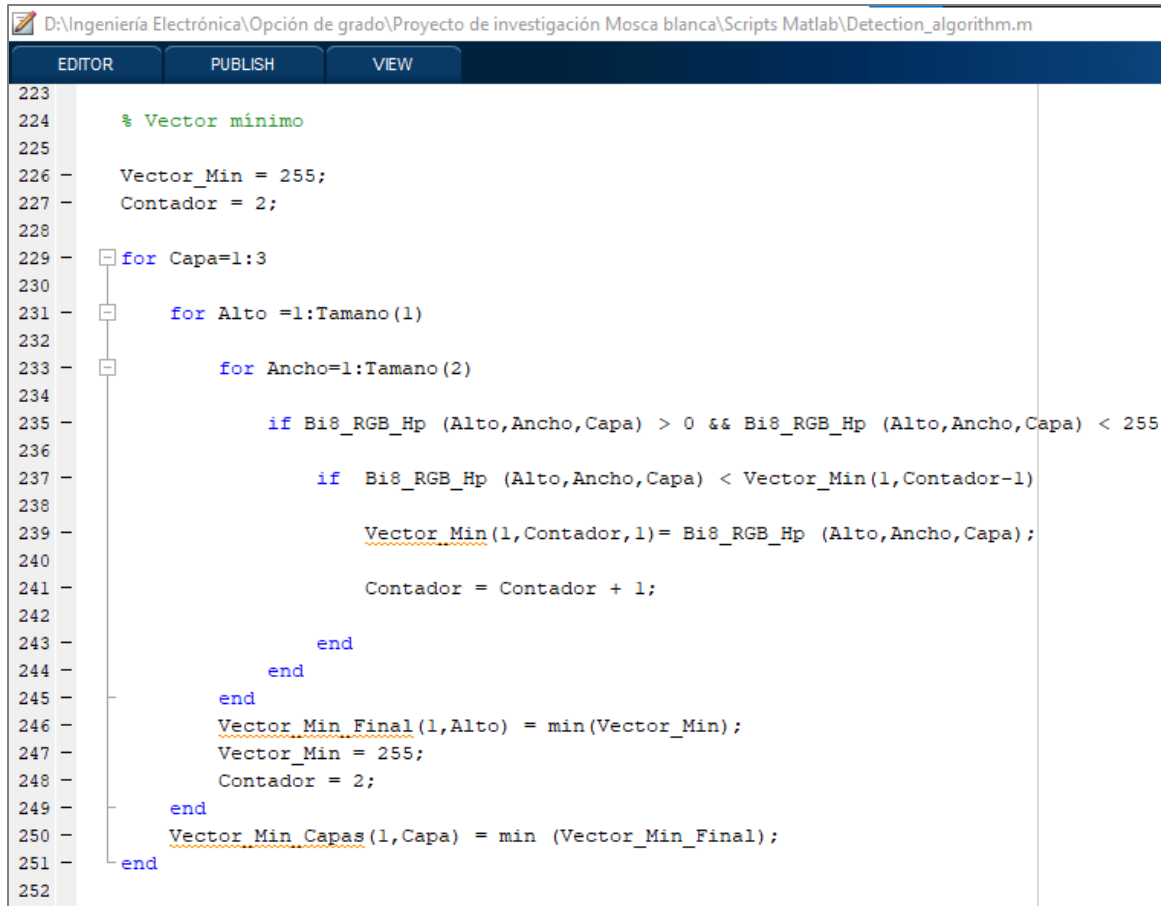


```

190 % % Remoción de pixeles con valores 255
191 % Vector máximo
192
193 - Tamano = size(Bi8_RGB_Hp);
194
195 - Vector_Max = 0;
196 - Contador = 2;
197
198 - for Capa=1:3
199
200 -     for Alto =1:Tamano(1)
201
202
203 -         for Ancho=1:Tamano(2)
204
205 -             if Bi8_RGB_Hp (Alto,Ancho,Capa) > 0 && Bi8_RGB_Hp (Alto,Ancho,Capa) < 255
206
207 -                 if Bi8_RGB_Hp (Alto,Ancho,Capa) > Vector_Max(1,Contador-1)
208
209 -                     Vector_Max(1,Contador,1)= Bi8_RGB_Hp (Alto,Ancho,Capa);
210
211 -                     Contador = Contador + 1;
212
213 -                 end
214 -             end
215 -         end
216 -         Vector_Max_Final(1,Alto) = max(Vector_Max);
217 -         Vector_Max = 0;
218 -         Contador = 2;
219 -     end
220 -     Vector_Max_Capas(1,Capa) = max (Vector_Max_Final);
221 - end

```

Figura 28. Determinación de vector de valores máximos. (Autor).



```

223
224 % Vector mínimo
225
226 - Vector_Min = 255;
227 - Contador = 2;
228
229 - for Capa=1:3
230
231 -     for Alto =1:Tamano(1)
232
233 -         for Ancho=1:Tamano(2)
234
235 -             if Bi8_RGB_Hp (Alto,Ancho,Capa) > 0 && Bi8_RGB_Hp (Alto,Ancho,Capa) < 255
236
237 -                 if Bi8_RGB_Hp (Alto,Ancho,Capa) < Vector_Min(1,Contador-1)
238
239 -                     Vector_Min(1,Contador,1)= Bi8_RGB_Hp (Alto,Ancho,Capa);
240
241 -                     Contador = Contador + 1;
242
243 -                 end
244 -             end
245 -         end
246 -         Vector_Min_Final(1,Alto) = min(Vector_Min);
247 -         Vector_Min = 255;
248 -         Contador = 2;
249 -     end
250 -     Vector_Min_Capas(1,Capa) = min (Vector_Min_Final);
251 - end
252

```

Figura 29. Determinación de vector de valores mínimos. (Autor).

Los valores obtenidos en el vector de máximos y mínimos serán utilizados para determinar las características de análisis del algoritmo. Dichas características son el promedio, rango, desviación estándar y varianza, calculados por el software Matlab como se indica en la Figura 30.

```

254
255 % % Promedio
256
257 Promedio_Max = mean(Vector_Max_Capas);
258 Promedio_Min = mean(Vector_Min_Capas);
259
260
261 % % Rango
262
263 Rango_Max = range(Vector_Max_Capas);
264 Rango_Min = range(Vector_Min_Capas);
265
266
267 % % Desviación estándar
268
269 Desv_STD_Max = std(Vector_Max_Capas);
270 Desv_STD_Min = std(Vector_Min_Capas);
271
272
273 % % Varianza
274
275 Varianza_Max = var(Vector_Max_Capas);
276 Varianza_Min = var(Vector_Min_Capas);
277

```

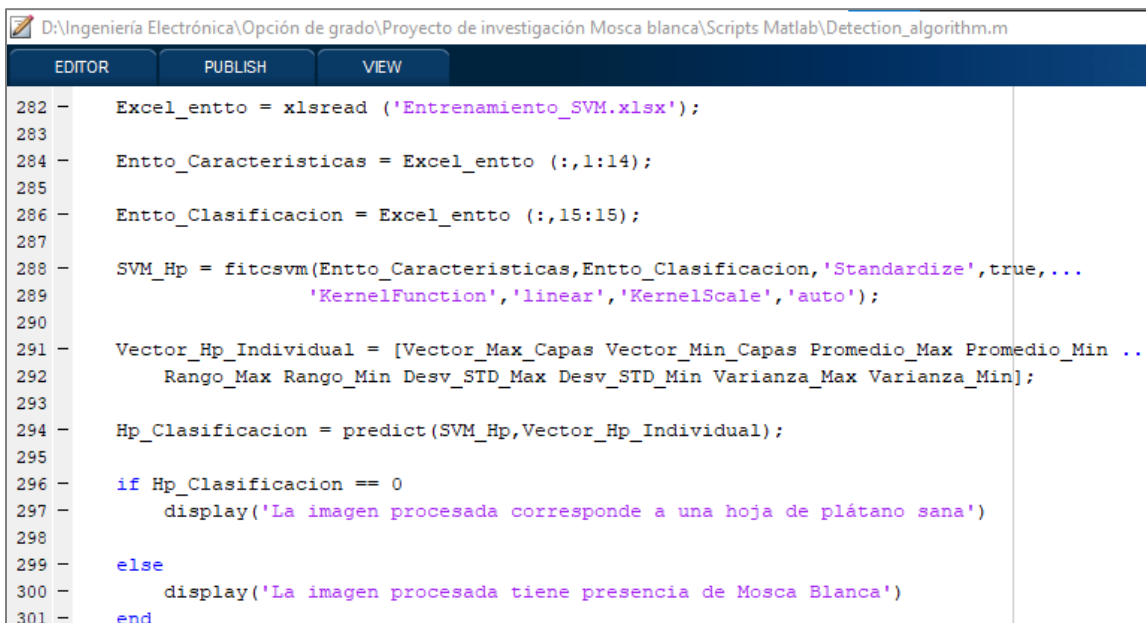
Figura 30. Cálculo de características de análisis. (Autor).

En la quinta etapa se ejecutará el entrenamiento del sistema de detección mediante una base de datos elaborada en Excel con cinco imágenes enfermas y cinco imágenes sanas de hoja de plátano, dichas imágenes fueron obtenidas en internet. En este documento de Excel se definirá las características de clasificación de acuerdo a cada imagen, definiendo el valor 0 para las hojas sanas y el valor 1 para las hojas con presencia de plaga mosca blanca (Figura 31).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Imagen	Máx_R	Máx_G	Máx_B	Min_R	Min_G	Min_B	Prom_Máx	Prom_Min	Rango_Máx	Rango_Min	Desv_Std_Máx	Desv_Std_Min	Var_Máx	Var_Min	Clasificación		
2	Hoja_Sana_1	178	252	209	44	98	74	213	72	74	54	37.16180835	27.05549852	1381	732	0		
3	Hoja_Sana_2	246	254	178	93	107	29	226	76.33333333	76	78	41.76122604	41.5852538	1.74E+03	1.73E+03	0		
4	Hoja_Sana_3	96	152	53	27	85	1	100.33333333	37.66666667	99	84	49.64205207	43.00387579	2.46E+03	1.85E+03	0		
5	Hoja_Sana_4	0	0	0	255	255	255	0	255	0	0	0	0	0	0	0		
6	Hoja_Sana_5	232	252	236	42	79	1	240	40.66666667	20	78	10.58300524	39.01709027	112	1522.333333	0		
7	Mosca_Blanca_1	254	254	254	26	36	1	254	21	0	35	0	18.02775638	0	325	1		
8	Mosca_Blanca_2	212	231	211	70	95	55	218	73.33333333	20	40	11.26942767	20.20725942	127	408.3333333	1		
9	Mosca_Blanca_3	182	227	177	56	79	36	195.33333333	57	50	43	27.53785274	21.51743479	758.3333333	463	1		
10	Mosca_Blanca_4	250	238	238	23	29	2	242	18	12	27	6.92820323	14.17744688	48	201	1		
11	Mosca_Blanca_5	235	247	178	79	90	27	220	65.33333333	69	63	36.86461718	33.65016097	1359	1132.333333	1		

Figura 31. Excel de entrenamiento de la máquina de soporte vectorial. (Autor).

Para el aprendizaje supervisado fue implementada una máquina de soporte vectorial (SVM – Support Vector Machine) que es de mucha utilidad en problemas binarios como es el caso de la detección de plaga mosca blanca, ya que se requiere definir si hay o no presencia de la plaga. El algoritmo genera un hiperplano a partir del cual separa los datos asignándoles un espacio dimensional mediante funciones denominadas Kernel, que para el caso específico del algoritmo de detección es de tipo lineal que contribuye a la clasificación entre dos clases (Figura 32).



```

D:\Ingeniería Electrónica\Opción de grado\Proyecto de investigación Mosca blanca\Scripts Matlab\Detection_algorithm.m
EDITOR PUBLISH VIEW
282 - Excel_entto = xlsread ('Entrenamiento_SVM.xlsx');
283
284 - Entto_Caracteristicas = Excel_entto (:,1:14);
285
286 - Entto_Clasificacion = Excel_entto (:,15:15);
287
288 - SVM_Hp = fitsvm(Entto_Caracteristicas,Entto_Clasificacion,'Standardize',true,...
289                 'KernelFunction','linear','KernelScale','auto');
290
291 - Vector_Hp_Individual = [Vector_Max_Capas Vector_Min_Capas Promedio_Max Promedio_Min ...
292                         Rango_Max Rango_Min Desv_STD_Max Desv_STD_Min Varianza_Max Varianza_Min];
293
294 - Hp_Clasificacion = predict(SVM_Hp,Vector_Hp_Individual);
295
296 - if Hp_Clasificacion == 0
297 -     display('La imagen procesada corresponde a una hoja de plátano sana')
298
299 - else
300 -     display('La imagen procesada tiene presencia de Mosca Blanca')
301 - end

```

Figura 32. Entrenamiento – Máquina de soporte vectorial. (Autor).

Para que pueda implementarse la SVM se requiere que el software Matlab lea la base de datos de entrenamiento en Excel. Iniciando con la lectura de las características alojadas en las primeras 14 columnas, seguido de la lectura de la columna 15 que es la que define la clasificación de la imagen.

La SVM requiere la base de datos de entrenamiento y una base de datos de prueba, también desarrollada en Excel, donde se alojan datos de nueve imágenes sanas y nueve imágenes con presencia de plaga como se muestra en la Figura 33.

Imagen	Máx_R	Máx_G	Máx_B	Min_R	Min_G	Min_B	Prom_Máx	Prom_Min	Rango_Máx	Rango_Min	Desv_Std_Máx	Desv_Std_Min	Var_Máx	Var_Min
Hoja_Sana_6	0	0	0	255	255	255	0	255	0	0	0	0	0	0
Hoja_Sana_7	0	0	0	255	255	255	0	255	0	0	0	0	0	0
Hoja_Sana_8	215	233	182	135	160	103	210	132,66667	51	57	25,86503431	28,57154762	669	816,33333
Hoja_Sana_9	0	0	0	255	255	255	0	255	0	0	0	0	0	0
Hoja_Sana_10	198	245	179	137	204	93	207,33333	144,66667	66	111	33,97548136	55,89573627	1154,3333	3124,3333
Hoja_Sana_11	0	0	0	255	255	255	0	255	0	0	0	0	0	0
Hoja_Sana_12	149	204	51	54	113	18	134,66667	61,66667	153	95	77,50053763	47,96179035	6006,3333	2300,3333
Hoja_Sana_13	246	254	178	93	107	29	226	76,333333	76	78	41,76122604	41,5852538	1744	1729,3333
Hoja_Sana_14	0	0	0	255	255	255	0	255	0	0	0	0	0	0
Mosca_Blanca_6	254	254	254	1	10	1	254	4	0	9	0	5,196152423	0	27
Mosca_Blanca_7	254	254	254	58	64	32	254	51,333333	0	32	0	17,0098011	0	289,33333
Mosca_Blanca_9	254	254	254	70	84	49	254	67,666667	0	35	0	17,61628035	0	310,33333
Mosca_Blanca_10	216	254	253	78	124	85	241	95,666667	38	46	21,65640783	24,78574859	469	614,33333
Mosca_Blanca_11	254	254	254	1	5	1	254	2,3333333	0	4	0	2,309401077	0	5,3333333
Mosca_Blanca_12	246	250	249	1	12	3	248,33333	5,3333333	4	11	2,081665999	5,859465277	4,3333333	34,3333333
Mosca_Blanca_13	238	210	182	77	73	35	210	61,666667	56	42	28	23,18045153	784	537,33333
Mosca_Blanca_14	254	254	253	133	164	142	253,66667	146,33333	1	31	0,577350269	15,94783162	0,3333333	254,33333
Mosca_Blanca_15	209	222	215	46	61	26	215,33333	44,333333	13	35	6,506407099	17,55942292	42,333333	308,33333

Figura 33. Excel de prueba para la máquina de soporte vectorial. (Autor).

Finalmente, si el valor obtenido en la clasificación corresponde a 0 el usuario recibirá el mensaje de que la imagen procesada es de una hoja sana. Si el valor es 1 el usuario sabrá que la imagen procesada contiene presencia de mosca blanca.

Resultados

Imágenes procesadas

A continuación, se muestran los resultados de cada etapa del procesamiento de las imágenes de hoja de plátano tanto enfermas como sanas.

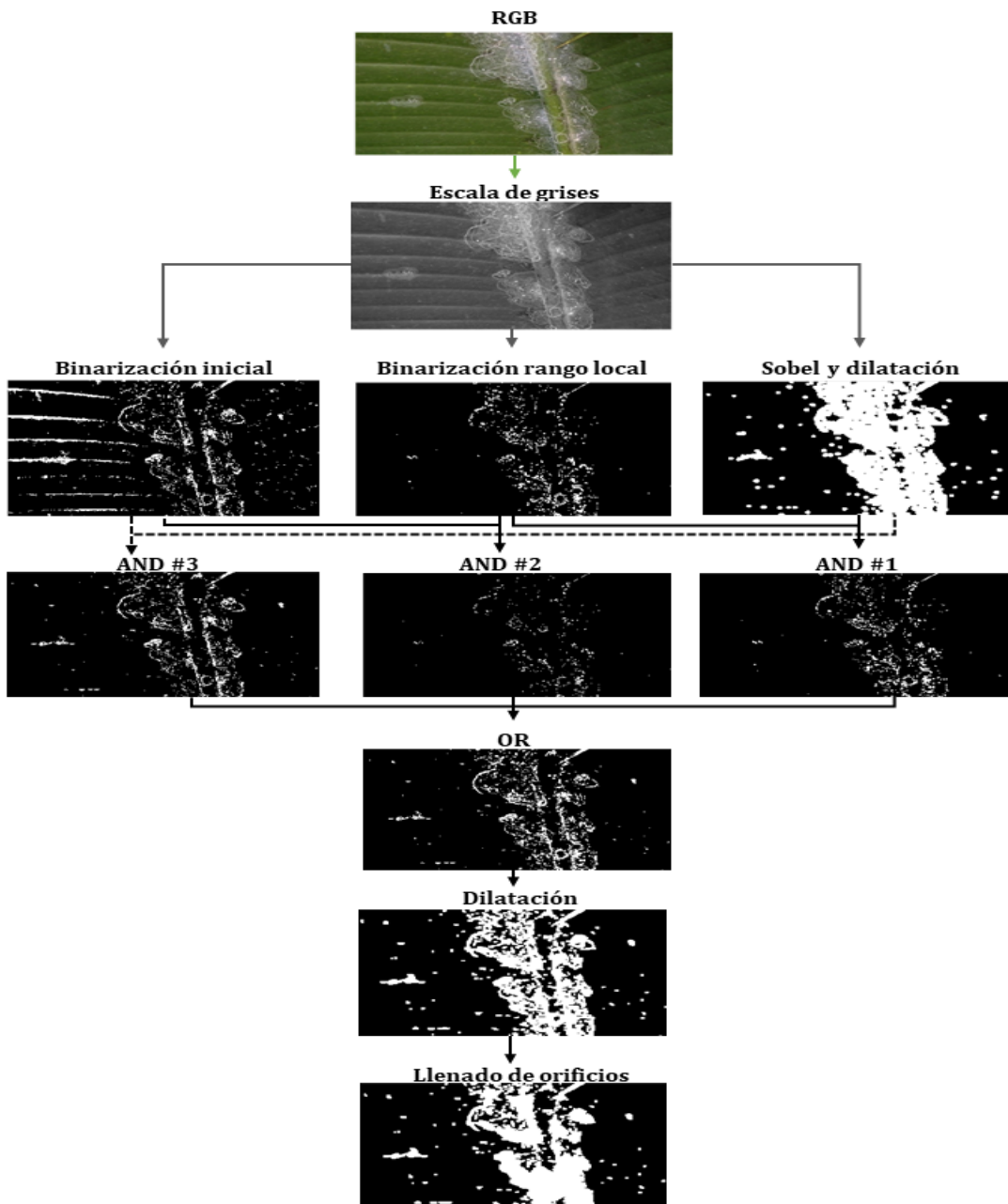


Figura 34. Resultados imagen Mosca_Blanca_4 – Procesamiento 1. (Autor).

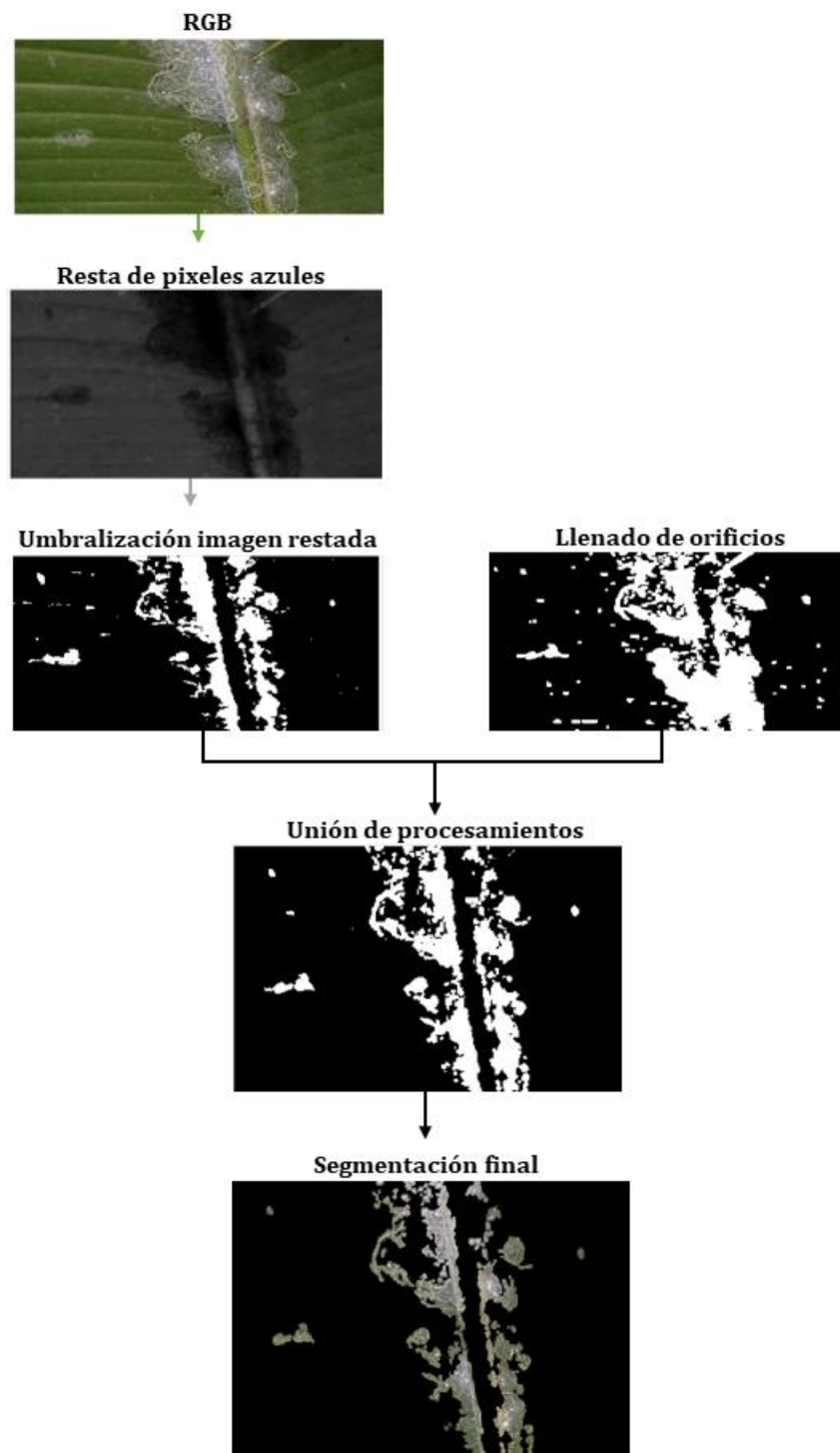


Figura 35. Resultados imagen Mosca_Blanca_4 – Procesamiento 2 e imagen final procesada.

(Autor).

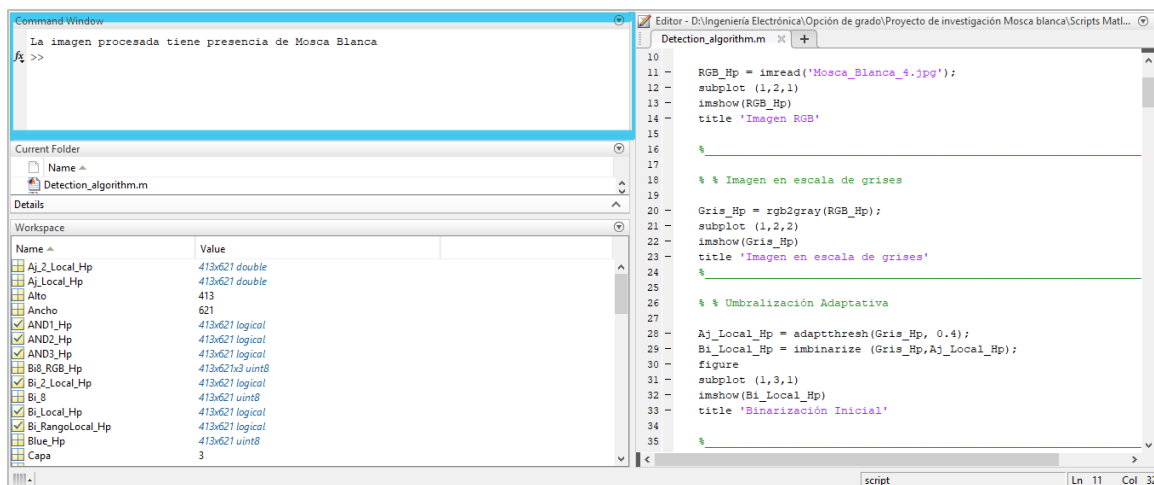


Figura 36. Resultado de clasificación de la imagen Mosca_Blanca_4. (Autor).

En los diagramas de las Figuras 34, Figura 35, Figura 37 y Figura 38 se muestran los pasos de procesamiento de la imagen que conllevan a la segmentación final, donde se observa el área afectada por la plaga, mosca blanca.

Por otra parte, en las Figuras 36 y Figura 39 se evidencia la clasificación de la hoja, que para el caso indica al usuario que la imagen procesada tiene presencia de mosca blanca.

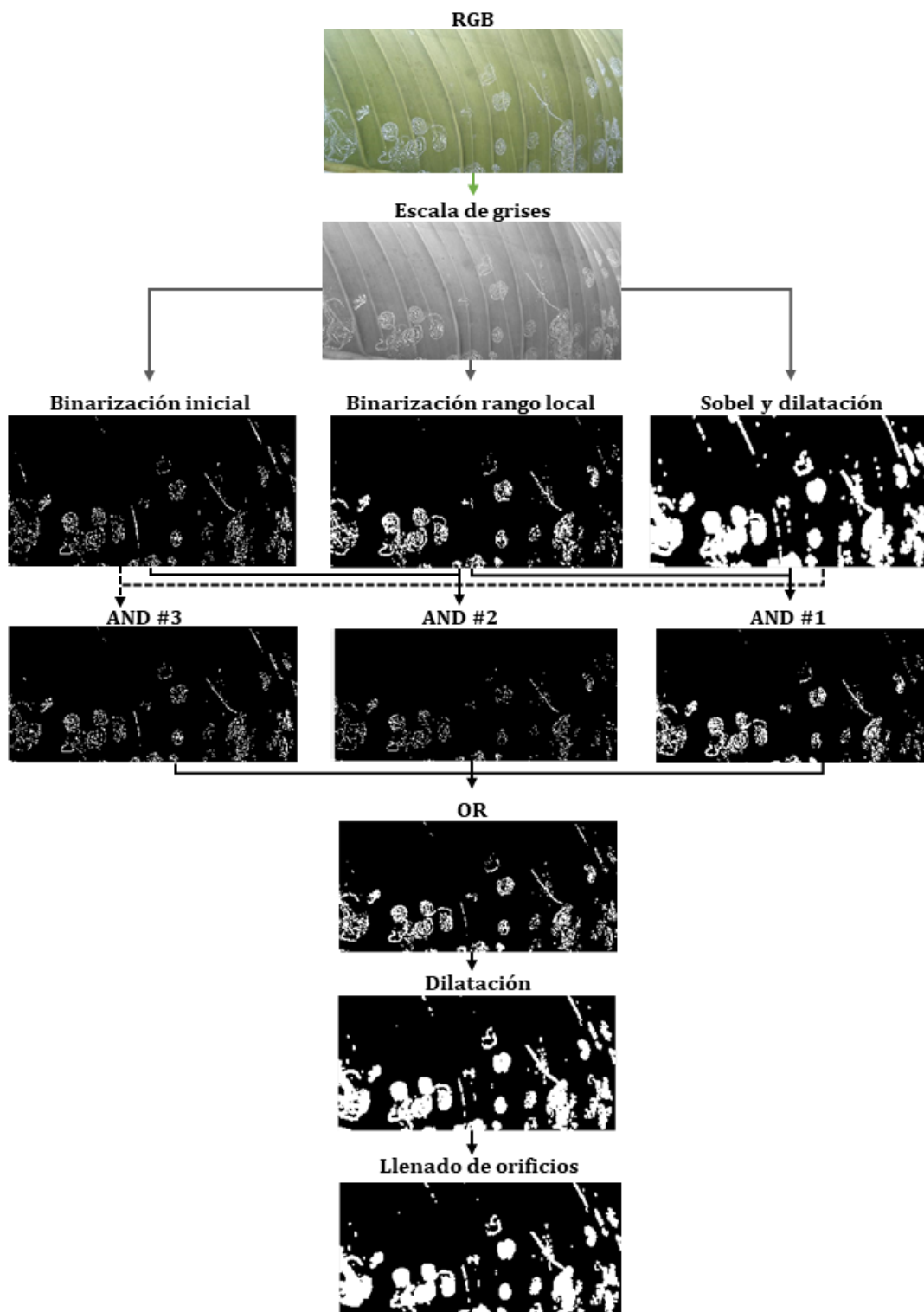


Figura 37. Resultados imagen Mosca_Blanca_9 – Procesamiento 1. (Autor).

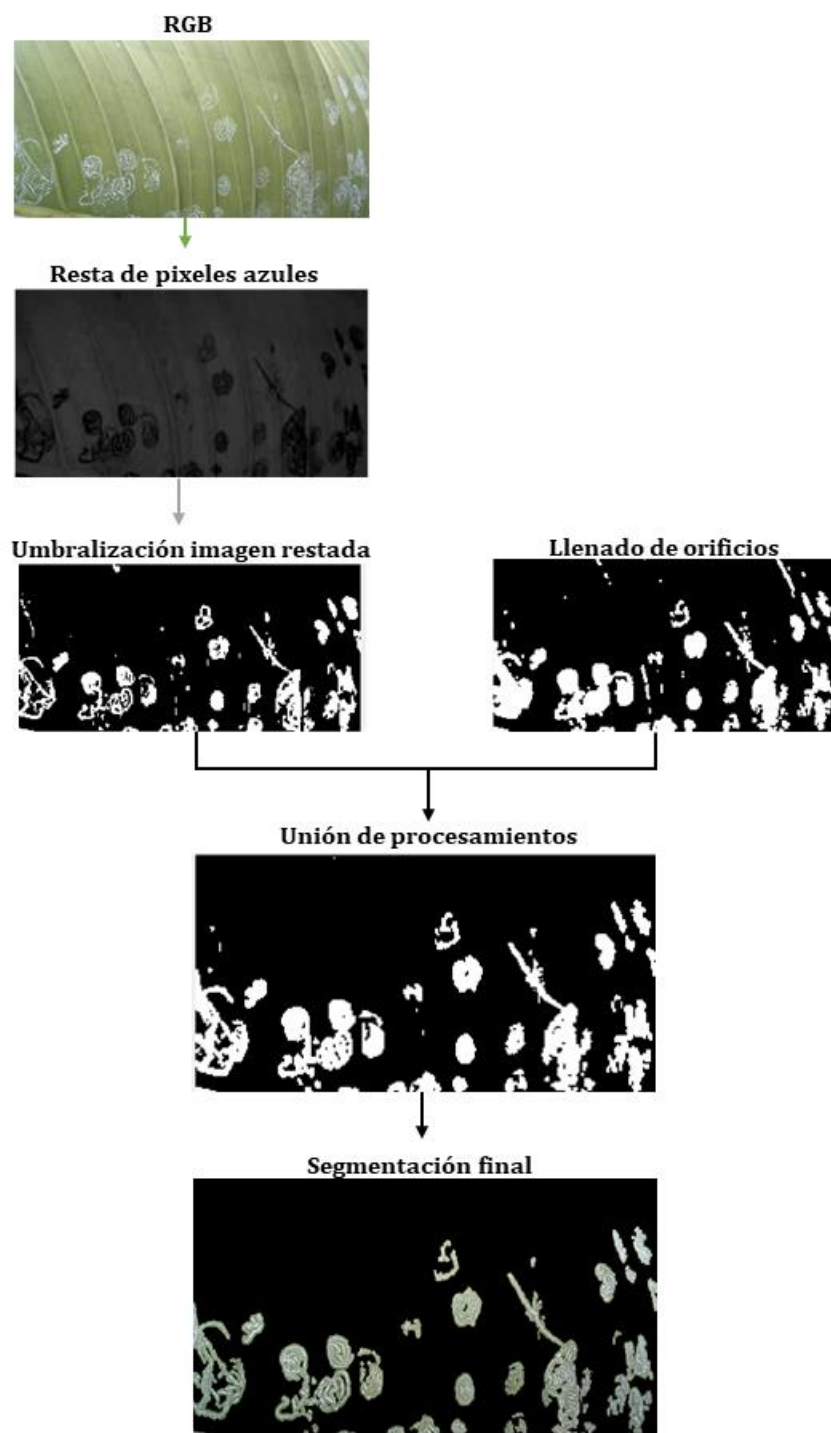


Figura 38. Resultados imagen Mosca_Blanca_9 – Procesamiento 2 e imagen final procesada.

(Autor).

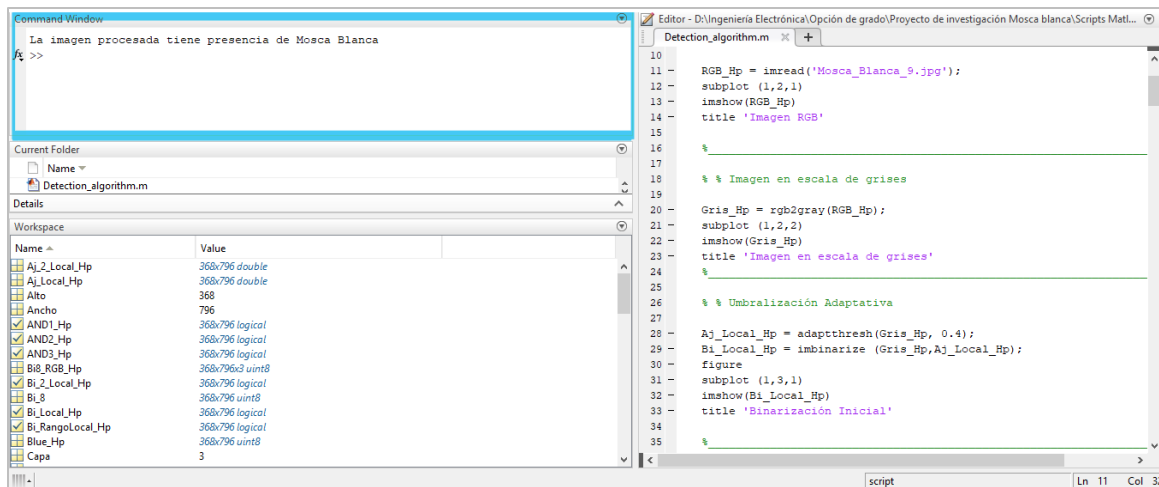


Figura 39. Resultado de clasificación de la imagen Mosca_Blanca_9. (Autor).

Continuando con la muestra de resultados se expone en la Figura 40, Figura 41, Figura 43 y Figura 44, el procesamiento realizado a hojas no afectadas por la plaga, obteniendo una imagen sin ninguna área resaltada.

En cuanto a la clasificación se muestra en la Figura 42 y Figura 45, que la clasificación otorgada es de una hoja sana,

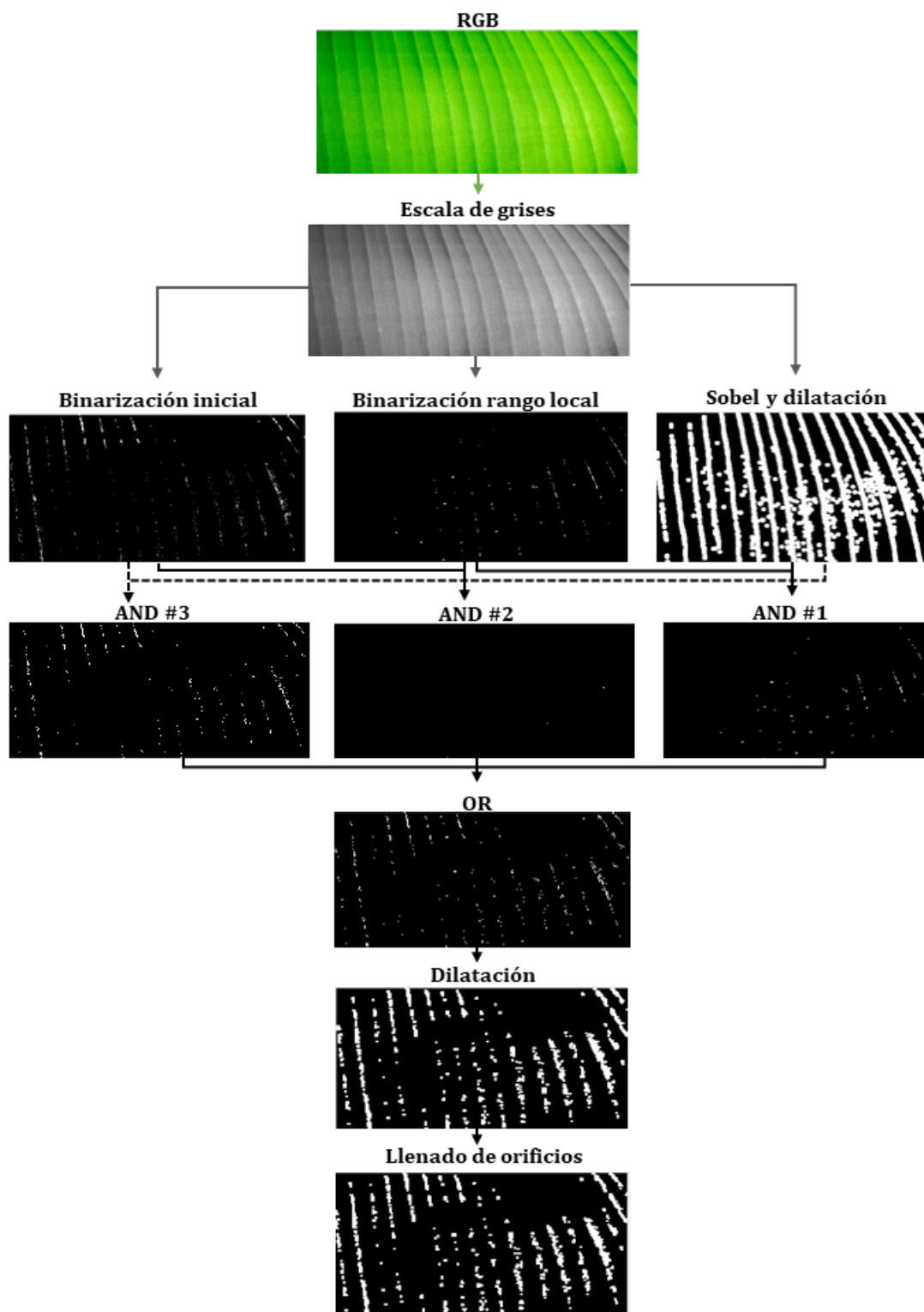


Figura 40. Resultados imagen Hoja_Sana_7 – Procesamiento 1. (Autor).

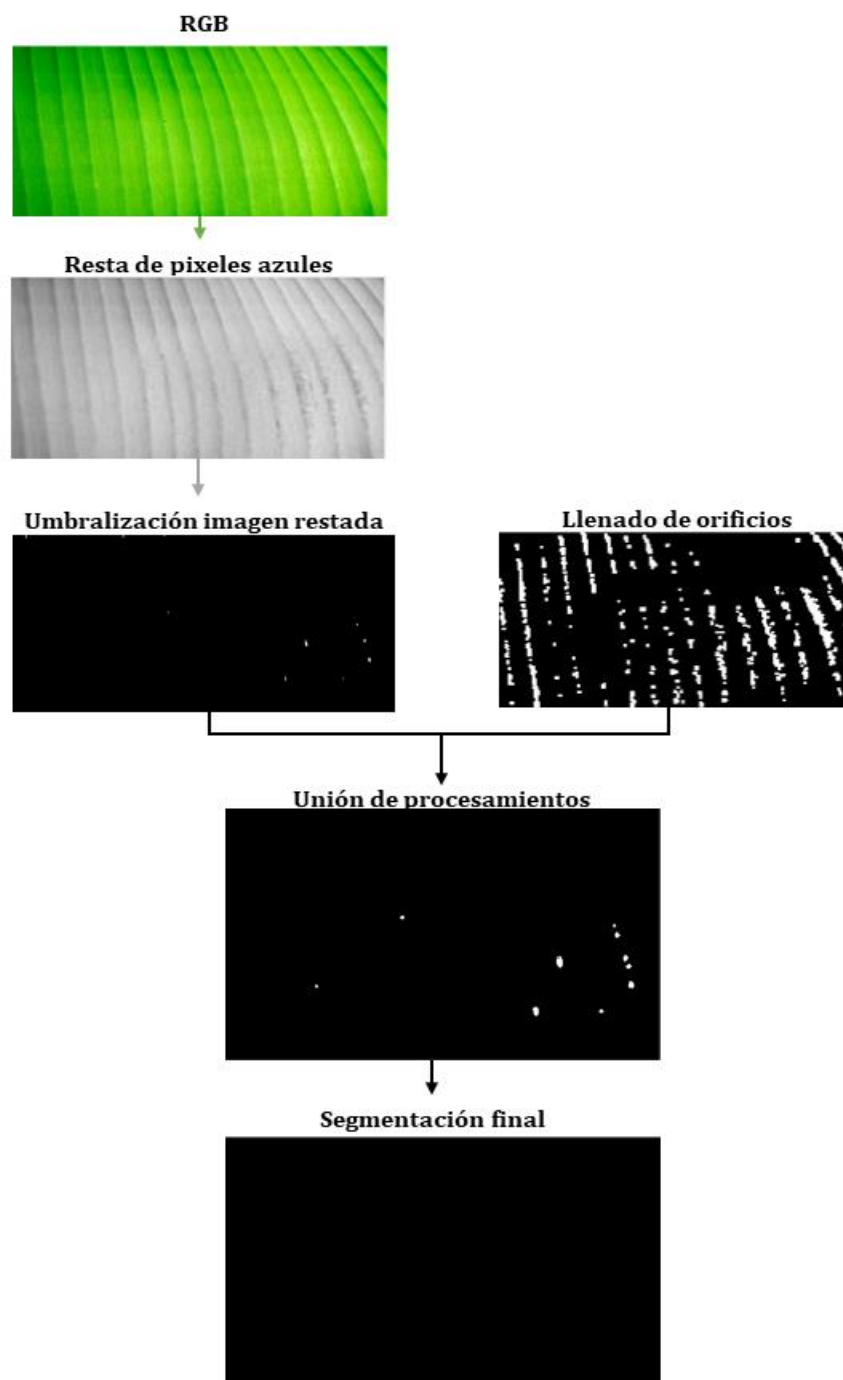


Figura 41. Resultados imagen Hoja_Sana_7 – Procesamiento 2 e imagen final procesada.

(Autor).

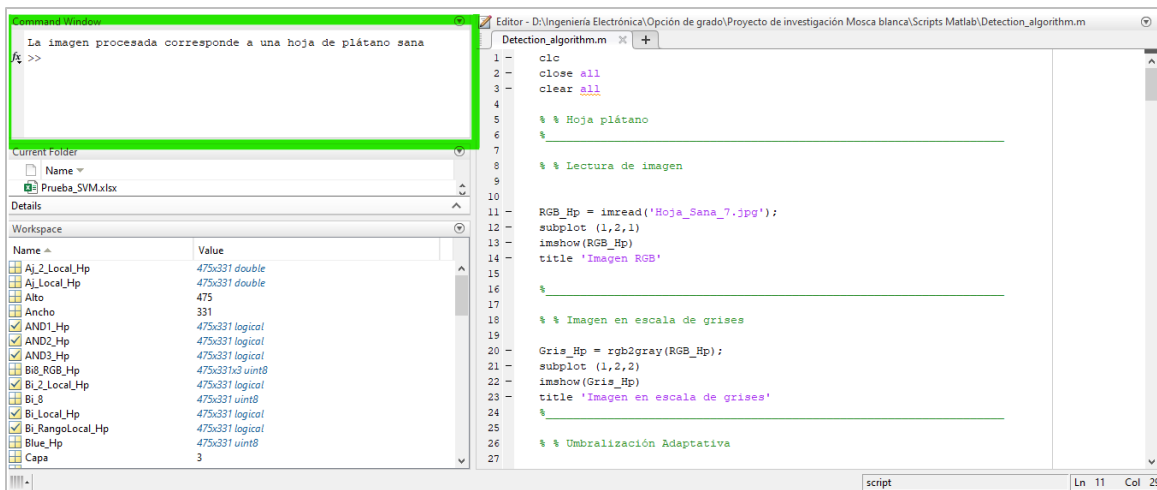


Figura 42. Resultado de clasificación de la imagen Hoja_Sana_7. (Autor).

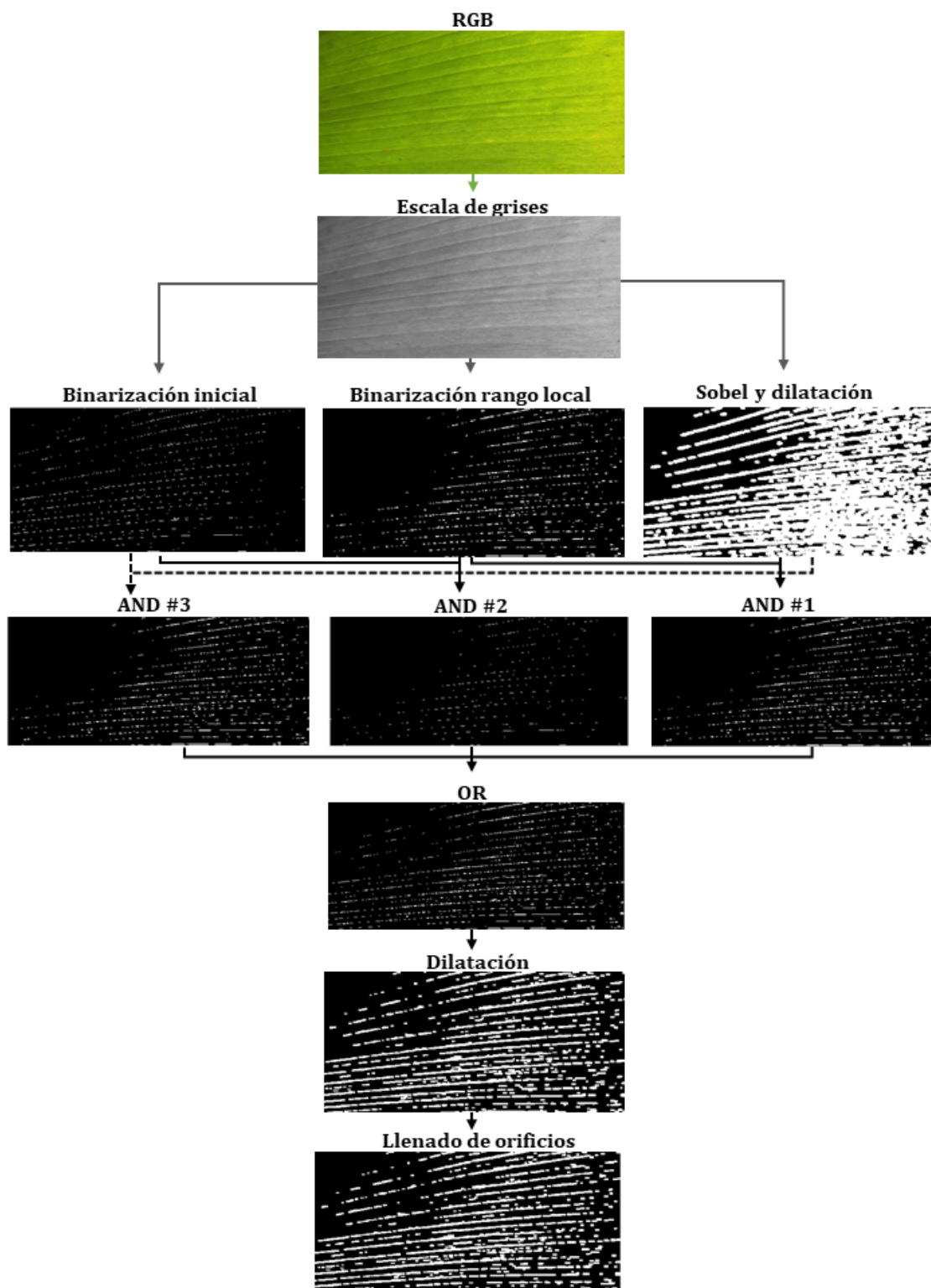


Figura 43. Resultados imagen Hoja_Sana_9 – Procesamiento 1. (Autor).

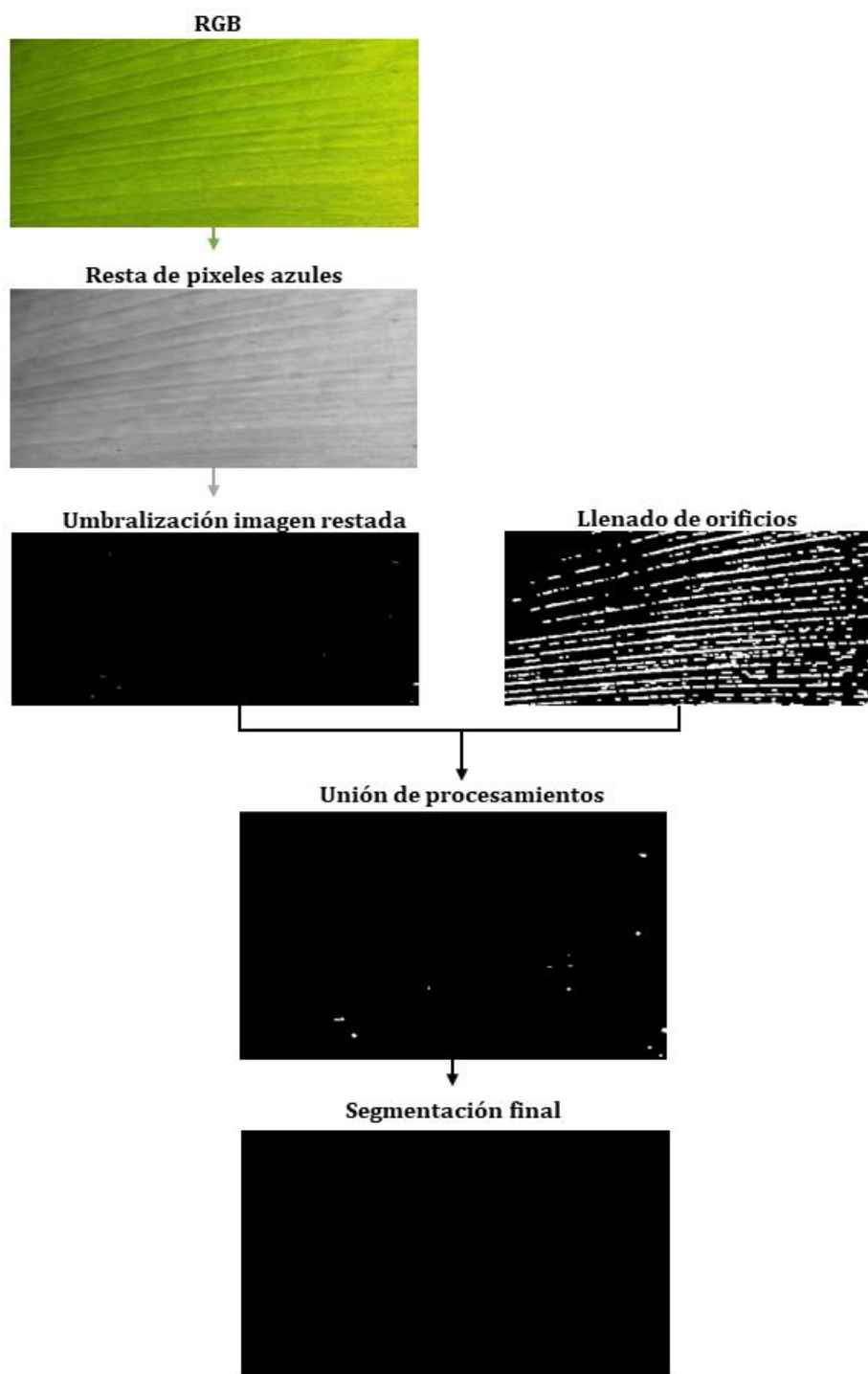


Figura 44. Resultados imagen Hoja_Sana_9 – Procesamiento 2 e imagen final procesada.

(Autor).

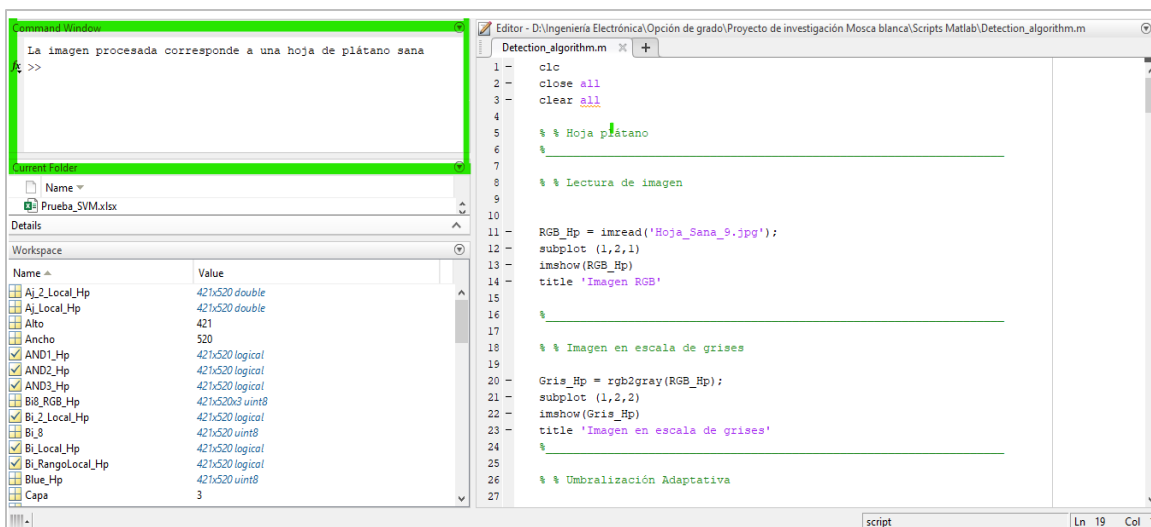


Figura 45. Resultado de clasificación de la imagen Hoja_Sana_9. (Autor).

Análisis del modelo

La curva ROC muestra el rendimiento de un modelo de clasificación en todos los umbrales de clasificación. Esta curva requiere dos parámetros que son la tasa de verdaderos positivos y la tasa de falsos positivos.

La tasa de verdaderos positivos (TPR – True Positive Rate) es representada de la siguiente forma:

$$TPR = \frac{TP}{TP + FN}$$

La tasa de falsos positivos (FPR – False Positive Rate) se define así:

$$FPR = \frac{FP}{FP + TN}$$

TP = True Positive – Verdadero Positivo

FP = False Positive – Falso Positivo

TN = True Negative – Verdadero Negativo

FN = False Negative – Falso Negativo

Resultado **9** **8** **1** **0**

Tabla 2.

Matriz de confusión del modelo.

		Predicción	
		1	0
Realidad	1	TP	FP
	0	FN	TN

		Predicción	
		1	0
Realidad	1	8	1
	0	0	9

En la tabla 2 se realiza la comparación entre la realidad y la predicción del modelo, a partir de la condición de imagen de hoja sana con valor 0 y con presencia de plaga con valor 1, a partir de la sumatoria de las condiciones se determinarán los principales aspectos de análisis de resultados.

La tasa de verdaderos positivos TPR es conocida también como sensibilidad, su valor será la probabilidad de clasificar, las etiquetas positivas tanto de la realidad como de la predicción.

$$TPR = \frac{8}{8 + 0}$$

$$TPR = \frac{8}{8}$$

$$TPR = 1$$

La tasa de verdaderos positivos TFR al igual que la especificidad permiten conocer la efectividad del clasificador respecto a la identificación negativa.

$$FPR = \frac{1}{1+9}$$

$$FPR = \frac{1}{10}$$

$$FPR = 0,1$$

$$\textit{Especificidad} = \frac{TN}{TN + FP}$$

$$\textit{Especificidad} = \frac{9}{9 + 1}$$

$$\textit{Especificidad} = \frac{9}{10}$$

$$\textit{Especificidad} = 0.9$$

$$FPR = 1 - \textit{Especificidad} = 1 - 0.9$$

$$1 - \textit{Especificidad} = 0.1$$

La precisión evalúa los resultados positivos de aquellos posibles en esta condición, es decir que tanto se acertó como hoja enferma dentro de aquellas que cumplían esta condición.

$$\textit{Precisión} = \frac{TP}{TP + FP}$$

$$\textit{Precisión} = \frac{8}{8 + 1}$$

$$\text{Presición} = \frac{8}{9}$$

$$\text{Presición} = 0.888888889$$

La exactitud describe que tan acertado es el modelo en cuanto a todos los resultados generados ya sea para la identificación de hojas sanas como para la de hojas de plátano con presencia de plaga mosca blanca.

$$\text{Exactitud} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Exactitud} = \frac{8 + 9}{8 + 9 + 1 + 0}$$

$$\text{Exactitud} = \frac{17}{18}$$

$$\text{Exactitud} = 0.944444444$$

La tasa de error muestra los fallos del clasificador.

$$\text{Error} = \frac{FP + FN}{TP + TN + FP + FN}$$

$$\text{Error} = \frac{1 + 0}{8 + 9 + 1 + 0}$$

$$\text{Error} = \frac{1}{18}$$

$$\text{Error} = 0,055555555$$

En la tabla 3 se muestran los resultados y sus porcentajes.

Tabla 3.

Parámetros obtenidos del modelo clasificador.

TPR Sensibilidad	FPR (1- Especificidad)	Especificidad	Precisión	Exactitud	Error
1	0,1	0,9	0,888	0,944	0,055
Porcentaje %	Porcentaje %	Porcentaje %	Porcentaje %	Porcentaje %	Porcentaje %
100	10	90	88,8	94,4	5,55

Finalmente, la curva ROC es representada en un gráfico de dos ejes en los cuales la tasa de verdaderos positivos (TPR) es denominada como sensibilidad y la tasa de falsos positivos (FPR) es definida como 1- especificidad. En la Figura 46, se muestra los puntos obtenidos en el algoritmo de acuerdo a cada tasa calculada, mostrando que el modelo tiene una exactitud de 94,4% y una precisión de 88,8%.

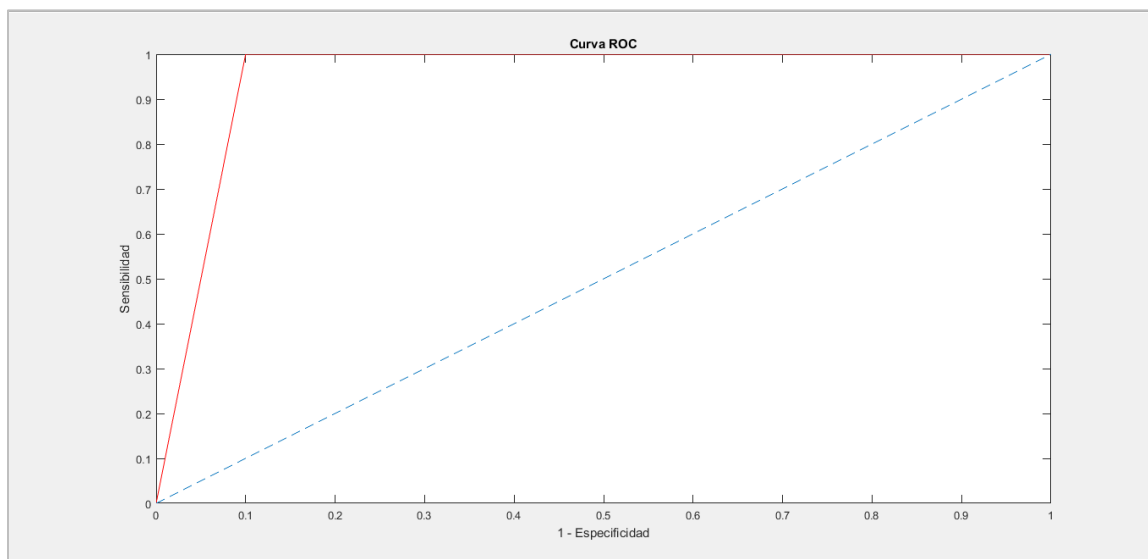


Figura 46. Curva ROC del algoritmo. (Autor).

Conclusiones

La segmentación de imágenes es una herramienta importante para la extracción de información específica como lo fue la presencia de plaga mosca blanca en las hojas de plátano.

La construcción del algoritmo necesita conocimientos generales de programación del software Matlab acompañado de las funciones que permiten procesar imágenes y adicionalmente requiere de estudios conceptuales de tratamiento de imágenes para la comprensión de cada instrucción.

La resta de pixeles contribuyo a la eliminación del fondo en las imágenes sanas disminuyendo la presencia de ruido en la segmentación.

Para la obtención de una binarización adecuada se requiere la utilización de diferentes técnicas.

La implementación de las compuertas lógicas permitió mayor obtención de información de cada una de las imágenes procesadas de igual forma las operaciones morfológicas fueron fundamentales para complementar dicha información resultante.

El análisis generado mediante la implementación de la máquina de soporte vectorial proporciono una correcta clasificación en la mayoría de las imágenes procesadas.

Los resultados obtenidos en el análisis de la matriz de confusión demostraron una efectividad del 94,4% y una precisión de 88.8% del modelo.

Referencias

- 6Letras. (2016, Diciembre 18). *YouTube*. Retrieved from <https://www.youtube.com/watch?v=p1PCPkZV1jU>
- Calderón, A., & Hurtado, H. (2019, Diciembre 26). Machine learning en a detección de enfermedades en plantas. *Tecnología, Investigación y Academia*, 55 - 61. Retrieved from <https://revistas.udistrital.edu.co/index.php/tia/article/view/15685/15932>
- Centeno, I. (2019, Mayo 06). *Xeridia*. Retrieved from xeridia.com/blog/la-vision-artificial-y-el-procesamiento-de-imagenes
- CERTIS . (2017, Marzo 29). *CERTIS - Growing Together*. Retrieved from <https://www.certiseurope.es/noticias/detalle/news/mosca-blanca-que-es-y-como-se-combate/>
- DANE. (2016, Septiembre). *DANE Información para todos* . Retrieved from https://www.dane.gov.co/files/investigaciones/agropecuario/sipsa/Bol_Insumos_sep_2016.pdf
- Devaraj, A., Rathan, K., Indira, K., & Jaahnavi, S. (2019). Identification of Plant Disease using Image Processing Technique. *2019 International Conference on Communication and Signal Processing (ICCSP)*, (pp. 749 - 753). Chennai. doi:10.1109/ICCSP.2019.8698056
- Francis, J., Sahaya, A., & Anoop, B. (2016). Identification of leaf diseases in pepper plants using soft computing techniques. *Conference on Emerging Devices and Smart Systems (ICEDSS)*, (pp. 168 - 173). Namakkal. Retrieved from <https://ieeexplore.ieee.org/document/7587787/authors#authors>

- Global, C. (2019, Julio 31). *Charter Global* . Retrieved from <https://www.charterglobal.com/what-is-image-binarization-in-ai/>
- Hurwitz, J & Kirsch D. (2018). IBM. Retrieved from <https://www.ibm.com/downloads/cas/GB8ZMQZ3>
- Kitpo, N., & Inoue, M. (2018). Early Rice Disease Detection and Position Mapping System using Drone and IoT Architecture. *12th South East Asian Technical University Consortium (SEATUC)*. Yogyakarta. doi:10.1109/SEATUC.2018.8788863
- López, J. (2019, Julio 23). *Agronegocios*. Retrieved from <https://www.agronegocios.co/agricultura/mas-de-90-del-banano-local-es-de-exportacion-2887702>
- Martinez, J. (2020, Octubre 02). *Artificial.net*. Retrieved from <https://www.iartificial.net/regresion-lineal-con-ejemplos-en-python/>
- MathWorks. (2021). *MathWorks*. Retrieved from <https://la.mathworks.com/help/images/image-segmentation.html?lang=en>
- MathWorks. (2021). *MathWorks*. Retrieved from <https://www.mathworks.com/help/images/ref/edge.html>
- MathWorks. (2021). *MathWorks*. Retrieved from <https://www.mathworks.com/help/stats/support-vector-machines-for-binary-classification.html>
- Minagricultura. (2018). *SIOC*. Retrieved from Sistema de Información de Gestión y Desempeño de las Organizaciones de Cadenas : <https://sioc.minagricultura.gov.co/Platano/Documentos/2018-10-30%20Cifras%20Sectoriales.pdf>

Sharma, P. (2019, Abril 01). *Analytics Vidhya's* . Retrieved from

<https://www.analyticsvidhya.com/blog/2019/04/introduction-image-segmentation-techniques-python/>

Simmonds, I. (2016, Octubre 11). *Agronegocios*. Retrieved from

<https://www.agronegocios.co/aprenda/mosca-blanca-una-plaga-que-preocupa-a-plataneros-del-pais-2622216>