

**Algoritmo De Detección De Sigatoka Negra En Las Hojas De Plátano Por
Medio De Inteligencia Artificial**

Juan David Betancur Cardona

Tesis para optar al título de Ingeniería en Telecomunicaciones

Asesora

Magister en Ingeniería Electrónica Paola Andrea Mateus Abaunza

Universidad Nacional Abierta y a Distancia - UNAD
Escuela de Ciencia Básicas Tecnología e Ingeniería – ECBIT
Ingeniería en Telecomunicaciones


Medellín

2021

Página de Aceptación

Nota de aceptación

Aprobado por el comité de grados y
en cumplimiento de los requisitos
exigidos por la Universidad Nacional
Abierta y a Distancia – UNAD, para
optar al título de Ingeniero de
Telecomunicaciones



Manuel Enrique Wagner Mendivelso

Jurado

Bogotá, 30 de abril de 2021

Agradecimientos

Expreso mi gratitud a Andrés Felipe Rúa Ortiz, Paola Andrea Mateus Abaunza y Julian Andrés Osorno Giraldo pues aportaron con su conocimiento y experiencia desde el área de la ingeniería y me acompañaron en el desarrollo de este proceso en aquellos momentos en los que la incertidumbre y el desdén por este desarrollo se hacía presente, la fortaleza, la compañía, el conocimiento y la pregunta acertada siempre estaban presentes en las personas a las que hoy agradezco por su disposición y altruismo con mi persona.

¡Gracias!

Contenido

Página de Aceptación	2
Agradecimientos	3
Tabla de contenido de imágenes	5
Glosario	6
Resumen	7
Abstract	8
Introducción	9
Objetivos	10
Objetivo General	10
Pregunta de investigación:	11
Marco teórico	12
Metodología	23
Preprocesamiento	23
<i>Lectura en espacio de color RGB</i>	24
<i>Conversión a espacio de color CIEL *a*b</i>	25
<i>Separación capas CIEL *a*b</i>	25
<i>k-means clustering en capas L*a*b</i>	26
Clasificación Supervisada	33
<i>Entrenamiento</i>	33
<i>Prueba</i>	37
Resultados	39
Conclusiones	43
Anexo 1	45
Bibliografía	47

Tabla de contenido de imágenes

<i>Imagen 1. Hoja de plátano con Sigatoka Negra.</i>	9
<i>Imagen 2. Hoja de planta de papa enferma</i>	12
<i>Imagen 3. Análisis en espacio de color L^*a^*b</i>	13
<i>Imagen 4. Hoja de planta de papa sana</i>	14
<i>Imagen 5. Segmento de hoja de planta de papa enferma</i>	15
<i>Imagen 6. Hojas de distintas plantas enfermas</i>	16
<i>Imagen 7. Zona cultivo de arroz</i>	17
<i>Imagen 8. Imágenes ‘mejoradas’ al aplicar ecualización de histograma</i>	18
<i>Imagen 9. Zonas de las imágenes en las que se aísla patrón de enfermedad</i>	18
<i>Imagen 10. Espacio de color L^*a^*b</i>	19
<i>Imagen 11. Diagrama de cromaticidad del espacio de color L^*a^*b</i>	20
<i>Imagen 12. Imagen en clasificador SVM</i>	22
<i>Imagen 13. Curva ROC de un test diagnóstico hipotético</i>	22
<i>Imagen 14. Preprocesamiento de imagen</i>	23
<i>Imagen 15. Hoja de plátano enferma en RGB por capas</i>	24
<i>Imagen 16. Hoja de plátano enferma en espacio de color CIEL *a*b</i>	25
<i>Imagen 17. Comparación capa a CIEL*a*b y RGB</i>	26
<i>Imagen 18. Hoja de plátano enferma segmentada con k-means</i>	26
<i>Imagen 19. Imagen enferma en espacio de color RGB y CIEL *a*b</i>	27
<i>Imagen 20. Multiplicación de capa a enferma por imagen Zona 1 k-means</i>	28
<i>Imagen 21. Multiplicación de capa a enferma por imagen Zona 2 k-means</i>	28
<i>Imagen 22. Multiplicación de capa a enferma por imagen Zona 3 k-means</i>	29
<i>Imagen 23. Resultado multiplicación de matrices</i>	29
<i>Imagen 24. Hoja de plátano en RGB y separada por capas</i>	30
<i>Imagen 25. Hoja de plátano sana en el espacio de color CIEL *a*b</i>	30
<i>Imagen 26. Hoja de plátano sana capas a y b CIEL *a*b</i>	31
<i>Imagen 27. Hoja de plátano sin Sigatoka Negra segmentada con k-means</i>	31
<i>Imagen 28. Multiplicación de capa a sana por imagen Zona 1 k-means</i>	32
<i>Imagen 29. Multiplicación de capa a sana por imagen Zona 2 k-means</i>	32
<i>Imagen 30. Multiplicación de capa a sana por imagen Zona 3 k-means</i>	32
<i>Imagen 31. Capa a hoja de plátano enferma en espacio de color CIEL *a*b</i>	33
<i>Imagen 32. Matriz convertida en valores binarios capa a CIEL *a*b</i>	34
<i>Imagen 33. Zonas enfermas de la imagen</i>	34
<i>Imagen 34. Características iniciales base de datos entrenamiento</i>	35
<i>Imagen 35. Características adicionales base de datos entrenamiento</i>	36
<i>Imagen 36. Entrenamiento</i>	37
<i>Imagen 37. Segmento base de datos de prueba</i>	38
<i>Imagen 38. Procesamiento</i>	38
<i>Imagen 39. Ejecución de función de SVM con kernel lineal en IDE MATLAB</i>	39
<i>Imagen 40. Clasificación de SVM en R3</i>	39
<i>Imagen 41. Curva ROC Algoritmo Inteligencia Artificial</i>	42

Glosario

CIEL*A*B: conocido es CIELAB (estrictamente CIE 1976 L*a*b*). Las siglas LAB se refieren al espacio de color tridimensional, en donde L o L* es luminosidad de negro a blanco, A o a* va de rojo a verde y B o b* va de azul a amarillo.

k-means clustering: es un algoritmo de clasificación no supervisada (clusterización) que agrupa objetos en k grupos basándose en sus características. El agrupamiento se realiza minimizando la suma de distancias entre cada objeto y el centroide de su grupo o cluster. Se suele usar la distancia cuadrática.

RGB: es un modelo de color basado en la síntesis aditiva, con el que es posible representar un color mediante la mezcla por adición de los tres colores de luz primarios. El modelo de color RGB no define por sí mismo lo que significa exactamente rojo, verde o azul, por lo que los mismos valores RGB pueden mostrar colores notablemente diferentes en distintos dispositivos que usen este modelo de color. Aunque utilicen un mismo modelo de color, sus espacios de color pueden variar considerablemente.

Support Vector Machines (SVM): son un conjunto de algoritmos de aprendizaje supervisado desarrollados por Vladimir Vapnik y su equipo en los laboratorios AT&T. una SVM construye un hiperplano o conjunto de hiperplanos en un espacio de dimensionalidad muy alta (o incluso infinita) que puede ser utilizado en problemas de clasificación o regresión. Una buena separación entre las clases permitirá una clasificación correcta.

Resumen

La planta de plátano es un gran impulsor de la economía a nivel mundial de los países productores de banano y plátano, en este contexto, la Sigatoka Negra es una enfermedad que se presenta en las plantas de plátano reduciendo la capacidad de producción de esta. Como consecuencia, la planta genera el nacimiento de racimos más pequeños a causa de la afectación, que se da por el impacto negativo que produce esta enfermedad en el proceso de fotosíntesis. En Colombia, esto representa pérdidas alrededor del 13% sobre los costos totales de producción. Con base en esto, se plantea desarrollar un algoritmo mediante inteligencia artificial y apoyado en un Entorno de Desarrollo Integrado (IDE) como MatLab. Para esto, se implementarán técnicas de procesamiento de imágenes a partir de una base de datos de imágenes enfermas y sanas. Esto se logra, analizando las imágenes en dos espacios de color (**RGB** y **CIEL*a*b**) para obtener características relevantes en ellas y permitir una segmentación de estas características, asilándolas en conjuntos o clústeres a través de la técnica conocida como **k-means**. Luego, se realizará un “entrenamiento” al algoritmo en pro de obtener un procesamiento adecuado y a su vez, permitir la identificación de los elementos más relevantes a través de un clasificador supervisado, que podrá detectar las hojas enfermas o sanas, con la intención de automatizar la detección de la enfermedad al través del tratamiento de imágenes.

Palabras clave: SIGATOKA NEGRA, INTELIGENCIA ARTIFICIAL, ALGORITMO, PLANTA DE PLÁTANO, TRATAMIENTO DE IMÁGENES.

Abstract

The banana plant is a major driver of the global economy of banana and banana producing countries, in this context, Sigatoka Negra is a disease that occurs in banana plants reducing banana production capacity. As a result, the plant generates the birth of smaller clusters due to the involvement, which is caused by the negative impact that this disease produces on the process of photosynthesis. In Colombia, this represents losses of about 13% on total production costs. Based on this, it is proposed to develop an algorithm using artificial intelligence and supported by an Integrated Development Environment (IDE) such as MatLab. For this, image processing techniques will be implemented from a database of sick and healthy images. This is accomplished by analyzing the images in two color spaces (RGB and CIEL*a*b) to obtain relevant characteristics in them and allow segmentation of these characteristics, removing them into assemblies or clusters through the technique known as k-means. The algorithm will then be "training" for proper processing and in turn allowing the identification of the most relevant elements through a supervised classifier, which will be able to detect diseased or healthy leaves, with the intention of automating the detection of the disease through imaging.

Keywords: SIGATOKA NEGRA, ARTIFICIAL INTELLIGENCE, ALGORITHM, BANANA PLANT, IMAGE TREATMENT.

Introducción

La enfermedad de la Sigatoka negra genera grandes perjuicios en un cultivo de plátano pues la presencia de esta genera afectación en el proceso de fotosíntesis de la planta lo que implica menos energía en esta; por ende, menos producción del fruto que esta planta produce (Arzanlou et al., 2007). Detectar esta enfermedad a tiempo, puede permitir aislar de manera preventiva la enfermedad por sectores en un cultivo o en la planta que presenta la afectación; sin embargo, el proceso de detección de este tipo de enfermedades puede volverse bastante tediosos pues los cultivos de la planta de plátano en ambientes productivos ocupan varias hectáreas, lo que implica un proceso de revisión manual poco eficiente y productivo.

En este contexto, se propone un desarrollo tecnológico que permita minimizar las pérdidas económicas que produce la Sigatoka Negra apoyado en la automatización y detección temprana de la enfermedad a través del análisis de imágenes.

Cuando la enfermedad se presenta en la planta de plátano, sus hojas comienzan a tomar colores distintos al verde; desde un amarillo pálido, pasando por el café y terminando en un negro opaco (*Imagen 1*)

Imagen 1. Hoja de plátano con Sigatoka Negra.



fuentes propia procesada en MATLAB

Al aplicar distintas técnicas de detección de imágenes, es posible aislar zonas de la imagen analizada y realizar un tratamiento a esta, de manera que sea posible 'caracterizar' las imágenes enfermas y las sanas. La aplicación de estas técnicas con el aprendizaje automático (*machine learning*) (Islam et al., 2017) ofrece una solución práctica al problema de producción de plátano y a su vez ayuda a reducir el uso de químicos que tanto daño producen en la tierra.

Objetivos

Objetivo General

Desarrollar un algoritmo de inteligencia artificial que permita la detección de Sigatoka Negra en hojas de plátano.

Objetivos Específicos

- Identificar la evolución y tendencias investigativas de la Sigatoka Negra, a través de una revisión del estado del arte.
- Adquirir una base de datos que contenga imágenes enfermas con Sigatoka Negra e imágenes sanas de la hoja de plátano.
- Implementar técnicas de tratamiento de imágenes con el fin de identificar las características de las imágenes de la base de datos recolectada.
- Desarrollar un clasificador que permita la identificación temprana de imágenes sanas y enfermas en las hojas de plátano.

Pregunta de investigación:

¿Qué proceso tecnológico permitiría la detección de la Sigatoka Negra en las hojas de plátano?

Marco teórico

Teniendo claro los objetivos, es procedente identificar bibliografía relacionada con temas orientados al tratamiento digital de imágenes. El objeto de dicha búsqueda está orientado a brindar una perspectiva de las investigaciones o impactos logrados con el tratamiento digital de imágenes; por tanto, es procedente referenciar los siguientes estudios realizados o analizar como desarrollan sus objetivos:

En un estudio realizado por el Departamento de Ingeniería Eléctrica e Informática de la universidad de Saskatchewan ubicada en Saskatoon, Canadá; aplican el tratamiento digital de imágenes teniendo como objetivo, detectar una enfermedad presente en las hojas de la planta papa (*Imagen 2*).

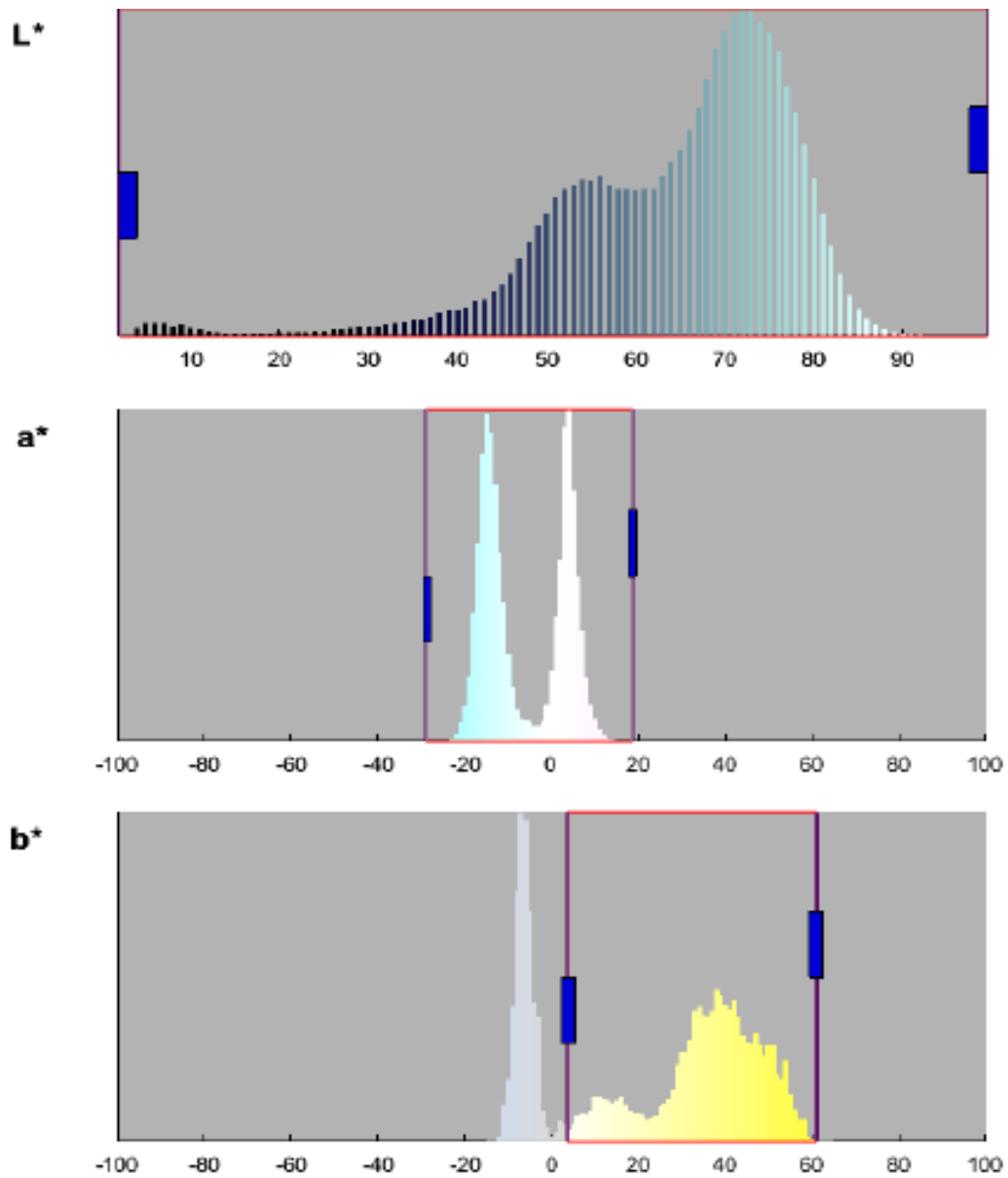
Imagen 2. Hoja de planta de papa enferma



Tomado de (Islam et al., 2017)

Esta enfermedad; al igual que en las plantas de plátano, afecta el proceso fotosintético de las plantas reduciendo su potencial productivo en proporciones difíciles de ignorar. El proyecto básicamente realiza lectura de las imágenes en un espacio de color muy reconocido actualmente (RGB) (*Imagen 2*) para luego hacer la conversión al espacio de color **CIEL*a*b** (*Imagen 3* (Islam et al., 2017)).

Imagen 3. Análisis en espacio de color **L*a*b**



Tomado de (Islam et al., 2017)

Así, logran obtener características digitales de la imagen las cuales, como se observa en la *Imagen 3*, permite aislar elementos de esta en procura de segmentar la región poseedora de singularidades propias de la 'asimetría' en el color que allí se presenta.

Para lograr detectar esta 'asimetría, se hace necesario determinar cuál es el opuesto a esta. Para ello, procuran realizar lecturas de imágenes sanas (*Imagen 4*) con intención de realizar comparaciones de patrones obtenidos de ambos extremos (imágenes enfermas o sanas)

Imagen 4. Hoja de planta de papa sana

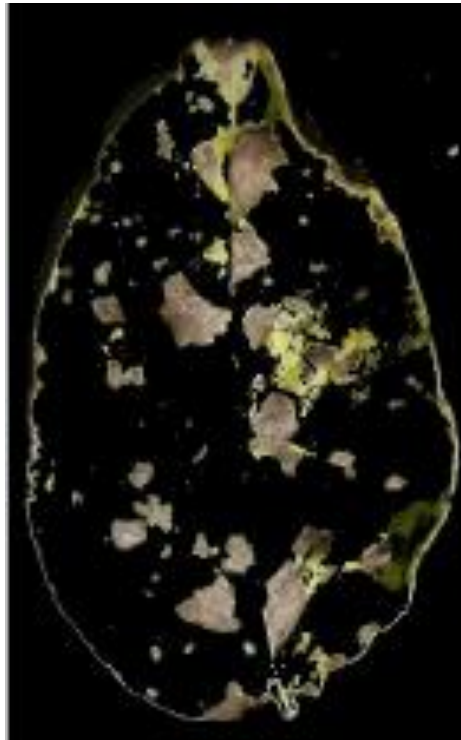


Tomado de (Islam et al., 2017)

*La imagen 2 a simple vista, ofrecen diferencias que logran procesar a través del espacio de color **CIEL*a*b** y, sobre este resultado, determinan que 'umbrales' obedecen o caracterizan homogeneidad en las imágenes que se procesan.*

Con base en esta detección, se logran aislar los patrones que no presentan uniformidad en el color, obteniendo conjuntos como los dispuestos en la Imagen 5.

Imagen 5. Segmento de hoja de planta de papa enferma



Tomado de (Islam et al., 2017)

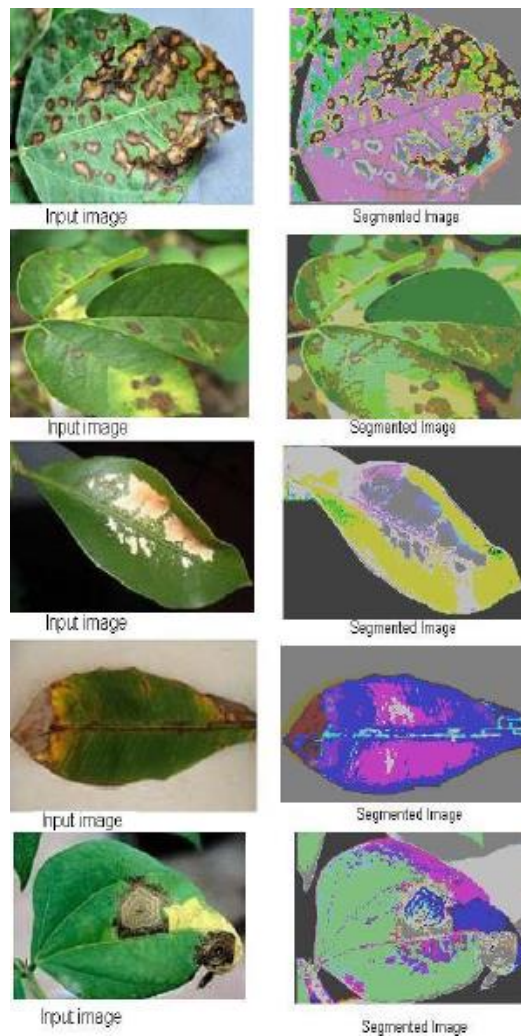
Al lograr clasificar las imágenes enfermas de las sanas (por un experto humano) se establece un valor con el cual realizan una automatización supervisada con el método Vector Support Machines (**SVM**). Este básicamente, establece un hiperplano sobre el cual hay un margen y ubica en sus extremos los elementos que poseen patrones o características propias de cada lado de este hiperplano logrando ofrecer una clasificación automática de los elementos evaluados en este (Islam et al., 2017).

En otro estudio expuesto en la Conferencia Internacional Sobre Avances en Ingeniería y Aplicaciones Informáticas (ICACEA) en el 2015 (Singh et al., 2015), nos muestran como la aplicación de distintas técnicas de segmentación de imágenes, arrojan ciertos resultados respecto a la enfermedad analizada en los tipos de hojas enfermas a las que se les aplica el estudio como lo son las hojas de las plantas de plátano, frijoles, yaca, limón, mango, papa, tomate y sapote (*Imagen 6*). Al aplicar

distintos métodos de segmentación de imágenes, procuran obtener resultados que permita determinar qué tipo de enfermedad está presente en la planta.

El espacio de color **CIEL*a*b** se usa como una herramienta que permite corregir errores en la imagen que se ingresa en el proceso de segmentación y, en el desarrollo del estudio, encuentran que a través del método conocido como **k-menas clustering**, es posible detectar distintos tipos de enfermedades. En los varios métodos que utilizan, encuentran que los parámetros que se utilizan para el Vector Support Machines (**SVM**) deben ser muy concretos respecto a las características de las imágenes analizadas por lo que la técnica de segmentación juega un papel fundamental.

Imagen 6. Hojas de distintas plantas enfermas



Tomado de (Singh et al., 2015)

En este estudio buscan, respecto al resultado con la técnica de umbralización utilizada (Umbral de Outzu) sobre la imagen ya segmentada, determinar que patrones obedecen a qué tipo de enfermedades.

La segmentación de imágenes permite obtener datos que pueden ser utilizados para entrenar un algoritmo de inteligencia artificial que permita automatizar un proceso de detección de patrones (enfermedades, por ejemplo) en hojas de plantas.

En el Sysposium del Consorcio de Universidades Técnicas del Sudeste Asiático (SEATUC), Yogyakarta, Indonesia realizado en 2018, exponen un estudio realizado para la detección de enfermedades en la hoja de arroz (Kitpo & Inoue, 2018).

En este, el ***k-means clustering*** se utiliza para formar conjuntos de patrones de las imágenes que se procesan y esta información la disponen en una red neuronal artificial utilizando el método conocido como Fuzzy Logic el cual consiste en arrojar dos posibles valores como resultado del análisis aplicado.

Los drones se encargan de obtener la materia prima para el procesamiento de las imágenes: las imágenes (Kitpo & Inoue, 2018). Estas se obtendrán programando a los drones para que realicen recorridos en un campo de cultivo de arroz. A los drones se les asignarán cuadrantes los cuales serán delimitados a través de un sistema de posicionamiento global (GPS). Las imágenes (fotos) que obtengan los drones, serán marcadas con las coordenadas en las cuales fueron obtenidas para determinar la imagen a que sector del cultivo pertenece (*Imagen 7*).

Imagen 7. Zona cultivo de arroz

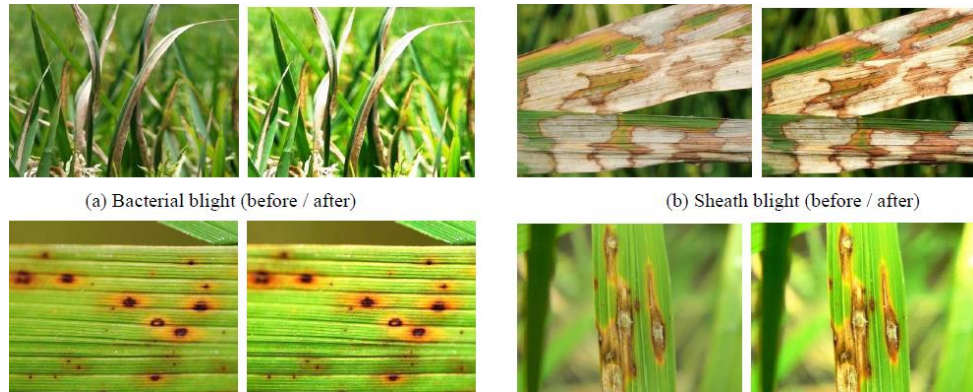


Tomado de (Kitpo & Inoue, 2018)

En este estudio utilizan el espacio de color **HSV**. En esta técnica, se realizan lecturas del tinte o matiz, saturación y brillo. Usan este espacio de color pues consideran que, al tomar las imágenes en un ambiente sometidos a cambios de

intensidad de luz, es más fácil segmentar la imagen a través de esta técnica y a este resultado realizar una ecualización del histograma (*Imagen 8*) de la imagen permitiendo resaltar patrones en la imagen que ayuden a segmentar la enfermedad que buscan detectar.

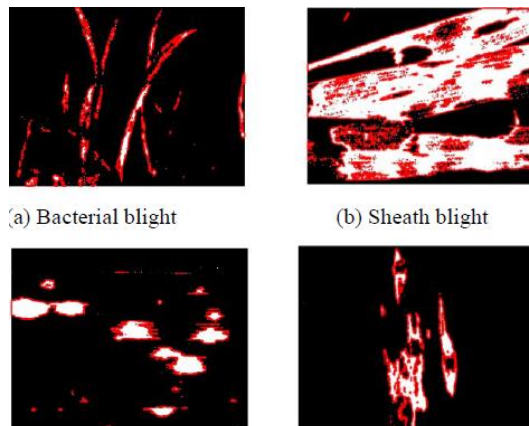
Imagen 8. Imágenes ‘mejoradas’ al aplicar ecualización de histograma



Tomado de (Kitpo & Inoue, 2018)

Luego de realizada esta ecualización, se obtiene una imagen más nítida por lo que aplican la técnica de segmentación **HSV** asilando las áreas de la imagen que poseen la enfermedad (*Imagen 9*).

Imagen 9. Zonas de las imágenes en las que se aísla patrón de enfermedad



(c) Brown spot

(d) Leaf blast

Tomado de (Kitpo & Inoue, 2018)

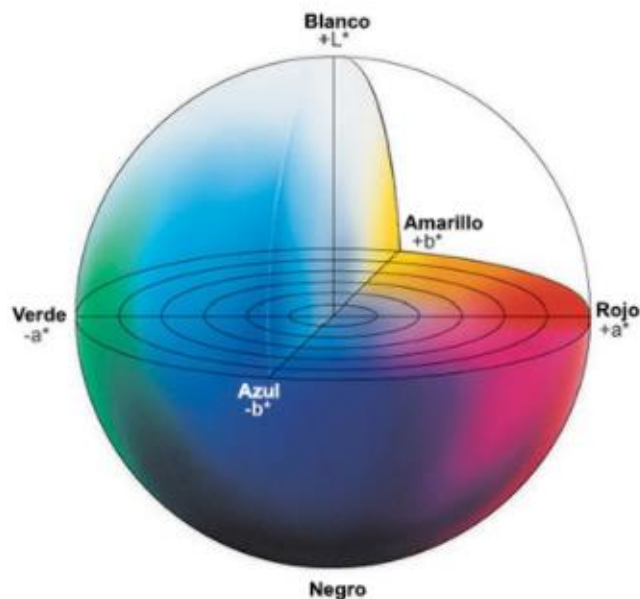
Una de las conclusiones que obtienen y la cual resalta es que consideran más apropiado el uso del Vector Support Machines (**SVM**) pues el atributo de hiperplano permite una clasificación más exacta de las características con las cuales se entrena este.

La implementación de técnicas orientadas al procesamiento de imágenes permite aplicación a gran variedad de campos en la actualidad como lo es la medicina, la agricultura, seguridad, entre otros. Teniendo en cuenta el objetivo principal de este proyecto, es necesario reconocer de manera general, en qué consisten las técnicas de tratamiento digital de imágenes que se implementarán para el desarrollo de los objetivos específicos buscando brindar claridad respecto a las metodologías implementadas.

La detección de patrones en imágenes que se capturan inicialmente basadas en el modelo de color **RGB**, brinda información importante pues permite clasificación en tres colores distintos (rojo, verde y azul). Un píxel en cada 'capa', contiene un valor entre 0 y 255 lo que indica la intensidad del color en cada capa generando matrices con $M \times N$ píxeles. (Thangadurai & Padmavathi, 2014).

Entre los modelos de color, el modelo **CIEL*a*b** (conocido como **CIELab**) estándar de la Comisión Internacional sobre Iluminación (CIE), define tres parámetros: **L**: varianza entre el blanco y el negro (luminiscencia), **a**: cambios de color entre el **rojo/verde** y **b**: cambios de color entre **amarillo/azul**. (imagen 10) (Minolta, n.d.)

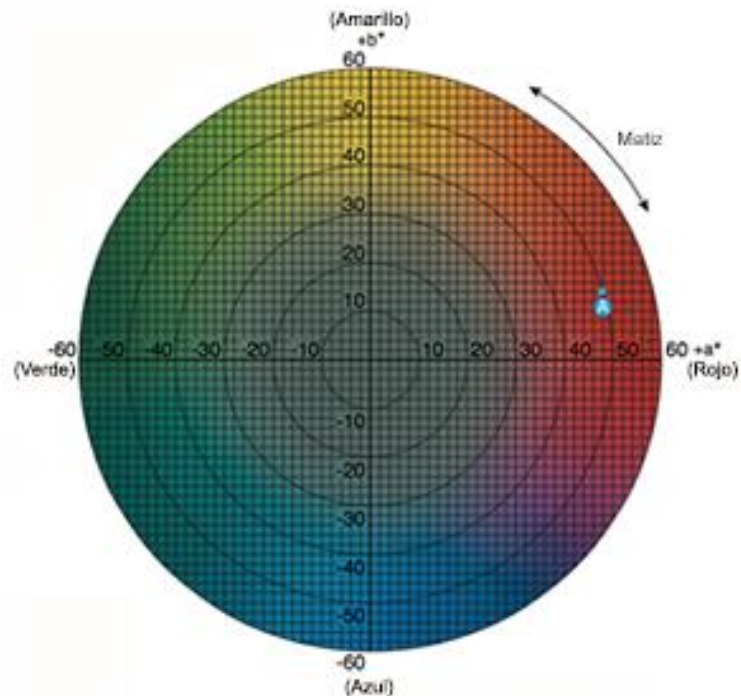
Imagen 10. Espacio de color L^*a^*b



Tomado de (Minolta, n.d.)

El espacio de color L^*a^*b establece básicamente que dos colores no pueden ser rojo y verde al mismo tiempo al igual que el amarillo y azul. Al aplicar este estándar de color al tratamiento digital de una imagen, podemos asignar una representación numérica a cada píxel en la imagen.

Imagen 11. Diagrama de cromaticidad del espacio de color L^*a^*b



Tomado de (Minolta, n.d.)

Una de las bondades del espacio de color L^*a^*b , consiste en la manera que representa numéricamente el píxel en cada una de sus capas. Los valores poseen cifras decimales permitiendo obtener mayores características de los píxeles de una imagen.

Con el espacio de color L^*a^*b , se obtienen datos que pueden ser fácilmente clasificados en conjuntos o clúster lo que permite, aislar áreas específicas en la imagen analizada.

La **segmentación basada en clústeres** consiste en separar una imagen en varios grupos con píxeles de características similares, tratando de seleccionar los centros de los clústeres de forma aleatoria. Luego, se le asigna a cada imagen un clúster que minimice la varianza entre el píxel y el centro de éste, después se recalcula el centro del clúster por medio de la media de los píxeles almacenados; esto se repite varias veces, hasta conseguir una convergencia o que los píxeles no

cambien de clúster (Rodríguez Rieiro, 2008). Luego de obtener estos conjuntos de datos; deben ser sometidos a una evaluación.

La **evaluación cualitativa**, consiste en identificar u obtener un patrón determinado dentro una imagen. La evaluación cualitativa generalmente se realiza por uno o varios expertos (humanos) que están en la capacidad de determinar las características de una imagen con el fin de identificar los objetos más relevantes de éstas (Nixon & Aguado, 2002). Al obtener esta evaluación (cualitativa) es posible realizar una evaluación cuantitativa de la información obtenida en las matrices resultantes de la lectura de la imagen en el espacio de color **L*a*b** aplicando la segmentación basada en clústeres. El método conocido como **k-means clustering** permite una cuantificación vectorial el cual, busca patrones similares en una matriz y los agrupa en conjuntos o clúster (Aristidis, Likas Vlassis, Nikos Verbeek, 2003), permitiendo aislar un conjunto de datos o de información de un todo.

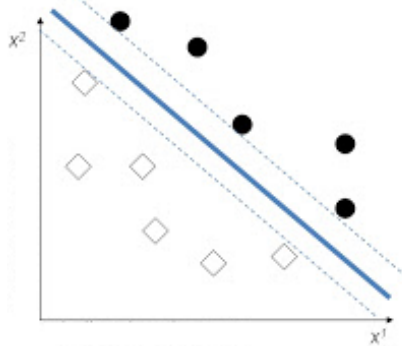
Los métodos de automatización supervisada como el **Vector Support Machines (SVM)** son algoritmos de aprendizaje o clasificación supervisada. Básicamente éste se encarga, con base en el aprendizaje supervisado, de clasificar un objeto o dato que ingresa al proceso de aprendizaje y lo 'ubica' según la parametrización del algoritmo de aprendizaje. En términos simples, la información que entra al clasificador es tenida en cuenta como un vector. Este algoritmo, se desarrolló en los años de 1990 como método de clasificación binaria. Hoy en día se utiliza para resolver problemas de clasificación múltiple siendo muy utilizado en el aprendizaje estadístico y en el *machine learning*. Los **SVM** se fundamentan en **Maximal Margin Classifier** que, a su vez, utilizan el concepto de hiperplano. Los SVM dependen en demasía del proceso de aprendizaje; pues inicialmente, se le debe indicar a que conjunto, dentro de un hiperplano, pertenecen los resultados obtenidos en el análisis realizado por el algoritmo. Cuando el clasificador (**SVM**) está debidamente entrenado, éste realiza de manera automática el proceso de selección, convirtiéndose en una especie de caja que devuelve la respuesta ante un nuevo caso.

Para los **SVM** se debe tener en cuenta un concepto muy importante llamado margen. Éste básicamente trata sobre el umbral o 'espacio' respecto a un límite que existe entre los datos resultantes del clasificador. En pocas palabras; hace referencia a la distancia que existe, según la línea (separación) definida en nuestro hiperplano, según la cual, separa los elementos en distintos conjuntos. (Orestes, 2008)

A continuación, el siguiente ejemplo: para entrenar un algoritmo que detecta pacientes con diabetes y sin diabetes, se debe clasificar la información resultante. Lo que hace el **SVM** básicamente es ubicar en un plano (x,y) el resultado de cada paciente; es decir, si es o no portador de la enfermedad. Para ello, establece un límite en el plano y de un lado ubica los pacientes positivos con la enfermedad y los que no, al otro extremo del límite. Dependiendo del entrenamiento del algoritmo, los

resultados, se ubicarán más cerca o no de la línea o límite. El margen corresponde a los datos en ambos extremos más cercanos al límite:

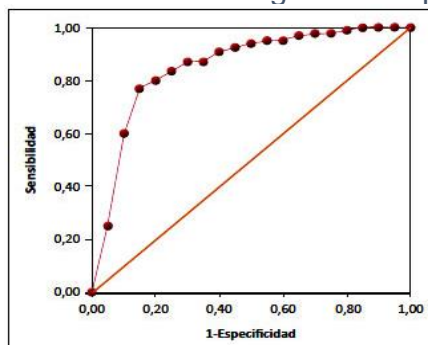
Imagen 12. Imagen en clasificador SVM



Tomado de (Orestes, 2008)

Luego de obtenidos los datos y haber logrado una clasificación, es importante evaluar que tan fiables son los resultados obtenidos en el proceso de automatización. La *curva de característica operativa del receptor* o *curva ROC* permite una representación gráfica de la sensibilidad, frente a una cantidad de datos que tienen como objetivo, establecer un sistema clasificador binario, según se varía en un umbral determinado a través de un proceso de clasificación previo. La curva ROC es la representación del promedio o razón de verdaderos positivos (VPR = Razón de Verdaderos Positivos), frente a la razón de falsos positivos (FPR = Razón de Falsos Positivos), según se varía el umbral de distinción entre ambos casos (valor a partir del cual decidimos que un caso es un positivo). ROC también significa: Relative Operating Characteristic (Característica Operativa Relativa) pues es una comparación de dos características (VPR y FPR), según cambiamos el umbral definido. (Cerde & Cifuentes, 2012).

Imagen 13. Curva ROC de un test diagnóstico hipotético



Tomado de: (Cerde & Cifuentes, 2012)

Metodología

Para el desarrollo de este proyecto se cuenta con dos bases de datos. La primera se refiere a una base de datos de imágenes enfermas y la segunda, a imágenes sanas. Estas bases de datos de imágenes se obtienen de una finca platanera ubicada en el municipio de Guarne (Antioquia). Esta finca posee un total de 104 imágenes entre las que se encuentran 52 imágenes con presencia de la Sigatoka Negra (enfermas) y 52 imágenes con ausencia de esta enfermedad (sanas).

El proceso de obtención de estas imágenes se realiza tomando fotos de las hojas de plátano en el espacio de color **RGB** en formato jpg y jpeg con resolución de 720 x 1080 pixels.

Las imágenes obtenidas, deberán pasar por tres etapas descritas a continuación:

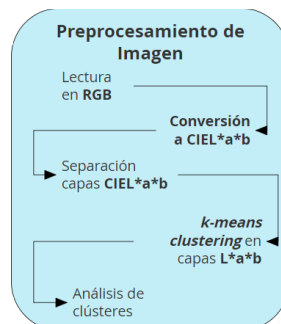
- Preprocesamiento
- Entrenamiento
- Prueba

Preprocesamiento

Las 104 imágenes se dividen en dos grupos: uno, se utilizará en el proceso de entrenamiento el cual contiene 12 imágenes enfermas y 8 sanas. El grupo de prueba, contiene 40 imágenes enfermas y 44 sanas.

Una vez ejecutada esta acción, se procede a realizar el preprocesamiento de cada una de las imágenes con objeto de determinar cuáles serán las características apropiadas para el entrenamiento del algoritmo de inteligencia artificial.

Imagen 14. Preprocesamiento de imagen

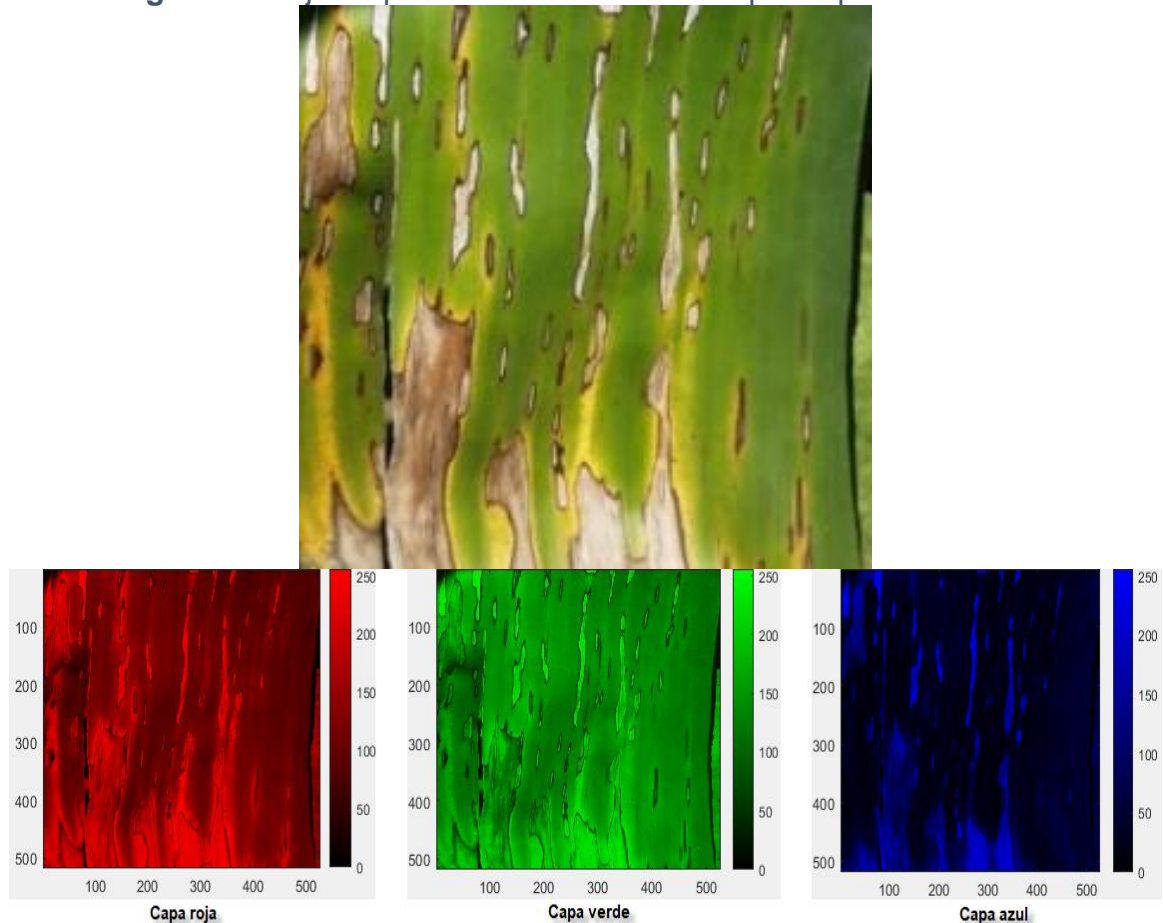


Elaboración propia

Lectura en espacio de color **RGB**

Se procede a subir cada imagen al IDE MATLAB realizando la lectura de esta en el espacio de color **RGB** con la función `imread`. Esta información se almacena en una variable y con la cual se da inicio al preprocesamiento. Este proceso arroja como resultado, una imagen con tres capas (**Red, Green, Blue**) (*Imagen 15*)

Imagen 15. Hoja de plátano enferma en **RGB** por capas



Fuente propia procesada en MATLAB

Al procesar digitalmente la imagen, se obtienen tres matrices de $N \times M$ dimensiones (depende de la cantidad de píxeles en la imagen). Las tres matrices corresponden a cada uno de los colores del espacio de color **RGB** (**Red** (rojo), **Green** (verde), **Blue** (azul)).

Conversión a espacio de color **CIEL*a*b**

A la imagen en el espacio de color **RGB**, se le realiza conversión al espacio de color **CIEL*a*b** (*Imagen 16*) en el IDE MATLAB.

Imagen 16. Hoja de plátano enferma en espacio de color **CIEL*a*b**



Fuente propia procesada en MATLAB

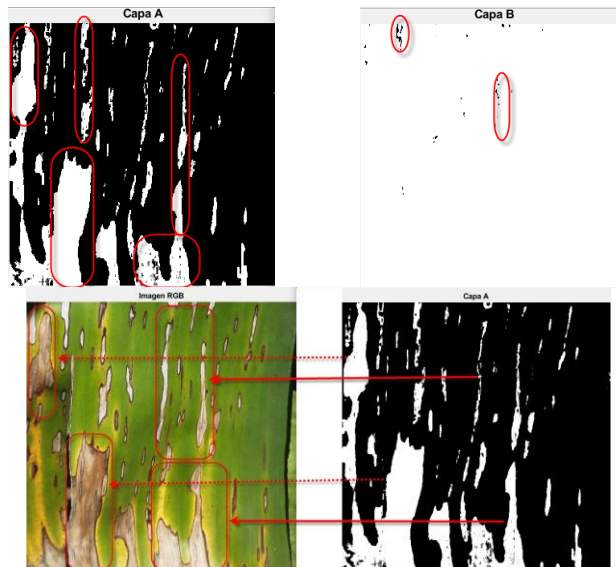
Esta conversión, arroja la misma cantidad de matrices (3) correspondientes a cada espacio de 'color' (**L** = luminiscencia: varianza entre el blanco y el negro, **a**: cambios de color entre el **rojo/verde** y **b**: cambios de color entre **amarillo/azul**).

Una de las bondades de la digitalización de la imagen, consiste en la segmentación de cada uno de sus componentes. Para el ojo humano, el objeto (la imagen) se comporta como un todo; pero esto, no es más que una mera ilusión. La imagen se construye de manera tal, que pueda ser interpretada por el observador como un elemento comprensible a su realidad circundante. Realmente las imágenes, son un compendio de capas que a su vez están construidas por unidades llamadas pixels. Estos se convierten en la materia prima al momento de realizar un tratamiento digital de imágenes.

*Separación capas **CIEL*a*b***

En tanto estas características, se procede a tomar cada capa del espacio de color **CIEL*a*b** (*Imagen 17*) buscando áreas o zonas correspondientes a la presencia de la enfermedad de la Sigatoka Negra; para esto, se hace necesario separar cada una de las capas y analizar los datos que estas arrojan. Se observa que la **capa a** posee más información respecto a la presencia de la Sigatoka Negra en la imagen, mientras que en la **capa b** no se observan características relevantes respecto a la enfermedad (*Imagen 17*).

Imagen 17. Comparación capa a CIEL*a*b y RGB

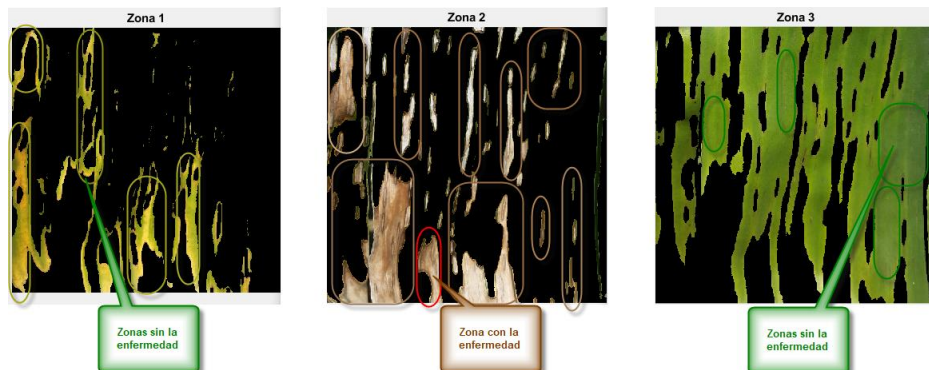


Fuente propia procesada en MATLAB

k-means clustering en capas L^*a^*b

Teniendo las imágenes segmentadas en el espacio de color $CIEL^*a^*b$, es necesario reunir los elementos de estas capas, en conjuntos o clúster que posean características similares. Para este proceso, se aplica el método conocida como *k-means clustering* lo que permitirá detectar en la imagen procesada, las zonas con presencia o ausencia de la enfermedad (*Imagen 18*).

Imagen 18. Hoja de plátano enferma segmentada con *k-means*



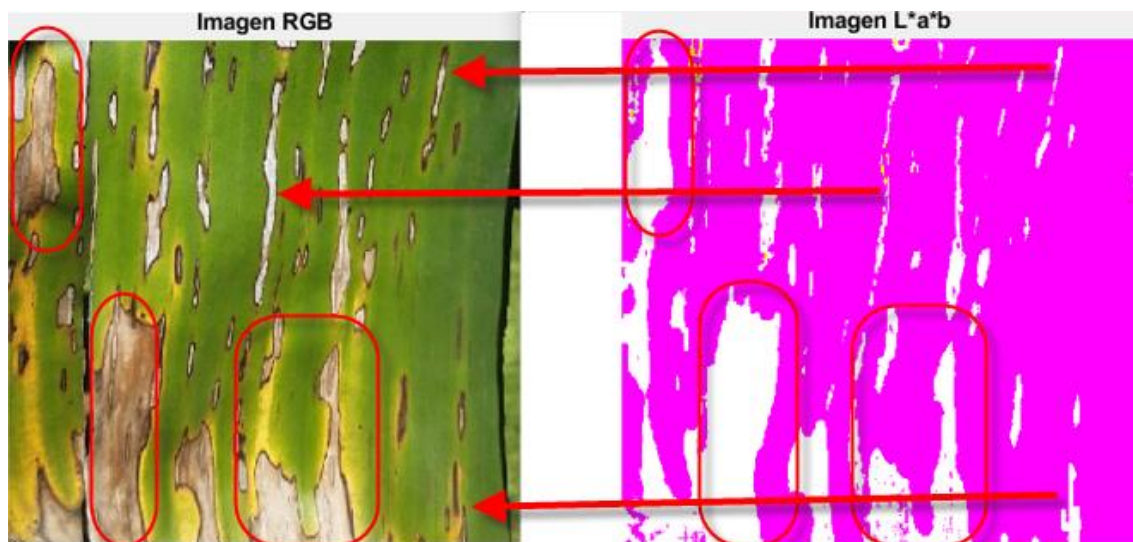
Fuente propia procesada en MATLAB

Se observa como el método del *k-means*, realiza una ‘separación’ de los elementos de la imagen previamente procesada en el espacio de color **CIEL*a*b**. Con estos elementos, se realizan ‘comparaciones’ que permiten entrenar al algoritmo de inteligencia artificial.

Análisis de clústeres

En el análisis previo respecto a las imágenes obtenidas en los distintos espacios de color (**RGB** y **CIEL*a*b**. (*Imagen 15* e *Imagen 17*), se evidencian las zonas que corresponde con la enfermedad. Al ser evidente este patrón en la imagen enferma analizada, es prudente realizar una ‘multiplicación’ entre las ‘imágenes’ con el objeto de suprimir las zonas que no poseen la enfermedad y así poder obtener solo las características propias de esta (*Imagen 19*).

Imagen 19. Imagen enferma en espacio de color **RGB** y **CIEL*a*b**



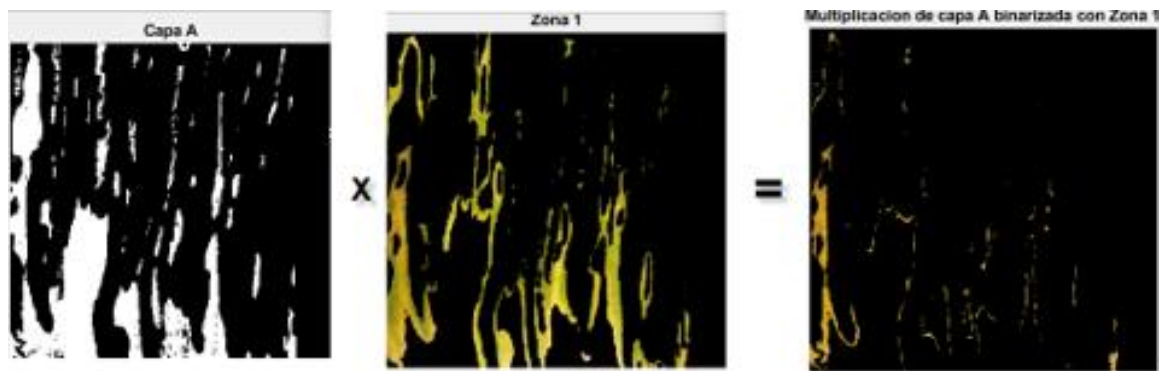
Fuente propia procesada en MATLAB

Lo que se hace realmente, es una multiplicación de matrices. Recordemos que las imágenes digitalizadas se convierten en matrices con valores correspondientes al espacio de color tratado; por ende, es procedente realizar operaciones entre las matrices resultantes de las conversiones realizadas en los pasos anteriormente descritos.

Esta multiplicación se lleva a cabo entre la **capa a** del espacio de color **CIEL*a*b** y las matrices resultantes de la segmentación con *k-means* (*Imagen 18*). Previo a esto se deben convertir en valores binarios los datos la **capa a**; esto, con el objeto de obtener valores de cero (0) en los pixeles que no posean la enfermedad

y valores de uno (1) en los pixeles que poseen la enfermedad (*Imagen 21*). La razón de esto obedece a lo siguiente: en la *Imagen 20a*, se observa que la zona en la **capa a** que no posee la enfermedad de la Sigatoka Negra carece de características (áreas de color negro); sin embargo, esta misma capa presenta una similitud inconfundible con la imagen en el espacio de color **RGB** (*Imagen 19*) respecto a las áreas donde se presenta la enfermedad (áreas de color blanco). Al realizar la multiplicación de la **capa a** (convertida en valores binarios) por la Zona 1 (*Imagen 20*), Zona 2 (*Imagen 21*) y Zona 3 (*Imagen 22*), se podrá brindar características al algoritmo de inteligencia artificial que le permitan determinar si una hoja está enferma o sana (*Imagen 23*).

Imagen 20. Multiplicación de **capa a** enferma por imagen Zona 1 **k-means**



Fuente propia procesada en MATLAB

Imagen 21. Multiplicación de **capa a** enferma por imagen Zona 2 **k-means**



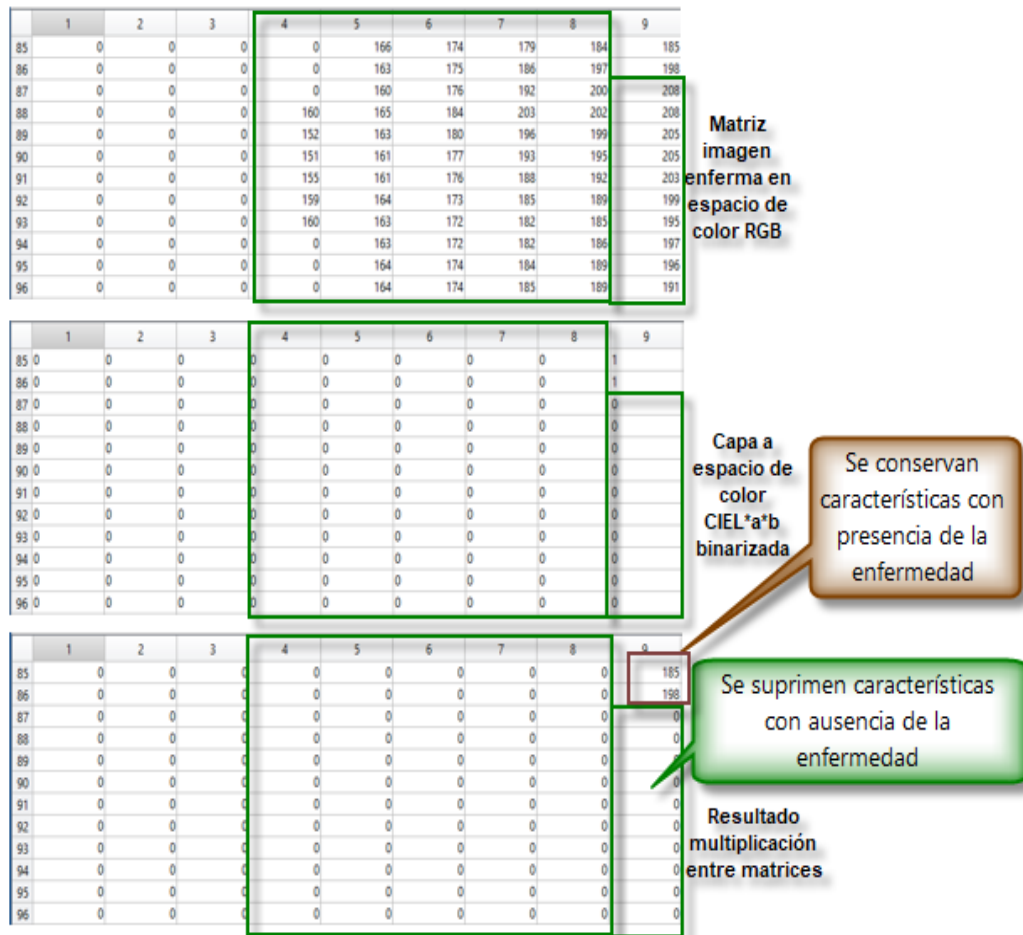
Fuente propia procesada en MATLAB

Imagen 22. Multiplicación de capa a enferma por imagen Zona 3 k-means



Fuente propia procesada en MATLAB

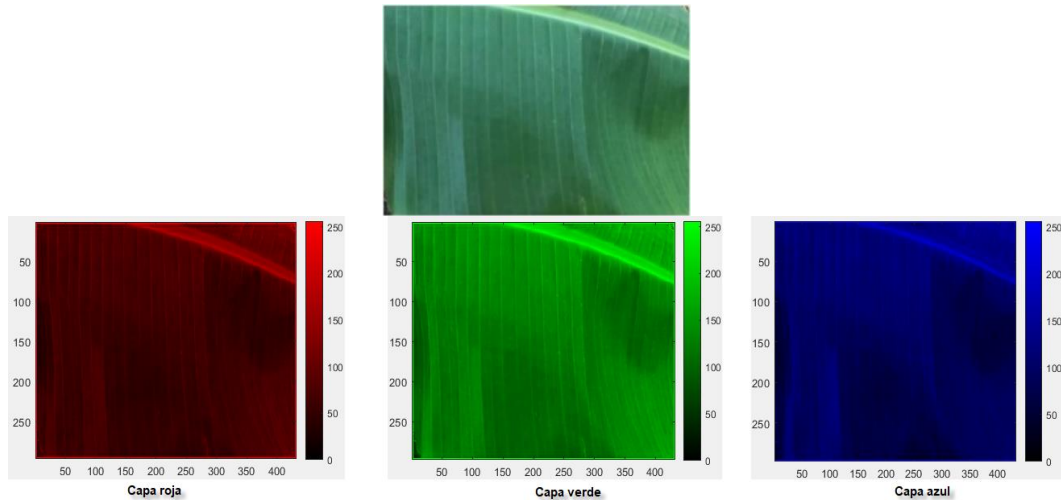
Imagen 23. Resultado multiplicación de matrices



Fuente propia procesada en MATLAB

Al procesar las imágenes sanas (*Imagen 24*) en el espacio de color **RGB**, obtenemos características con similitudes en las tres capas las cuales, se usan para el entrenamiento del algoritmo de inteligencia artificial que permita determinar esta característica (imagen sana).

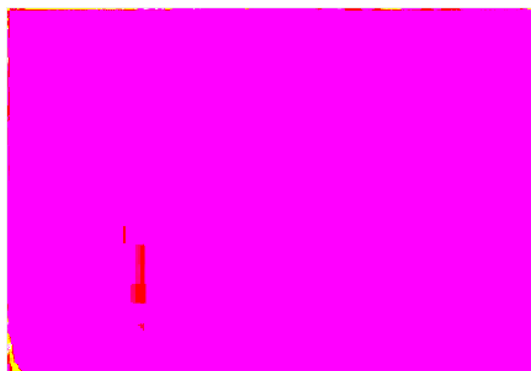
Imagen 24. Hoja de plátano en **RGB** y separada por capas



Fuente propia procesada en MATLAB

Al realizar el procesamiento de la *imagen 24* (leer la imagen en el espacio de color **RGB** y procesarla en el espacio de color **CIEL*a*b**), se obtiene el resultado expuesto en la *imagen 25*.

Imagen 25. Hoja de plátano sana en el espacio de color **CIEL*a*b**



Fuente propia procesada en MATLAB

Al segmentar la *imagen 25*, se evidencia que entre las **capas a y b** (*Imagen 26*), la diferencia es mínima.

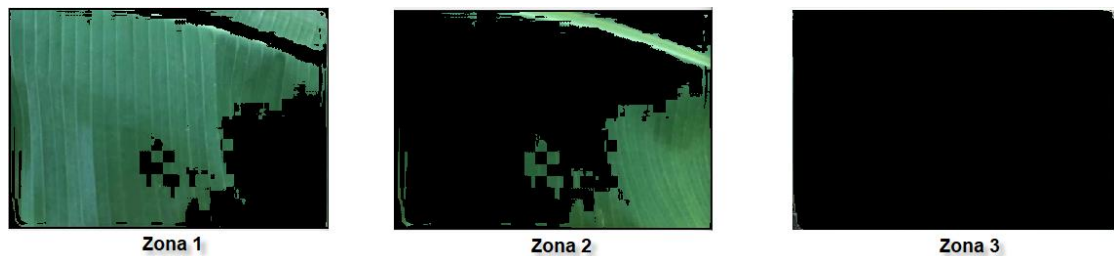
Imagen 26. Hoja de plátano sana capas **a** y **b** CIEL*a*b



Fuente propia procesada en MATLAB

En la **capa a**, se observa uniformidad respecto al área o zona que posee el color verde en la *Imagen 24*. Al separar por clústeres esta imagen, se obtendrán los distintos tonos de verde (*Imagen 27*) de esta.

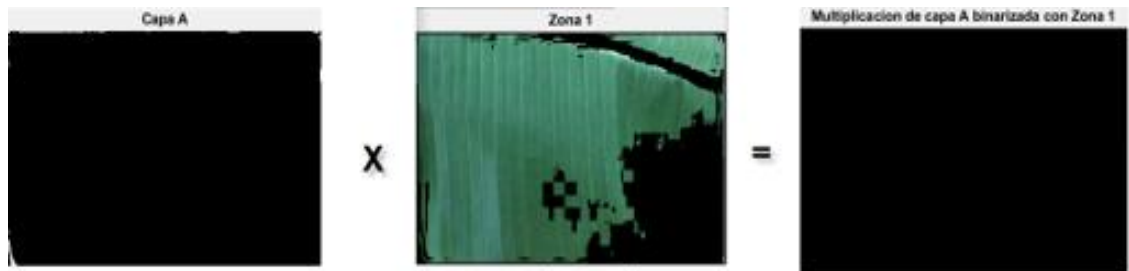
Imagen 27. Hoja de plátano sin Sigatoka Negra segmentada con **k-means**



Fuente propia procesada en MATLAB

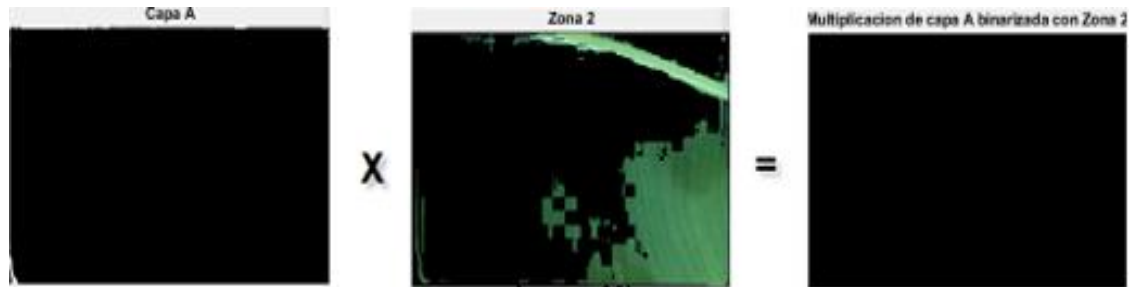
Al realizar la multiplicación de matrices entre la **capa a** del espacio de color **CIEL*a*b**, entre las imágenes a las que se les aplico el método de **k-means**, se obtiene un resultado casi homogéneo. Este resultado, permitirá obtener datos que servirán para caracterizar una imagen enferma con Sigatoka Negra o no. Para este caso, la enfermedad no está presente (*Imagen 28*, *Imagen 29*, *Imagen 30*).

Imagen 28. Multiplicación de **capa a sana** por imagen Zona 1 **k-means**



Fuente propia procesada en MATLAB

Imagen 29. Multiplicación de **capa a sana** por imagen Zona 2 **k-means**



Fuente propia procesada en MATLAB

Imagen 30. Multiplicación de **capa a sana** por imagen Zona 3 **k-means**



Fuente propia procesada en MATLAB

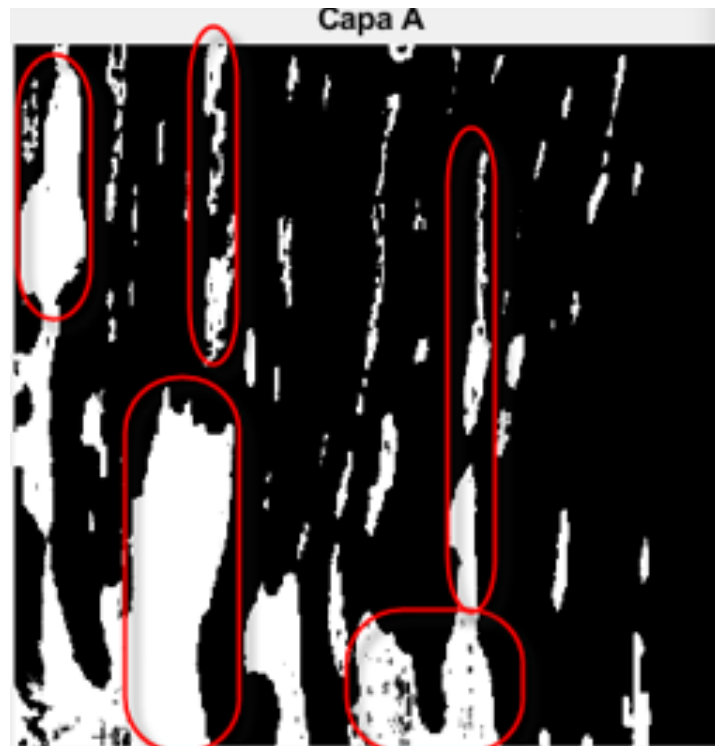
La metodología de lectura de las imágenes es uniforme para ambos casos (imágenes enfermas o sanas) lo que es fundamental pues le permite simplicidad al algoritmo de inteligencia artificial y fácil clasificación, según el parámetro determinado en la evaluación cualitativa dispuesta por el experto al momento de revisión de la base de datos de imágenes por este.

Clasificación Supervisada

Entrenamiento

Hasta este punto, se han aplicado técnicas que permiten segmentar las imágenes, procurando determinar características en ellas que le brinden elementos necesarios al algoritmo de inteligencia artificial para realizar la clasificación deseada. En el proceso de preprocesamiento, se observa que las imágenes procesadas en el espacio de color **CIEL*a*b** en la **capa a** (*imagen 31*) identifica las zonas de la imagen que poseen la enfermedad.

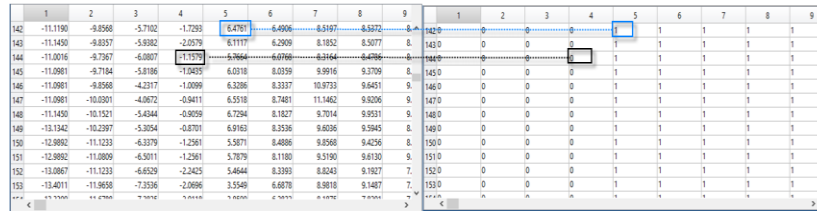
Imagen 31. Capa a hoja de plátano enferma en espacio de color **CIEL*a*b**



Fuente propia procesada en MATLAB

De esta capa se obtendrán las características que enseñarán al algoritmo de inteligencia artificial a detectar la enfermedad en las hojas de plátano. Para ello, se procede a convertir en valores binarios la **capa a** (*imagen 31*) lo que convierte a los valores iguales o menores a cero en el valor del conjunto unitario (0) y los valores mayores que cero, en uno (1) (*Imagen 32*).

Imagen 32. Matriz convertida en valores binarios capa a CIEL*a*b



1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9
142	-11.190	-8.9568	-5.7102	-1.7293	6.4797	-6.4906	-6.5197	-6.5372	0	0	0	0	1	1	1	1	1
143	-11.1430	-8.8937	-5.8982	-2.0579	6.1117	6.2909	8.1852	8.5077	0	0	0	0	1	1	1	1	1
144	-11.0016	-8.7967	-5.8087	-1.9179	5.7864	-6.0768	-8.3164	-8.4786	0	0	0	0	1	1	1	1	1
145	-11.0981	-8.7184	-5.8186	-1.2435	6.0318	8.0359	9.9916	9.3709	0	0	0	0	1	1	1	1	1
146	-11.0981	-8.9568	-4.2317	-1.0099	6.3286	8.3337	10.9733	9.6451	0	0	0	0	1	1	1	1	1
147	-11.0981	-10.0501	-4.0672	-0.9411	6.5518	8.7481	11.1462	9.8206	0	0	0	0	1	1	1	1	1
148	-11.1430	-10.1521	-3.4344	-0.3059	6.7246	8.1827	9.7814	9.9531	0	0	0	0	1	1	1	1	1
149	-13.1342	-10.2397	-3.3054	-0.8791	6.9163	8.3368	9.8036	9.5945	0	0	0	0	1	1	1	1	1
150	-12.9892	-11.1233	-6.3379	-1.2561	5.5871	8.4886	9.8568	9.4236	0	0	0	0	1	1	1	1	1
151	-12.9892	-11.0809	-6.5011	-1.2561	5.7879	8.1180	9.5190	9.6130	0	0	0	0	1	1	1	1	1
152	-13.0867	-11.1233	-6.6529	-2.2425	5.4644	8.3393	8.8243	9.1927	0	0	0	0	1	1	1	1	1
153	-13.4011	-11.9658	-7.3536	-2.0696	3.5549	6.6878	8.8818	9.1487	0	0	0	0	1	1	1	1	1

Fuente propia procesada en MATLAB

Con el resultado de la multiplicación de la **capa a** por cada uno de los clústeres obtenidos con **k-means**, se eliminan las características que no son relevantes en la imagen (zonas sin la enfermedad como el color verde), pues en la **capa a**, las zonas sanas están en negro (0) y al multiplicar este valor por cualquier otro en las demás capas, arrojará resultado en cero (0) y a su vez, las posiciones en la matriz con las características de la enfermedad quedan intactas, pues al multiplicarse por uno (1), su valor no cambia; así, se logra eliminar las características con ausencia de la enfermedad en la imagen, obteniendo solo las posiciones de la matriz (píxeles) con las características pertinentes a la enfermedad (*Imagen 33*).

Imagen 33. Zonas enfermas de la imagen



Fuente propia procesada en MATLAB

Realizado este proceso, se debe determinar qué características entrenaran al algoritmo de inteligencia artificial, pues estas le permitirán a dicho algoritmo una clasificación apropiada respecto a la presencia de la enfermedad de Sigatoka Negra o no en la imagen que se procese. Si bien un experto a simple vista puede determinar si la imagen procesada posee la enfermedad o no, el algoritmo no tiene esta capacidad, por lo que se deben obtener las características de cada clúster, pero para ello, se debe de leer cada posición en la matriz resultante (cada pixel) en el espacio de color **RGB**. Al realizar esta lectura, se procede a obtener las siguientes características del vector resultante (*Imagen .34*):

- Valor máximo y mínimo de la capa que contiene el color rojo
- Valor máximo y mínimo de la capa que contiene el color verde
- Valor máximo y mínimo de la capa que contiene el color azul

Imagen 34. Características iniciales base de datos entrenamiento

Imagen	maximo_rojo	maximo_verde	maximo_azul	minimo_rojo	minimo_verde	minimo_azul
enferma1	254	0	254	108	0	211
enferma2	253	254	254	6	18	13
enferma3	254	254	254	249	34	32
enferma4	254	254	0	35	69	0
enferma5	254	0	254	24	0	81
enferma6	0	0	254	0	0	87
enferma15	254	254	254	27	21	2
enferma18	0	231	243	0	36	5
enferma22	0	0	182	0	0	4
enferma26	84	242	0	56	114	0
enferma28	242	0	0	26	0	0
enferma34	0	254	0	0	65	0
sana2	254	0	0	167	0	0
sana3	0	0	0	0	0	0
sana4	0	242	0	0	54	0
sana5	244	0	0	232	0	0
sana11	0	0	194	0	0	70
sana20	0	0	254	0	0	253
sana25	0	0	0	0	0	0
sana26	0	0	0	0	0	0

Fuente propia

Se determinan estas características iniciales ya que cuando existe la enfermedad de Sigatoka Negra, la imagen contiene mayor cantidad de colores amarillos y cafés, siendo el rojo el más predominante en la imagen. Al contrario de esto, las imágenes sanas poseen un color verde en la totalidad de esta, lo que arroja una diferencia determinante respecto a los colores rojos, permitiendo un punto importante en la diferenciación de las características con lo que el algoritmo logra clasificar una imagen como enferma o sana. También es importante poder realizar una clasificación apropiada con el mínimo de características posibles; así, se logra que el algoritmo procese de manera más eficiente la información que se le ingresa permitiendo un resultado acertado.

De las características anteriormente nombradas, se obtienen otras adicionales (Imagen 35):

- Rango de los valores máximos de las tres capas obtenidas anteriormente
- Rango de los valores mínimos de las tres capas obtenidas anteriormente
- Desviación estándar de los valores máximos de las tres capas obtenidas anteriormente
- Desviación estándar de los valores mínimos de las tres capas obtenidas anteriormente
-
-

Imagen 35. Características adicionales base de datos entrenamiento

Imagen	maximo_rojo	maximo_verde	maximo_azul	minimo_rojo	minimo_verde	minimo_azul	rango_max	rango_min	dev_std_max	dev_std_min
enferma7	254	213	254	53	83	117	41	64	23,6714	32,0208
enferma8	254	0	254	10	0	175	254	175	146,647	98,2768
enferma9	218	254	227	43	9	86	36	77	18,735	38,5876
enferma10	254	254	0	51	76	0	254	76	146,647	38,7341
enferma11	254	251	254	69	24	69	3	45	1,7321	25,9808
enferma12	254	0	254	104	0	39	254	104	146,647	52,5389
enferma13	254	0	254	114	0	29	254	114	146,647	59,2481
enferma14	254	254	0	34	88	0	254	88	146,647	44,3772
enferma16	254	254	228	32	6	7	26	26	15,0111	14,7309
enferma17	254	0	254	43	0	87	254	87	146,647	43,501
enferma19	0	247	243	0	51	82	247	82	141,465	41,4045
enferma20	195	0	215	111	0	98	215	111	118,7785	60,6822
enferma21	0	253	254	0	77	61	254	77	146,3591	40,6325
enferma23	0	178	254	0	66	130	254	130	130,3687	65,0026
enferma24	248	0	233	50	0	71	248	71	139,0551	36,4737
sana1	0	0	254	0	0	254	254	254	146,647	146,647
sana6	0	0	0	0	0	0	0	0	0	0
sana7	0	0	0	0	0	0	0	0	0	0
sana8	0	0	0	0	0	0	0	0	0	0
sana9	0	0	0	0	0	0	0	0	0	0
sana10	40	0	0	8	0	0	40	8	23,094	4,6188
sana12	0	0	0	0	0	0	0	0	0	0
sana13	0	0	0	0	0	0	0	0	0	0
sana14	0	0	0	0	0	0	0	0	0	0
sana15	0	0	0	0	0	0	0	0	0	0

Fuente propia

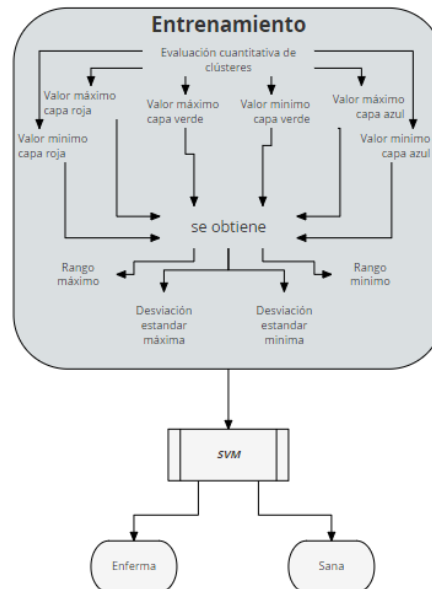
Teniendo en cuenta que el rango es la dispersión entre los valores del vector obtenido, esta, cuando una imagen posee la enfermedad (presencia de colores cafés y amarillos) es alta, por lo que el rango de esta es alto. Cuando la imagen esta sana (presencia del color verde en la totalidad de la imagen) el rango es mínimo (tiende a cero) pues el color es uniforme en la imagen.

Con la desviación estándar se busca reforzar el valor obtenido con el rango pues al realizar esta operación, se ratifica la dispersión entre el promedio obtenido de los valores sometidos a la operación contrastados contra cada uno de ellos.

Todas estas características le permiten al algoritmo de inteligencia artificial determinar si una imagen posee la enfermedad o no. El proceso de entrenamiento es fundamental, pues permite determinar que características son las más apropiadas para este. El proceso de entrenamiento le permiten al algoritmo de inteligencia artificial a través del **SVM** (Support Vector Machine), detectar que imagen posee la enfermedad o no con gran precisión. Estas características se eligieron, pues estas varían bastante de una imagen enferma a una sana lo que le permite al **SVM** clasificar de manera más efectiva.

En este proceso de entrenamiento, se toman 12 imágenes enfermas y 8 imágenes sanas de las bases de datos de imágenes.

Imagen 36. Entrenamiento



Fuente propia

Prueba

Para determinar la efectividad del algoritmo de inteligencia artificial (*anexo 1*), se aplicará el mismo proceso de *entrenamiento*, pero con imágenes distintas a las utilizadas en este, por lo que se analizarán un total de 40 imágenes enfermas y 44 sanas. Con este proceso, se obtienen las características de cada una de las imágenes de la base de datos de prueba (*Imagen 37*), con las cuales se evalúa el desempeño del algoritmo de inteligencia artificial.

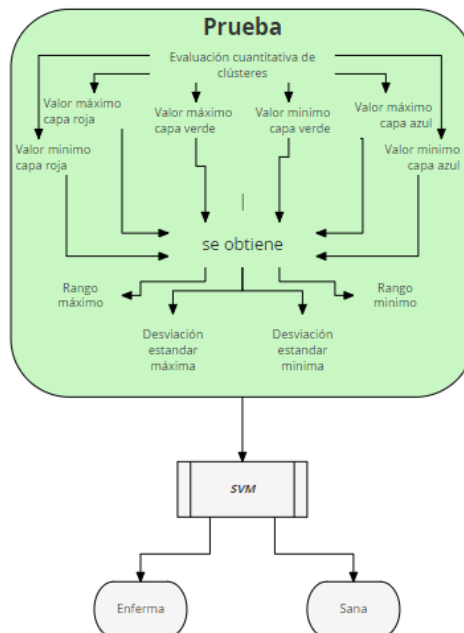
En esta fase se analiza la efectividad del algoritmo de inteligencia artificial, pues este, tendrá que determinar la condición de presencia o ausencia de la enfermedad de Sigatoka Negra en la hoja de plátano por sí solo. El algoritmo al final deberá imprimir un mensaje en el que indique esta condición.

Imagen 37. Segmento base de datos de prueba

Imagen	maximo_rojo	maximo_verde	maximo_azul	minimo_rojo	minimo_verde	minimo_azul	rango_max	rango_min	desv_stand_max	desv_stand_min
enfema7	254	213	254	53	83	117	41	64	23,6714	32,0208
enfema8	254	0	254	10	0	175	254	175	146,647	39,2166
enfema9	218	254	227	43	9	86	36	77	18,735	38,5876
enfema10	254	254	0	51	76	0	254	76	146,647	38,7341
enfema11	254	251	254	63	24	63	3	45	1,7321	25,9808
enfema12	254	0	254	104	0	39	254	104	146,647	52,5389
enfema13	254	0	254	114	0	29	254	114	146,647	59,2481
enfema14	254	254	0	34	88	0	254	88	146,647	48,3772
enfema16	254	254	228	32	6	71	26	26	15,0111	18,7308
enfema17	254	0	254	43	0	87	254	87	146,647	43,501
enfema19	0	247	243	0	51	82	247	82	141,465	41,4045
enfema20	195	0	215	111	0	98	215	111	118,7785	60,6822
enfema21	0	253	254	0	77	61	254	77	146,3591	40,6325
enfema23	0	178	254	0	66	130	254	130	130,3687	65,0026
enfema24	246	0	233	50	0	71	246	71	139,0551	36,4737
sana1	0	0	254	0	0	254	254	254	146,647	146,647
sana6	0	0	0	0	0	0	0	0	0	0
sana7	0	0	0	0	0	0	0	0	0	0
sana8	0	0	0	0	0	0	0	0	0	0
sana9	0	0	0	0	0	0	0	0	0	0
sana10	40	0	0	8	0	0	40	8	23,094	4,6188
sana12	0	0	0	0	0	0	0	0	0	0
sana13	0	0	0	0	0	0	0	0	0	0
sana14	0	0	0	0	0	0	0	0	0	0
sana15	0	0	0	0	0	0	0	0	0	0
sana16	0	0	0	0	0	0	0	0	0	0
sana17	0	0	0	0	0	0	0	0	0	0
sana18	0	0	0	0	0	0	0	0	0	0
sana19	0	0	0	0	0	0	0	0	0	0
sana21	0	0	0	0	0	0	0	0	0	0
sana22	0	0	0	0	0	0	0	0	0	0

Fuente propia

Imagen 38. Procesamiento



Fuente propia

Resultados

EL algoritmo propuesto se aplicó a una base de datos de 104 imágenes de las cuales se tomaron 12 imágenes enfermas y 8 sanas para el proceso de entrenamiento. Para la etapa de prueba se tomaron 40 imágenes enfermas y 44 sanas.

Para el proceso de clasificación con el **SVM**, se utilizó el kernel lineal (*Imagen 39*), pues la cantidad de características es mayor a dos (para este proyecto se utilizaron 10 características) por lo que se debe expandir las dimensiones del espacio original.

Imagen 39. Ejecución de función de **SVM** con kernel lineal en IDE MATLAB

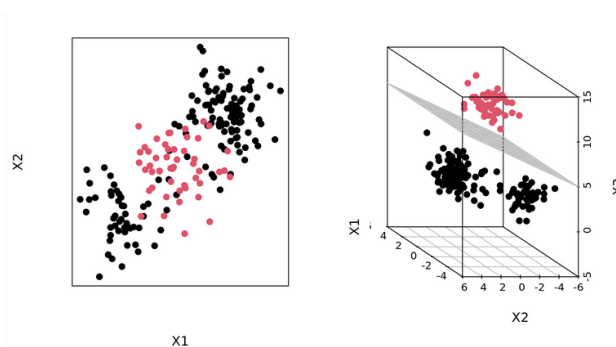
```
maximo_rojo  maximo_verde  maximo_azul  minimo_rojo  minimo_verde  minimo_azul  dev_stand_min
svmStruct = fitcsvm(caracteristicas,experto,'Standardize',true,
'KernelFunction','linear','KernelScale','auto');
```

Fuente propia procesada en MATLAB

El hecho de que dos grupos no sean linealmente separables no implica que no sean separables en un espacio de más dimensiones (*Imagen 40*).

Si los grupos (sanas o enfermas) no son linealmente separables en un espacio original (R^2) no implica que no lo pueda ser en un espacio con más dimensiones (R^n); por ende, añadir más dimensiones puede permitir una separación óptima de los elementos en dos grupos respecto al margen (*Imagen 40*)

Imagen 40. Clasificación de SVM en R^3



Tomada de (Joaquín Amat Rodrigo, n.d.)

Dicho esto, las características seleccionadas ofrecen al **SVM** elementos para la construcción de un hiperplano apropiado según su funcionalidad. Para este caso, las características seleccionadas permiten una separación entre los grupos (enfermas o sanas) muy acertada.

Para el caso del ejemplo en la *imagen 40*, se observa que, al agregar una dimensión extra, es posible realizar una clasificación de los elementos de manera más optima permitiendo ubicar los elementos clasificados en los extremos del hiperplano generado por el **SVM**.

Luego de ejecutado el proceso de clasificación realizado por el algoritmo (*anexo 1*), se procede a llenar una matriz de confusión (*tabla 1*) con el objeto de visualizar el desempeño del algoritmo de inteligencia artificial que se emplea en el **SVM**. La matriz de confusión muestra el nivel de efectividad de la clasificación del algoritmo.

Los **falsos positivos (FP)** son las imágenes que el algoritmo clasifica como enfermas pero realmente están sanas.

Los **falsos negativos (FN)** son las imágenes que el algoritmo clasifica como sanas, pero realmente están enfermas.

Los **verdaderos positivos (VP)** son las imágenes que el algoritmo clasifica como imágenes enfermas y realmente están enfermas.

Los **verdaderos negativos (VN)** son las imágenes que el algoritmo clasifica como sanas y realmente son sanas.

La matriz de confusión nos muestra los niveles de exactitud, precisión, sensibilidad y especificidad de la clasificación del algoritmo.

La exactitud se refiere al porcentaje de predicciones correctas respecto al total de la muestra; es decir, habla de los positivos predichos por el algoritmo versus los positivos reales.

La exactitud se calcula con la siguiente ecuación:

$$exactitud = \frac{VN + VP}{VN + FP + FN + VP}$$

La precisión se refiere al porcentaje de predicciones positivas correctas y se calcula con la siguiente ecuación:

$$precisión = \frac{VP}{VP + FP}$$

La sensibilidad expresa la cantidad de **VP** detectados por el algoritmo correctamente y la especificidad, el porcentaje de **VN** detectados correctamente por el algoritmo. Para hallar estos valores, se aplican las siguientes ecuaciones:

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

$$Especificidad = \frac{VN}{VN + FP}$$

La cantidad de casos correspondientes a estas métricas en la base de datos de prueba son los siguientes:

$$FP = 0 (b)$$

$$FN = 1 (c)$$

$$VP = 39 (a)$$

$$VN = 44 (d)$$

Dicho esto, se procede a evaluar los resultados procesados por el algoritmo de inteligencia artificial en la base de datos de prueba, obteniendo el resultado dispuesto en la *tabla 1*:

Tabla 1. Matriz de confusión

	Negativo	Positivo		
Negativo	A	b	Exactitud	98.8%
Positivo	c	d	Precisión	100%
	Sensibilidad		Especificidad	
	97.5%		100%	

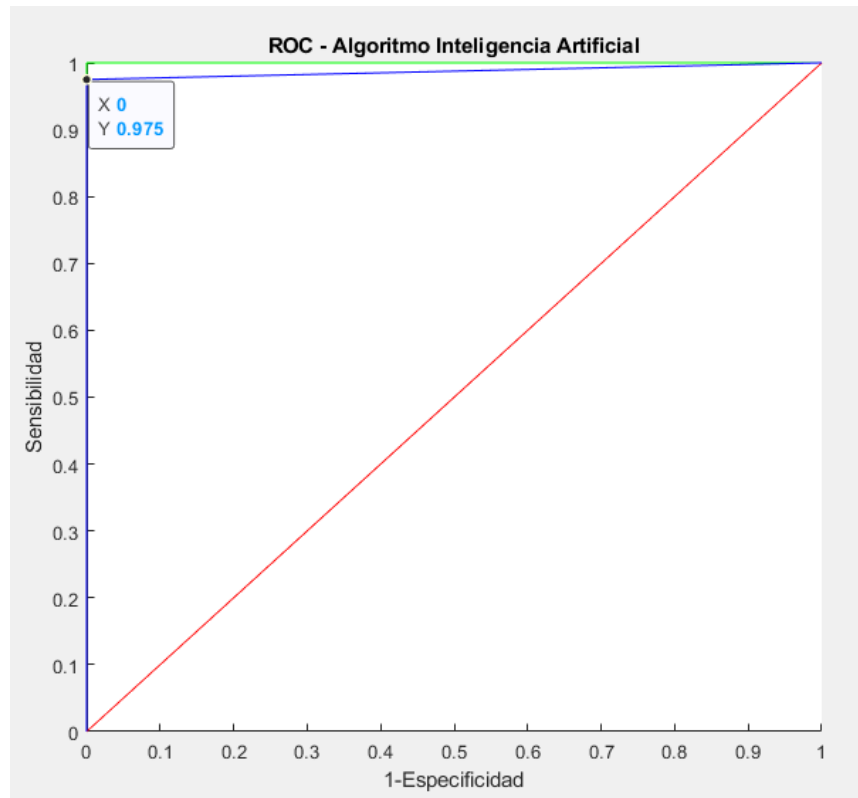
Fuente Propia

Al evaluar el rendimiento de los resultados de la clasificación obtenidos en la matriz de confusión (*tabla 1*), se evidencia que el algoritmo logra una distinción entre las dos clases posibles muy efectiva.

Al representar gráficamente estos resultados (curva ROC (*Imagen 41*)) se observa la relación de la sensibilidad frente a la especificidad respecto al umbral

determinado por el **SVM**, ratificando así la precisión del algoritmo de inteligencia artificial pues la **sensibilidad** que denota la curva ROC está muy por encima de la **línea de no discriminación**.

Imagen 41. Curva ROC Algoritmo Inteligencia Artificial



Fuente propia procesada en MATLAB

Conclusiones

Las técnicas utilizadas para el procesamiento de las imágenes según la metodología propuesta ofrecen información importante para lograr los objetivos propuestos. El espacio de color **CIEL*a*b** permite obtener mucha más información del píxel analizado lo que permitió al **k-means** ubicar en clústeres, las zonas que se deseaban aislar de la imagen analizada y así poder brindar características al algoritmo de inteligencia artificial para la clasificación de una imagen enferma o sana.

El **Vector Support Machines (SVM)** se convierte en un mecanismo altamente eficiente para el algoritmo de inteligencia artificial, pues permite una clasificación muy acertada respecto a los datos que se le proporcionan logrando como resultado, un hiperplano con un margen muy acertado respecto a las dos posibles conclusiones del proceso.

Las métricas obtenidas en la matriz de confusión muestran un alto desempeño del algoritmo de inteligencia artificial logrando una precisión perfecta y una exactitud del 98.8%. Estos valores demuestran que el algoritmo desarrollado en este proyecto (*anexo 1*) posee un alto nivel de exactitud al momento de clasificar las imágenes con presencia de la enfermedad de Sigatoka Negra en la hoja de plátano

Efectivamente se observa que las características utilizadas para entrenar al algoritmo de inteligencia artificial ofrecen información suficiente para que el algoritmo logre una clasificación efectiva de las imágenes procesadas, pues permite al clasificador ubicar en los extremos del hiperplano, las imágenes con presencia de la enfermedad de Sigatoka Negra y, al otro extremo de este, las imágenes que no poseen dicha enfermedad

El porcentaje de sensibilidad (97.5%) del algoritmo de inteligencia artificial permite concluir un alto desempeño de este; sin embargo, al analizar la única falla que este presentó, se observa que no logró dar un **verdadero positivo** a la imagen procesada y esto se debe, a la 'poca' presencia de la enfermedad en la imagen analizada por el algoritmo de inteligencia artificial. La causa determinante en esta falla se presentó en la detección de características de la imagen en el espacio de color **CIEL*a*b** en la **capa a**, por lo que, si se quisiera mejorar el algoritmo, se debe implementar otro tipo de técnica para la obtención de características con imágenes de este tipo, como segmentar las zonas enfermas con la técnica conocida como el **umbral de Otsu**.

La aplicación de este algoritmo de inteligencia artificial en el contexto colombiano podría traer como consecuencia, mayor eficiencia en cultivos de plátano traducida en reducción de costos por insumos químicos, mayor aprovechamiento de los suelos y menor impacto químico sobre estos.

La inclusión de este desarrollo tecnológico impulsa la industria 4.0 en los procesos productivos agropecuarios en Colombia, permitiendo obtener mayor eficiencia apoyados en la automatización de procesos.

Anexo 1

```
enferma=imread('enferma 41.jpeg');
enfermaLAB= rgb2lab(enferma);
figure;
imshow(enferma)
title('Imagen RGB');
```

```
capaA=enfermaLAB(:,:,2);
figure;
imshow(capaA)
title('Capa A');
```

```
capaA_bin=im2bw(capaA);
figure;
imshow(capaA_bin)
title('Capa 2 CIELab binarizada');
```

```
enfermaK = enfermaLAB(:,:,2:3);%
enfermaK = im2single(enfermaK);
colores = 3;
agrupa_colores = imsegkmeans(enfermaK,colores,'NumAttempts',3);

figure;
imshow(agrupa_colores)
title("Colores");
```

```
capaA_x_cluster1 = cluster1 .* uint8(capaA_bin);
figure;
imshow(capaA_x_cluster1)
title("Multiplicacion de capa A binarizada con Zona 1")
capaA_x_cluster2 = cluster2 .* uint8(capaA_bin);
figure;
imshow(capaA_x_cluster2)
title("Multiplicacion de capa A binarizada con Zona 2")
capaA_x_cluster3 = cluster3 .* uint8(capaA_bin);
figure;
imshow(capaA_x_cluster3)
title("Multiplicacion de capa A binarizada con Zona 3")
```

```

for j=1:n
    if capa(i,j) > max_cluster_1(1,k) && capa(i,j) < 255
        max_cluster_1(1,k) = capa(i,j);
        %filmax=i;
        %colmax=j;
    end
    if capa(i,j) > (min_cluster_1(1,k) > 0)
        min_cluster_1(1,k) = capa(i,j);
        %filmin=i;
        %colmin=j;
    end
end
end

maximo_final(1)=max(max_cluster_1);
maximo_final(2)=max(max_cluster_2);
maximo_final(3)=max(max_cluster_3);

minimo_final(1)=min(min_cluster_1);
minimo_final(2)=min(min_cluster_2);
minimo_final(3)=min(min_cluster_3);

rango_max=range(maximo_final);
rango_min=range(minimo_final);

svmStruct = fitcsvm(caracteristicas,experto,'Standardize',true,
'KernelFunction','linear','KernelScale','auto');

prueba1=[maximo_final(1),maximo_final(2),maximo_final(3),minimo_final(1),
minimo_final(2),minimo_final(3),rango_max,rango_min,desv_max,desv_min];

clasificacion=predict(svmStruct,prueba1);
if clasificacion == 1
    disp('La hoja posee sigatoka')
else
    disp('La hoja esta sana')
end
end

```

Nota: algoritmo implementado para detección de Sigatoka Negra

Bibliografía

- ARISTIDIS Likas, VLASSIS Nikos, VERBEEK Jakob. *The global k-means clustering algorithm*. {En línea}. {10 septiembre 2020} disponible en: <https://www-sciencedirect-com.bibliotecavirtual.unad.edu.co/science/article/abs/pii/S0031320302000602>
- ARZANLOU Mahdi, ABELN Edwin, GERT Kema., WAALWIJK Cees, CARRIER Jean., DE VRIES Ineke, GUZMAN Mauricio y CROUS Pedro. Molecular diagnostics for the Sigatoka disease complex of banana. *Phytopathology*. {En línea}. {20 junio 2020} disponible en: <https://doi.org/10.1094/PHYTO-97-9-1112>
- CERDA Jaime y CIFUENTES Lorena. Uso de curvas ROC en investigación clínica. Aspectos teórico-prácticos. {En línea}. {22 junio 2020} disponible en: https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0716-10182012000200003
- ISLAM Monzurul, DINH Ahn, WAHID Khan y BHOWMIK Panka. Detection of potato diseases using image segmentation and multiclass support vector machine. *Canadian Conference on Electrical and Computer Engineering*. {En línea}. {28 junio 2020} disponible en: <https://doi.org/10.1109/CCECE.2017.7946594>
- RODRIGO Joaquín. *Máquinas de Vector Soporte (Support Vector Machines, SVMs)*. {En línea}. {10 agosto 2020} disponible en: https://www.cienciadedatos.net/documentos/34_maquinas_de_vector_soporte_support_vector_machines
- KITPO Nuttakarn y MASAHIRO Inoue. Early rice disease detection and position mapping system using drone and IoT architecture. *Proceedings*. {En línea}. {20 agosto 2020} disponible en: <https://doi.org/10.1109/SEATUC.2018.8788863>
- MINOLTA Konica. Entendiendo El Espacio de Color CIE L*A*B*. {En línea}. {21 agosto 2020} disponible en: <https://sensing.konicaminolta.us/mx/blog/entendiendo-el-espacio-de-color-cie-lab/>
- NIXON Mark. *Feature Extraction and Image Processing*. {En línea}. {22 agosto 2020} disponible en: <http://eds.a.ebscohost.com.bibliotecavirtual.unad.edu.co/eds/detail/detail?vid=0&sid=aeb884fb-513a-43ab-bd5e-03b45602da0f%40sessionmgr4008&bdata=Jmxhbm9ZXMmc2l0ZT1lZHMtbGl2ZS5y29wZT1zaXRI#AN=204193&db=nlebk>
- ORESTES. *Optimization & Machine Learning*. {En línea}. {2 septiembre 2020}

disponible en: <https://optimachine.blogspot.com/2008/03/qu-son-las-svm.html>

RODRÍGUEZ Rieiro. *Segmentación de imágenes en color y textura basada en el modelo de los bordes locales*. {En línea}. {3 septiembre 2020} disponible en: bibing.us.es/proyectos/abreproy/11527/fichero/2%252FCapítulo+2.pdf

SINGH Vijai, VARSHA y MISRA. Detection of unhealthy region of plant leaves using image processing and genetic algorithm. *Conference Proceeding - 2015 International Conference on Advances in Computer Engineering and Applications*. {En línea}. {5 septiembre 2020} disponible en: <https://doi.org/10.1109/ICACEA.2015.7164858>

THANGADURAI y PADMAVATHI. Computer vision image enhancement for plant leaves disease detection. *Proceedings - 2014 World Congress on Computing and Communication Technologies*. {En línea}. {10 septiembre 2020} disponible en: <https://doi.org/10.1109/WCCCT.2014.39>