

DIAGNÓSTICO DE ENFERMEDADES EN PLANTAS

T E S I S

Que para obtener el grado de
**Maestro en Modelación y Optimización de
Procesos**

Presenta

Omar Antonio Mercado Ocampo

Director de Tesis:

Dra. Lili Guadarrama Bustos
Dr. Carlos Alberto Paredes Orta



Autorización de la versión final

Agradecimientos.

A mis padres y hermanos por todo su apoyo.

A mis asesores de tesis por su guía y consejos en todo el proceso.

A mis profesores y compañeros por la transmisión de conocimientos.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por la beca de maestría que me permitió dedicarme de forma exclusiva al posgrado.

Índice

1. INTRODUCCIÓN.	5
2. ANTECEDENTES.	6
3. MARCO TEÓRICO	8
3.1. ENFERMEDADES EN PLANTAS.	8
3.2. RED NEURONAL - MULTI LAYER PERCEPTRON (MLP) [57].	8
3.3. ESPACIO DE COLOR HSV.	11
3.4. GAUSSIAN MIXTURE MODELS [53].	12
3.4.1. FORMULACIÓN DEL MODELO.	12
3.4.2. MÁXIMA VEROSIMILITUD.	13
3.5. ALGORITMO EXPECTATION MAXIMIZATION (EM).	14
3.5.1. Inicializar valores.	14
3.5.2. Paso de esperanza.	14
3.5.3. Paso de Maximización.	14
3.5.4. Evaluar la verosimilitud.	15
3.6. PROMINENCIA BAYESIANA [19].	16
3.7. MÉTODO DE OTSU [54].	18
3.8. ANÁLISIS MORFOLÓGICO DE IMÁGENES [56].	20
3.8.1. EROSIÓN.	20
3.8.2. DILATACIÓN.	21
3.8.3. APERTURA.	21
3.8.4. CERRADURA.	22
3.9. FILTRO GAUSSIANO [58].	23
4. METODOLOGÍA.	24
4.1. RED NEURONAL PARA EL RECONOCIMIENTO DE COLORES.	26
4.2. SEGMENTACIÓN CON POCO RUIDO EN EL FONDO.	28
4.2.1. APLICACIÓN DE UN FILTRO GAUSSIANO.	28
4.2.2. OBTENCIÓN DE PARÁMETROS INICIALES POR K-MEANS.	28
4.2.3. CLASIFICACIÓN DE LOS PÍXELES EN 2 SEGMENTOS HACIENDO USO DE GMM.	29
4.2.4. IDENTIFICACIÓN DE LA PLANTA U HOJA.	31
4.2.5. CERRADURA DEL SEGMENTO QUE PERTENECE A LA HOJA.	31
4.2.6. PSEUDO CÓDIGO.	33
4.3. SEGMENTACIÓN PARA EL CASO DE FONDOS CON RUIDO.	34
4.3.1. CODIFICACIÓN LA IMAGEN HACIENDO USO DE LA RED NEURONAL.	34
4.3.2. OBTENCIÓN DEL MAPA DE PROMINENCIA.	35
4.3.3. SEGMENTACIÓN DEL MAPA DE PROMINENCIA MEDIANTE OTSU.	39
4.3.4. APLICACIÓN DE APERTURA Y CERRADURA A LA IMAGEN SEGMENTADA.	39
4.3.5. PSEUDO CÓDIGO.	41
4.4. CUANTIFICACIÓN DE PIGMENTACIÓN	43
5. RESULTADOS.	44
5.0.1. PRESENTACIÓN DE RESULTADOS.	44
5.0.2. EVALUACIÓN DE RESULTADOS.	45
6. CONCLUSIONES.	47
7. TRABAJO FUTURO.	48
7.0.1. DETERMINACIÓN DE LA DIMENSIÓN DEL KERNEL PARA OPERACIONES DE MORFOLOGÍA.	48
7.1. SELECCIÓN AUTOMÁTICA DEL MODELO DE SEGMENTACIÓN	49
8. ANEXOS.	50
8.1. IMÁGENES DE PRUEBA.	50
8.2. IMPLEMENTACIÓN DE MODELOS GMM.	59
8.3. IMPLEMENTACIÓN DE PROMINENCIA BAYESIANA.	64
9. BIBLIOGRAFÍA.	68

Índice de cuadros

1.	Tiempos de ejecución con un procesador Intel i7 @ 1.90 GHz 2.40 GHz y 8 gb de RAM.	45
2.	Reducción del esfuerzo computacional para los casos de segmentación por Prominencia.	45
3.	Evaluación de resultados.	46

1. INTRODUCCIÓN.

La agricultura y el cuidado de la vegetación son claves para el bienestar de la sociedad, en específico dados dos grandes retos a los cuales se enfrenta la humanidad hoy en día que son el cambio de temperatura del medio ambiente y la necesidad de alimentar a una población enorme reduciendo las emisiones de efecto invernadero que dicha actividad provoca, por lo cual aparece la inquietud de mejorar la productividad del cultivo orientado a dietas que permitan cuidar el medio ambiente.

Existen diversas formas de mejorar la productividad de los cultivos, pero también existen muchas amenazas. Una de ellas son las enfermedades que pueden presentarse en las plantas, las cuales pueden poner en riesgo su salud no sólo para aquellas que son utilizadas para la agricultura, pero también en aquellas que conforman las selvas y bosques que contribuyen a la generación del oxígeno que la humanidad respira.

La pronta identificación de las enfermedades en plantas es clave para realizar actividades correctivas que permitan cuidar la salud de las mismas con el objetivo de que puedan contribuir al éxito de la agricultura y a las necesidades relacionadas con la conservación de la vegetación.

Entre la enorme diversidad de enfermedades que pueden presentarse en algunas plantas, el trabajo presente se enfocará en la detección de las siguientes: necrosis, clorosis y puntos blancos.

La inspección visual es una de las formas de realizar actividades de diagnóstico para poder tomar acciones que logren proteger la salud de la planta en caso de ser necesario. Una de las desventajas de este tipo de diagnóstico es tener que emplear a seres humanos para realizar tareas repetitivas mientras pueden realizar tareas que agreguen mayor valor al objetivo en turno y que hagan mejor uso de las cualidades que poseemos los seres humanos, además, dada la naturaleza de la inspección visual, se corre el riesgo de tener variabilidad en el juicio de una persona con respecto a otra o tener variabilidad en el juicio de una misma persona a través del tiempo, normalmente ocasionado por la fatiga visual que conlleva una inspección.

Dado lo mencionado con anterioridad se considera que la automatización de la tarea del diagnóstico de enfermedades en plantas ofrece beneficios que pueden incrementar la productividad en las actividades de agricultura o pueden ayudar a la conservación de la vegetación de una zona determinada. Para lograrlo se deben implementar técnicas de visión computacional y clasificación de patrones para que mediante de una imagen de una hoja de la planta o la misma planta en su totalidad se tenga la capacidad de realizar un diagnóstico cierto y consistente con respecto a la presencia de la clorosis, necrosis o de puntos blancos en la superficie de la hoja.

Para poder ejecutar el objetivo de automatizar el diagnóstico de enfermedades se hará uso de técnicas de clasificación con el objetivo de segmentar la hoja del fondo y así proceder a realizar una clasificación de los píxeles que componen el segmento correspondiente a la hoja. Una técnica es la de Gaussian Mixture Models (GMM) en la cual se parte del principio que se quiere clasificar los píxeles de una imagen en dos principales segmentos uno para el fondo y otro para la planta, en el caso de que las imágenes tengan mucho ruido en el fondo se hará uso de otro tipo de técnica bayesiana para la segmentación de la planta y del fondo, la cual es conocida como Prominencia Bayesiana, que tiene como objetivo detectar un objeto prominente, es decir un objeto que destaque en una imagen.

Se consideran los colores asociados a las enfermedades mencionadas con anterioridad y el color verde, que representa el color de una planta sana para el alcance del trabajo que se presenta. Para la clasificación del color de cada píxel se hace uso de una red neuronal analizando la pigmentación de un píxel determinado con el espacio de color HSV, lo cual al ser realizado para cada píxel permite aproximar el porcentaje del tejido de una planta que tiene necrosis, clorosis o puntos blancos y evidentemente el porcentaje sano.

Al emplear técnicas para la clasificación de patrones de forma secuencial, primero para la segmentación y reconocimiento de lo que es una planta en una imagen y posteriormente la clasificación de colores que permite aproximar la presencia de las enfermedades mencionadas con anterioridad en la planta, se genera el potencial de tener un alto impacto al sentar las bases teóricas y una implementación funcional de una herramienta que permita servir de apoyo para mejorar la productividad en la agricultura y en el cuidado del medio ambiente.

Dados los argumentos mencionados con anterioridad el objetivo de la presente tesina es desarrollar e implementar algoritmos o técnicas que permitan el diagnóstico correcto y consistente de la porción de presencia de clorosis, necrosis y puntos blancos en plantas.

2. ANTECEDENTES.

El diagnóstico de enfermedades en plantas es un tema de gran importancia y como es de esperar se han realizado trabajos similares. Uno de ellos realizado por la Universidad Autónoma de Querétaro y el cual fue en gran parte la fuente de inspiración para el trabajo de tesina que se presenta. Este trabajo trata sobre el desarrollo de diversos algoritmos para el diagnóstico de las enfermedades de clorosis, necrosis, puntos blancos y deformaciones en las plantas [1]. Es importante mencionar que en dicho trabajo se proponen algoritmos específicos para cada problema y existen ciertas restricciones que controlan el ambiente en el cual se aplican los algoritmos para cada caso, particularmente. Para el caso de la clorosis se hace uso de un fondo blanco dado que dicho fondo tiene una tendencia a incrementar la efectividad del algoritmo específico diseñado para la detección de la presencia de clorosis en las plantas, lo cual es positivo en una parte ya que eso incrementa la efectividad de los algoritmos pero genera limitaciones en las aplicaciones que se pueden tener en diversos ambientes y en la práctica.

El desarrollo de algoritmos específicos para cada enfermedad es algo positivo desde el enfoque del desarrollo de herramientas que se encuentren orientadas a realizar pruebas con las enfermedades en las plantas en ambientes controlados. Sin embargo limita la potencial extensión de las herramientas para el diagnóstico cuando se tienen ambientes no controlados o la adaptación para realizar el diagnóstico para otro tipo de enfermedades, tomando en cuenta que muchos de ellos se basan en la observación de la pigmentación de la planta.

La metodología implementada en el presente trabajo de tesina para el diagnóstico de enfermedades en plantas se divide en dos grandes etapas, la primera de ellas es la segmentación de la imagen para separar la planta del fondo y la posterior que es la clasificación y cuantificación de los colores que forman parte del segmento correspondiente a la planta. A continuación se presentan algunos de los trabajos previos que han servido de inspiración para realizar la tarea mencionada con anterioridad.

Se ha estudiado el problema de segmentación haciendo uso de modelos GMM, mediante el uso de los espacios de color del tipo RGB o de la escala de grises, incorporando el algoritmo Expectation Maximization (EM) para la estimación de los parámetros del modelo que permiten realizar la segmentación [2]. Existen líneas de investigación que realizan la segmentación de imágenes mediante la combinación de características de la imagen con algún tipo de regularización o conocimiento a priori que logre capturar la noción de una segmentación aceptable, muchos de los trabajos realizados para la segmentación de imágenes pueden separarse en dos líneas de investigación [2]:

a) Extracción de características de la imagen y modelos de características que logren ser tan informativos como sean posibles para el objetivo de segmentación. Algunos de los estudios más recientes logran combinar la intensidad, textura y contorno para generar características que ayuden en la segmentación con el objetivo específico de imitar de alguna manera u otra, la forma en la cual los seres humanos realizan la segmentación [3]. Otro de los enfoques es el trabajo en el cual se hacen uso de medidas multi escala de intensidad [4]. Algunos ejemplos clásicos que incorporan la segmentación basada en texturas, como es el caso de los trabajos que hacen uso de los Filtros de Gabor [5], basadas en transformaciones de onda [6], matrices de co-ocurrencia [7], características derivadas de modelos de Cadenas de Markov para campos aleatorios de modelos de textura locales [8], [9]. Otros trabajos han realizado segmentación haciendo uso de modelos que incorporan medidas estadísticas no paramétricas de la similitud de las texturas mediante clustering a pares [10]. La literatura de los modelos que incorporan características de textura es muy vasta y se ofrece un resumen muy completo en [11].

b) Desarrollo de metodologías que imponen algún tipo de regularización espacial a la segmentación, por ejemplo, que integran características locales a una segmentación que sea globalmente coherente. Los métodos basados en gráficas alcanzan dicho objetivo mediante la formulación de la segmentación de una imagen como la partición de una gráfica [12], [13], [14]. La coherencia espacial se puede alcanzar también mediante la restricción de la clase de particiones de imágenes que sean consideradas por el algoritmo de segmentación, por ejemplo, [15] y [16] lo realizan considerando jerarquías de particiones polinomiales o de particiones del tipo árbol, o mediante la imposición de algún tipo de prior sobre la longitud o la suavidad de las fronteras de las regiones [17]. En un enfoque probabilístico del tipo bayesiano la preferencia para algún tipo de regularización espacial se formula de manera usual haciendo uso de Campos Aleatorios de Markov o Markov Random Fields (MRF), se detalla un conjunto completo de referencias en [18].

Otra propuesta para estudiar la segmentación son las técnicas Bayesianas, dentro de las cuales, existen propuestas centradas en medir la prominencia de cada uno de los píxeles de una imagen, la idea es que un objeto será prominente mientras que el fondo no lo será, por lo tanto ayudando a alcanzar el objetivo de separar una región de interés de su respectivo fondo, implementando algoritmos que logren cuantificar la prominencia de un objeto para separarlo de su fondo [19].

Normalmente en una imagen existe una región o cuerpo de mayor interés y existe una gran cantidad de píxeles que corresponden a un fondo que no tienen mayor interés para las personas que observan la imagen. En el caso de la tesina, la región de interés son las plantas u hojas y el fondo puede ser muy variado dependiendo del contexto en el cual se ha tomado la imagen, el concepto de prominencia o saliency incorpora esa habilidad natural que tenemos los seres humanos para centrar nuestra visión y atención en los objetos de mayor importancia al momento de realizar alguna observación.

Comúnmente la separación del fondo se realiza mediante la detección de objetos en movimiento en contra de un fondo estático [20], [21], [22], [23], [24]. Estas técnicas son efectivas en ciertos contextos pero este enfoque tiene varios problemas cuando las escenas son dinámicas o cuando la cámara no es estática; estas situaciones han sido tratadas mediante la compensación de los movimientos de la cámara y mediante la actualización constante del fondo en el modelo [25], [26]. Otro enfoque es el uso de métodos que permiten la detección de objetos que son entrenados para reconocer objetos o formas específicas como lo pueden ser personas [27], [28], [29] y se han obtenido resultados prometedores, sin embargo uno de los inconveniente de dicha metodología es que requieren una cantidad enorme de fotos para realizar los entrenamientos y que en ocasiones no se generaliza de forma robusta cuando se presentan casos que difieren de forma significativa a los materiales que fueron utilizados para el entrenamiento.

Un enfoque alternativo es el de las detecciones de prominencia que se inspiran en el mecanismo de visión de los seres humanos, como se menciona con anterioridad, que nos permiten detectar regiones de interés de forma automática y sin mucho entrenamiento previo e inclusive se puede detectar a un objeto novedoso prominente ante un fondo ordinario que nunca ha sido observado con anterioridad, algo que es muy complicado de realizar con los algoritmos de redes neuronales que tienen problemas al encontrarse con escenarios que son muy diferentes a los que fueron utilizados para el entrenamiento. Para el caso de los métodos de prominencia muchos de ellos miden el contraste local de una área con respecto a sus alrededores haciendo uso de características como la escala de grises, color y orientación de gradiente [30], [31], [32], se han presentado métodos un poco más novedosos haciendo uso de características de bajo nivel de color e iluminación con resultados prometedores en [33]. Otro tipo de detectores de prominencia implementan el espectro de fase de la Transformada de Fourier de la imagen [34], [35], la implementación de dichos métodos es sencilla pero la precisión de dichos métodos es un poco limitada, cabe mencionar que existen detectores de prominencia que se han configurado o diseñado específicamente para la detección de prominencia en videos [36], [37].

Uno de los enfoques más recientes para la detección de prominencia en distintos contextos hace uso de las aplicaciones de aprendizaje profundo. Uno de ellos se enfoca en la prominencia visual basada en características multi escala profundas [38], otro realiza la detección de prominencia mediante aprendizaje profundo multi contexto [39], mediante la aplicación de redes neuronales convolucionales profundas [40], [41], además se han realizado aplicaciones de redes neuronales para la detección de prominencia en video [42], [43], [44].

3. MARCO TEÓRICO

Dada la naturaleza interdisciplinaria del objetivo del trabajo presente, el marco teórico se conforma de un conjunto diverso de conceptos y herramientas que son utilizadas para lograr un diagnóstico adecuado de las enfermedades mencionadas con anterioridad, a continuación se procede a describir el fundamento teórico de los principales conceptos que son aplicados para el alcance del objetivo.

3.1. ENFERMEDADES EN PLANTAS.

Se explican las 3 enfermedades que son consideradas en el trabajo, los colores asociados y sus principales causas.

Necrosis: Es una enfermedad en plantas que se caracteriza por la muerte prematura de las células de un tejido, normalmente la necrosis es causada mediante infecciones por un patógeno, toxinas o por trauma. Las lesiones ocasionadas por la necrosis en las hojas generan manchas foliares en las cuales se produce un colapso de células, su contenido se degenera y la porción del tejido afectado disminuye o se oscurece, la zona central de una mancha posee células muertas y la zona en las periferias todavía tiene células vivas pero que están dañadas por la toxina o patógeno [45], se asocia con colores negros o cafés.

Clorosis: Es una condición anormal en la que el follaje produce insuficiente clorofila, al ocurrir, las hojas no tienen una coloración normal verde. La coloración es un verde muy pálido, amarillo o amarillo blanquecino. Las plantas que tienen dicha afectación tienen una capacidad disminuida de generar carbohidratos y pueden morir en caso de que la enfermedad no sea tratada, la clorosis puede corregirse suplementando la planta con hierro, magnesio y nitrógeno, aunque al igual puede deberse a un exceso de calcio, algunos pesticidas, defectos de riego, varios parásitos y enfermedades infecciosas [46], [47].

Puntos blancos: Son provocados cuando la debilidad en la planta deshabilita el proceso fotosintético y las funciones de respiración [48].

3.2. RED NEURONAL - MULTI LAYER PERCEPTRON (MLP) [57].

Una red neuronal es una familia de funciones inspirada por redes neuronales naturales con los principales objetivos de poder clasificar o de realizar ciertas predicciones. Una red neuronal se compone por la conexión de un diverso número de neuronas donde cada neurona recibe un número de entradas y tiene asociados el mismo número de pesos más un bias. La topología de una red del tipo MLP es una capa de entrada que se conecta a todas las entradas de datos y un número de capas intermedias que reciben como entrada de información la salida generada por las neuronas anteriores, finalmente se tiene una capa de salida que tiene como objetivo generar una clasificación o predicción.

Cada neurona realiza un ajuste de sus pesos y esa suma ponderada de los pesos multiplicados por las entradas sirve de entrada para una función de activación que en la mayoría de las ocasiones es del tipo sigmoideal (logística), tanh o relu. Se tiene que realizar este paso debido a que si no se realiza sin importar el número de neuronas que se tengan, la salida se comportará como una gran neurona, éste se debe a que cualquier combinación de funciones lineales produce una función lineal.

Las figuras 1 y 2 corresponden a la representación gráfica de una red neuronal en la cual la primera capa se compone por las entradas, es decir los datos que alimentan a la red y que serán procesadas por las neuronas que componen las capas intermedias, en las capas intermedias se encuentran neuronas que realizan la suma ponderada basada en el valor de las entradas de la neurona y los pesos de las conexiones, para los casos de neuronas que se encuentran en las capas intermedias las salidas son las entradas de otras capas intermedias que al mismo tiempo serán procesadas por otras neuronas. Cabe mencionar que la salida de cada neurona se procesa mediante una función que como ya lo mencionamos anteriormente es logística o tangencial, para el caso de la neurona de salida es en la cual se realiza la predicción o la clasificación dependiendo del tipo de problema que se esté tratando, en ocasiones la neurona de salida no tiene una función, es decir la salida es exclusivamente la suma ponderada de las entradas multiplicadas por sus respectivos pesos.

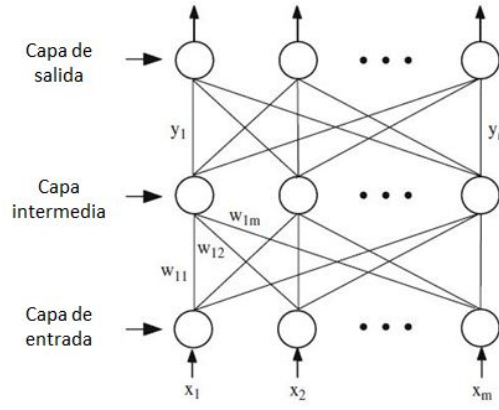


Figura 1: Perceptron del tipo MLP - Fuente [57]

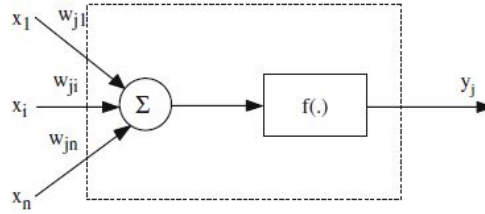


Figura 2: Detalles de una neurona - Fuente [57]

Una de las maneras más comunes de medir el desempeño de la red neuronal es haciendo uso del error cuadrático medio (MSE) en el cual para cada observación en la base de datos de entrenamiento, en caso de tratarse del MSE de entrenamiento y posteriormente el MSE de la predicción, se realiza una resta entre el valor generado por el modelo (\hat{y}_k) y el valor real (y_k), dicha resta se eleva al cuadrado, se realiza la sumatoria de todas las observaciones y se divide entre el número de observaciones consideradas, ya sea para el entrenamiento y predicción, esto es,

$$MSE = \frac{\sum_{k=1}^n (y_k - \hat{y}_k)^2}{n} \quad (1)$$

El entrenamiento, es decir la actualización de los pesos de las conexiones se realiza mediante un algoritmo que hace uso del gradiente, que determina para una función multivariada la dirección de máximo ascenso o descenso respectivamente.

Tomando en cuenta que la salida de una neurona j , es la siguiente.

$$y_j = f\left(\sum w_{ji}x_i\right) \quad (2)$$

Donde f es una función de activación, w_{ji} es el peso de la conexión entre la neurona j y la neurona i . Mientras que x_i es la salida de la neurona i , si corresponde a una neurona de la capa de entrada entonces x_i simplemente es un valor de entrada. La modificación para el peso de una neurona j en conexión con una neurona i se describe a continuación.

$$\Delta w_{ji} = \eta \delta_j x_i \quad (3)$$

Donde η es el parámetro denominado como tasa de aprendizaje, x_i es el valor de la salida de la neurona i , en caso de ser una neurona de la capa de entrada x_i es simplemente un valor de entrada. El parámetro δ_j depende si la neurona j es una neurona de salida o una neurona que pertenece a las capas anteriores. Para las neuronas de capa de salida lo definimos como.

$$\delta_j = \left(\frac{\partial f}{\partial net_j}\right)(y_j^t - y_j) \quad (4)$$

Donde net_j es la suma total ponderada de las señales de entrada a la neurona j y la variable y_j^t es la salida objetivo para la neurona j .

Para el caso de las neuronas que pertenecen a las capas intermedias lo definimos de la siguiente manera.

$$\delta_j = \left(\frac{\partial f}{\partial net_j}\right) \sum_q w_{qj} \delta_q \quad (5)$$

Dado que no existen valores objetivos para las neuronas intermedias, la diferencia entre el objetivo y la salida de una neurona j se reemplaza por la suma ponderada de los términos δ_q que se ya se obtuvieron para las neuronas q conectadas a la salida de j . Por lo tanto, de forma iterativa, empezando por la capa de salida, el término δ se calcula para las neuronas en todas las capas y se determinan los pesos para todas las conexiones.

3.3. ESPACIO DE COLOR HSV.

El espacio de color Hue, Saturation and Brightness (HSV) se denomina así debido a que integra los componentes de Matiz, Saturación y Valor. Para definir la representación de un color, existen otros espacios de colores como el espacio Red, Blue and Red (RGB) que hace uso del concepto de síntesis aditiva en el cual se considera que es posible representar un color mediante la suma de los tres colores de luz primarios. Otro ejemplo es la Escala de Grises en la cual se tiene un sólo canal, el valor representa sólo la cantidad de luz, es decir la información relacionada a la intensidad, para todos los casos mencionados es importante destacar que en el análisis de imágenes la información de los espacios de colores se encuentra en cada uno de los píxeles de la imagen siendo un pixel la menor unidad homogénea de color de una imagen digital.

El espacio HSV es una transformación no lineal del espacio de color RGB, creado por Alvy Ray Smith, describe de manera circular la frecuencia de cada color en el espectro visible en la componente H, con la componente S que representa la pureza del color y si se realiza un símil en términos numéricos es el contraste entre las componentes RGB con el valor mínimo y el valor máximo, mientras que V representa la cercanía con del pixel negro y el pixel blanco [49], [50], [51].

En la figura 3 se puede observar en la parte superior al componente H que realiza la descripción de las frecuencias de cada color, mientras que en el triángulo a la derecha se observan a detalle las componentes S y V.

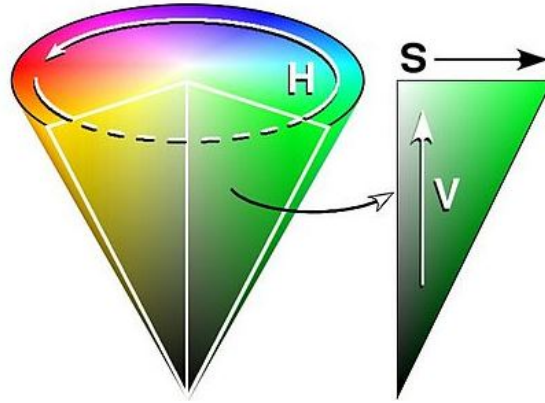


Figura 3: Representación gráfica del modelo HSV - Fuente [52]

Sean MAX y MIN el valor máximo y mínimo de los componentes RGB respectivamente, los componentes del espacio HSV se pueden determinar de la siguiente manera.

$$H = \begin{cases} \text{indefinido} & \text{si } MAX = MIN \\ 60^\circ \times \frac{G - B}{MAX - MIN} + 0^\circ & \text{si } MAX = R \text{ y } G \geq B \\ 60^\circ \times \frac{G - B}{MAX - MIN} + 360^\circ & \text{si } MAX = R \text{ y } G < B \\ 60^\circ \times \frac{B - R}{MAX - MIN} + 120^\circ & \text{si } MAX = G \\ 60^\circ \times \frac{R - G}{MAX - MIN} + 240^\circ & \text{si } MAX = B \end{cases} \quad (6)$$

$$S = \begin{cases} 0 & MAX = 0 \\ 1 - \frac{MIN}{MAX} & \text{en otro caso} \end{cases} \quad (7)$$

$$V = MAX \quad (8)$$

En la práctica el uso del espacio HSV para el procesamiento de imágenes puede generar algoritmos o metodologías más robustas que el uso del espacio de color del tipo RGB, ya que el espacio HSV es más robusto a la presencia de brillos en las imágenes, siendo una ventaja para analizar imágenes en exteriores.

3.4. GAUSSIAN MIXTURE MODELS [53].

Son modelos de agrupamiento o clustering, donde se asume que las variables de interés se pueden modelar con un conjunto de distribuciones gaussianas. En nuestro caso de estudio, dado que se tiene el propósito de realizar la segmentación de la planta y el fondo, el número de gaussianas será de 2 y dependiendo si la segmentación se realiza en escala de grises, RGB o HSV. Las gaussianas serán univariadas o multivariadas respectivamente. La asignación de una observación a algunas de las distribuciones gaussianas será determinada mediante el peso de la observación para cada una de las distribuciones, la distribución para la cual una observación tenga el mayor peso será a la cual se le asigne la pertenencia.

3.4.1. FORMULACIÓN DEL MODELO.

Un modelo GMM es una superposición lineal de distribuciones gaussianas, formulado mediante el uso de variables discretas latentes. Un modelo GMM univariado se describe de la siguiente forma,

$$p(x) = \sum_{k=1}^K \pi_k \mathcal{N}(x | \mu_k, \sigma_k^2) \quad (9)$$

Donde π_k, μ_k, σ_k^2 son los coeficientes de mezcla, media y varianza respectivamente de la distribución k .

Ahora introducimos una variable aleatoria binaria con dimensión K que llamaremos z teniendo 1 en K representaciones, en las cuales un elemento particular $z_k = 1$ mientras el resto son igual a 0, por lo tanto los valores de z_k satisfacen que $z_k \in \{0, 1\}$ y $\sum_k z_k = 1$. Existen K estados posibles para el vector z dependiendo del elemento que sea diferente de 0, cuando $z_k = 1$ representa el escenario en el cual la observación x ha sido resultado de un muestreo de la distribución k .

Se define la distribución conjunta $p(x, z)$ en términos de la distribución marginal $p(z)$ y una distribución condicional $p(x|z)$, a continuación se representa de forma visual lo mencionado con anterioridad.



Figura 4: Representación de un GMM general con la distribución $p(x, z) = p(z)p(x|z)$. - Fuente [53]

La distribución marginal sobre z se especifica en términos de los coeficientes de mezcla π_k , tal que,

$$p(z_k = 1) = \pi_k \quad (10)$$

Donde los parámetros $\{\pi_k\}$, deben satisfacer lo siguiente,

$$0 \leq \pi_k \leq 1 \quad (11)$$

$$\sum_{k=1}^K \pi_k = 1 \quad (12)$$

Dado que z hace uso de 1 de k representaciones, se puede escribir esta distribución en la siguiente forma,

$$p(z) = \prod_{k=1}^K \pi_k^{z_k} \quad (13)$$

De forma similar, la distribución condicional de x dado un valor particular para z es una gaussiana,

$$p(x|z_k = 1) = \mathcal{N}(x|\mu_k, \sigma_k^2) \quad (14)$$

Lo cual puede escribirse de la siguiente manera,

$$p(x|z) = \prod_{k=1}^K \mathcal{N}(x|\mu_k, \sigma_k^2)^{z_k} \quad (15)$$

La distribución conjunta está dada por $p(z)p(x|z)$ y la distribución marginal de x se obtiene sumando todos los valores posibles de z para la distribución conjunta, dando el siguiente resultado,

$$p(x) = \sum_z p(z)p(x|z) = \sum_{k=1}^K \pi_k \mathcal{N}(x|\mu_k, \sigma_k^2) \quad (16)$$

Por lo tanto la distribución marginal de x es una mezcla de gaussianas. Si se tienen varias observaciones $x_1, x_2, x_3, \dots, x_N$, entonces dado que se ha representado la distribución marginal de la forma $p(x) = \sum_z p(x, z)$ en consecuencia para cada observación x_n existe una variable latente correspondiente z_n .

Una cantidad que tiene un rol importante es la probabilidad condicional de z dado x . Se hará uso de la variable $\gamma(z_k)$ para expresar $p(z_k = 1|x)$, cuyo valor es el siguiente,

$$\gamma(z_k) = p(z_k = 1|x) = \frac{p(z_k = 1)p(x|z_k = 1)}{\sum_{j=1}^K p(z_j = 1)p(x|z_j = 1)} \quad (17)$$

$$\gamma(z_k) = \frac{\pi_k \mathcal{N}(x|\mu_k, \sigma_k^2)}{\sum_{j=1}^K \pi_j \mathcal{N}(x|\mu_j, \sigma_j^2)} \quad (18)$$

Se entiende a π_k como la probabilidad a priori de que $z_k = 1$ y a $\gamma(z_k)$ como la probabilidad posterior después de haber observado a x , al igual $\gamma(z_k)$ se entiende como el peso que la distribución gaussiana k tiene para explicar la observación x .

3.4.2. MÁXIMA VEROSIMILITUD.

Suponiendo que existen N observaciones x_1, x_2, \dots, x_N y se desea modelar dichas observaciones mediante una mezcla de gaussianas, con sus correspondientes N variables latentes z . Si se asume que las observaciones son independientes e idénticamente distribuidas, entonces la función logarítmica de verosimilitud es la siguiente,

$$\ln p(X|\pi, \mu, \sigma^2) = \sum_{n=1}^N \ln \sum_{k=1}^K \pi_k \mathcal{N}(x_n|\mu_k, \sigma_k^2) \quad (19)$$

A continuación se muestra una representación visual del modelo GMM considerando las N observaciones.

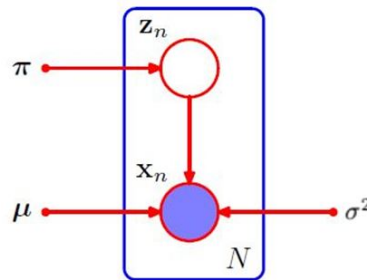


Figura 5: Representación visual del modelo GMM para N observaciones donde la observación x_n cuenta con su correspondiente z_n variable latente - Fuente [53]

3.5. ALGORITMO EXPECTATION MAXIMIZATION (EM).

Un método elegante y poderoso para encontrar soluciones de máxima verosimilitud para modelos con variables latentes es el algoritmo de Expectation (Esperanza) Maximization (Maximización). En el paso de Esperanza se hace uso de los valores disponibles de los parámetros para calcular las probabilidades o responsabilidades posteriores, descritas en la ecuación (18). Se hace uso de esas probabilidades en el paso de Maximización para re-estimar las medias, varianzas y los pesos de las distribuciones.

Es importante destacar que el modelo GMM es un modelo incompleto, ya que al analizar los datos (X), no podemos observar a las variables latentes (Z), pero las variables latentes tienen un rol muy importante en explicar la generación de las observaciones, ya que para cada observación existe una variable latente relacionada al conjunto de K distribuciones gaussianas que tiene distintas responsabilidades de explicar o de haber generado dicha variable. Como ya se mencionó con anterioridad $\pi_k = p(z_{nk} = 1)$ es la probabilidad a priori de que la distribución k sea la responsable de explicar la observación x_n , mientras que $\gamma(z_{nk}) = p(z_{nk} = 1|x)$ se interpreta como la probabilidad posterior o la responsabilidad que tiene la distribución k de explicar la observación x_n .

Dado que en la práctica sólo se observan los datos (X) entonces se considera la esperanza, con respecto a la distribución posterior de las variable latentes. Por lo tanto el valor esperado de la variable indicadora z_{nk} es.

$$\mathbb{E}[z_{nk}] = \frac{\pi_k \mathcal{N}(x_n | \mu_k, \sigma_k^2)}{\sum_{j=1}^K \pi_j \mathcal{N}(x_n | \mu_j, \sigma_j^2)} = \gamma(z_{nk}) \quad (20)$$

El algoritmo EM nos permite encontrar los valores de los parámetros que mejor ajusten el modelo GMM a las observaciones disponibles, a continuación se presentan los pasos del algoritmo.

3.5.1. Inicializar valores.

Inicializar los valores para las medias μ_k , las varianzas σ_k^2 y los pesos π_k y evaluar el valor inicial para la función logarítmica de verosimilitud.

3.5.2. Paso de esperanza.

Evaluar los pesos que tiene cada una de las distribuciones para una observación x_n .

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n | \mu_k, \sigma_k^2)}{\sum_{j=1}^K \pi_j \mathcal{N}(x_n | \mu_j, \sigma_j^2)} \quad (21)$$

3.5.3. Paso de Maximización.

Re-estimar los parámetros haciendo uso de los pesos actuales.

$$\mu_k^{new} = \frac{1}{N_K} \sum_{n=1}^N \gamma(z_{nk}) x_n \quad (22)$$

$$\sigma_k^{2new} = \frac{1}{N_K} \sum_{n=1}^N \gamma(z_{nk}) (x_n - \mu_k^{new})^2 \quad (23)$$

$$\pi_k^{new} = \frac{N_k}{N} \quad (24)$$

Donde.

$$N_k = \sum_{n=1}^N \gamma(z_{nk}) \quad (25)$$

3.5.4. Evaluar la verosimilitud.

Evaluar la verosimilitud haciendo uso de los nuevos parámetros y revisar si existe convergencia, en caso contrario repetir los pasos de esperanza y maximización.

$$\ln p(X|\pi, \mu, \sigma^2) = \sum_{n=1}^N \ln \sum_{k=1}^K \pi_k \mathcal{N}(x|\mu_k \sigma_k^2) \quad (26)$$

3.6. PROMINENCIA BAYESIANA [19].

En análisis de imágenes, la prominencia se refiere a la propiedad que tiene un objeto para destacarse del fondo. La prominencia probabilística asigna a cada pixel en una imagen la probabilidad de ser prominente.

Siendo x una propiedad del pixel en turno ya sea escala de grises o índice RGB, lo cual se ejemplifica mediante la siguiente imagen.

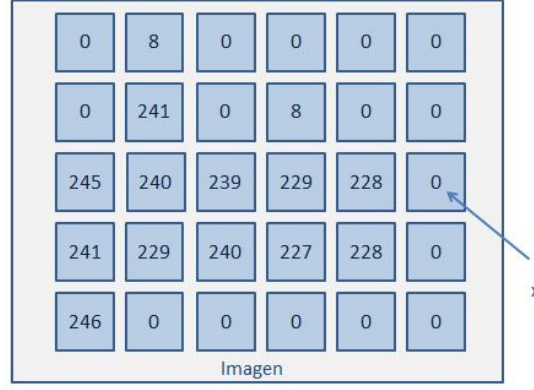


Figura 6: Representación de un pixel como variable x

Se tienen dos hipótesis, siendo h_0 la hipótesis nula y h_a la hipótesis alterna, especificadas a continuación.

h_0 : x no es prominente

h_1 : x es prominente

Los valores iniciales para h_0 y h_1 incorporan toda la información previa que se tiene posible al momento.

Para obtener los nuevos estimados de la probabilidad de prominencia de un pixel $p(x)$ se tiene lo siguiente,

$$p(x|h_1) = \frac{p(x \cap h_1)}{p(h_1)} \quad (27)$$

$$p(h_1|x) = \frac{p(x \cap h_1)}{p(x)} \quad (28)$$

Mediante Probabilidad total se sabe que.

$$p(x) = p(x|h_0)p(h_0) + p(x|h_1)p(h_1) \quad (29)$$

Aplicando el Teorema de Bayes se tiene que el nuevo estimado de prominencia de un pixel es el siguiente.

$$p(h_1|x) = \frac{p(x|h_1)p(h_1)}{p(x|h_0)p(h_0) + p(x|h_1)p(h_1)} \quad (30)$$

Donde

$p(x|h_0)$ es la probabilidad de observar el valor x dado que no es prominente

$p(x|h_1)$ es la probabilidad de observar el valor x dado que es prominente

$p(h_1|x)$ es la probabilidad de que el pixel sea prominente dado el valor de la observación x .

Para formar las distribuciones h_0 y h_1 respectivamente se construyen histogramas de los valores de una ventana

que se desliza por la imagen, teniendo cuidado de que los píxeles de cada ventana se traslapen con el objetivo de actualizar los valores de prominencia, con los histogramas se procede a calcular el valor para $p(x|h_o)$ y $p(x|h_1)$. La cantidad de píxeles que se traslapen depende de la distancia que exista entre las ventanas de prominencia, lo cual es un parámetro que determina el usuario.

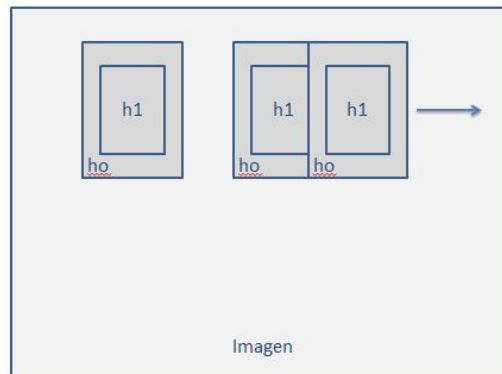


Figura 7: Deslizamiento tradicional de las ventanas de prominencia.

Una forma diferente, es hacer uso de ventanas de prominencia aleatorizando su posición en la imagen que representa la matriz.

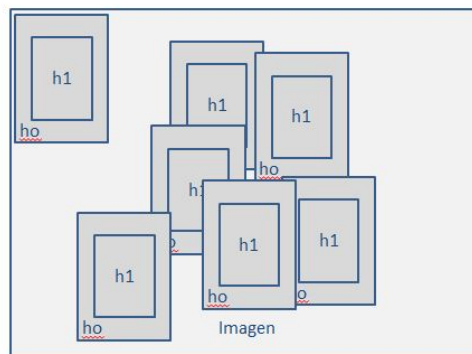


Figura 8: Posicionamiento aleatorio de las ventanas de prominencia.

Existe un conjunto de parámetros que son de mucha importancia para la implementación de la técnica de prominencia bayesiana. Uno de ellos es la determinación de las dimensiones de ancho y alto que debe de tener la ventana de prominencia que se desliza recorriendo la imagen o que se coloca de forma aleatoria en la imagen, todo depende del conocimiento que se tenga del tipo de imágenes que serán analizadas, ya que eso permite tener una estimación más adecuada de la cantidad de los píxeles que pertenecen al objeto de interés.

Otro parámetro que es de vital importancia es la determinación del umbral que permite clasificar aquellos valores que se consideren prominentes y los que no se consideren así, se realizó una investigación en la cual se determina que el valor de 0.70 es un umbral muy razonable para realizar la clasificación de la prominencia de un objeto en la imagen [26], existen varias propuestas para determinar este umbral, por ejemplo, una de ellas es una técnica muy conocida para la binarización de una imagen, denominada Método de Otsu.

A continuación se presenta una representación visual de un mapa de prominencia, que es una matriz con las mismas dimensiones de la imagen original, pero en cada celda de la matriz, se tiene la probabilidad de que un píxel determinado sea considerado prominente.



Figura 9: Representación gráfica de un mapa de prominencia.

3.7. MÉTODO DE OTSU [54].

El método de Otsu tiene la función de determinar de forma automática el umbral óptimo para realizar la binarización de imágenes, tradicionalmente se usa cuando las imágenes son analizadas en escalas de grises o intensidad, pero en el caso del trabajo de tesina presente, se hará uso de dicho método para encontrar el umbral óptimo para las probabilidades o valores de prominencia.

El método de Otsu funciona de mejor manera cuando el histograma de la imagen es bimodal y existe un valle entre los dos picos del histograma. En dicho caso el método tiene un muy buen funcionamiento, en casos contrarios se puede tener un desempeño que dista de ser óptimo, dado que se asume que la técnica de Prominencia Bayesiana logrará generar un mapa de prominencia adecuado, entonces se asume que se tendrán dos picos, uno que sea para valores altos y otro para valores bajos de prominencia.

El algoritmo prueba distintos valores para los umbrales y selecciona aquel que tenga la menor suma de las varianzas dentro de los dos grupos segmentados por el umbral determinado. A continuación se presentan los cálculos necesarios para poder cuantificar las varianzas para el caso en el cual se trata con el valor de la intensidad de un píxel.

Sea $P(i)$ el valor de probabilidad del histograma para el valor i correspondiente al valor de intensidad $i = 0, \dots, l$.

$$P(i) = \frac{\#pixeles = i}{total} \quad (31)$$

Es decir el número de píxeles con valor igual a i entre el número total de píxeles.

El Método de Otsu busca encontrar el valor óptimo t^* , es decir el valor de intensidad que logre dividir a los píxeles en dos grupos T_0 y T_1 , minimizando la varianza dentro de los grupos, los píxeles que pertenezcan a T_0 son aquellos que tengan valores de intensidad que van de 0 a t , mientras que aquellos píxeles que pertezcan a T_1 son aquellos que tengan valores de intensidad que van de $t + 1$ a l .

Sean $w_b(t)$, $\mu_b(t)$, $\sigma_b^2(t)$ el peso, la media y la varianza respectivamente para el fondo (background) y sean $w_f(t)$, $\mu_f(t)$, $\sigma_f^2(t)$ el peso, la media y la varianza respectivamente para el frente (foreground).

Los cálculos para los pesos $w_b(t)$ y $w_f(t)$ que corresponden al fondo y al frente, respectivamente, se realizan de la siguiente manera.

$$w_b(t) = \sum_{i=1}^t P(i) \quad (32)$$

$$w_f(t) = \sum_{i=t+1}^l P(i) \quad (33)$$

Al haber calculado los pesos, se procede a realizar los cálculos para $\mu_b(t)$ y $\mu_f(t)$.

$$\mu_b(t) = \frac{\sum_{i=1}^t i * P(i)}{w_b(t)} \quad (34)$$

$$\mu_f(t) = \frac{\sum_{i=t+1}^l i * P(i)}{w_f(t)} \quad (35)$$

Los cálculos para $\sigma_b^2(t)$ y $\sigma_f^2(t)$ son los siguientes.

$$\sigma_b^2(t) = \frac{\sum_{i=1}^t (i - \mu_b(t))^2 * P(i)}{w_b(t)} \quad (36)$$

$$\sigma_f^2(t) = \frac{\sum_{i=t+1}^l (i - \mu_f(t))^2 * P(i)}{w_f(t)} \quad (37)$$

Así $\sigma_w^2(t)$ es la suma ponderada de ambas varianzas, se quiere encontrar el mejor valor para el umbral, es decir, el valor t^* que permita minimizar las varianzas dentro clases, que en conjunto con sus respectivos pesos permitan minimizar la expresión que se presenta a continuación.

$$\sigma_w^2(t) = w_b(t) * \sigma_b^2(t) + w_f(t) * \sigma_f^2(t) \quad (38)$$

En la práctica se hace uso de la varianza entre clases y se busca conocer el valor que logre maximizarla, ya que es computacionalmente más eficiente.

$$\sigma_B^2 = \sigma^2 - \sigma_w^2 \quad (39)$$

$$= w_b(t) * (\mu_b(t) - \mu)^2 + w_f(t) * (\mu_f(t) - \mu)^2 \quad (40)$$

$$= w_b(t) * w_f(t) * (\mu_b(t) - \mu_f(t))^2 \quad (41)$$

Donde μ y σ^2 son la media y la varianza total respectivamente.

A continuación se presentan dos imágenes para que pueda ser apreciado el resultado de aplicar el Método de Otsu para binarizar una imagen.



Figura 10: Imagen original e imagen binarizada por Otsu - Fuente [55].

3.8. ANÁLISIS MORFOLÓGICO DE IMÁGENES [56].

Las transformaciones morfológicas son transformaciones sencillas basadas en la forma de la imagen. Se suelen realizar normalmente en imágenes binarias y se necesitan dos entradas para ser ejecutadas, una de ellas es la imagen y otra el kernel, que tradicionalmente es una matriz cuyos valores pueden ser 0 o 1 en el caso de imágenes binarizadas. Los valores que toma el kernel son los valores que tienen los píxeles al momento de posicionar el kernel por encima de ellos, a continuación se presenta una representación visual.

1	0	1	0	0	1	1	0	1	0	0	1
1	1	1	1	1	0	1	1	1	1	0	0
1	0	1	0	0	1	1	0	1	0	0	1
0	1	1	1	1	0	0	1	1	1	1	0
0	0	0	0	1	1	0	0	0	0	1	1
0	1	1	0	0	0	0	1	1	0	0	1
1	0	1	0	0	1	1	0	1	0	0	1
0	1	0	1	1	0	0	1	0	1	1	0
1	0	1	0	0	1	1	0	1	0	0	1
0	1	0	1	0	0	0	1	0	1	0	0
1	0	1	0	1	1	1	1	1	1	0	1
1	1	1	0	0	0	1	1	1	0	0	1

0	0	0
1	1	0
0	1	0

Kernel 3x3

Figura 11: Representación de un Kernel 3x3.

Dos operaciones básicas son las de erosión y dilatación con las cuales se construyen las operaciones de apertura y de cerradura.

3.8.1. EROSIÓN.

La operación de erosión tiene la intención de funcionar de una forma similar a la erosión física, erosionando los bordes de la imagen original, si se centra el kernel en un píxel. Dicho píxel será considerado un 1 si todos los píxeles en el kernel son 1, caso contrario el píxel se considera 0.



Figura 12: Imagen de prueba - Fuente [56]



Figura 13: Efecto de la erosión en la imagen de prueba - Fuente [56]

3.8.2. DILATACIÓN.

Es la operación opuesta a la erosión en el cual si se centra el kernel en un píxel, dicho píxel será considerado 1, si al menos uno de los píxeles del kernel es 1.



Figura 14: Efecto de la dilatación en la imagen de prueba - Fuente [56]

3.8.3. APERTURA.

Una apertura es la operación compuesta por una erosión seguida de una dilatación, normalmente es utilizada para eliminar puntos en una imagen binaria.



Figura 15: Efecto de la apertura en la imagen de prueba - Fuente [56]

3.8.4. CERRADURA.

Una cerradura es la operación compuesta por una dilatación seguida de una erosión, normalmente se usa cuando se quieren cerrar pequeños hoyos o puntos en algún objeto.



Figura 16: Efecto de la cerradura en la imagen de prueba - Fuente [56]

3.9. FILTRO GAUSSIANO [58].

Un filtro gaussiano es una técnica que se utiliza para suavizar una imagen, en la cual un píxel resulta modificado en función a un conjunto de píxeles que lo rodean. En concreto ese conjunto de píxeles que rodean al píxel que se busca suavizar y las ponderaciones del peso que tiene cada uno se representan a través de un kernel gaussiano.

Aplicar un filtro gaussiano a una imagen es realizar una convolución de la imagen con una función Gaussiana.

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \quad (42)$$

Para el caso de dos dimensiones la función es la siguiente.

$$G(x) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (43)$$

Donde x es la distancia desde el origen en el eje horizontal y en el caso de y es la distancia desde el origen en el eje vertical. El parámetro σ representa la desviación del filtro y entre mayor sea el valor de σ mayor será el filtrado que se aplica a la imagen, es decir resultará en una imagen más borrosa y suavizada.

A continuación se presenta un ejemplo de un kernel gaussiano, con $\sigma = 0.84089642$.

$$\begin{bmatrix} 0.00000067 & 0.00002292 & \mathbf{0.00019117} & 0.00038771 & \mathbf{0.00019117} & 0.00002292 & 0.00000067 \\ 0.00002292 & 0.00078633 & 0.00655965 & 0.01330373 & 0.00655965 & 0.00078633 & 0.00002292 \\ \mathbf{0.00019117} & 0.00655965 & 0.05472157 & 0.11098164 & 0.05472157 & 0.00655965 & \mathbf{0.00019117} \\ 0.00038771 & 0.01330373 & 0.11098164 & \mathbf{0.22508352} & 0.11098164 & 0.01330373 & 0.00038771 \\ \mathbf{0.00019117} & 0.00655965 & 0.05472157 & 0.11098164 & 0.05472157 & 0.00655965 & \mathbf{0.00019117} \\ 0.00002292 & 0.00078633 & 0.00655965 & 0.01330373 & 0.00655965 & 0.00078633 & 0.00002292 \\ 0.00000067 & 0.00002292 & \mathbf{0.00019117} & 0.00038771 & \mathbf{0.00019117} & 0.00002292 & 0.00000067 \end{bmatrix}$$

Figura 17: Kernel gaussiano con $\sigma = 0.84089642$ - Fuente [58].

4. METODOLOGÍA.

Para realizar un correcto diagnóstico de la porción de la planta u hoja que tiene clorosis, necrosis y puntos blancos se tienen dos estrategias principales.

- Segmentación: separar la hoja o planta del fondo mediante el uso de GMM o Prominencia Bayesiana.
- Cuantificación de pigmentación: cuantificar la porción de píxeles de la superficie total de la hoja que se encuentren relacionados con algunas de las enfermedades haciendo uso de una Red Neuronal (NN).

El caso de la segmentación mediante uso de GMM se aplica cuando el fondo de la imagen no tenga mucho ruido, es decir, sea algún color sólido. Posteriormente a la segmentación se procede a realizar la cuantificación de los colores de los píxeles que pertenecen al segmento identificado como la hoja, mediante la red neuronal, con el objetivo de estimar la porción de la presencia de las enfermedades. Para el caso en el cual se tengan imágenes con fondos que contengan ruido, como por ejemplo, una imagen tomada en el exterior con un fondo de tierra, primero se hace uso de la red neuronal para identificar aquellos píxeles que puedan pertenecer a una hoja o planta (blanco, negro, verde, amarillo o café) y aquellos que a priori no puedan pertenecer a una hoja. Es decir se realiza un primer supuesto del segmento para posteriormente refinar mediante el uso de las técnicas de Prominencia Bayesiana con el objetivo de destacar o reconocer el objeto prominente en una imagen, que en este caso se espera sea la planta u hoja. Posteriormente se cuantifican los colores del segmento y se estima la porción de la presencia de alguna de las enfermedades en la planta.

A continuación se presenta un diagrama que describe de forma general la metodología y los puntos mencionados con anterioridad.

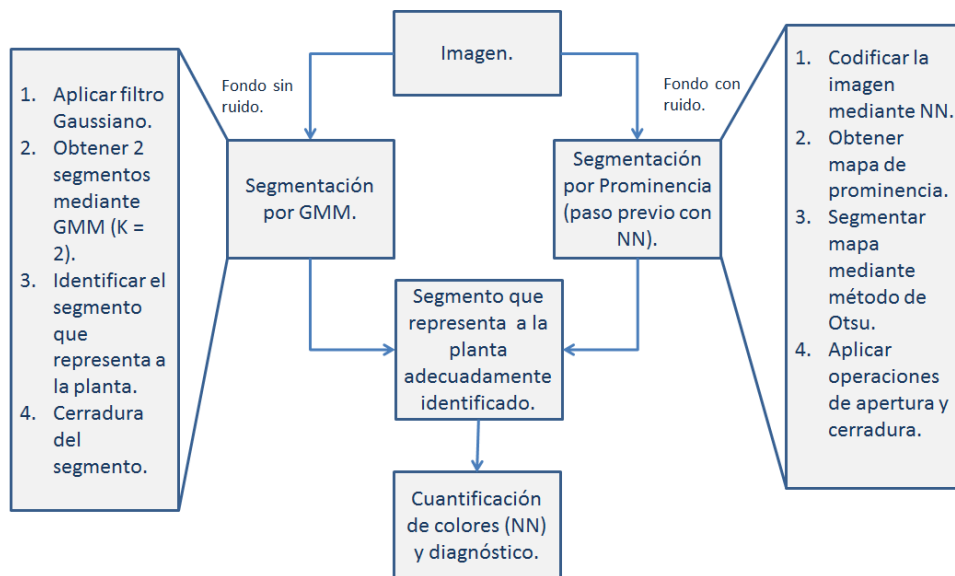


Figura 18: Diagrama de la estrategia general.

Para la estrategia de segmentación es importante considerar la imagen que se observa como un efecto y la causa es la distribución Gaussiana que genera la imagen para el caso de los fondos con poco ruido y el mapa de prominencia para el caso de los fondos con ruido, por lo tanto se generan las siguientes preguntas que serán contestadas en el desarrollo de la metodología.

- Dado que se tiene una imagen con poco ruido. ¿Cuáles son los parámetros de las distribuciones que podrían generarla?, ¿cuáles describen la distribución que genera a la planta?, ¿cómo se calculan?
- Dado que se tiene una imagen con ruido. ¿Cuál es el mapa de prominencia que mejor la describe?, ¿cómo se calcula?

Es importante destacar que la identificación automática del tipo de ruido y en consecuencia la metodología de segmentación a usar se contempla como una extensión a futuro del trabajo actual.

Además se incluye un diagrama con un nivel de detalle mayor en la descripción de los pasos y en el cual se especifican aquellos pasos con parámetros que tiene que definir el usuario.

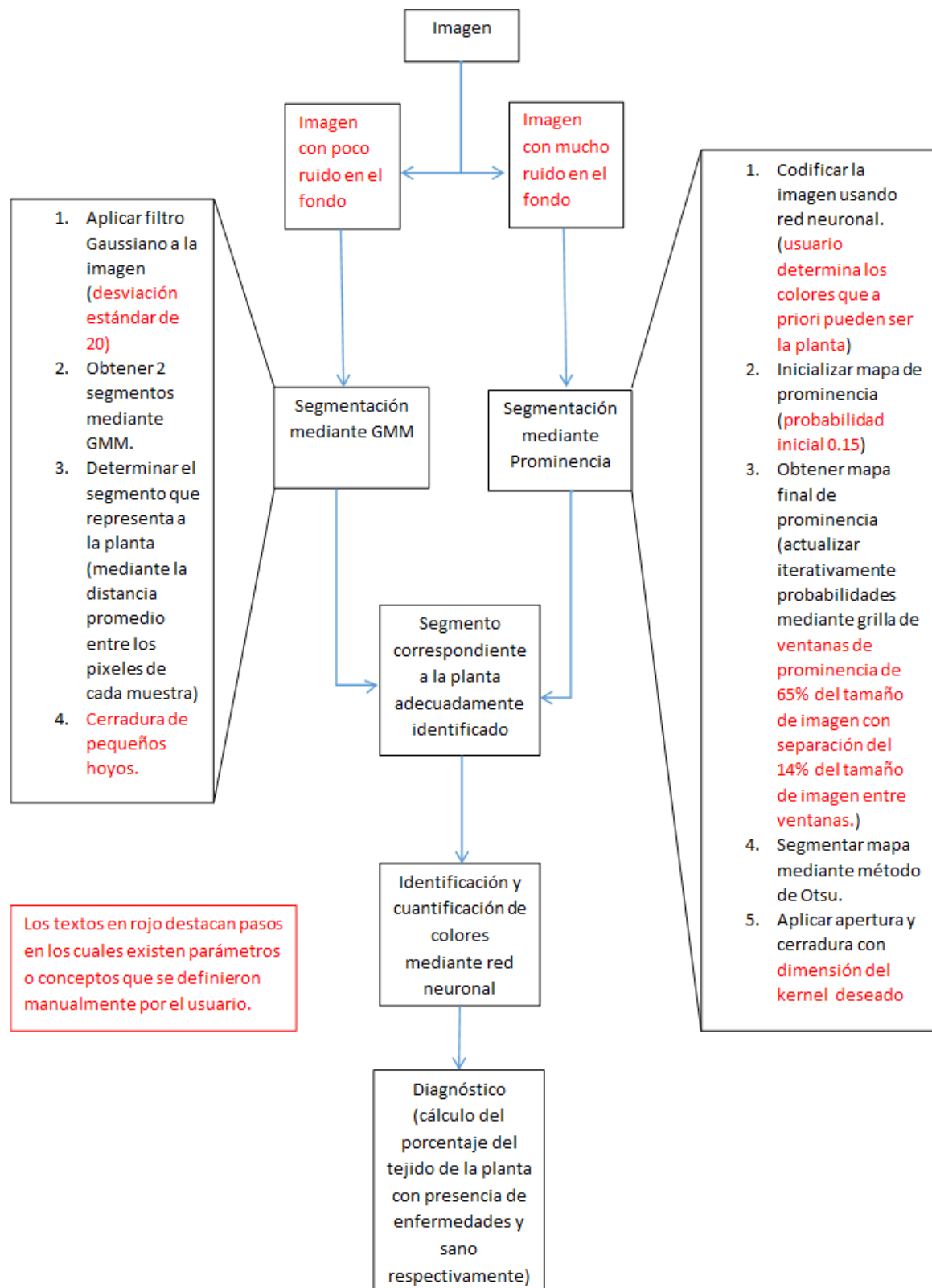


Figura 19: Diagrama detallado.

4.1. RED NEURONAL PARA EL RECONOCIMIENTO DE COLORES.

Como se mencionó con anterioridad se entrenó y diseñó una red neuronal para el reconocimiento de colores, se reconocen en total 5 tonalidades o colores que se consideran suficientes para poder realizar los reconocimientos de las enfermedades que se contemplan en la tesina, los colores y las enfermedades a las que se asocian son:

- Blanco: relacionado con la presencia de puntos blancos en una planta.
- Negro: relacionado con la presencia de necrosis en una planta.
- Verde: relacionado con una planta sana.
- Amarillo: relacionado con la presencia de clorosis.
- Café: relacionado con la presencia de necrosis.

Para realizar la etapa de entrenamiento de la red neuronal se construyó una base de datos de aproximadamente 350 registros, en los cuales cada renglón de la base de datos representa el índice rgb de un color y una etiqueta, la etiqueta sirve para identificar el color que se busca reconocer mediante el uso de la red neuronal, a continuación se presenta la codificación que fue utilizada.

- Blanco: 0
- Negro: 1
- Verde: 2
- Amarillo: 3
- Café: 4

La base de datos ya etiquetada fue dividida en 2 partes un 80 % de la base fue utilizada para la etapa de entrenamiento y un 20 % de la base fue utilizada para la etapa de predicción de la red neurona, se hizo uso del lenguaje de programación R y de la paquetería neural net, generando la siguiente red, a pesar de que la base de datos se construyó originalmente con la codificación de color RGB, la red se entrenó haciendo uso de la codificación HSV, ya que mostró ser mucho más robusta en fotos tomadas en exteriores.

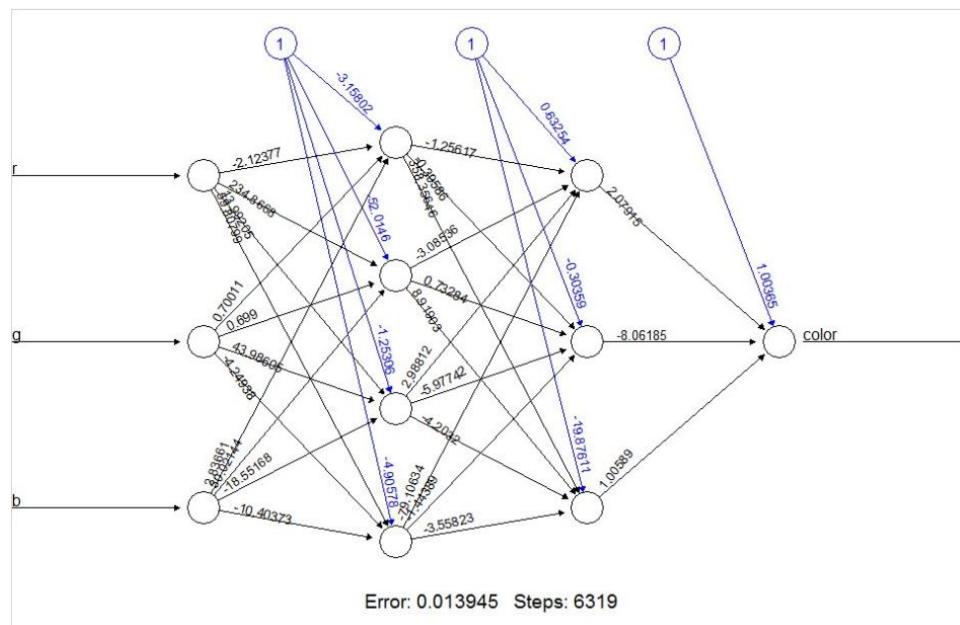


Figura 20: Representación de la red neuronal generada para clasificar colores.

La red neuronal alcanzó un error cuadrático medio de 0.008 y es utilizada en la etapa de clasificación de colores para posteriormente estimar la porción de presencia de alguna de las enfermedades consideradas en el alcance de la tesina.

A pesar de que el entrenamiento de la red se realizó en R, en gran medida por la calidad de las representaciones visuales de las redes entrenadas, pero dado que el resto del código de la implementación de la tesina fue realizado en Python, por las amplias paqueterías para procesamiento de imágenes, la red se tuvo que codificar manualmente en Python, a continuación se presenta la red neuronal que es utilizada en el presente trabajo de tesina.

```
def getColorNN2(rgb):
    color = 0
    r = rgb[0]/255
    g = rgb[1]/255
    b = rgb[2]/255
    h,s,v = colorsys.rgb_to_hsv(r,g,b)

    o11 = 1/(1 + math.exp(-1*(-3.158 - 2.123*h + 0.700*s + 2.836*v)))
    o12 = 1/(1 + math.exp(-1*(-52.014 + 234.866*h + 0.699*s + 30.021*v)))
    o13 = 1/(1 + math.exp(-1*(-1.253 + 13.992*h + 43.986*s - 18.551*v)))
    o14 = 1/(1 + math.exp(-1*(-4.905 + 59.807*h - 4.249*s - 10.403*v)))
    o21 = 1/(1 + math.exp(-1*(0.632 - 1.256*o11 - 3.085*o12 + 2.988*o13 - 79.106*o14)))
    o22 = 1/(1 + math.exp(-1*(-0.303 - 0.395*o11 + 0.732*o12 - 5.977*o13 - 1.443*o14)))
    o23 = 1/(1 + math.exp(-1*(-19.876 + 358.356*o11 + 8.919*o12 - 4.203*o13 - 3.558*o14)))
    output = round(1.003 + 2.079*o21 - 8.061*o22 + 1.005*o23,0)
    if output <= 0:
        color = 0
    elif output == 1:
        color = 1
    elif output == 2:
        color = 3
    elif output == 3:
        color = 2
    elif output == 4:
        color = 1
    else:
        color = 4
    return color
```

4.2. SEGMENTACIÓN CON POCO RUIDO EN EL FONDO.

En el caso de que las imágenes tengan poco ruido de fondo, es decir que tengan un fondo sólido, lo cual es común cuando las imágenes se toman en ambientes controlados se tiene el siguiente proceso para poder realizar la segmentación de la planta, para posteriormente clasificar los colores y estimar la presencia de enfermedades, el proceso de segmentación está basado principalmente en un modelo GMM. Es importante destacar que las implementaciones se hicieron en Python y haciendo uso de la paquetería `sklearn.mixture` y de los métodos `GaussianMixture`, ya que ofrecieron mayor velocidad que las implementaciones programadas de forma independiente, el proceso es el siguiente.

- Preparar la imagen mediante la aplicación de un filtro gaussiano.
- Obtener parámetros iniciales por k-means.
- Clasificar los píxeles en 2 segmentos haciendo uso de un modelo GMM.
- Identificar el segmento que corresponde a la planta u hoja.
- Cerradura del segmento que corresponde a la hoja.

Se hará uso de la siguiente imagen de una hoja que presenta coloraciones blanco y negro, representando presencia de las enfermedades de puntos blancos y necrosis, respectivamente, para poder ejemplificar los resultados de algunas de las etapas que forman parte de la implementación para los casos de fondos con poco ruido.



Figura 21: Hoja con simulación de puntos blancos y necrosis.

4.2.1. APLICACIÓN DE UN FILTRO GAUSSIANO.

La imagen se convierte a escala de grises y se realiza una copia de la misma, a dicha copia se le aplica un filtro gaussiano con la intención de considerar la información del vecindario de cada píxel en el momento de aplicar los modelos GMM, se obtiene una tercera imagen realizando la suma ponderada de la imagen original y la imagen a la cual se le ha aplicado el filtro gaussiano, se da un peso de 0.5 a la imagen original y de 0.5 a la imagen a la cual se le ha aplicado el filtro gaussiano.

4.2.2. OBTENCIÓN DE PARÁMETROS INICIALES POR K-MEANS.

Los píxeles de la imagen en canal de grises se separan en 2 grupos haciendo uso del conocido algoritmo de k-means, el cual genera una asignación dura mientras que GMM realiza una asignación suave. K - means pertenece al tipo de algoritmos no supervisados en el cual aleatoriamente se generan 2 centroides y en cada iteración del algoritmo los píxeles que se encuentran más cercanos a un centroide forman un clúster, se calcula la media de los clústers y dicha media pasa a ser el nuevo centroide para posteriormente volver a repetir el proceso hasta satisfacer un criterio de convergencia.

4.2.3. CLASIFICACIÓN DE LOS PÍXELES EN 2 SEGMENTOS HACIENDO USO DE GMM.

Una vez que se tienen los centroides obtenidos por el algoritmo de k-means con sus respectivas medias y desviaciones estándar, se procede a realizar la segmentación por medio de un modelo GMM que se compone de dos distribuciones gaussianas univariadas, la variable x corresponde al valor en la escala de grises de un pixel, en el contexto en el cual se busca segmentar la imagen en planta o fondo, el modelo GMM consta de 2 distribuciones gaussianas, a continuación se expresa el modelo que describe el objetivo de este paso del proceso de segmentación.

$$p(x) = \pi_1 \mathcal{N}(x|\mu_1, \sigma_1^2) + \pi_2 \mathcal{N}(x|\mu_2, \sigma_2^2) \quad (44)$$

Donde

π_1, μ_1, σ_1^2 son el peso, la media y la varianza respectivamente de la distribución 1.

π_2, μ_2, σ_2^2 son el peso, la media y la varianza respectivamente de la distribución 2.

Para encontrar el valor óptimo de los parámetros presentados dado una cantidad N de píxeles $X = \{x_1, x_2, \dots, x_n\}$ se hace uso de la función logarítmica de verosimilitud, la cual se busca maximizar con el objetivo de encontrar los parámetros que permitan encontrar el mejor ajuste del modelo a los datos disponibles, a continuación se presenta la función logarítmica de verosimilitud.

$$\ln p(X|\pi_1, \mu_1, \sigma_1^2, \pi_2, \mu_2, \sigma_2^2) = \sum_{n=1}^N \ln(\pi_1 \mathcal{N}(x_n|\mu_1, \sigma_1^2) + \pi_2 \mathcal{N}(x_n|\mu_2, \sigma_2^2)) \quad (45)$$

El algoritmo EM sirve para obtener los parámetros que logren maximizar la función expresada con anterioridad, el algoritmo EM consta de dos pasos que se repiten hasta alcanzar una convergencia en la función de verosimilitud o algún criterio de paro, los pasos son cálculo de esperanza y maximización.

Paso de cálculo de esperanza.

En este paso se calculan los pesos o responsabilidades que tienen las distribuciones correspondientes a la distribución 1 $\gamma(z_{n1})$ o la distribución 2 $\gamma(z_{n2})$ para cada una de las observaciones x_n , dichos pesos servirán para el cálculo de los parámetros del modelo y al igual para poder asignar a cada observación a la distribución gaussiana para la cual tenga el mayor peso al finalizar el algoritmo, es importante destacar que z_{n1} es una variable latente binaria que toma el valor de 1 cuando se interpreta que x_n es el resultado de un muestreo de la distribución 1, con una responsabilidad $\gamma(z_{n1})$, aplica lo mismo para la distribución 2.

$$\gamma(z_{n1}) = \frac{\pi_1 \mathcal{N}(x_n|\mu_1, \sigma_1^2)}{\pi_1 \mathcal{N}(x_n|\mu_1, \sigma_1^2) + \pi_2 \mathcal{N}(x_n|\mu_2, \sigma_2^2)} \quad (46)$$

$$\gamma(z_{n2}) = \frac{\pi_2 \mathcal{N}(x_n|\mu_2, \sigma_2^2)}{\pi_1 \mathcal{N}(x_n|\mu_1, \sigma_1^2) + \pi_2 \mathcal{N}(x_n|\mu_2, \sigma_2^2)} \quad (47)$$

Donde

$\gamma(z_{n1})$ es el peso que tiene la distribución 1 para la observación n .

$\gamma(z_{n2})$ es el peso que tiene la distribución 2 para la observación n .

Paso de maximización.

Ahora en este paso se calculan los valores para los parámetros que caracterizan el modelo GMM haciendo uso de los pesos que se calcularon en el paso de la esperanza, primero se calcula los nuevos valores para las medias de las distribuciones.

$$\mu_1^{new} = \frac{1}{N_1} \sum_{n=1}^N \gamma(z_{n1}) x_n \quad (48)$$

$$\mu_2^{new} = \frac{1}{N_2} \sum_{n=1}^N \gamma(z_{n2}) x_n \quad (49)$$

Ya que se calcularon las nuevas medias, se procede a calcular las nuevas varianzas para la distribuciones.

$$\sigma_1^{2^{new}} = \frac{1}{N_1} \sum_{n=1}^N \gamma(z_{n1}) (x_n - \mu_1^{new})^2 \quad (50)$$

$$\sigma_2^{2^{new}} = \frac{1}{N_2} \sum_{n=1}^N \gamma(z_{n2}) (x_n - \mu_2^{new})^2 \quad (51)$$

Se actualizan los valores de los pesos de cada distribución.

$$\pi_1^{new} = \frac{N_1}{N} \quad (52)$$

$$\pi_2^{new} = \frac{N_2}{N} \quad (53)$$

Donde.

$$N_1 = \sum_{n=1}^N \gamma(z_{n1}) \quad (54)$$

$$N_2 = \sum_{n=1}^N \gamma(z_{n2}) \quad (55)$$

Con los nuevos parámetros se evalúa la función logarítmica de verosimilitud hasta alcanzar una convergencia o hasta llegar a un criterio de paro.

$$\ln p(X|\pi_1, \mu_1, \sigma_1^2, \pi_2, \mu_2, \sigma_2^2) = \sum_{n=1}^N \ln(\pi_1 \mathcal{N}(x_n|\mu_1, \sigma_1^2) + \pi_2 \mathcal{N}(x_n|\mu_2, \sigma_2^2)) \quad (56)$$

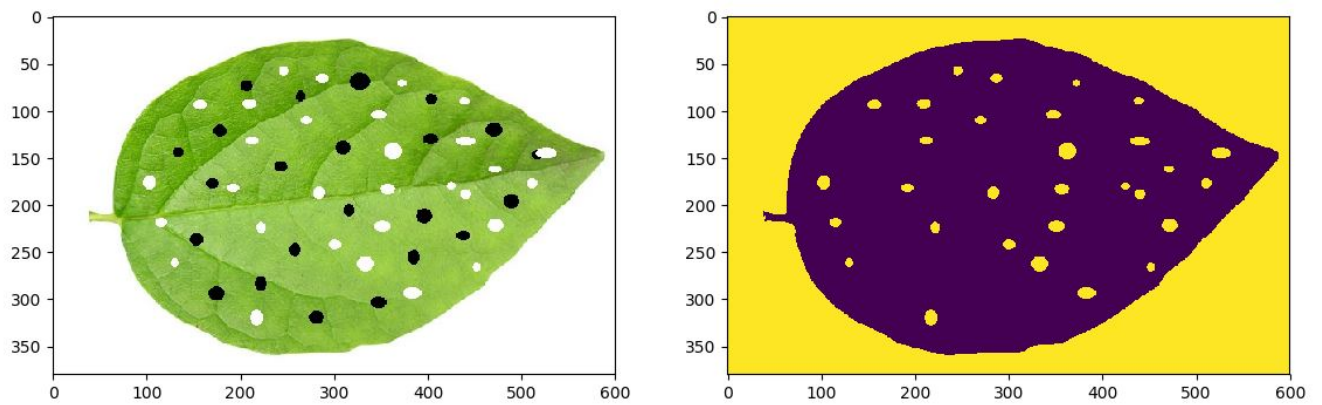


Figura 22: Ejemplo de la segmentación alcanzada mediante GMM.

4.2.4. IDENTIFICACIÓN DE LA PLANTA U HOJA.

Una vez que se tiene la imagen binarizada, es decir separada en 2 segmentos, se tiene que reconocer o identificar de alguna manera el que pertenece a la hoja y el que pertenece al fondo. Se generó un algoritmo basado en el principio de que dado que una hoja o planta generaría un segmento unido, concentrado y con cierta esfericidad en su forma, si se genera aleatoriamente una muestra de los píxeles que corresponden al segmento 1 y otra muestra de los píxeles que corresponden al segmento 2, para posteriormente en base a las coordenadas de los píxeles calcular la distancia promedio de las muestras de cada segmento y en base a esas distancias elegir aquella que tenga el menor valor para ser reconocido como el segmento que representa a la planta u hoja.

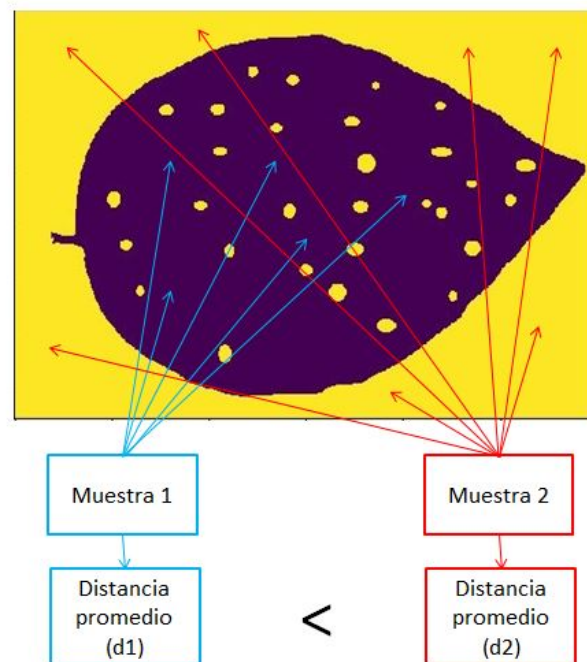


Figura 23: Representación del principio para identificar el segmento que pertenece a a planta.

4.2.5. CERRADURA DEL SEGMENTO QUE PERTENECE A LA HOJA.

Ya que se tiene identificado el segmento que pertenece a la hoja, se procede a realizar la cerradura de los pequeños hoyos que pueden existir en el segmento, una manera de hacerlo es aplicando las técnicas morfológica

matemática del análisis de imágenes tradicional, aplicando la operación de cerradura.

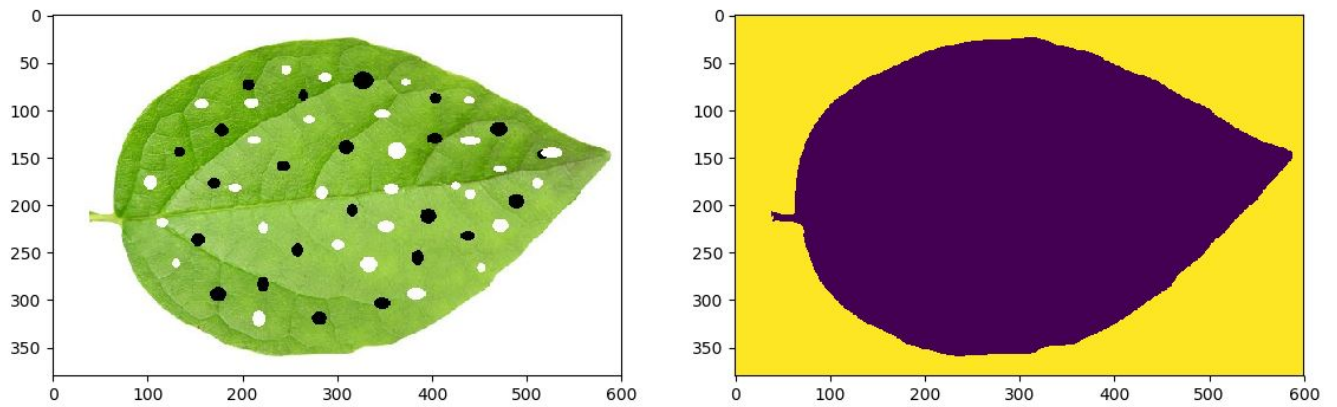


Figura 24: Ejemplo del resultado alcanzado por la cerradura, se observa que los hoyos han desaparecido.

Se puede observar que se ha logrado una cerradura adecuada de la hoja, ya no se tienen los hoyos y se puede proceder a estimar la porción de la hoja que representan las enfermedades mencionadas con anterioridad.

A continuación se presenta una secuencia de imágenes con el objetivo de observar la evolución de la segmentación en cada uno de los pasos principales de la estrategia.

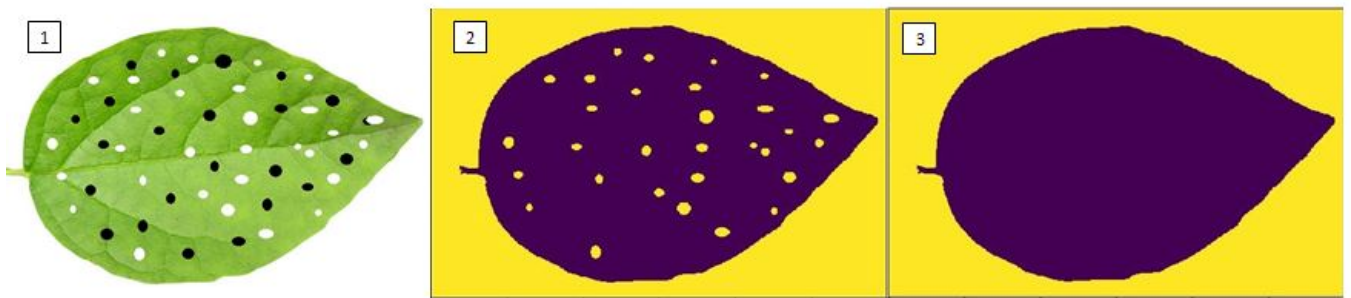


Figura 25: Evolución de la segmentación 1) Imagen original, 2) Segmentación por GMM , 3) Segmentación después de cerradura

4.2.6. PSEUDO CÓDIGO.

A continuación se presenta un pseudo código para el proceso de segmentación mediante GMM.

1. imagen \leftarrow aplicar filtro gaussiano a la imagen inicial
2. $N \leftarrow$ número de pixeles de la imagen, $C \leftarrow$ criterio de convergencia
3. inicializar parámetros mediante k means

$$\mu_1, \sigma_1^2, \pi_1, \mu_2, \sigma_2^2, \pi_2$$

4. paso de esperanza.

for n in N

$$\gamma(z_{n1}) = \frac{\pi_1 \mathcal{N}(x_n | \mu_1, \sigma_1^2)}{\pi_1 \mathcal{N}(x_n | \mu_1, \sigma_1^2) + \pi_2 \mathcal{N}(x_n | \mu_2, \sigma_2^2)}$$

$$\gamma(z_{n2}) = \frac{\pi_2 \mathcal{N}(x_n | \mu_2, \sigma_2^2)}{\pi_1 \mathcal{N}(x_n | \mu_1, \sigma_1^2) + \pi_2 \mathcal{N}(x_n | \mu_2, \sigma_2^2)}$$

5. paso de maximización.

$$N_1 = \sum_{n=1}^N \gamma(z_{n1})$$

$$N_2 = \sum_{n=1}^N \gamma(z_{n2})$$

$$\pi_1^{new} = \frac{N_1}{N}$$

$$\pi_2^{new} = \frac{N_2}{N}$$

$$\mu_1^{new} = \frac{1}{N_1} \sum_{n=1}^N \gamma(z_{n1}) x_n$$

$$\mu_2^{new} = \frac{1}{N_2} \sum_{n=1}^N \gamma(z_{n2}) x_n$$

$$\sigma_1^{2new} = \frac{1}{N_1} \sum_{n=1}^N \gamma(z_{n1}) (x_n - \mu_1^{new})^2$$

$$\sigma_2^{2new} = \frac{1}{N_2} \sum_{n=1}^N \gamma(z_{n2}) (x_n - \mu_2^{new})^2$$

6. verificar convergencia.

$$\ln p(X|\theta) = \sum_{n=1}^N \ln(\pi_1 \mathcal{N}(x_n | \mu_1, \sigma_1^2) + \pi_2 \mathcal{N}(x_n | \mu_2, \sigma_2^2))$$

$$\ln p(X|\theta^{new}) = \sum_{n=1}^N \ln(\pi_1^{new} \mathcal{N}(x_n | \mu_1^{new}, \sigma_1^{2new}) + \pi_2^{new} \mathcal{N}(x_n | \mu_2^{new}, \sigma_2^{2new}))$$

If $|\ln p(X|\theta^{new}) - \ln p(X|\theta)| < C$ Then

for n in N

If $\gamma(z_{n1}) \geq \gamma(z_{n2})$ Then

$$s_1 \leftarrow x_n$$

Else

$$s_2 \leftarrow x_n$$

$$\theta \leftarrow \theta^{new}$$

ir a paso 7

Else

$$\theta \leftarrow \theta^{new}$$

ir a paso 4

7. identificar el segmento que representa a la planta

$$m_1 \leftarrow \text{muestra de } s_1$$

$$m_2 \leftarrow \text{muestra de } s_2$$

$$d_1 \leftarrow \text{promedio secuencial de distancia de } m_1$$

$$d_2 \leftarrow \text{promedio secuencial de distancia de } m_2$$

If $d_1 \leq d_2$ Then

$$s_1 \leftarrow \text{planta}$$

Else

$$s_2 \leftarrow \text{planta}$$

8. aplicar cerradura a planta

9. fin

4.3. SEGMENTACIÓN PARA EL CASO DE FONDOS CON RUIDO.

En el caso de que las imágenes hayan sido tomadas en exteriores, es muy probable que el fondo contenga mucho ruido al provenir de ambientes que no están controlados por lo cual los modelos GMM no ofrecen un desempeño aceptable, es en dicho escenario que se realiza la segmentación mediante el uso de Prominencia Bayesiana, a continuación se describe el proceso general.

- Codificar la imagen haciendo uso de la red neuronal.
- Obtener el mapa de prominencia.
- Segmentar el mapa de prominencia con el Método de Otsu.
- Aplicar apertura y cerradura a la imagen segmentada.

Con el objetivo de que se pueda visualizar el impacto que tiene cada uno de los pasos en la segmentación de una hoja con un fondo ruidoso, se hará uso de la siguiente imagen para poder apreciar la evolución en la calidad de la segmentación alcanzada al ir progresando en cada uno de los pasos.



Figura 26: Imagen a ser segmentada.

4.3.1. CODIFICACIÓN LA IMAGEN HACIENDO USO DE LA RED NEURONAL.

Dado que la aplicación de las técnicas de Prominencia Bayesiana dieron resultados poco satisfactorios en algunos casos haciendo uso de la escala de grises como medio de codificación o representación de la imagen se generó la idea de incorporar el hecho de que la tesina está enfocada en la identificación de un número limitado de enfermedades en plantas y por lo tanto el número limitado de colores que se reconocen en la red son de interés para lograr el objetivo, todos los colores restantes que en este caso generan ruido, son una fuente de complejidad de la cual se puede prescindir. Por lo tanto la idea es codificar la imagen, pasar de una matriz que representa la imagen en RGB a tener la siguiente codificación haciendo uso de la red neuronal, con el objetivo de que todo lo que sea ruido sea codificado con un 0, así mejorando de forma considerable el desempeño de las técnicas de Prominencia Bayesiana.

- Blanco - 0
- Negro - 0/1 (Dependiendo del caso).
- Verde - 1
- Amarillo - 1
- Café - 0/1 (Dependiendo del caso).

- Color no identificado - 0 (Ruido).

Como se puede observar los colores café y negro pueden ser considerados como ruido o no, dependiendo del caso de la imagen, normalmente son considerados como ruido, el caso del color blanco siempre es considerado como ruido ya que normalmente está presentes en puntos blancos que serán cerrados haciendo uso del paso de la aplicación de apertura y cerradura.

Se observa que la red neuronal codifica todo aquel color que no puede reconocer con un 0, con el objetivo de estandarizar el ruido y reducir la probabilidad de que se considere prominente algo del fondo, a continuación se presenta el resultado de dicha codificación, que puede ser interpretada como la creencia inicial de lo que es la planta u hoja prominente.

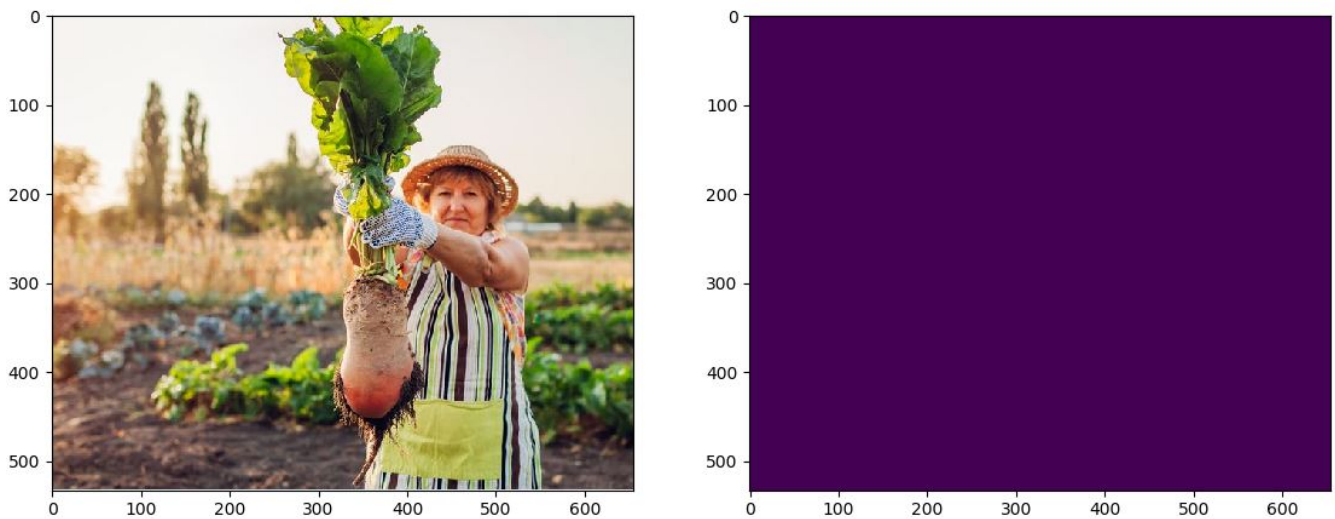


Figura 27: Imagen original y supuesto inicial de la planta, codificado por la red neuronal.

4.3.2. OBTENCIÓN DEL MAPA DE PROMINENCIA.

Se implementan los conceptos de Prominencia considerando que se inicializa el mapa de prominencia con una probabilidad de ser prominente de 0.15 para todos los píxeles. A continuación se muestra una representación visual del mapa inicial de prominencia.

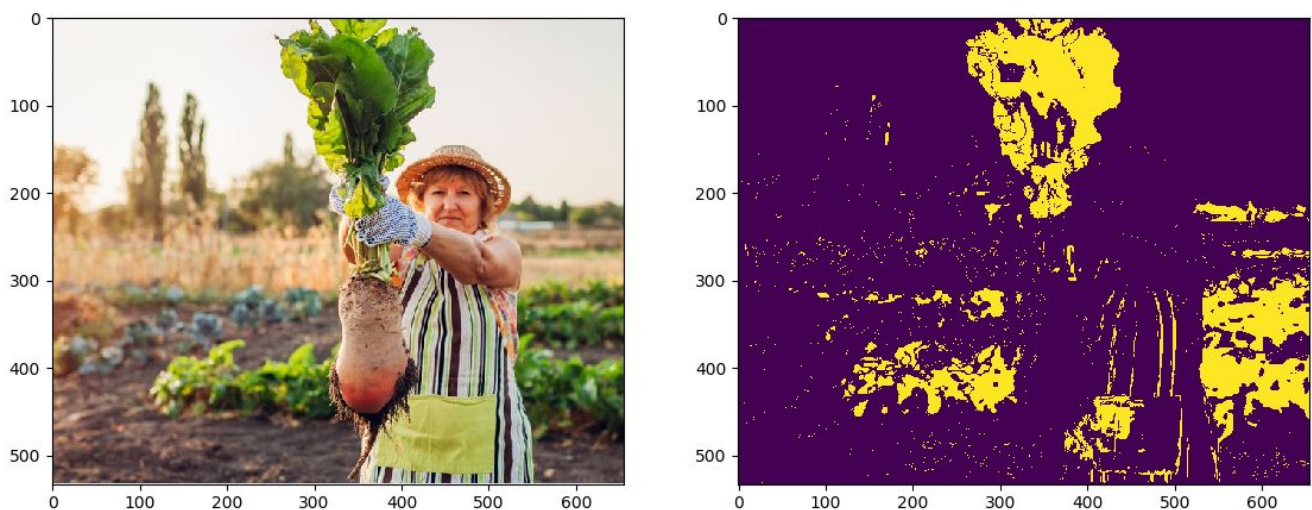


Figura 28: Imagen original y mapa inicial de prominencia, todos los píxeles tienen el mismo valor.

Tomando el valor promedio del renglón y de la columna de los píxeles que a priori se consideran como parte de una planta, se genera un patrón de ventanas de prominencia en forma de grilla que se puede visualizar a continuación.

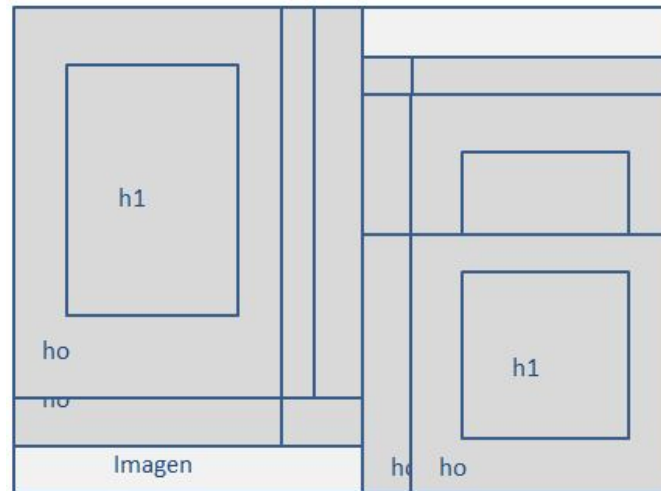


Figura 29: Representación visual de una posible ubicación de las ventanas de prominencia.

Los dos tipos de prominencias se tratan de una prominencia local, que es la que se calcula comparando lo que está dentro de la ventana de prominencia (K) contra el marco de la misma ventana (W) y la prominencia global que se calcula comparando lo que está dentro de la ventana de prominencia (K) contra lo que está fuera de la ventana, es decir el resto de la imagen (I), se le da un peso de 0.8 a la prominencia local y un peso de 0.20 a la prominencia global, cabe destacar que sólo los píxeles que estén codificados con 1 serán considerados en el cálculo de prominencia, ya que el hecho de estar etiquetados con 1 demuestra una creencia inicial de que pueden ser parte de una planta u hoja, en caso de que tengan la codificación de 0 se considera que no pueden ser parte de la planta y que forman parte del ruido, es decir aquello desconocido.

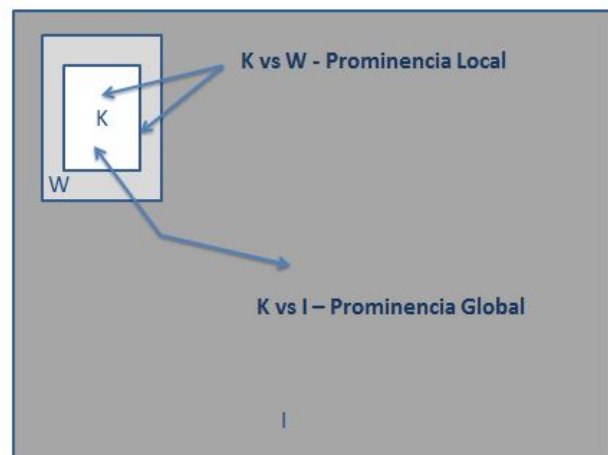


Figura 30: Representación visual de los conceptos de prominencia local y global.

A continuación se presentan las expresiones que en el contexto considerado permiten generar un mapa de prominencia que considere tanto la prominencia local y la prominencia global.

Se tiene que:

h_0 cuando el píxel no es considerado prominente, se asocia con I y W.

h_1 cuando el píxel es considerado prominente se asocia con K.

Por lo tanto para el caso de la prominencia local, se calcula la probabilidad de prominencia de un píxel de la siguiente manera.

$$p_l(h_1|x) = \frac{p(x|h_1)p(h_1)}{p(x|h_0)p(h_0) + p(x|h_1)p(h_1)} \quad (57)$$

Donde.

$p(h_0)$ es la probabilidad a priori de que el píxel no sea prominente.

$p(h_1)$ es la probabilidad a priori de que el píxel sea prominente.

$p(x|h_0)$ es la probabilidad de observar la codificación del píxel dado que no es prominente.

$p(x|h_1)$ es la probabilidad de observar la codificación del píxel dado que es prominente.

Para el cálculo de las probabilidades condicionales se tiene lo siguiente.

$$p(x|h_0) = \frac{W_1}{W_T} \quad (58)$$

$$p(x|h_1) = \frac{K_1}{K_T} \quad (59)$$

Donde.

W_1 es el número de píxeles con codificación 1 en W.

W_T es el número total de píxeles en W.

K_1 es el número total de píxeles con codificación 1 en K.

K_T es el número total de píxeles en K.

Para el caso de la prominencia global, se tienen las siguientes expresiones.

$$p_g(h_1|x) = \frac{p(x|h_1)p(h_1)}{p(x|h_0)p(h_0) + p(x|h_1)p(h_1)} \quad (60)$$

Donde.

$p(h_0)$ es la probabilidad a priori de que el píxel no sea prominente.

$p(h_1)$ es la probabilidad a priori de que el píxel sea prominente.

$p(x|h_0)$ es la probabilidad de observar la codificación del píxel dado que no es prominente.

$p(x|h_1)$ es la probabilidad de observar la codificación del píxel dado que es prominente.

Para el cálculo de las probabilidades condicionales se tiene lo siguiente.

$$p(x|h_0) = \frac{I_1}{I_T} \quad (61)$$

$$p(x|h_1) = \frac{K_1}{K_T} \quad (62)$$

Donde.

I_1 es el número de píxeles con codificación 1 en W.

I_T es el número total de píxeles en W.

K_1 es el número total de píxeles con codificación 1 en K.

K_T es el número total de píxeles en K.

Por lo tanto ya teniendo los valores para la probabilidad de prominencia local $p_l(h_1|x)$ y para la prominencia global $p_g(h_1|x)$ respectivamente el valor de la nueva prominencia es el siguiente.

$$p(h_1|x) = 0,8 * p_l(h_1|x) + 0,2 * p_g(h_1|x) \quad (63)$$

En cada una de las 15 iteraciones del algoritmo se modifica la probabilidad de que un pixel sea prominente, se guarda en un mapa de prominencia alterno el valor máximo alcanzado por cada uno de los píxeles que fueron considerados en las iteraciones del algoritmo, al finalizar el mapa de prominencias considerado es aquel compuesto por los valores máximos de prominencia.

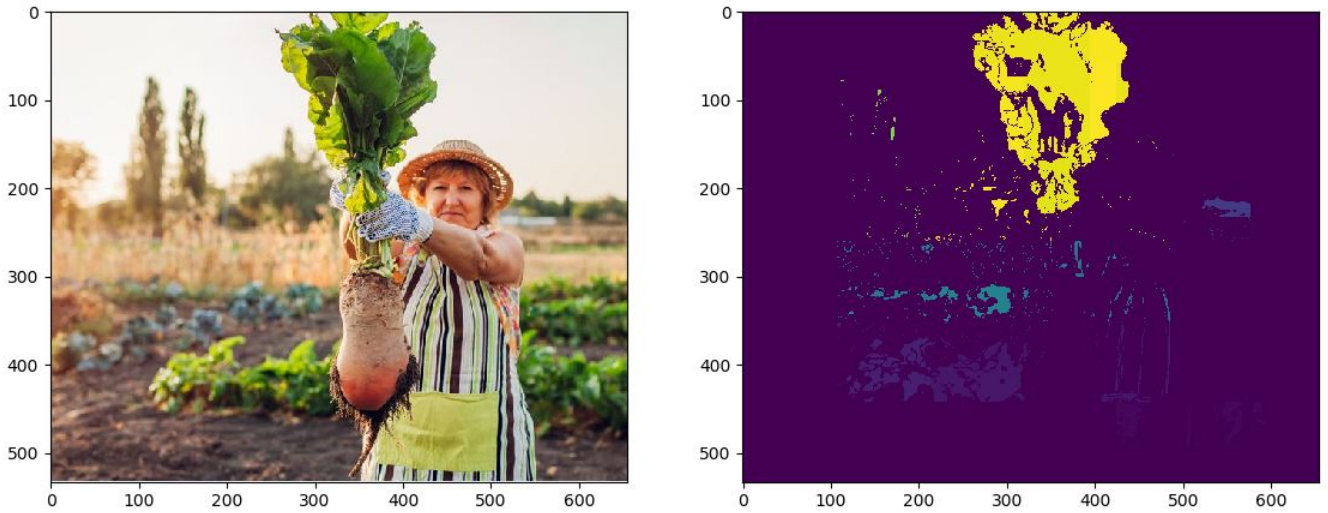


Figura 31: Imagen original y el mapa de prominencia obtenido.

A continuación se presenta el mapa de prominencia alcanzado, se puede observar el mapa de prominencia con distintos valores de probabilidad que están entre 0.15 que es la inicial y la mínima del mapa y 1, se observan unos píxeles en azul, otros en morado y otros en amarillo.

4.3.3. SEGMENTACIÓN DEL MAPA DE PROMINENCIA MEDIANTE OTSU.

Ya que se tiene el mapa de prominencia se tiene que realizar una binarización en 2 segmentos uno correspondiente al fondo, aquello que no es prominente y otro que pertenece a la planta u hoja, es decir aquello que sí es prominente, para realizar dicha tarea se tiene que encontrar un umbral entre 0 y 1 que logre segmentar de la mejor manera posible el mapa de prominencia, como se explicó en la sección del marco teórico existe una metodología muy buena para dichos casos, que se conoce como Método de Otsu el cual fue implementado en Python haciendo uso de la paquetería skimage y los métodos de filters.

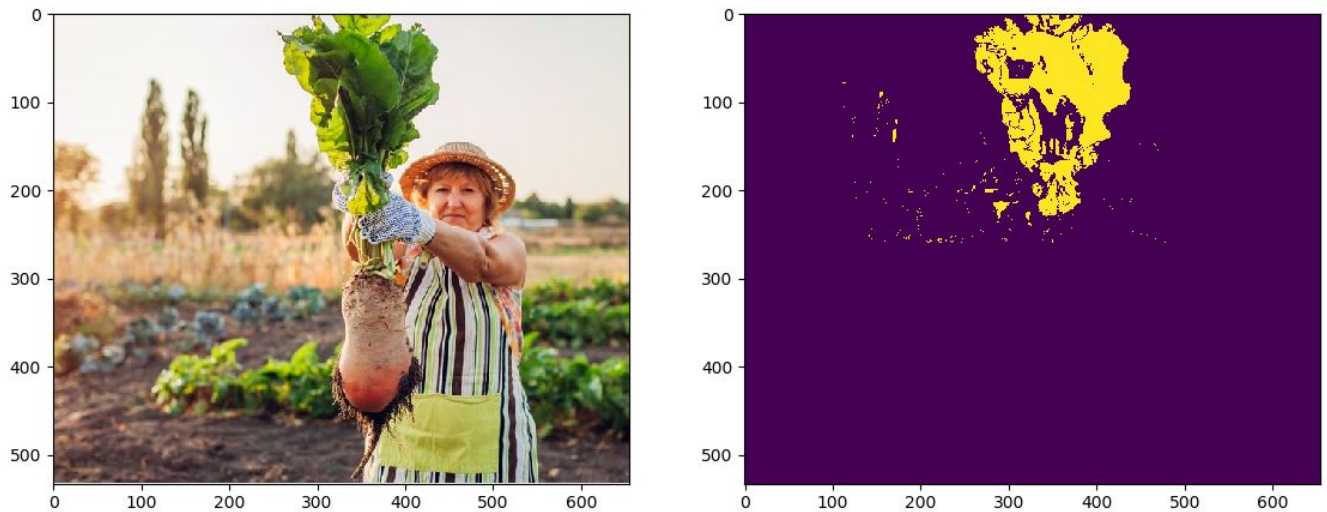


Figura 32: Imagen original y segmentación alcanzada al aplicar Otsu al mapa de prominencia.

Se observa que ya se tiene una imagen binarizada en la cual los píxeles amarillos corresponden al segmento de la planta y los píxeles morados al segmento del fondo, aunque existen algunos detalles, como aquellos puntos que se encuentran en el fondo, que serán eliminados haciendo uso de técnicas de morfología matemática de imágenes.

4.3.4. APLICACIÓN DE APERTURA Y CERRADURA A LA IMAGEN SEGMENTADA.

Ahora que se tiene una imagen segmentada, que en algunos de los casos puede llegar a tener pequeños hoyos en el segmento que corresponde a la hoja o puntos en el segmento que corresponde al fondo, se implementan operaciones de morfología matemática que son las operaciones de apertura y cerradura.

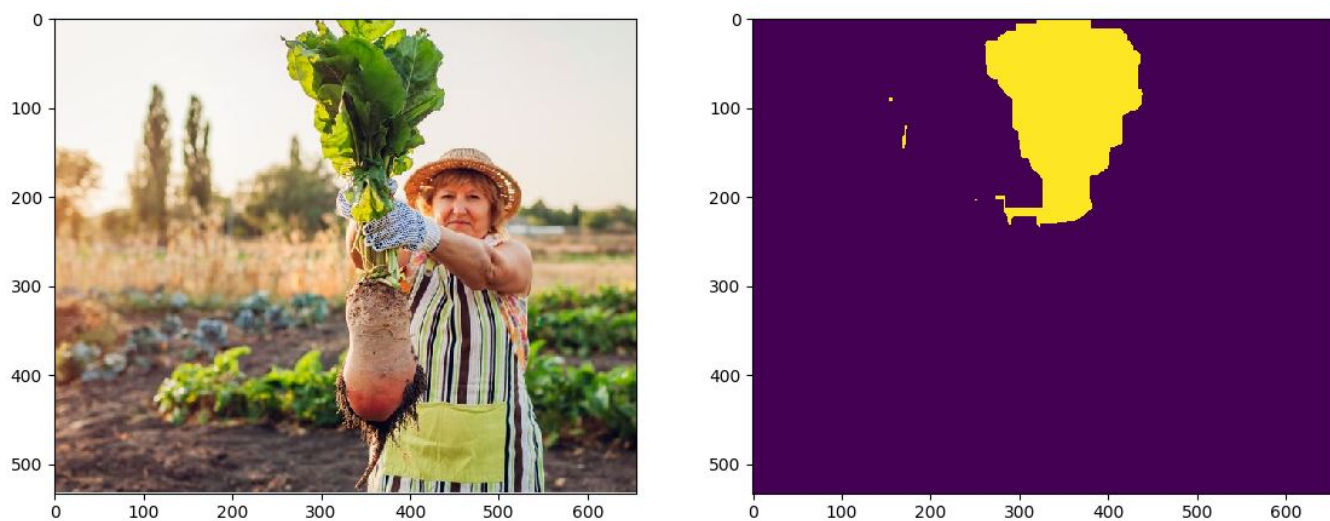


Figura 33: Imagen original y segmentación alcanzada al aplicar las operaciones de apertura y cerradura.

A continuación se presenta un conjunto de dos secuencias de imágenes con el objetivo de observar la evolución de la segmentación en cada uno de los pasos principales de la estrategia.



Figura 34: Evolución de la segmentación 1) Original 2) Codificación por red 3) Mapa inicial de prominencia , 4) Mapa final de prominencia , 5) Segmentación después de Otsu 6) Segmentación después de apertura y cerradura.

4.3.5. PSEUDO CÓDIGO.

A continuación se presenta un pseudo código para el proceso de segmentación mediante Prominencia.

1. $R \leftarrow$ número de renglones de la imagen (im), $C \leftarrow$ número de columnas de la imagen (im)
2. codificar la imagen, inicializar el mapa de prominencia (p) y obtener el valor del renglón y columnas promedio de los pixeles codificados con "1"

$r_p, c_p, I_1 \leftarrow -0$

for r in R

for c in C

$p[i, j] \leftarrow 0,15$

$h, s, v \leftarrow im[r, c]$

$im[r, c] \leftarrow NN(h, s, v)$

If $im[r, c] = 1$ Then

$r_p \leftarrow r_p + r$

$c_p \leftarrow c_p + c$

$I_1 \leftarrow I_1 + 1$

$r_p \leftarrow \text{redondear}(\frac{r_p}{I_1})$

$c_p \leftarrow \text{redondear}(\frac{c_p}{I_1})$

3. determinar los centros de las ventanas de prominencia mediante r_p, c_p y la separación deseada.

$step \leftarrow \text{redondear}(max(R, C))$

for i in $(0, 2)$

for j in $(0, 2)$

centro $\leftarrow [r_p - i * step, c_p - j * step]$

If $j > 0$ Then

centro $\leftarrow [r_p + i * step, c_p + j * step]$

4. actualizar valores de prominencia

$p_{max} \leftarrow p$

$l \leftarrow max(R, C)$

$I_T \leftarrow R * C$

$K_1, K_T, W_1, W_T \leftarrow 0$

for centro in centros

$r_c, c_c \leftarrow \text{centro}$

$minR \leftarrow \text{redondear}(max(0, r_c - 0,325 * l)), maxR \leftarrow \text{redondear}(min(R, r_c + 0,325 * l))$

$minC \leftarrow \text{redondear}(max(0, c_c - 0,325 * l)), maxC \leftarrow \text{redondear}(min(R, c_c + 0,325 * l))$

$minR_K \leftarrow \text{redondear}(1,075 * minR), maxR_K \leftarrow \text{redondear}(0,925 * maxR)$

$minC_K \leftarrow \text{redondear}(1,075 * minC), maxC_K \leftarrow \text{redondear}(0,925 * maxC)$

for i in $(minR, maxR)$

for j in $(minC, maxC)$

If $minR_K < i < maxR_K$ and $minC_K < j < maxC_K$ Then

$K_T \leftarrow K_T + 1$

If $im[i, j] = 1$ Then

$K_1 \leftarrow K_1 + 1$

Else

$W_T \leftarrow W_T + 1$

If $im[i, j] = 1$ Then

$W_1 \leftarrow W_1 + 1$

for i in $(minR_K, maxR_K)$

for j in $(minC_K, maxC_K)$

If $im[i, j] = 1$ Then

$p(h_0) \leftarrow 1 - p[i, j]$

$p(h_1) \leftarrow p[i, j]$

$p(x|h_1) \leftarrow \frac{K_1}{K_T}$

$p_l(x|h_0) \leftarrow \frac{W_1}{W_T}$

$p_g(x|h_0) \leftarrow \frac{I_1}{I_T}$

$$p_l(h_1|x) \leftarrow \frac{p(x|h_1)p(h_1)}{p(x|h_1)p(h_1) + p_l(x|h_0)p(h_0)}$$

$$p_g(h_1|x) \leftarrow \frac{p(x|h_1)p(h_1)}{p(x|h_1)p(h_1) + p_g(x|h_0)p(h_0)}$$

$$p(h_1|x) \leftarrow 0,8 * p_l(h_1|x) + 0,2 * p_g(h_1|x)$$

$$p[i, j] \leftarrow p(h_1|x)$$

If $p(h_1|x) \geq p_{max}[i, j]$ Then

$$p_{max}[i, j] \leftarrow p(h_1|x)$$

5. segmentar p_{max} mediante Método de Otsu.
6. aplicar operaciones de apertura y cerradura al segmento.
7. fin

4.4. CUANTIFICACIÓN DE PIGMENTACIÓN

Dado que ya se ha segmentado la planta u hoja haciendo uso de GMM o mediante Prominencia Bayesiana para los casos de fondo con poco o mucho ruido respectivamente se tiene que realizar la cuantificación de los colores con el objetivo de estimar la porción de presencia de alguna de las enfermedades consideradas, es decir del segmento que pertenece a la planta u hoja se van a contar el número de píxeles que corresponden a cada uno de los colores considerados para obtener la porción de cada uno de ellos en relación al total de píxeles en el segmento que corresponde a la planta.

- Porción Blanca: $\frac{PB}{PT}$
- Porción Negra: $\frac{PN}{PT}$
- Porción Verde: $\frac{PV}{PT}$
- Porción Amarilla: $\frac{PA}{PT}$
- Porción Café: $\frac{PC}{PT}$

Donde.

PT es el número de píxeles blancos.

PN es el número de píxeles negros.

PV es el número de píxeles verdes.

PA es el número de píxeles amarillos.

PC es el número de píxeles cafés.

PT es el número de píxeles totales de la hoja.

Ahora que se tienen las porciones de colores, se procede a calcular las porciones de presencia de cada una de las enfermedades consideradas.

- Porción de Puntos Blancos : Porción Blanca.
- Porción de Necrosis: Porción Negra + Porción Café.
- Porción Sana: Porción Verde.
- Porción de Clorosis: Porción Amarilla.

5. RESULTADOS.

5.0.1. PRESENTACIÓN DE RESULTADOS.

A continuación se presentan dos de las imágenes de prueba consideradas, en las cuales se puede observar un estimado de la porción de presencia de puntos blancos, necrosis, clorosis y tejido sano de la planta u hoja, ubicados en la parte superior de la imagen original y en la parte superior de la imagen que corresponde a la segmentación alcanzada, se observa el tiempo de ejecución en segundos.

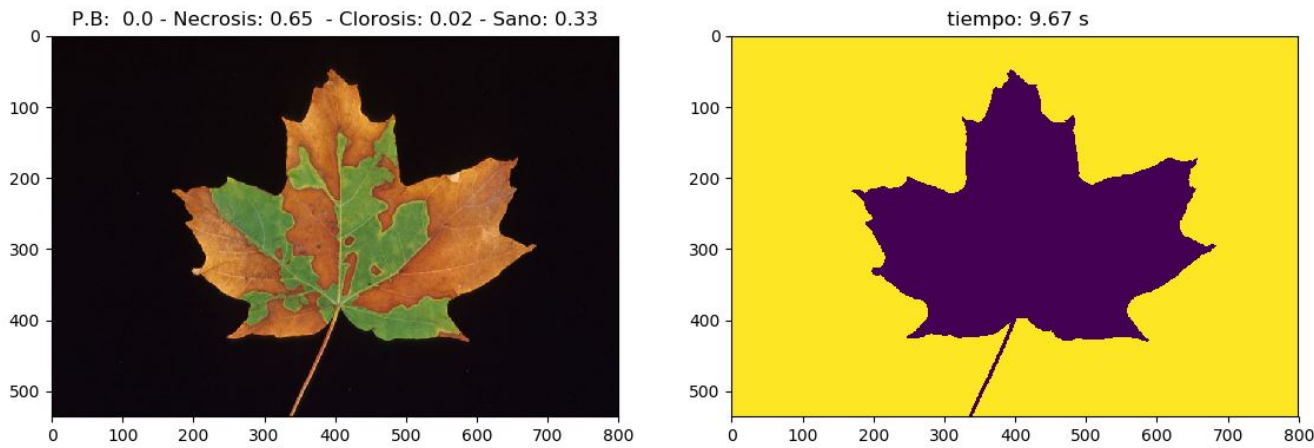


Figura 35: Resultados para la imagen de prueba 4.

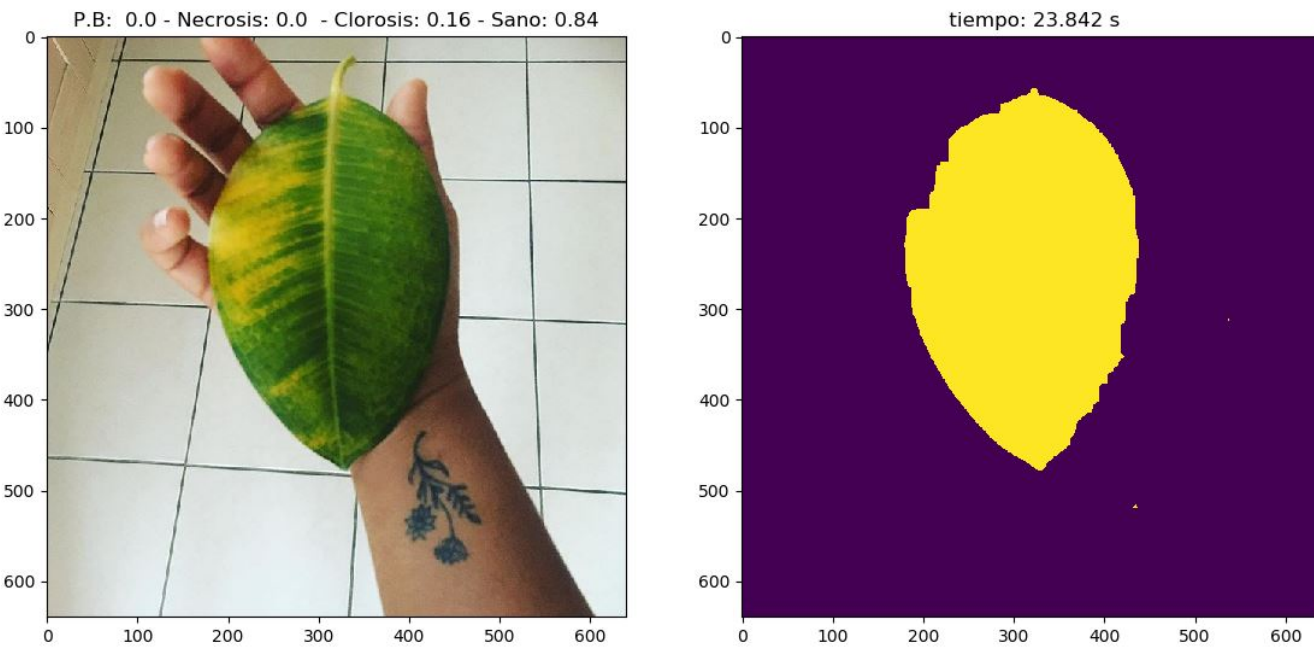


Figura 36: Resultados para la imagen de prueba 16.

A continuación se presenta una tabla en la cual se observan los tiempos de ejecución y la dimensión para cada una de las imágenes, además una tabla en la cual se presentan la eficiencia en esfuerzos computacionales para el caso de segmentación por Prominencia.

Tiempos de ejecución								
Imagen	Píxeles	Tiempo (s)	Imagen	Píxeles	Tiempo (s)	Imagen	Píxeles	Tiempo (s)
1	228000	7.916	9	432600	11.993	17	107532	6.143
2	240000	7.055	10	108778	6.282	18	1920000	96.565
3	2678346	89.28	11	96354	5.609	19	487200	23.172
4	428800	9.67	12	147356	7.797	20	324608	15.919
5	95052	3.992	13	171182	9.368	21	453604	24.64
6	249696	8.093	14	165900	8.651	22	113793	5.735
7	540000	18.163	15	229810	12.261	23	215865	11.737
8	157800	4.92	16	409600	23.842			

Cuadro 1: Tiempos de ejecución con un procesador Intel i7 @ 1.90 GHz 2.40 GHz y 8 gb de RAM.

Segmentación por Prominencia - Esfuerzo computacional			
Imagen	Píxeles considerados	Píxeles totales	%
10	12123	108778	11.14
11	15804	96354	16.40
12	25895	147356	17.57
13	28620	171182	16.72
14	22992	165900	13.86
15	50245	229810	21.86
16	71118	409600	17.36
17	28379	107532	26.39
18	288744	1920000	15.04
19	40838	487200	8.38
20	53780	324608	16.57
21	62912	453604	13.87
22	15365	113793	13.50
23	24790	215865	11.48

Cuadro 2: Reducción del esfuerzo computacional para los casos de segmentación por Prominencia.

Es importante mencionar que para alcanzar tiempos competitivos se hizo uso del poder que tiene la integración de conocimiento a priori para generar supuestos que logren reducir el esfuerzo computacional, como lo es el caso para la segmentación mediante Prominencia, en la cual al codificar la imagen mediante la red neuronal se tiene un supuesto inicial de lo que podría ser la planta y exclusivamente se consideran dichos píxeles para encontrar el segmento prominente que representa a la planta.

5.0.2. EVALUACIÓN DE RESULTADOS.

Con el objetivo de comparar el desempeño del diagnóstico automático de enfermedades en plantas con una inspección humana y debido a la nula disponibilidad de una base de datos que tuviera la cuantificación de la porción de planta cubierta con algunas de las enfermedades mencionadas con anterioridad, se realizó la encuesta para el público en general. Debido a que la encuesta se enfocó en un público que no posee experiencia en el diagnóstico de enfermedades en plantas, el instrumento fue diseñado para ser respondido haciendo uso de la opción múltiple con 4 opciones en total para cada planta, cada una de las opciones mostrando estimados de la porción de presencia de algunas de las enfermedades, para todos los casos una de las opciones corresponde al diagnóstico automático, por lo tanto el interés se encuentra en evaluar el porcentaje respuesta que tuvo cada una las opciones que corresponde al diagnóstico automático, se considera válida la evaluación por el público en general, puesto que es únicamente identificación de colores. A continuación se muestra una tabla con los respectivos porcentajes para cada una de las imágenes de prueba, donde es necesario especificar lo siguiente.

- % Respuesta: número de veces que se selecciona la opción generada automáticamente entre el número de respuestas totales.
- Ranking: la posición que tiene la opción generada automáticamente dentro de las 4 disponibles.

Imagen	% Respuesta	Ranking
1	66.70 %	1
2	46.70 %	1
3	33.30 %	2
4	66.70 %	1
5	33.30 %	2
6	60 %	1
7	46.70 %	1
8	60 %	1
9	46.70 %	1
10	46.70 %	1
11	60.00 %	1
12	46.70 %	1
13	40.00 %	1
14	33.30 %	1
15	53.30 %	1
16	26.70 %	2
17	26.70 %	2
18	60 %	1
19	40.00 %	1
20	60 %	1
21	33.30 %	2
22	46.7 %	1
23	33.3 %	2

Cuadro 3: Evaluación de resultados.

Se observa que en general el porcentaje de selección promedio fue de 46,83 % lo cual es considerablemente superior al porcentaje de selección de un 25 % que representa el hecho de tener 4 respuestas posibles, además se observa que en todos los casos la opción generada automáticamente tiene un ranking como mínimo de 2, siendo muy cercana la respuesta de la opción 1 en cada uno de los casos a la que se generó automáticamente. Además las opciones generadas automáticamente ocuparon el lugar número 1 del ranking en 17 de los 23 casos posibles, es decir en el 73,91 % de todos los casos.

Dados los análisis de resultados presentados con anterioridad y el hecho de que la respuestas de la encuesta fueron realizadas por el público general es apropiado declarar que los métodos generados presentan un desempeño que es al menos similar al de un ser humano no especializado en la tarea, algo muy interesante dada la gran variabilidad que existe en las 23 imágenes de prueba en las cuales se tienen distintas plantas y fondos, lo cual habla muy bien del poder de generalización que tienen el conjunto de supuestos simples incorporados a los modelos, logrando un desempeño similar al de un ser humano sin hacer uso de enormes cantidades de materiales para el entrenamiento de los métodos, debido al costo humano y financiero que representa su obtención.

6. CONCLUSIONES.

Se realizó una propuesta competitiva para el diagnóstico automatizado de la clorosis, necrosis y puntos blancos mediante la adecuada segmentación de la planta u hoja y la posterior estimación de la porción del tejido de la planta que presenta alguna de las enfermedades mediante la clasificación de colores haciendo uso de una red neuronal.

A diferencia del trabajo realizado en [1], el trabajo presente realiza el diagnóstico cuando las fotos son tomadas inclusive en exteriores, presentando mucho ruido lo que dificulta el uso de técnicas de segmentación tradicionales y se implementa el concepto de prominencia bayesiana, dado que el éxito de muchos de las técnicas bayesianas tienen que ver con la forma en la cual se incorpora la información a priori, se propone pasar de usar una codificación de la imagen en escala de grises o en RGB para pasar a una codificación haciendo uso de la red neuronal que está entrenada para reconocer los colores que se consideran que a priori pueden formar parte de una planta u hoja.

La propuesta de usar la red para pasar a una codificación que capture la información disponible con prioridad, es decir que codifica de una forma adecuada a aquellos pixeles que presuntamente pueden ser una planta mejoró de forma significativa el desempeño y la robustez del algoritmo propuesto, ya que en cierta manera busca asemejarse a la forma en la cual un ser humano encontraría una planta u hoja en una imagen, primero buscando los colores que considera son típicos en una planta y después decidiendo cuales de los objetos que tiene esas característica son en realidad prominentes, es decir en el contexto del trabajo actual, cuáles de ellos son planta.

Como se puede observar el algoritmo logra detectar de forma adecuada una hoja o planta en una diversidad considerable de fondos y para una diversidad considerable de tipos de planta, es decir es una propuesta que se generaliza muy bien ya que en escenarios muy distintos se observaron resultados competitivos, se considera que esa capacidad que se tiene para ofrecer buenos resultados en una diversidad de escenarios haciendo uso de un conjunto pequeño de supuestos que se transforman para ser incorporados en la información a priori que permite tener un buen punto de partida para las técnicas bayesianas es una de las principales fortalezas del trabajo presente en una era en la cual se hace uso de otro tipo de técnicas para la segmentación de imágenes como puede ser machine learning que hacen uso de enormes cantidades de imágenes de entrenamiento con los recursos que conlleva la preparación de dicho material de entrenamiento y que no siempre logran generalizarse de manera competitiva.

Por lo tanto se considera que la capacidad de generalización de la propuesta generada en el trabajo ya sea mediante el uso de segmentación por GMM cuando los fondos no tienen ruido o el uso de Prominencia Bayesiana con fondos que poseen mucho ruido es una de las fortalezas principales de la propuesta y que lo posicionan como un ejemplo interesante de lo que se puede lograr cuando se logran incorporar un número reducido de supuestos que de alguna manera u otra permitan que el algoritmo propuesto reaccione de forma adecuada ante una diversidad en el tipo de plantas y en el tipo de fondos observados.

7. TRABAJO FUTURO.

7.0.1. DETERMINACIÓN DE LA DIMENSIÓN DEL KERNEL PARA OPERACIONES DE MORFOLOGÍA.

Se considera que la siguiente línea de trabajo debe ser la determinación adecuada del tamaño del kernel usado para la operación de cerradura, se observa que si se trata de una hoja con ciertos hoyos por colores negros o blancos un kernel de un 0.04 es suficiente para cerrar esos hoyos, pero si se trata de segmentar una planta, dicha dimensión para el kernel provocaría una segmentación errónea al considerar los espacios entre las hojas como si fueran hoyos y no una parte del fondo, por lo tanto lo que se tiene que desarrollar es una metodología que distinga si la segmentación se hace a una planta o a una hoja, a continuación se ilustran un ejemplo de un caso favorable y un caso desfavorable.

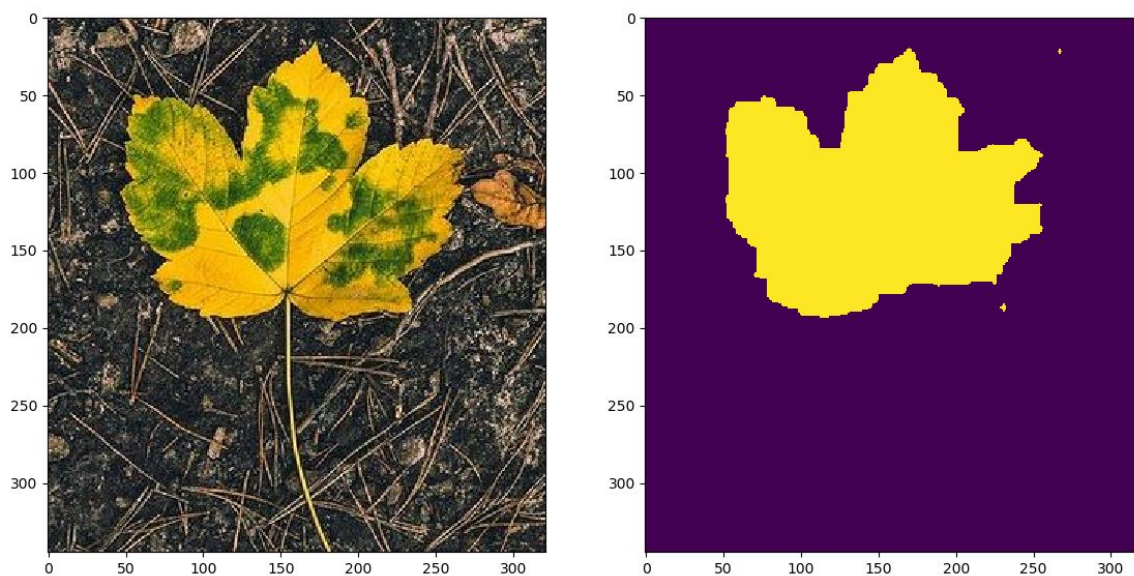


Figura 37: CASO FAVORABLE DE UN KERNEL GRANDE.

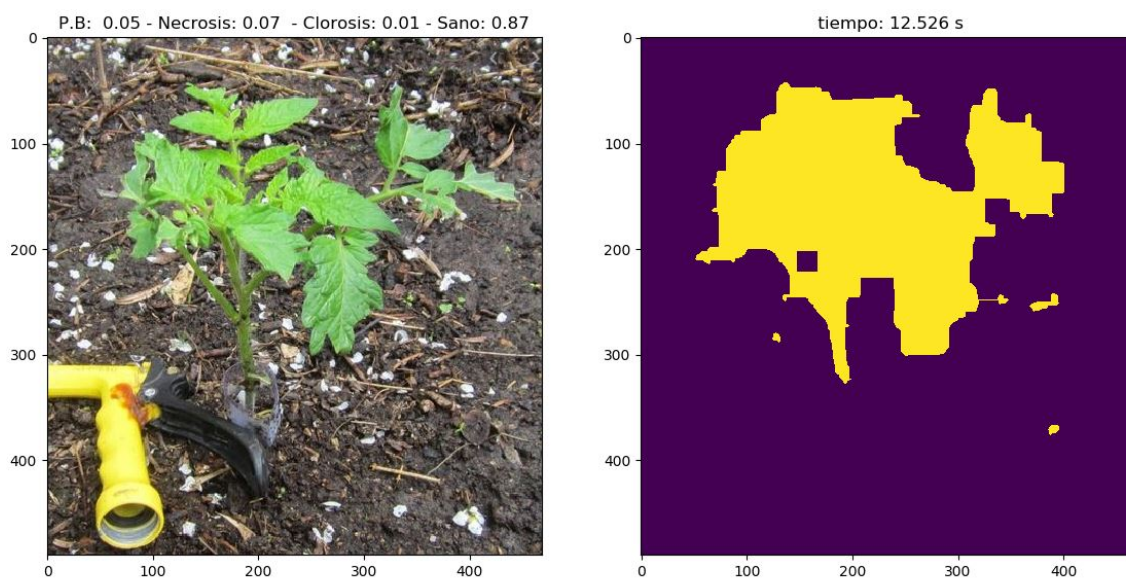


Figura 38: CASO DESFAVORABLE DE UN KERNEL GRANDE.

7.1. SELECCIÓN AUTOMÁTICA DEL MODELO DE SEGMENTACIÓN

Actualmente se determina de forma manual el tipo de modelo que se debe usar ya sea GMM o Prominencia Bayesiana, para los casos de fondos con poco ruido y con ruido considerable, respectivamente, se considera que se debe realizar un método que permita identificar de forma automatizada el tipo de método que debe de ser usado, para ello se tiene una idea preliminar de considerar la desviación estándar de un conjunto de muestras aleatorias de la imagen partiendo del supuesto de que las imágenes con mucho ruido tendrán mayor dispersión, otra línea puede ser el uso un método para obtener los contornos de una imagen y sobre de eso obtener muestras e identificar aquellas que tengan una cantidad moderada de contornos para ser clasificadas como imágenes de poco ruido y aquellas que tengan una cantidad considerable de contornos para ser clasificadas como imágenes con mucho ruido y posteriormente hacer uso de la metodología correspondiente.

8. ANEXOS.

8.1. IMÁGENES DE PRUEBA.

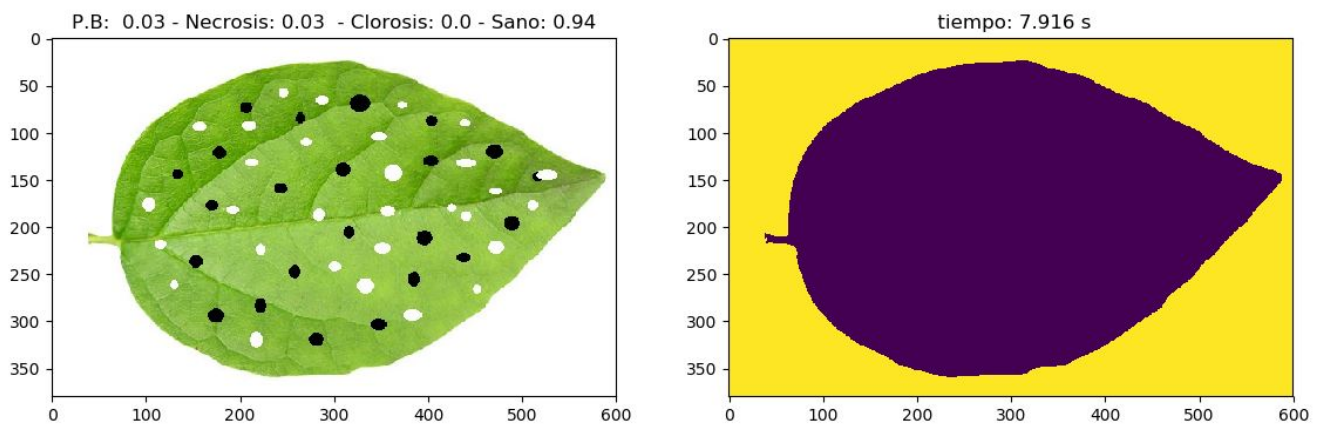


Figura 39: Resultados para la imagen de prueba 1.

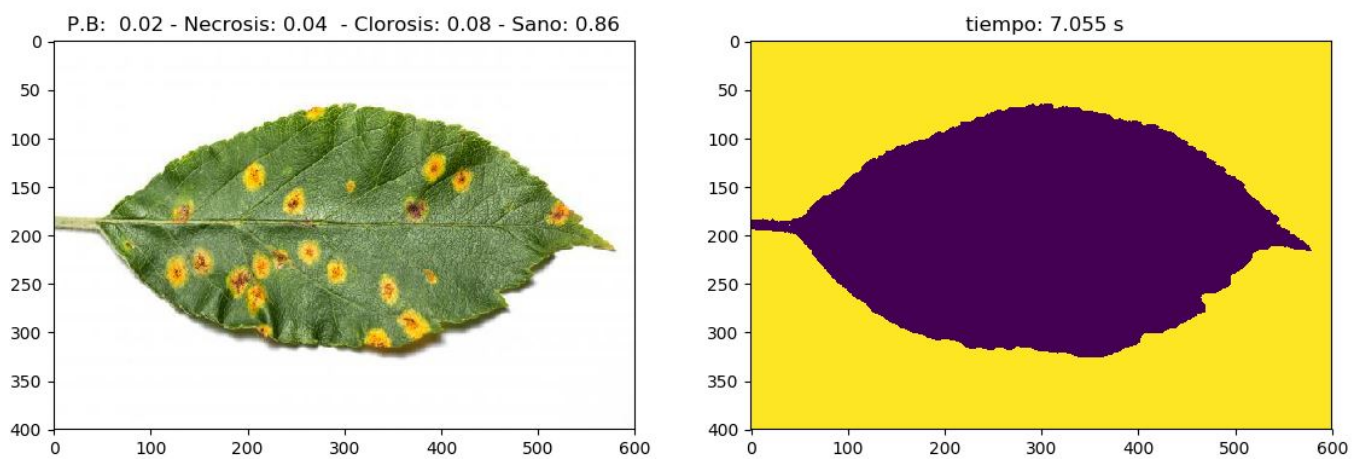


Figura 40: Resultados para la imagen de prueba 2.

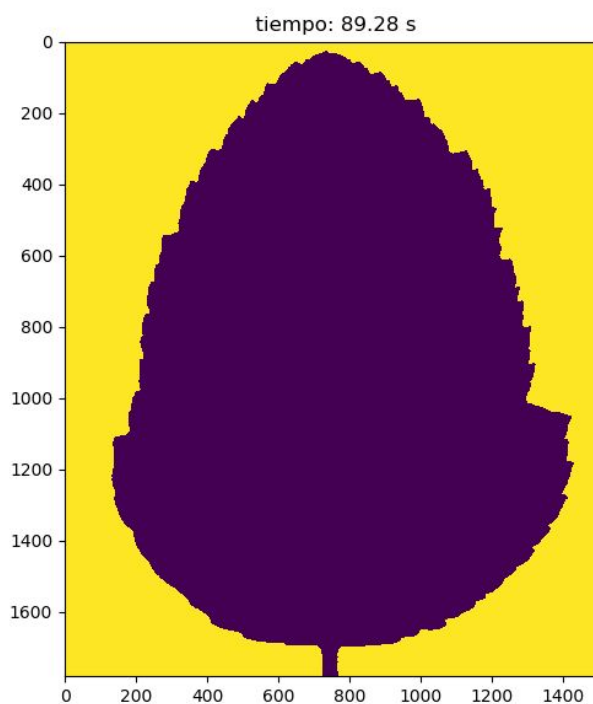
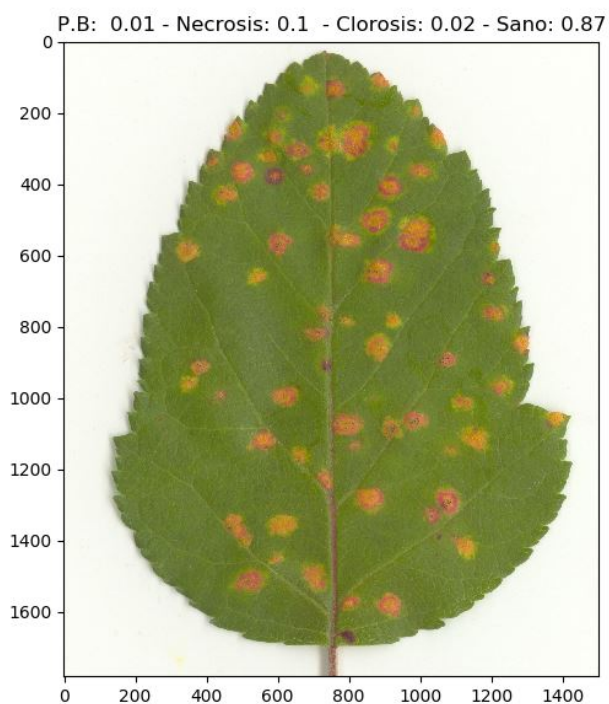


Figura 41: Resultados para la imagen de prueba 3.

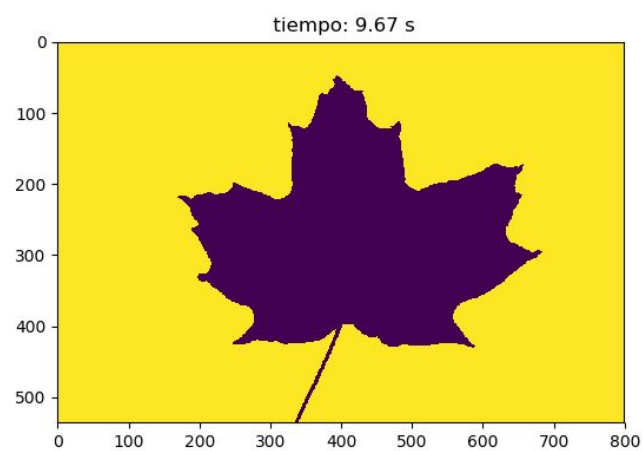
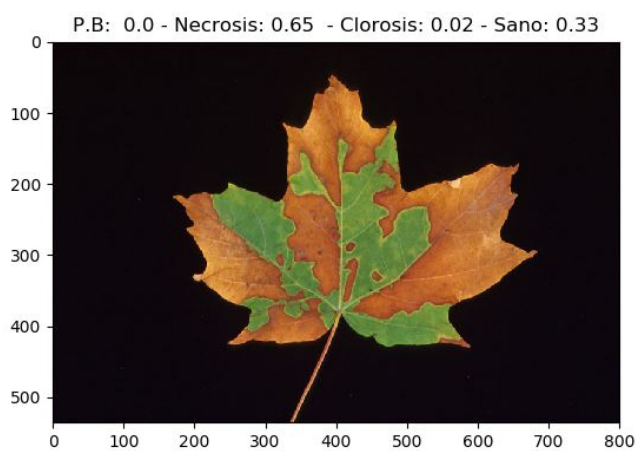


Figura 42: Resultados para la imagen de prueba 4.

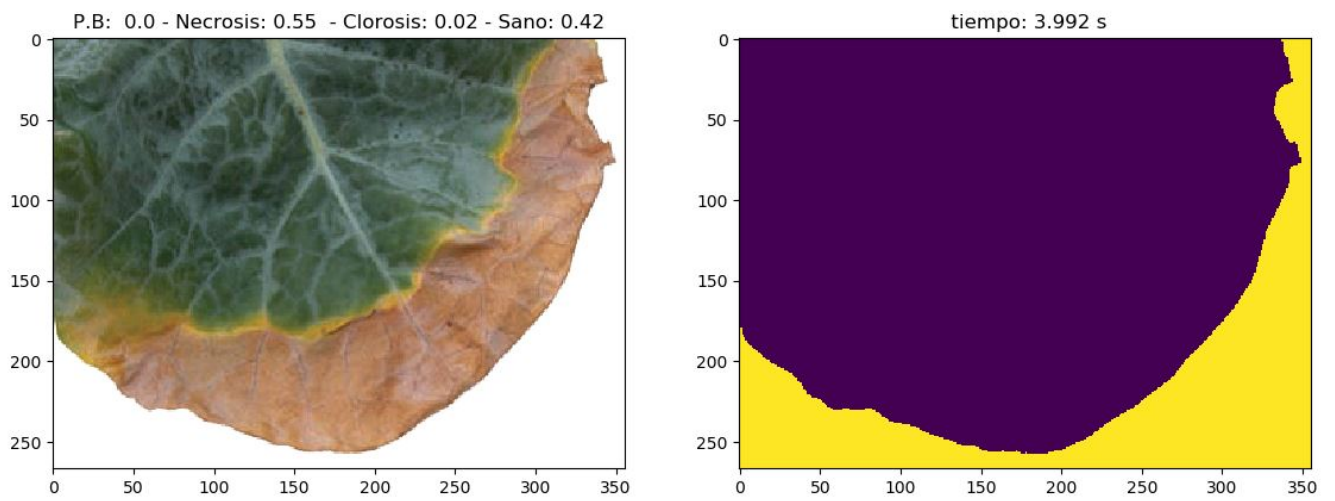


Figura 43: Resultados para la imagen de prueba 5.

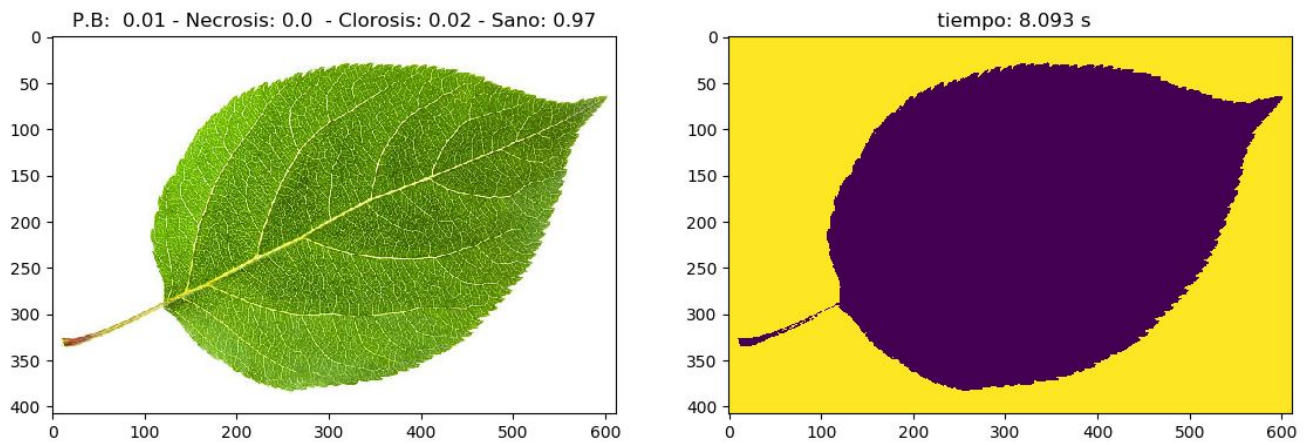


Figura 44: Resultados para la imagen de prueba 6.

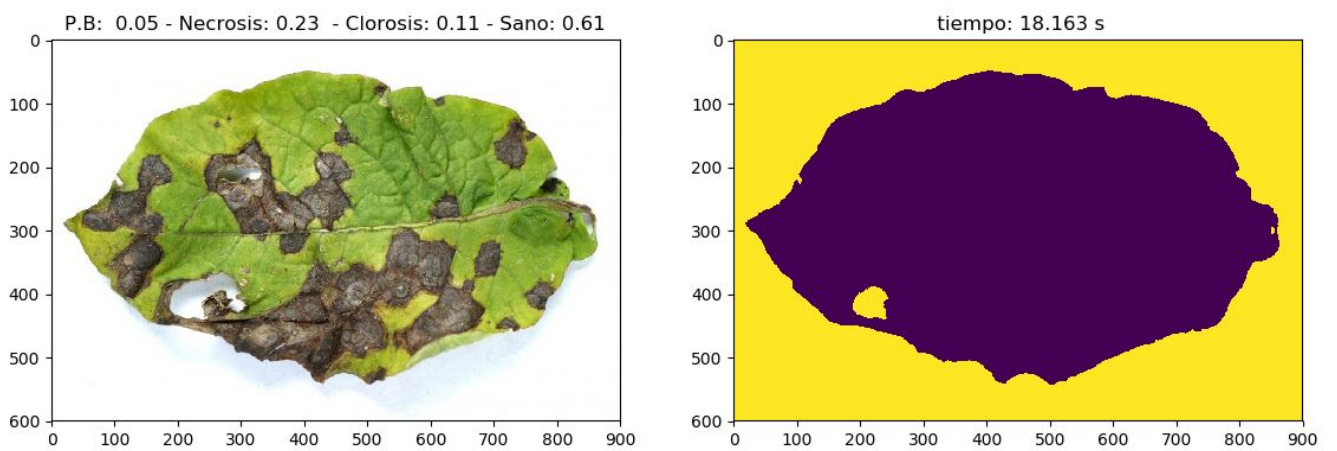


Figura 45: Resultados para la imagen de prueba 7.

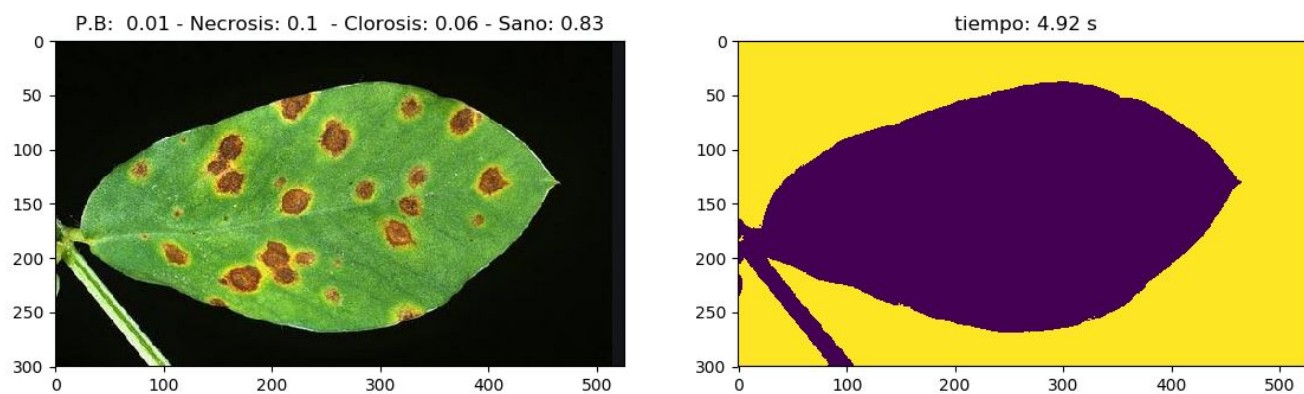


Figura 46: Resultados para la imagen de prueba 8.

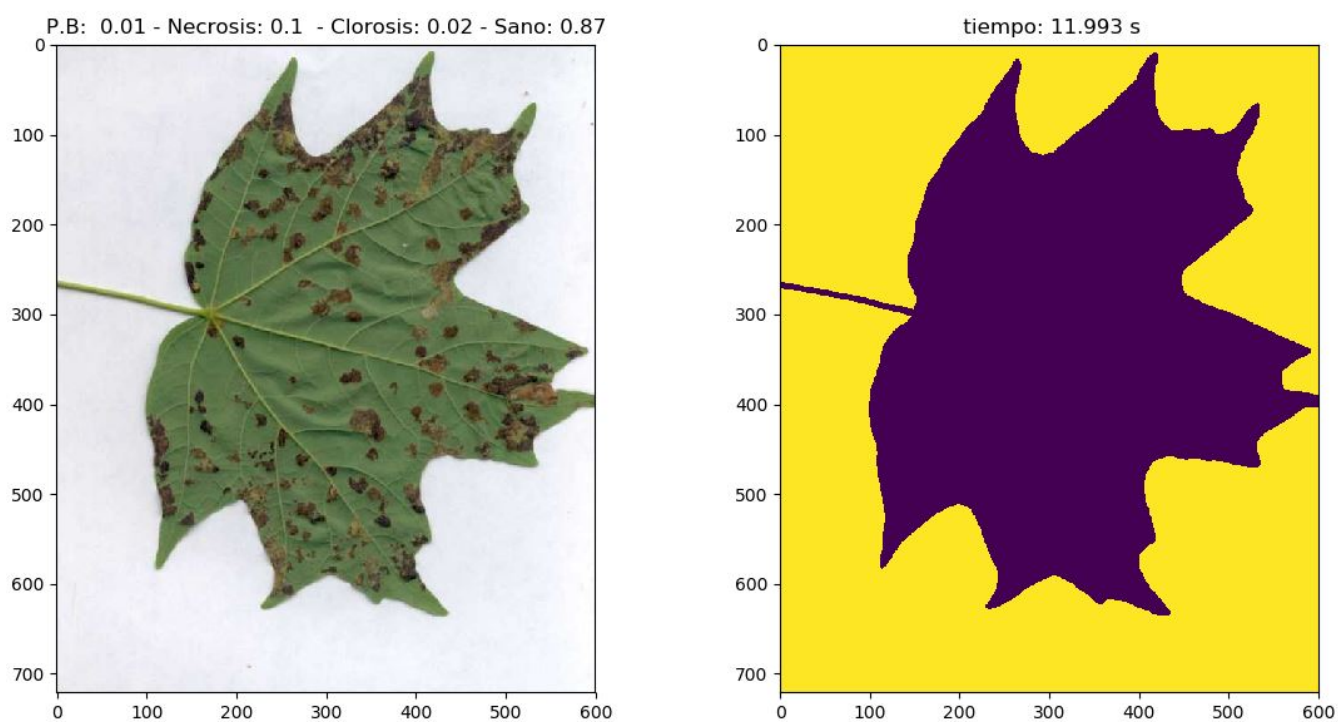


Figura 47: Resultados para la imagen de prueba 9.

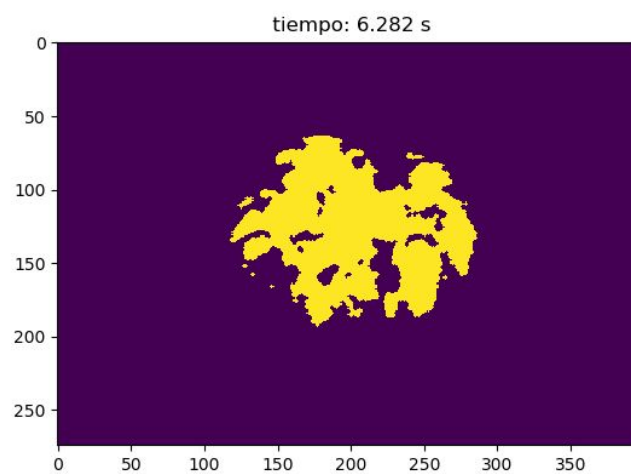
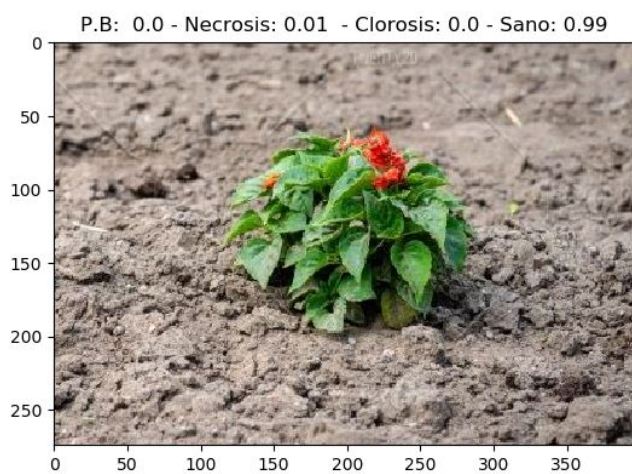


Figura 48: Resultados para la imagen de prueba 10.

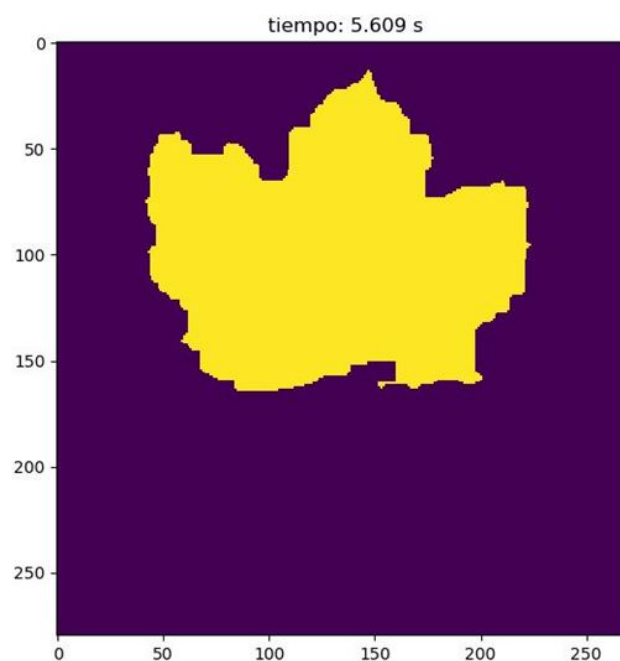
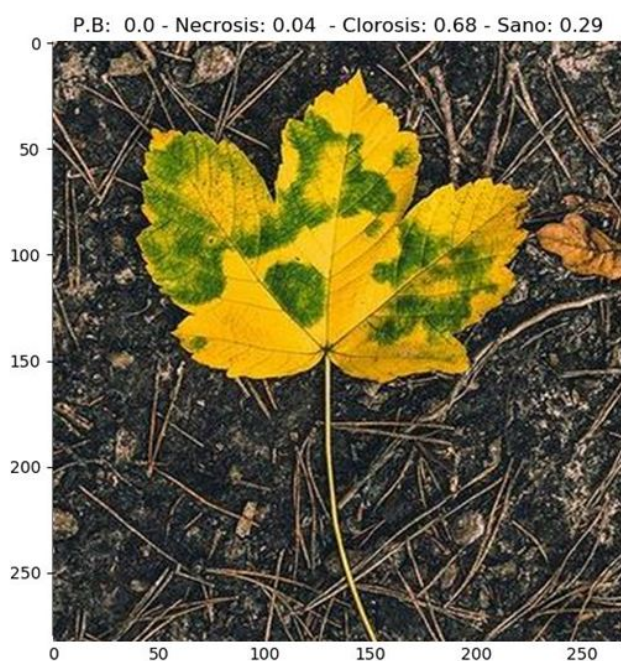


Figura 49: Resultados para la imagen de prueba 11.

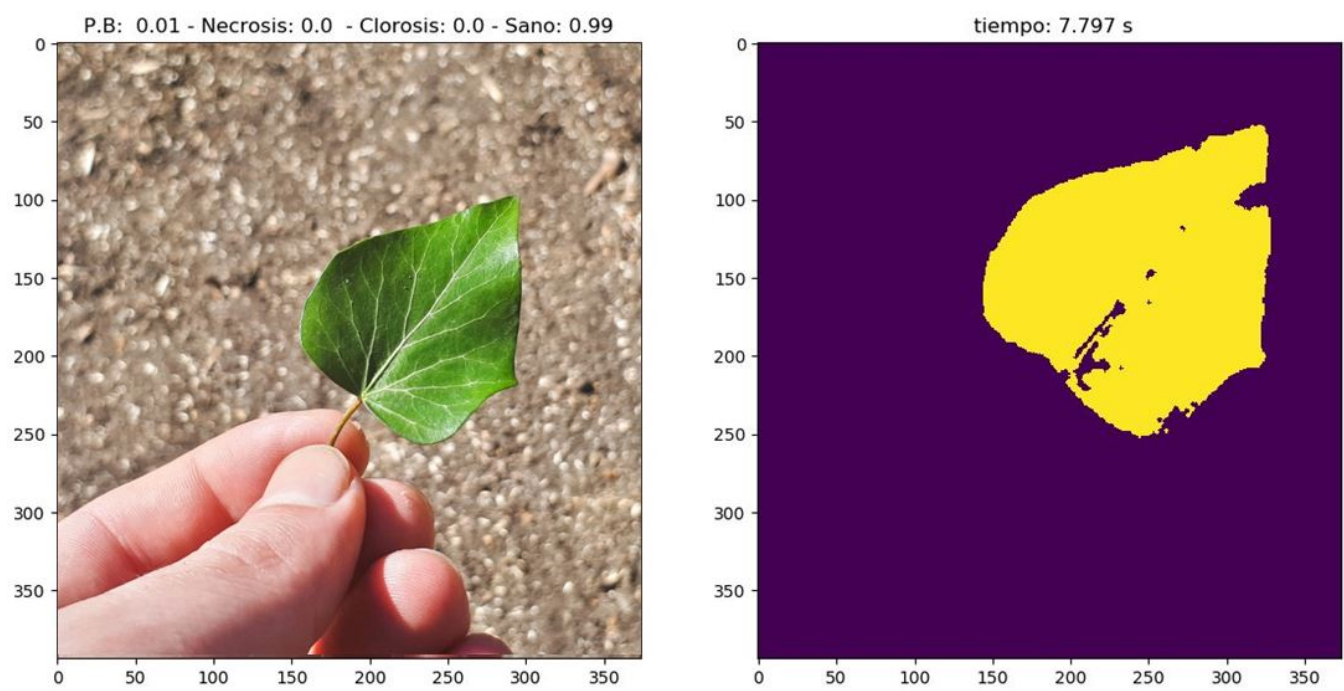


Figura 50: Resultados para la imagen de prueba 12.

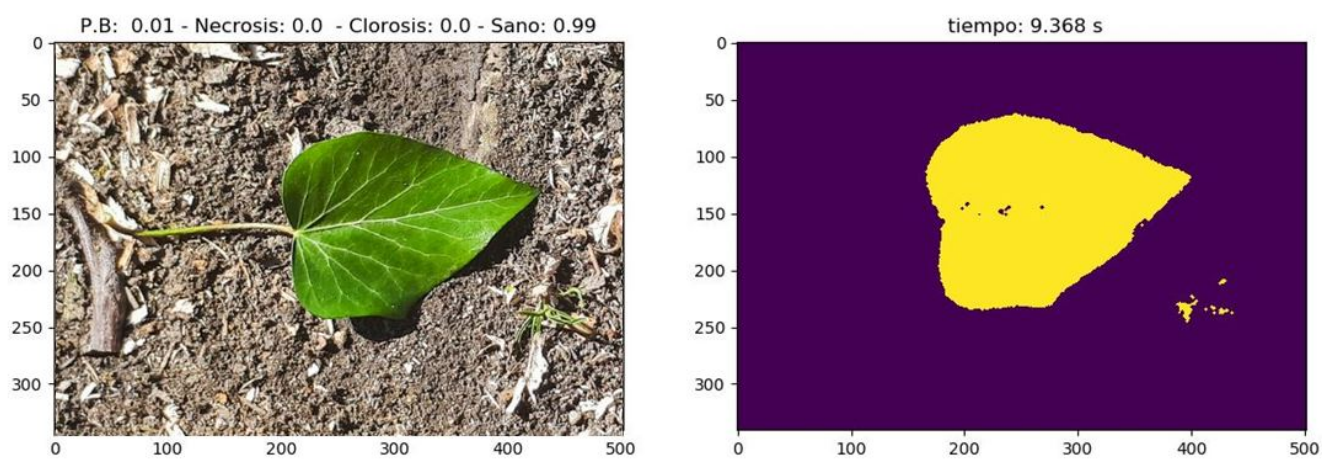


Figura 51: Resultados para la imagen de prueba 13.

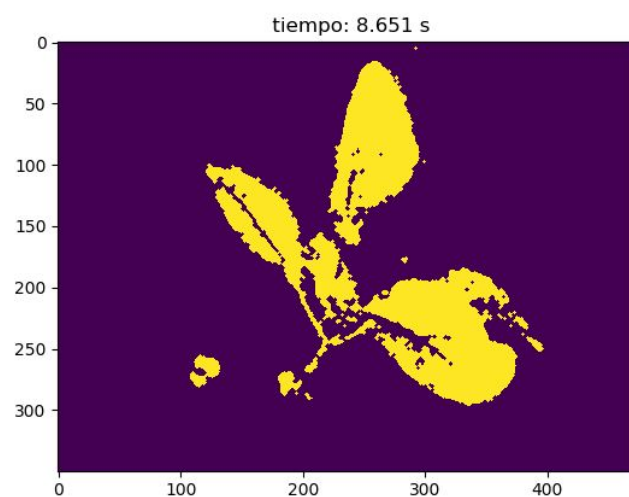
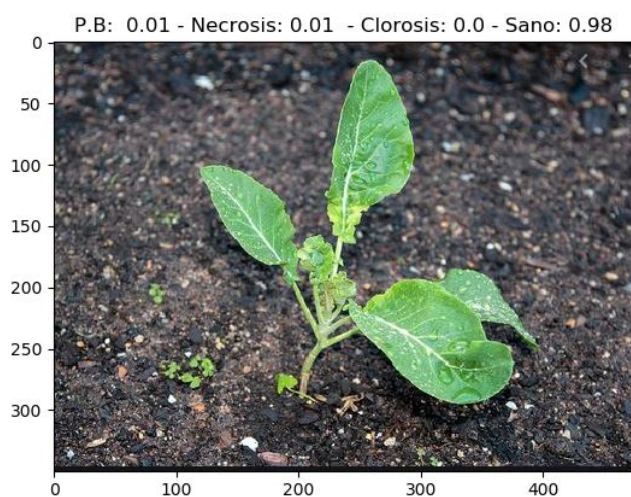


Figura 52: Resultados para la imagen de prueba 14.

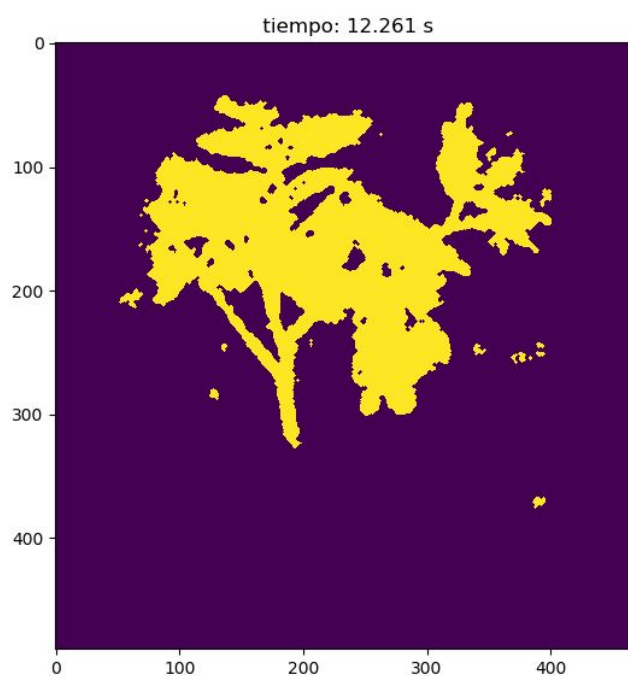
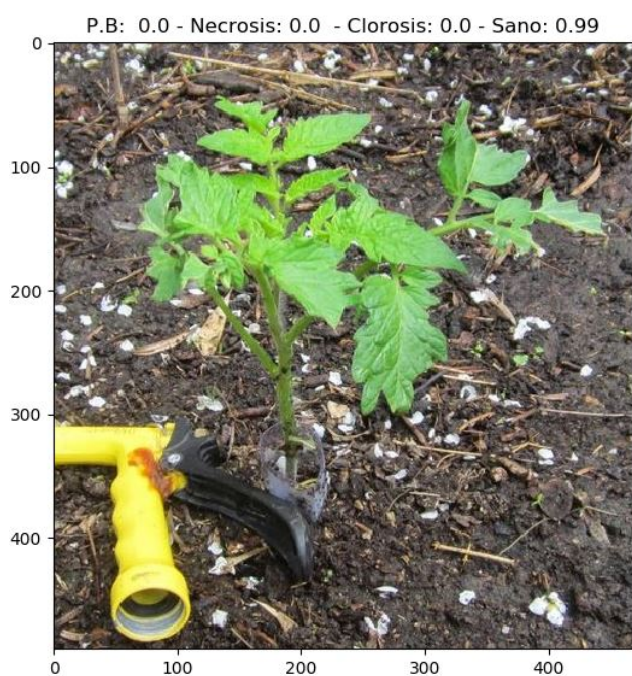


Figura 53: Resultados para la imagen de prueba 15.

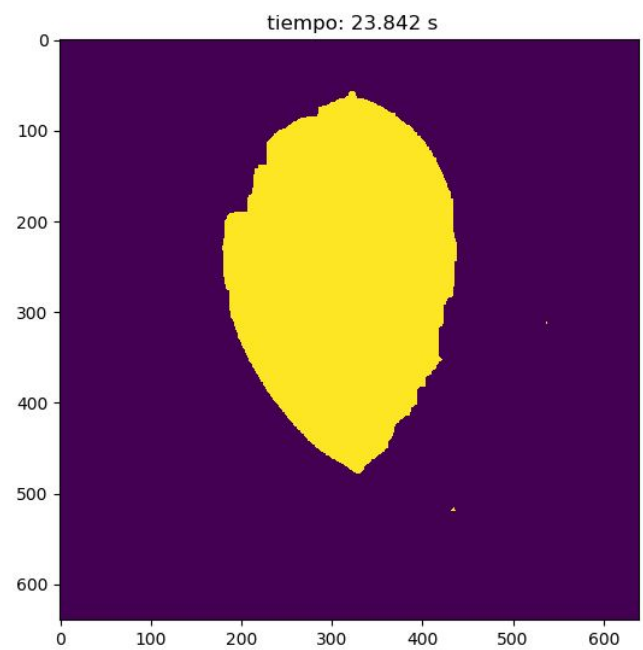
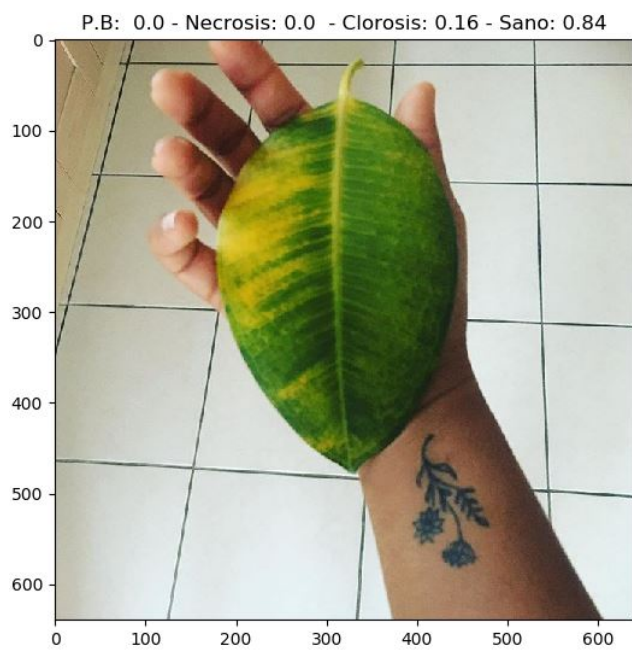


Figura 54: Resultados para la imagen de prueba 16.

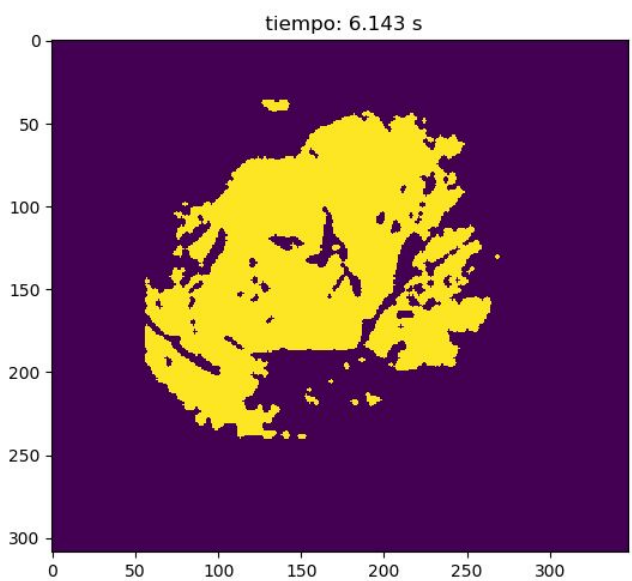
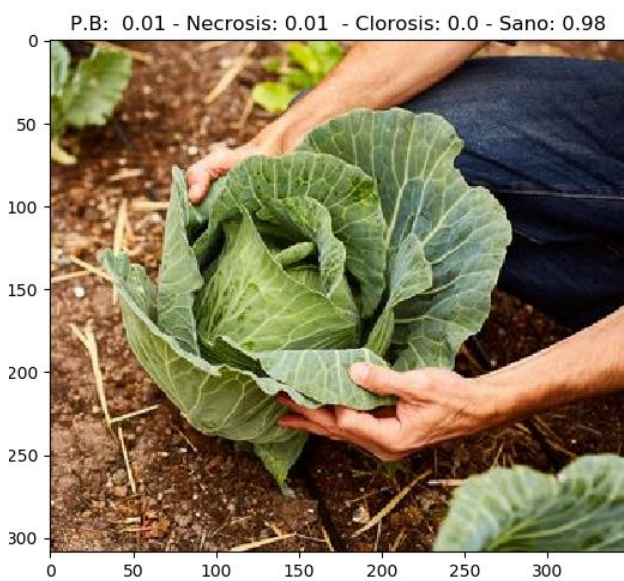


Figura 55: Resultados para la imagen de prueba 17.

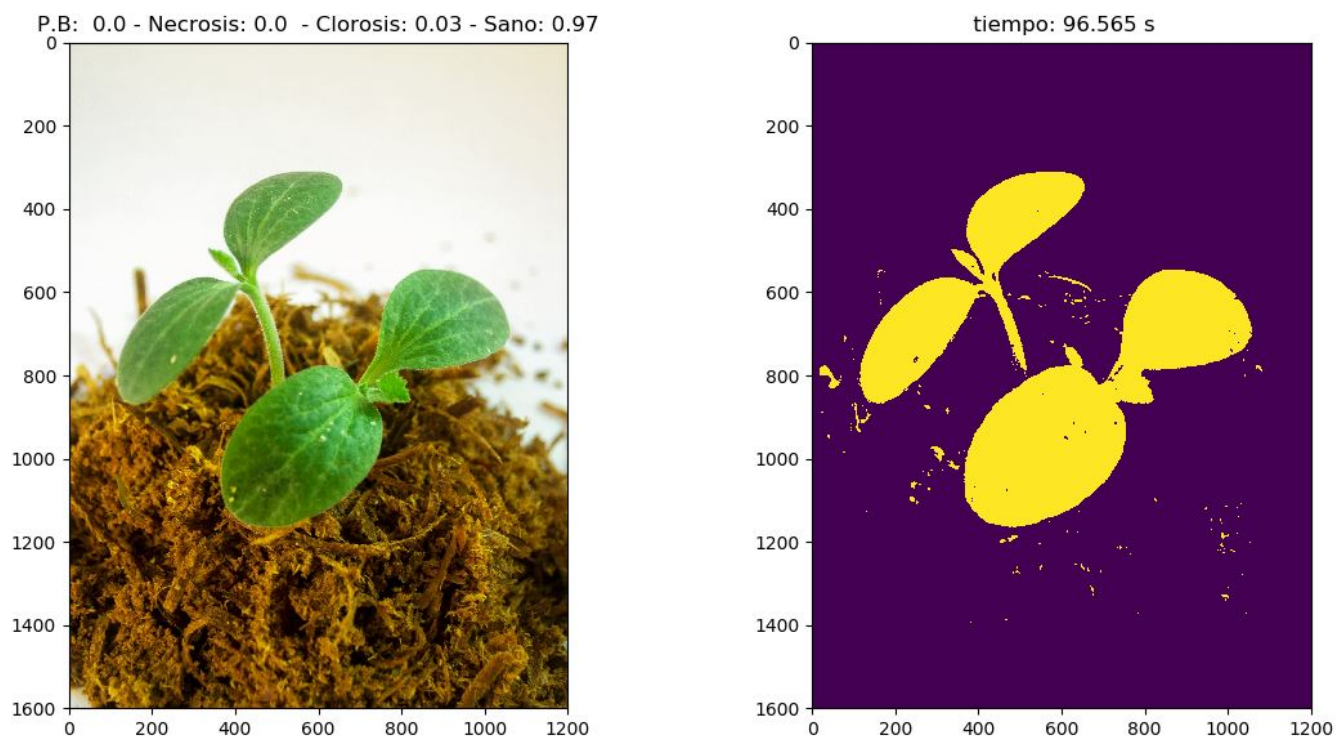


Figura 56: Resultados para la imagen de prueba 18.

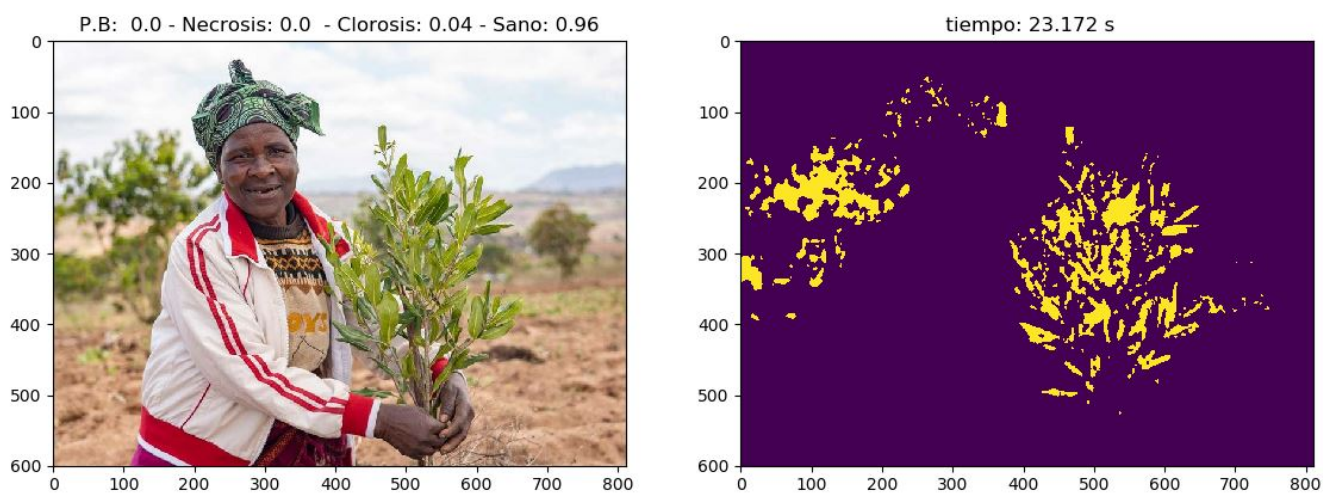


Figura 57: Resultados para la imagen de prueba 19.

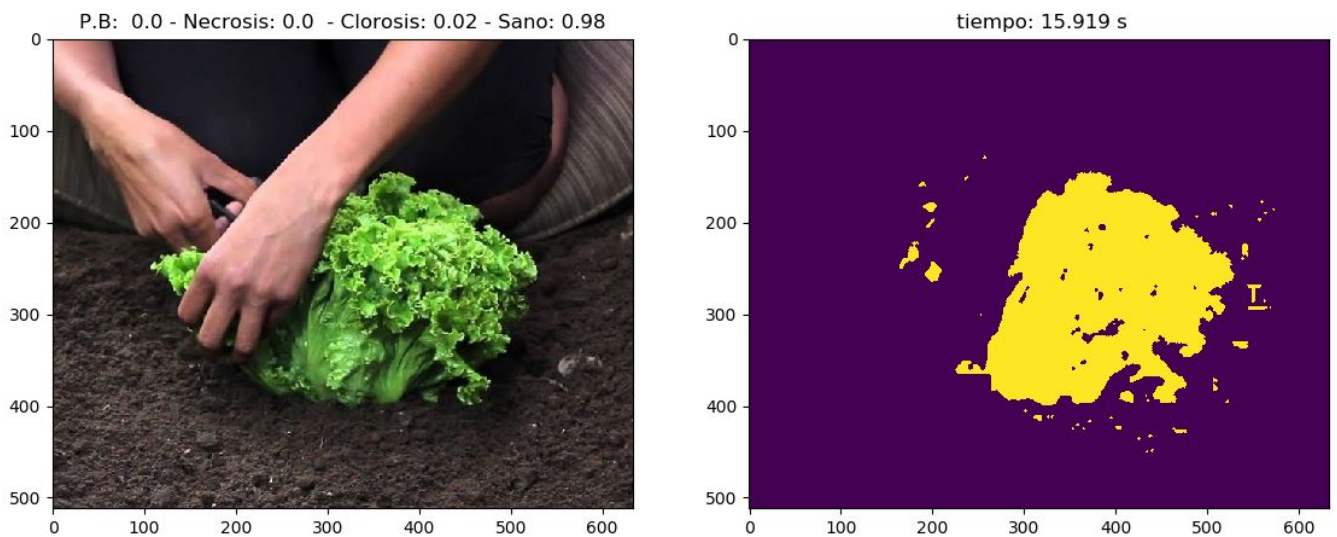


Figura 58: Resultados para la imagen de prueba 20.

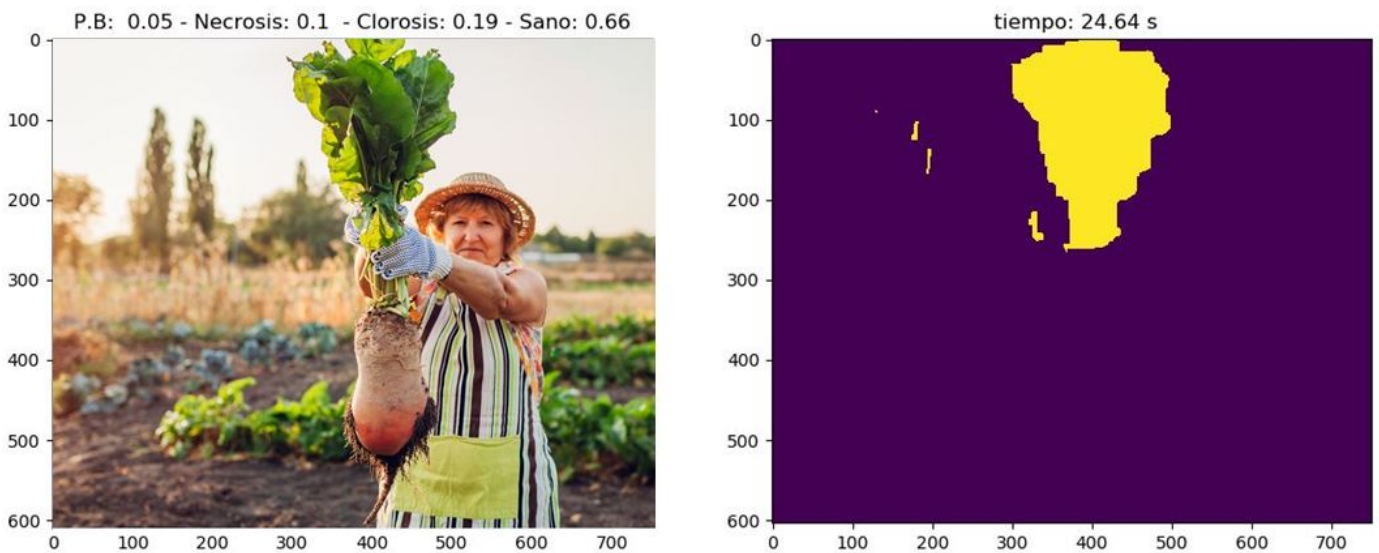


Figura 59: Resultados para la imagen de prueba 21.

8.2. IMPLEMENTACIÓN DE MODELOS GMM.

```
from PIL import Image, ImageFilter
import matplotlib.pyplot as plt
import numpy as np
import math
import cv2
from scipy import ndimage
from sklearn.mixture import GaussianMixture
import colorsys
import time
from skimage import filters, morphology

def getGaussianMixtureImage(strImage):

    imArray1 = cv2.cvtColor(np.array(Image.open(strImage)), cv2.COLOR_BGR2GRAY)
    imArray2 = ndimage.gaussian_filter(np.array(imArray1), sigma=4)
```


P.B: 0.01 - Necrosis: 0.01 - Clorosis: 0.02 - Sano: 0.96



tiempo: 5.735 s

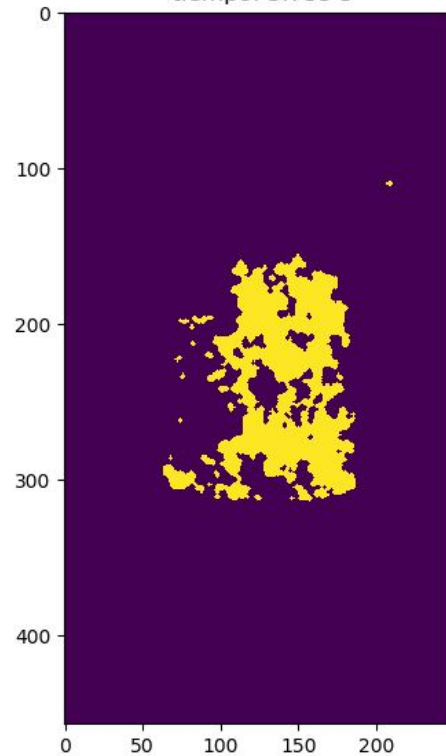


Figura 60: Resultados para la imagen de prueba 22.

P.B: 0.0 - Necrosis: 0.0 - Clorosis: 0.0 - Sano: 1.0



tiempo: 11.737 s

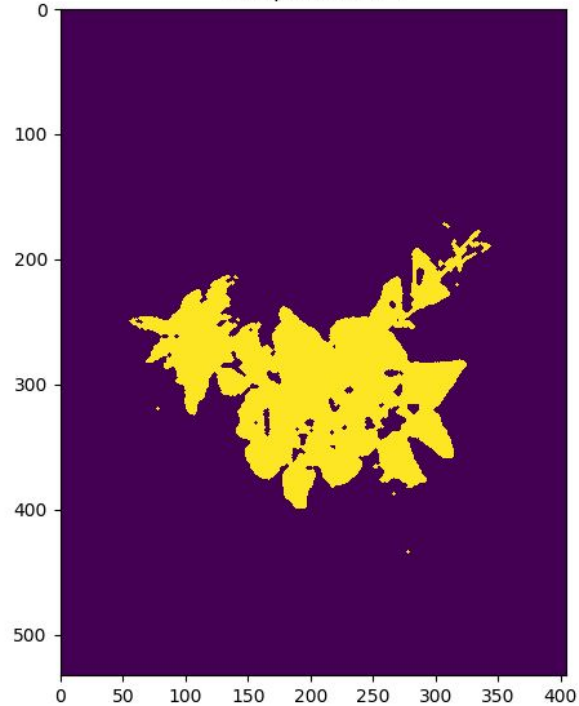


Figura 61: Resultados para la imagen de prueba 23.

```
imArray3 = 0.5*imArray1 + 0.5*imArray2
img = imArray3
```

```

classif = GaussianMixture(n_components=2)
classif.fit(img.reshape((img.size , 1)))
threshold = np.mean(classif.means_)
binary_img = img > threshold

```

```

return (binary_img)

```

```

def getBackGround(imageArray , center1 , center2 ):

```

```

    rows = imageArray.shape[0]
    cols = imageArray.shape[1]
    vecCenter1 = []
    vecCenter2 = []
    dist1 = 0
    dist2 = 0

```

```

    for k in range(200):
        i = np.random.randint(0,rows,1,int)
        j = np.random.randint(0,cols,1,int)
        pixel = imageArray[i,j]
        coordinates = []
        coordinates.append(i)
        coordinates.append(j)
        if (pixel == center1):
            vecCenter1.append(coordinates.copy())
        else:
            vecCenter2.append(coordinates.copy())

```

```

    for i in range(len(vecCenter1)- 1):
        x = vecCenter1[i]
        y = vecCenter1[i+1]
        dist = math.sqrt(math.pow(x[0]-y[0],2)+math.pow(x[1]-y[1],2))
        dist1 = dist1 + dist

```

```

    for i in range(len(vecCenter2)- 1):
        x = vecCenter2[i]
        y = vecCenter2[i+1]
        dist = math.sqrt(math.pow(x[0]-y[0],2)+math.pow(x[1]-y[1],2))
        dist2 = dist2 + dist

```

```

    dist1 = dist1/len(vecCenter1)
    dist2 = dist2/len(vecCenter2)

```

```

    if (dist1 > dist2):
        backGround = center1
    else:
        backGround = center2

```

```

    return (backGround)

```

```

def binaryToGrayScale(imageArray1 , backGround , strImage ):

```

```

    imageArray2 = np.array(Image.open(strImage).convert('L'))
    rows = imageArray1.shape[0]
    cols = imageArray1.shape[1]

```

```

for i in range(rows):
    for j in range(cols):
        pixel = imageArray1[i,j]
        if (pixel == backGround):
            imageArray2[i,j] = 255
        else:
            imageArray2[i,j] = 0

imageArray3 = ndimage.gaussian_filter(imageArray2,20)

return (imageArray1,imageArray2)

def binaryToGrayScale2(imageArray1,backGround,strImage):

    imageArray2 = imageArray1.copy()
    rows = imageArray1.shape[0]
    cols = imageArray1.shape[1]

    for i in range(rows):
        for j in range(cols):
            pixel = imageArray1[i,j]
            if (pixel == backGround):
                imageArray2[i,j] = False
            else:
                imageArray2[i,j] = True

    return (imageArray2)

def analyzeImage(imageArray2,strImage):

    imageArray3 = np.array(Image.open(strImage))
    rows = imageArray2.shape[0]
    cols = imageArray2.shape[1]
    area = 0
    colorCount = np.zeros(5)
    colorID = ["white spots","necrosis","chlorosis","healthy","unidentified"]
    index = 0

    for i in range(rows):
        for j in range(cols):
            pixel = imageArray2[i,j]
            if (pixel == True):
                area = area + 1
                rgb = imageArray3[i,j]
                index = getColorNN2(rgb)
                colorCount[index] = colorCount[index] + 1

    colorCount = colorCount * (1/area)
    end = time.time()
    plt.figure()
    plt.subplot(121)
    plt.title("P.B: " + str(round(colorCount[0],2)) + " - Necrosis: " + str(round(colorCo
- Chlorosis: " + str(round(colorCount[2],2)) + " - Sano: " + str(round(colorCount[3],2)) )
    plt.imshow(imageArray3)
    plt.subplot(122)
    plt.title(" tiempo: " + str(round(end - start,3)) + " s")
    plt.imshow(imageArray2)

```

```

plt.show()

return(colorID , colorCount)

def getColorNN2(rgb):

    color = 0

    r = rgb[0]/255
    g = rgb[1]/255
    b = rgb[2]/255

    h,s,v = colorsys.rgb_to_hsv(r,g,b)

    o11 = 1/(1 + math.exp(-1*(-3.1580227 - 2.1237742*h + 0.7001064*s + 2.8366065*v)))
    o12 = 1/(1 + math.exp(-1*(-52.0145977 + 234.8668015*h + 0.6990048*s + 30.0214358*v)))
    o13 = 1/(1 + math.exp(-1*(-1.253056 + 13.992049*h + 43.986046*s - 18.551682*v)))
    o14 = 1/(1 + math.exp(-1*(-4.905778 + 59.807990*h - 4.249384*s - 10.403728*v)))
    o21 = 1/(1 + math.exp(-1*(0.6325434 - 1.2561720*o11 - 3.0853635*o12 + 2.9881184*o13 -
    o22 = 1/(1 + math.exp(-1*(-0.3035949 - 0.3958577*o11 + 0.7328440*o12 - 5.9774242*o13 -
    o23 = 1/(1 + math.exp(-1*(-19.876105 + 358.356456*o11 + 8.919032*o12 - 4.203202*o13 -

    output = round(1.003647 + 2.079147*o21 - 8.061846*o22 + 1.005889*o23,0)

    if (output <= 0):
        color = 0
    elif(output == 1):
        color = 1
    elif(output == 2):
        color = 3
    elif(output == 3):
        color = 2
    elif(output == 4):
        color = 1
    else:
        color = 4

    return (color)

def codifyImage(binaryImage , background , rows , cols):

    binaryImage2 = binaryImage.copy()
    if background == True:
        for i in range(rows):
            for j in range(cols):
                pixel = binaryImage[i,j]
                if pixel == background:
                    binaryImage2[i,j] = False
                else:
                    binaryImage2[i,j] = True
    return binaryImage2.copy()

strImage = "h1.jpg"
start = time.time()

```

```

imageArray3 = np.array(Image.open(strImage))
rows = imageArray3.shape[0]
cols = imageArray3.shape[1]
binaryImage = (getGaussianMixtureImage(strImage))
backGround = getBackGround(binaryImage, True, False)
binaryImage2 = codifyImage(binaryImage, backGround, rows, cols)
#Para los casos en los que se tiene una hoja o planta sin espacios entre si.
#opening = morphology.opening(binaryImage)
closing = morphology.closing(binaryImage2, morphology.square(int(0.04*max(rows, cols))))
#Para los casos en los que se tiene una planta con espacios y no una hoja.
#closing = morphology.closing(binaryImage)
#backGround = getBackGround(closing, True, False)
print(analyzeImage(closing, strImage))

```

8.3. IMPLEMENTACIÓN DE PROMINENCIA BAYESIANA.

```

import numpy as np
from PIL import Image
from skimage import filters, morphology
import matplotlib.pyplot as plt
import math
import colorsys
import random
import time

def maps(strimage):
    start = time.time()
    array1 = np.array(Image.open(strimage))
    rows = array1.shape[0]
    cols = array1.shape[1]
    array2 = np.zeros([rows, cols])
    pt = 0
    xc = 0
    yc = 0
    for i in range(rows):
        for j in range(cols):
            pixel = getColorNN2(array1[i, j])
            array2[i, j] = pixel
            if pixel == 1:
                pt = pt + 1
                xc = xc + i
                yc = yc + j

    xc = xc/pt
    yc = yc/pt
    pt = pt/(rows*cols)
    s = np.zeros([rows, cols]) + 0.15
    sm = s.copy()
    side = int((0.65*max(rows, cols))/2)
    centers = []
    step = int(0.15*max(rows, cols))
    for i in range(3):
        for j in range(3):
            centers.append([max(int(xc-i*step), 0), max(int(yc-j*step), 0)])
            if j > 0:
                centers.append([min(int(xc+i*step), rows-1), min(int(yc+j*step), cols-1)])

    for k in range(15):

```



```

cr,cc = centers[k]

lr = max(cr-side, 0)
hr = min(cr+side, rows-1)
lc = max(cc-side, 0)
hc = min(cc+side, cols-1)
lrk = int(1.075*lr)
hrk = int(0.925*hr)
lck = int(1.075*lc)
hck = int(0.925*hc)

tk = 0
tw = 0
pk = 0
pw = 0

for i in range(lr, hr):
    for j in range(lc, hc):
        color = array2[i, j]
        if lrk < i < hrk and lck < j < hck:
            if color == 1:
                pk = pk + 1
                tk = tk + 1
        else:
            if color == 1:
                pw = pw + 1
                tw = tw + 1

pk = pk/tk
pw = pw/tw

for i in range(lrk, hrk):
    for j in range(lck, hck):

        color = array2[i, j]
        if color == 1:
            cs = s[i, j]
            ccs = 1-cs
            pkcs = pk*cs
            ns = 0.8*pkcs/(pkcs + pw*ccs) + 0.2*pkcs/(pkcs + pt*ccs)
            s[i,j] = ns
            if ns > sm[i,j]:
                sm[i,j] = ns

tre = filters.threshold_otsu(sm)
binary = sm > tre
opening = morphology.opening(binary)
#Para los casos en los que se tiene una hoja o planta sin espacios enete si.
closing = morphology.closing(opening, morphology.square(int(0.04*max(rows, cols))))
#Para los casos en los que se tiene una planta con espacios y no una hoja.
#closing = morphology.closing(opening)

area = 0
colorCount = np.zeros(5)
for i in range(rows):
    for j in range(cols):
        if (closing[i,j] == True):
            area = area + 1
            rgb= array1[i,j]

```

```

        color = getColorNN(rgb)
        colorCount[color] = colorCount[color] + 1

colorCount = colorCount * (1/area)

finish = time.time()

plt.figure()
plt.subplot(121)
plt.title("P.B: " + str(round(colorCount[0],2))
+ " - Necrosis: " + str(round(colorCount[1],2)) + " - Clorosis: "
+ str(round(colorCount[3],2)) + " - Sano: " + str(round(colorCount[2],2)) )
plt.imshow(array1)
plt.subplot(122)
plt.title(" tiempo: " + str(round(finish - start,3)) + " s")
plt.imshow(closing)
plt.show()

return ()

def getColorNN2(rgb):

    r = rgb[0]/255
    g = rgb[1]/255
    b = rgb[2]/255

    h,s,v = colorsys.rgb_to_hsv(r,g,b)

    o11 = 1/(1 + math.exp(-1*(-3.1580227 - 2.1237742*h
+ 0.7001064*s + 2.8366065*v)))
    o12 = 1/(1 + math.exp(-1*(-52.0145977 + 234.8668015*h
+ 0.6990048*s + 30.0214358*v)))
    o13 = 1/(1 + math.exp(-1*(-1.253056 + 13.992049*h
+ 43.986046*s - 18.551682*v)))
    o14 = 1/(1 + math.exp(-1*(-4.905778 + 59.807990*h
- 4.249384*s - 10.403728*v)))
    o21 = 1/(1 + math.exp(-1*(0.6325434 - 1.2561720*o11
- 3.0853635*o12 + 2.9881184*o13 - 79.1063440*o14)))
    o22 = 1/(1 + math.exp(-1*(-0.3035949 - 0.3958577*o11
+ 0.7328440*o12 - 5.9774242*o13 - 1.4438876*o14)))
    o23 = 1/(1 + math.exp(-1*(-19.876105 + 358.356456*o11
+ 8.919032*o12 - 4.203202*o13 - 3.558226*o14)))

    output = (1.003647 + 2.079147*o21 - 8.061846*o22 + 1.005889*o23)
    #Usualmente se usa 0.15 pero generaba falsos positivos, se observo
    que 0.04 genera mejores resultados.

    color = 0
    if abs(output - round(output,0)) < 0.1:
        output = round(output,0)
        if output <= 0:
            color = 0
        elif output == 1:
            color = 0
        elif output == 2:
            if g > r and g > b:
                color = 1
        elif output == 3:
            color = 1
        elif output == 4:

```

```

        color = 0
    else:
        color = 0
else:
    color = 0

return color

```

```
def getColorNN(rgb):
```

```

    color = 0

    r = rgb[0]/255
    g = rgb[1]/255
    b = rgb[2]/255

    h,s,v = colorsys.rgb_to_hsv(r,g,b)

    o11 = 1/(1 + math.exp(-1*(-3.1580227 - 2.1237742*h
+ 0.7001064*s + 2.8366065*v)))
    o12 = 1/(1 + math.exp(-1*(-52.0145977 + 234.8668015*h
+ 0.6990048*s + 30.0214358*v)))
    o13 = 1/(1 + math.exp(-1*(-1.253056 + 13.992049*h
+ 43.986046*s - 18.551682*v)))
    o14 = 1/(1 + math.exp(-1*(-4.905778 + 59.807990*h
- 4.249384*s - 10.403728*v)))
    o21 = 1/(1 + math.exp(-1*(0.6325434 - 1.2561720*o11
- 3.0853635*o12 + 2.9881184*o13 - 79.1063440*o14)))
    o22 = 1/(1 + math.exp(-1*(-0.3035949 - 0.3958577*o11
+ 0.7328440*o12 - 5.9774242*o13 - 1.4438876*o14)))
    o23 = 1/(1 + math.exp(-1*(-19.876105 + 358.356456*o11
+ 8.919032*o12 - 4.203202*o13 - 3.558226*o14)))

    output = round(1.003647 + 2.079147*o21 - 8.061846*o22 + 1.005889*o23,0)

    if (output <= 0):
        color = 0
    elif(output == 1):
        color = 1
    elif(output == 2):
        color = 2
    elif(output == 3):
        color = 3
    elif(output == 4):
        color = 1
    else:
        color = 4

    return (color)

```

```
maps(" farmer1.jpg")
```

9. BIBLIOGRAFÍA.

- [1] L.M. Contreras Medina*, I. Torres Pacheco, "Sensor Inteligente para la cuantificación de síntomas en plantas enfermas", Universidad Autónoma de Querétaro, 2012
- [2] A.T Figueredo Mário, "Bayesian Segmentation Using Gaussian Field Priors", Springer Verlag, 2005.
- [3] D. Martin, C. Fowlkes, and J. Malik. "Learning to detect natural image boundaries using local brightness, color and texture cues." *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 26, pp. 530-549, 2004.
- [4] E. Sharon, A. Brandt, R. Basri. "Segmentation and boundary detection using multiscale intensity measurements." *Proc. IEEE CVPR*, vol. I, pp. 469-476, Kauai, Hawaii, 2001.
- [5] A. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters." *Pattern Recognition*, vol. 24, pp. 1167-1186, 1991.
- [6] M. Unser, "Texture classification and segmentation using wavelet frames." *IEEE Trans. Image Proc.*, vol. 4, pp. 1549-1560, 1995.
- [7] R. Haralick, K. Shanmugan, and I. Dinstein, "Textural features for image classification." *IEEE Trans. Syst., Man, and Cybernetics*, vol. 8, pp. 610-621, 1973.
- [8] G. Cross and A. Jain. "Markov random field texture models." *IEEE Trans. Patt. Anal. and Mach. Intell.*, vol. 5, pp. 25-39, 1983.
- [9] H. Derin and H. Elliot. "Modelling and segmentation of noisy and textured images in Gibbsian random fields." *IEEE Trans. Patt. Anal. and Mach. Intell.*, vol. 9, pp. 39-55, 1987.
- [10] T. Hofmann, J. Puzicha, and J. Buhmann. "Unsupervised texture segmentation in a deterministic annealing framework." *IEEE Trans. Patt. Anal. and Mach. Intell.*, vol. 20, pp. 803-818, 1998.
- [11] T. Randen and J. Husoy. "Filtering for texture classification: a comparative study." *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 21, pp. 291-310, 1999.
- [12] J. Shi and J. Malik, "Normalized cuts and image segmentation." *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 22, pp. 888-905, 2000.
- [13] Y. Weiss, "Segmentation using eigenvectors: a unifying view." *Proc. Intern. Conf. on Computer Vision – ICCV'99*, pp. 975-982, 1999.
- [14] Z. Wu and R. Leahy, "Optimal graph theoretic approach to data clustering: theory and its application to image segmentation." *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 15, pp. 1101-1113, 1993.
- [15] L. Hermes and J. Buhmann, "A minimum entropy approach to adaptive image polygonization," *IEEE Trans. Image Proc.*, vol. 12, pp. 1243-1258, 2003.
- [16] R. Nowak and M. Figueiredo, "Unsupervised progressive parsing of Poisson fields using minimum description length criteria," *Proc. IEEE ICIP'99*, Kobe, Japan, vol. II, pp. 26-29, 1999.
- [17] S. C. Zhu and A. Yuille, "Region competition: unifying snakes, region growing, and Bayes/MDL for multi-band image segmentation," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 18, pp. 884-900, 1996.
- [18] S. Z. Li, *Markov Random Field Modelling in Computer Vision*. Springer Verlag, 2001. [19] Rahtu, E., Heikkilä, J.: A simple and efficient saliency detector for background subtraction. In: *Proc. the 9th IEEE International Workshop on Visual Surveillance (VS 2009)*, Kyoto, Japan, pp. 1137-1144 (2009)
- [20] Y. Sheikh and M. Shah. Bayesian modeling of dynamic scenes for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1778-1792, 2005.
- [21] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *Proc. European Conference of Computer Vision (ECCV)*, pages 751-757, 2000.
- [22] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh. Background modeling and subtraction of dynamic scenes. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1305-1312, 2003.
- [23] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 246-252, 1999.
- [24] M. Heikkilä and M. Pietikainen. A texture-based method for modeling the background and detecting moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):657-662, 2006.
- [25] A. Murray and D. Basu. Motion tracking with active camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):449-459, 1994.
- [26] Y. Ren, C. Chua, and Y. Ho. Motion detection with nonstationary background. *Machine Vision and Applications*, 13(5-6):332-343, 2003.
- [27] O. Tuzel, F. Porikli, and P. Meer. Region covariance: a fast descriptor for detection and classification. In *Proc. European Conference of Computer Vision (ECCV)*, volume 2, page 589600, 2006.
- [28] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, 2005.
- [29] O. Tuzel, F. Porikli, and P. Meer. Human detection via classification on riemannian manifolds. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.
- [30] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE*

- Transactions on Pattern Analysis and Machine Intelligence, 20(11):1254–1259, 1998.
- [31] <http://lcvwww.epfl.ch/achanta/SalientRegionDetection/SalientRegionDetection.html>.
- [32] Y. Hu, X. Xie, W. Ma, L. Chia, and D. Rajan. Salient region detection using weighted feature maps based on the human visual attention model. In In Proc. Advances in Multimedia Information Processing, pages 993–1000, 2004.
- [33] A. Radhakrishna, F. Estrada, P. Wils, and S. Susstrunk. Salient region detection and segmentation. In In Proc. International Conference on Computer Vision Systems (ICVS), pages 66–75, 2008.
- [34] X. Hou and L. Zhang. Saliency setction: a spectral residual approach. In In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007.
- [35] C. Guo, Q. Ma, and L. Zhang. Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform. In In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008.
- [36] V. Mahadevan and N. Vasconcelos. Background subtraction in highly dynamic scenes. In In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008.
- [37] A. Radhakrishna, F. Estrada, P. Wils, and S. Susstrunk. Salient region detection and segmentation. In In Proc. International Conference on Computer Vision Systems (ICVS), pages 66–75, 2008.
- [38] Guanbin Li, Yizhou Yu; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 5455-5463
- [39] Rui Zhao, Wanli Ouyang, Hongsheng Li, Xiaogang Wang; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1265-1274
- [40] Junting Pan, Elisa Sayrol, Xavier Giro-i-Nieto, Kevin McGuinness, Noel E. O'Connor; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 598-606
- [41] G. Li , Y. Yu , Visual saliency detection based on multi scale deep CNN features, IEEE Transon Image Process. 25 (11) (2016) 5012–5024 .
- [42] W. Wang, J. Shen, L. Shao, Deep learning for video saliency detection, 2017 arXiv preprint arXiv: 1702.00871 .
- [43] L. Wang , W. Ouyang , X. Wang , H. Lu. , Visual tracking with fully convolutional networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2015 .
- [44] W. Wang , J. Shen , L. Shao , Video salient object detection via fully convolu- tional networks, IEEE Trans. Image Process. 99 (2017) 1 .
- [45] Jauch, C. Patología Vegetal. Ed. El Ateneo, Buenos Aires, 1979.
- [46] Quirós, Alfredo Durán. .^{En}fermedades Y Otros Problemas de Las Plantas: Reconocimiento de Campo”. Editorial Universidad de Costa Rica, 1998.
- [47] Prendes, José Alberto Oliveira; Khouri, Elías Afif; López, Matías Mayor,.^{Análisis} de suelos y plantas y recomendaciones de abonado”, Universidad de Oviedo, 2006
- [48] Agrios, G.N. Plant Pathology. (ed), Elsevier Academic Press. London, UK. 2005.
- [49] Barranco Gutierrez, Amador- Salgado, "Segmentación de imágenes usando HSV y Gaussiana de Segundo Orden", 2015.
- [50] Igor K Amit, Ehud R, Çolor Invariants for Person Reidenti- cation”, IEE Transactions on Pattern Analysis and Machine Intelligence, vol 35, no 7, pp 165-170, 1998.
- [51] Batllea J, Casalsb, A. Freixeneta J, Marti J, .A review on strategies for recognizing natural objects in color images of outdoor scenes, Image and Vision Computing, 18, pp 515 - 530, 2000.
- [52] https://es.wikipedia.org/wiki/Modelo_de_color_HSV
- [53] Bishop - Pattern Recognition and Machine Learning - Springer 2006.
- [54] Otsu, N., .^A Threshold Selection Method from Gray-Level Histograms. IEEE Transactions on Systems, Man, and Cybernetics. Vol. 9, No. 1, 1979, pp. 62–66.
- [55] https://en.wikipedia.org/wiki/Otsus_method
- [56] https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html
- [57] CHAPTER 3 ARTIFICIAL NEURAL NETWORKS D. T. PHAM, M. S. PACKIANATHER, A. A. AFIFY Manufacturing Engineering Centre, Cardiff University, Cardiff CF24 3AA, United Kingdom
- [58] https://en.wikipedia.org/wiki/Gaussian_blur