



INSTITUTO TECNOLÓGICO DE CD. GUZMÁN  
PROGRAMA DE MESTRÍA EN CIENCIAS DE LA  
COMPUTACIÓN

**TESIS**

TEMA:

*“Identificación de la enfermedad Botrytis cinérea  
en la granada por medio del procesamiento de  
imágenes”*

QUE PARA OBTENER EL GRADO DE:

*MAESTRO EN CIENCIAS DE LA COMPUTACIÓN*

PRESENTA:

***VERONICA ESMERALDA MIRAMONTES VARO***

DIRECTORES:

*DRA. MARÍA GUADALUPE SÁNCHEZ CERVANTES*

*DR. SERGIO SANDOVAL PÉREZ*

*MC.FELIPE ALFONSO ORDOÑEZ GARCIA*



TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Cd. Guzmán

Cd. Guzmán, Jal. a 09/Agosto/2018

Oficio No. DEPI/61/18

ASUNTO : AUTORIZACIÓN DE IMPRESIÓN

**C. VERÓNICA ESMERALDA MIRAMONTES VARO**  
**N.C. M16290035**

En cumplimiento con el documento normativo de las disposiciones para la operación de estudios de posgrado del Tecnológico Nacional de México y con base en la aprobación del Comité Tutorial comisionado para su revisión; la División de Estudios de Posgrado e Investigación le otorga la autorización de impresión de su trabajo de tesis intitulado:

**"IDENTIFICACIÓN DE LA ENFERMEDAD BOTRYTIS CINÉREA EN LA GRANADA POR MEDIO DEL PROCESAMIENTO DE IMÁGENES"**

dirigido por la **Dra. María Guadalupe Sánchez Cervantes**, desarrollado como requisito parcial para la obtención del grado de Maestro en Ciencias de la Computación, de acuerdo al plan de estudios MCCOM-2011-05.

Sin otro asunto en particular, quedo de usted.

**ATENTAMENTE**

  
**DR. HUMBERTO BRACAMONTES DEL TORO**  
**JEFE DE DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN**



C.p. Archivo



Av. Tecnológico No. 100 C.P. 49100 A.P. 150  
 Cd. Guzmán, Jal. Tel. Conmutador (341) 5 75 20 50  
[www.itcg.edu.mx](http://www.itcg.edu.mx)



---

# Dedicatoria

Esta tesis es en honor a mi hija Yaretzi Erandi, por el amor, la comprensión y paciencia que ha tenido durante estos años.

---

# Agradecimientos

**A mi hija:** por todo el esfuerzo que ha hecho por mí y la fortaleza que me dio para continuar con mi superación profesional aún cuando existían momentos complicados.

**A mi familia:** a mis padres, quiero agradecer todo el apoyo y ayuda que me brindaron para seguir adelante, junto con mi hermano Alfonso, que siempre me apoyaron con el cuidado de mi hija y nunca la dejaron sola. Gracias por sus consejos y buenos deseos.

**A mis amigos y maestros:** mi agradecimiento y aprecio, a mis amigos por brindarme su ayuda y apoyo en todo momento, a mis maestros por su enseñanza, paciencia, guía, y por haber confiado en mí.

**A la empresa GEGEMA, S.P.R DE R.L:** por brindarme su calidez y confianza al dejarme entrar a su empresa, para realizar la investigación de campo, por facilitarme todos los recursos y sus conocimientos sobre el manejo de la granada en su empresa.

## Resumen

*Existen diferentes enfermedades que se presentan en las granadas, la Botrytis cinérea es una de ellas. Esta enfermedad afecta la producción de los derivados de la granada, ya que puede llegar a contaminar el producto final y por consiguiente acelerar su putrefacción. Debido a lo anterior, el objetivo del presente trabajo de tesis es identificar la enfermedad de la Botrytis cinérea en la granada a través del procesamiento de imágenes, en pos-cosecha, la etapa de selección previa a la producción. Para llevar a cabo lo anterior, se pre-procesa la imagen de entrada, la cual es pasada por varios procesos de procesamiento de imágenes para poder elegir el que brinde mejores resultados en la clasificación, estos métodos fueron método de Otsu, binarización, crecimiento de regiones, K-means clustering, punto interior a un triángulo, aplicación de operaciones morfológicas y base de datos de píxeles. Se usaron 3 clasificadores, TensorFlow, SVM y Random Forest. Donde los resultados favorables para TensorFlow fueron las imágenes con el filtro Gaussiano, segmentación manual y segmentación con K-means. Para el Random Forest la extracción de características estadísticas y para SVM la extracción de características con transformadas wavelet de las imágenes con el filtro Gaussiano.*

## Abstract

*There are different diseases that occur in pomegranates, Botrytis cinerea is one of them. This disease affects the production of the derivatives of the pomegranate, as it can contaminate the final product and consequently accelerate its putrefaction. Due to the above, the objective of this thesis work is to identify the disease of Botrytis cinerea in the pomegranate through the processing of images, in post-harvest, the stage of selection prior to production. To carry out the above, the input image is pre-processed, which is passed through several image processing processes in order to choose the one that provides the best results in the classification, these methods were Otsu method, binarization, growth of regions, K-means clustering, interior point to a triangle, application of morphological operations and pixel database. We used 3 classifiers, TensorFlow, SVM and Random Forest. Where the favorable results for TensorFlow were the images with the Gaussian filter, manual segmentation and segmentation with K-means. For the Random Forest the extraction of statistical characteristics and for SVM the extraction of characteristics with wavelet transforms of the images with the Gaussian filter.*

---

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Definición del problema . . . . .	2
1.3. Justificación . . . . .	3
1.4. Objetivos . . . . .	4
1.4.1. Objetivo general . . . . .	4
1.4.2. Objetivos específicos . . . . .	4
1.5. Hipótesis . . . . .	5
<b>2. Estado del arte</b>	<b>6</b>
2.1. Introducción . . . . .	6
2.2. Antecedentes . . . . .	7
2.2.1. Granada . . . . .	7
2.2.2. Fresas . . . . .	9
2.2.3. Uvas . . . . .	9
2.2.4. Cítricos . . . . .	10
2.2.5. Varios (uva, manzana, granada) . . . . .	11
2.2.6. Manzana . . . . .	12
2.2.7. Café . . . . .	13
<b>3. Fundamento Teórico</b>	<b>17</b>
3.1. Introducción . . . . .	18
3.2. Botrytis cinérea . . . . .	18
3.3. Inteligencia artificial . . . . .	20
3.4. Visión artificial . . . . .	21
3.4.1. Procesamiento de imágenes . . . . .	21
3.4.2. Filtros y operaciones morfológicas . . . . .	23
3.4.3. Imágenes a color . . . . .	27
3.4.4. Métodos de segmentación . . . . .	30
3.4.5. Extracción de características estadísticas . . . . .	36
3.4.6. Transformación del dominio Wavelets . . . . .	38
3.4.7. Reconocimiento de patrones . . . . .	40
3.5. Redes neuronales . . . . .	41
3.5.1. Neurona biológica . . . . .	41

3.5.2.	Neurona artificial . . . . .	43
3.5.3.	Tipos de función de activación . . . . .	44
3.5.4.	Arquitecturas fundamentales . . . . .	46
3.5.5.	Red neuronal convolucional . . . . .	50
3.6.	TensorFlow Network de Google . . . . .	52
3.7.	Máquina de Vectores de Soporte (SVM) . . . . .	54
3.8.	Clasificador Random Forest . . . . .	55
3.9.	Herramientas de programación . . . . .	57
3.9.1.	Python . . . . .	57
3.9.2.	Matlab . . . . .	57
3.9.3.	Weka . . . . .	57
<b>4.</b>	<b>Metodología</b>	<b>58</b>
4.1.	Introducción . . . . .	59
4.2.	Tipo de Investigación . . . . .	60
4.3.	Universo, población o unidades de análisis . . . . .	60
4.4.	Adquisición de la imagen . . . . .	61
4.5.	Pre-procesamiento de la imagen . . . . .	62
4.5.1.	Obtener área de interés . . . . .	62
4.5.2.	Filtros y operaciones morfológicas . . . . .	62
4.5.3.	Realce de contrastes . . . . .	64
4.6.	Segmentación . . . . .	65
4.6.1.	Segmentación manual . . . . .	65
4.6.2.	Binarización . . . . .	65
4.6.3.	Método de Otsu . . . . .	66
4.6.4.	Crecimiento de regiones . . . . .	67
4.6.5.	Método matemático de punto interior a un triángulo . . . . .	68
4.6.6.	Generación de base de datos de pixeles enfermos . . . . .	72
4.6.7.	K-means clustering . . . . .	73
4.7.	Extracción de características estadísticas . . . . .	74
4.8.	Extracción de características de frecuencia de dominio con transformadas wavelets . . . . .	75
4.9.	TensorFlow Network de Google . . . . .	75
4.9.1.	Instalar TensorFlow . . . . .	76
4.9.2.	Entrenamiento y clasificación con TensorFlow . . . . .	76
4.10.	Clasificador SVM y Random Forest . . . . .	77

<b>5. Resultados</b>	<b>83</b>
5.1. Introducción . . . . .	83
5.2. Adquisición de la imagen . . . . .	84
5.3. Pre-procesamiento de la imagen . . . . .	84
5.3.1. Obtener área de interés . . . . .	84
5.3.2. Filtros y operaciones morfológicas . . . . .	84
5.3.3. Realce de contrastes . . . . .	87
5.4. Segmentación . . . . .	87
5.4.1. Segmentación manual . . . . .	87
5.4.2. Binarización . . . . .	88
5.4.3. Método de Otsu . . . . .	89
5.4.4. Crecimiento de regiones . . . . .	91
5.4.5. Método matemático de punto interior a un triángulo . . . . .	91
5.4.6. Generación de base de datos de pixeles enfermos . . . . .	92
5.4.7. K-means clustering . . . . .	92
5.5. Extracción de características estadísticas . . . . .	101
5.6. Extracción de características en el dominio de la frecuencia con transformadas Wavelets . . . . .	102
5.7. Clasificación de la Botrytis cinérea . . . . .	102
5.8. Clasificación con SVM y Random Forest . . . . .	103
<b>6. Conclusiones</b>	<b>105</b>
<b>A. Productos obtenidos</b>	<b>107</b>
A.1. Estadía profesional . . . . .	107
A.2. Ponencia de un artículo en la Academia Journals Morelia 2018 . . . . .	108
<b>B. Actividades de apoyo</b>	<b>109</b>
B.1. Sistema Administrativo para la empresa GEGEMA S.P.R. DE R.L. . . . .	109
<b>C. Glosario</b>	<b>110</b>
<b>Bibliografía</b>	<b>118</b>



---

# Índice de figuras

3.1. Granada agria con hongo. . . . .	19
3.2. Granada visiblemente sana, pero tiene Botrytis cinérea por dentro. . . . .	19
3.3. Granada dulce con grieta, piel con corteza contaminada y arilo sano. . . . .	20
3.4. Granada dulce con grieta, arilos con hongo. . . . .	20
3.5. Representación del espacio de color RGB como cubo unitario. . . . .	28
3.6. Histograma para RGB. a) Imagen original, b) Histogramas concatenados de los planos RGB. . . . .	29
3.7. Modelo de color HSV. . . . .	30
3.8. a) esquema del algoritmo de filtrado por filas y columnas para la descomposición de la imagen original en cuatro subimágenes; b) simbología utilizada. . .	40
3.9. Representación de una neurona biológica. . . . .	42
3.10. Sinapsis biológica. . . . .	43
3.11. Representación de una neurona artificial tipo McCulloch-Pitts. . . . .	43
3.12. Función escalón o umbral . . . . .	44
3.13. Función lineal a tramos . . . . .	45
3.14. Función sigmoidea . . . . .	45
3.15. Red neuronal unicapa . . . . .	47
3.16. Red neuronal multicapa . . . . .	47
3.17. Red neuronal convolutiva . . . . .	50
3.18. Convolución . . . . .	51
3.19. Max pooling . . . . .	51
3.20. Conección de las neuronas . . . . .	52
3.21. Estructura de Inception V3 . . . . .	53
3.22. Hiperplano óptimo de separación entre dos clases en el espacio transformado.	54
4.1. Metodología utilizada para la identificación de la Botrytis en la granada. . . .	59
4.2. Cabina fotográfica de luz blanca . . . . .	62
4.3. Gráficas de los pixeles de los planos $R$ y $G$ . . . . .	68
4.4. Gráficas de los pixeles de los planos $R$ y $B$ . . . . .	68
4.5. Gráficas de los pixeles de los planos $G$ y $B$ . . . . .	69
4.6. Gráficas de los pixeles de los planos $S$ y $V$ . . . . .	69
4.7. Gráficas de los pixeles de los planos $a^*$ y $b^*$ . . . . .	69
4.8. Trazo de rectángulos en el espacio de color RGB en los planos $R$ y $G$ . . . . .	70
4.9. Trazo de rectángulos en el espacio de color RGB en los planos $R$ y $B$ . . . . .	70

4.10. Trazo de rectángulos en el espacio de color RGB en los planos G y B. . . . .	70
4.11. Trazo de cuadrados en el espacio de color HSV en los planos S y V. . . . .	71
4.12. Trazo de cuadrados en el espacio de color CIELab en los planos a* y b*. . . . .	71
4.13. Punto interior a un triángulo. . . . .	72
4.14. Clasificación con Weka. . . . .	80
4.15. Seleccionar SVM. . . . .	81
4.16. Seleccionar Random Forest . . . . .	81
4.17. Clasificación con Weka. . . . .	82
5.1. Obtener área de interés. a) imagen de granada con Botrytis, b) reducción al área de interés. . . . .	84
5.2. Aplicación de filtros. a) imagen original, b) Gauss, c) media, d) mediana. . .	85
5.3. Aplicación de filtros. e) Laplaciant, f) Prewitt, g) Sobel, h) Gauss en escala de grises. . . . .	85
5.4. Aplicación de filtros. i) media en escala de grises, j) Laplaciant en escala de grises, k) Prewitt en escala de grises y l) Sobel en escala de grises. . . . .	85
5.5. Evaluación MAE para los filtros de Gauss, Media y Mediana. . . . .	86
5.6. Evaluación PSNR para los filtros de Gauss, Media y Mediana. . . . .	86
5.7. Operaciones morfológicas. a) imagen original, b) binarización, c) ecualización del histograma, d) dilatación, e) erosión, f) dilatación – erosión, g) erosión – dilatación. . . . .	86
5.8. Realce de contraste. . . . .	87
5.9. Segmentación manual de la granada. a) Enfermedad de la Botrytis, b) Parte sin Botrytis. . . . .	87
5.10. Segmentación por binarización. a) imágenes originales, b) binarización acep- table, b) imágenes originales, d) binarización rechazada. . . . .	88
5.11. Resultado del primer algoritmo del método de Otsu. . . . .	89
5.12. Resultados del segundo algoritmo del método de Otsu. . . . .	90
5.13. Crecimiento de regiones. . . . .	91
5.14. Resultados del método de punto interior a un triángulo. a) imagen original, b) planos R y G, c) planos R y B, d) planos G y B de RGB, e) planos H y S del espacio de color HSV y f) planos a* y b* del espacio de color CIELab. . . . .	91
5.15. Búsqueda en base de datos de pixeles. a) imagen original, b) imagen segmentada.	92
5.16. Resultado de la operación morfológica de la suma de diferencia absoluta. . .	93
5.17. Resultado de la operación morfológica de distancia euclideana. . . . .	93
5.18. Resultado de la operación morfológica de coseno. . . . .	94
5.19. Resultado de la operación morfológica de la suma de diferencia absoluta. . . .	94

5.20. Resultado de la operación morfológica de distancia euclideana. . . . .	95
5.21. Resultado de la operación morfológica de coseno. . . . .	95
5.22. Resultado de la operación morfológica de la suma de diferencia absoluta. . . .	96
5.23. Resultado de la operación morfológica de distancia euclideana. . . . .	96
5.24. Resultado de la operación morfológica de coseno. . . . .	96
5.25. Resultado de la operación morfológica de correlación. . . . .	97
5.26. Resultado de la operación morfológica de la suma de diferencia absoluta. . . .	97
5.27. Resultado de la operación morfológica de distancia euclideana. . . . .	98
5.28. Resultado de la operación morfológica de coseno. . . . .	98
5.29. Resultado de la operación morfológica de la suma de diferencia absoluta. . . .	99
5.30. Resultado de la operación morfológica de distancia euclideana. . . . .	99
5.31. Resultado de la operación morfológica de coseno. . . . .	99
5.32. Resultado de la operación morfológica de la suma de diferencia absoluta. . . .	100
5.33. Resultado de la operación morfológica de distancia euclideana. . . . .	100
5.34. Resultado de la operación morfológica de coseno. . . . .	100
5.35. Resultado de la operación morfológica de correlación. . . . .	101
5.36. Extracción de características estadísticas. . . . .	101
5.37. Extracción de características en el dominio de la frecuencia con transformadas Wavelets. . . . .	102
A.1. Memorias Academia JOurnals Morelia 2018 . . . . .	108

---

# Índice de tablas

3.1. Ejemplos de vectores base Haar y Daubechies para filtrado Paso Bajo y Paso Alto. . . . .	39
3.2. Bandas de frecuencia en una TDW con cuatro bandas. . . . .	40
5.1. Resultados con TensorFlow. . . . .	103
5.2. Resultados SVM y Random Forest. . . . .	104

---

---

## CAPÍTULO 1

---

# Introducción

### Contenido

---

<b>1.1. Motivación</b> . . . . .	<b>1</b>
<b>1.2. Definición del problema</b> . . . . .	<b>2</b>
<b>1.3. Justificación</b> . . . . .	<b>3</b>
<b>1.4. Objetivos</b> . . . . .	<b>4</b>
1.4.1. Objetivo general . . . . .	4
1.4.2. Objetivos específicos . . . . .	4
<b>1.5. Hipótesis</b> . . . . .	<b>5</b>

---

### 1.1. Motivación

La granada es un fruto que aporta muchos beneficios para la salud, su masa está compuesta por 80 % de agua, es una fruta baja en calorías, que contiene potasio, vitamina A, vitamina C y vitamina B9. El tamaño de una granada puede oscilar entre el de una naranja a una toronja. Las granadas también están expuestas a enfermedades, estas enfermedades son, la podredumbre del fruto que puede ser causada por dos tipos de hongos *Botrytis cinérea* y *Alternaria spp*, otra enfermedad es el Cribado (*Clasterosporium carpophilum*) la cual se manifiesta con manchas negras en la piel del fruto. La podredumbre del fruto es la enfermedad más importante que afecta la producción de granada. La *Botrytis cinérea* es uno de los hongos que la causan el cual entra por el pistilo y va extendiéndose dentro del fruto hasta podrir los granos completamente e incluso la corteza. La *Alternaria spp* es otro hongo que pudre la granada, un sintoma es la presencia de micelio negro al interior de la fruta atacando los arilos, comenzando por el calyx, mientras que la corteza es dura y correosa pero con apariencia saludable. Ambas enfermedades son favorecidas por las lluvias [Mondragón J., 2008]. Actualmente la tarea de seleccionar la granada en el estado óptimo para ser procesada y sacar el mayor provecho es ardua, debido a que se elabora de manera manual y a criterio

del ser humano. Ya que no se cuenta con un sistema eficiente para la selección de esta fruta y el almacenamiento de la información, se realizó una aplicación para la identificación de la enfermedad de la Botrytis cinérea en la granada por medio del reconocimiento de imágenes de manera eficaz, esta aplicación estará vinculada a la empresa GEGEMA, S.P.R DE R.L dedicada al cultivo y procesamiento de la granada.

Esta tesis está estructurada y organizada de la siguiente manera:

Capítulo 2 ESTADO DEL ARTE: en este capítulo se presentan los proyectos existentes, en los cuales se trabajó sobre el procesamiento de imágenes para la detección de enfermedades en las frutas.

Capítulo 3 FUNDAMENTO TEÓRICO: en este capítulo se describen todos los métodos, técnicas y herramientas a utilizar para el desarrollo del proyecto de investigación.

Capítulo 4 METODOLOGÍA: en este capítulo se detalla el desarrollo del proyecto, desde la adquisición de las mismas hasta la clasificación de las mismas.

Capítulo 5 RESULTADOS: en este capítulo se muestran los resultados obtenidos en cada uno de los procesos descritos en la metodología.

Capítulo 6 CONCLUSIONES: en este capítulo se describen las conclusiones obtenidas del proyecto.

## 1.2. Definición del problema

A pesar de que la detección de las enfermedades en la granada es un campo de investigación presente, se considera una tarea difícil el obtener un patrón detallado de las enfermedades debido a la variabilidad de cada fruto a procesar. Las empresas dedicadas a la producción de los derivados de la granada, en la región sur de Jalisco no cuentan con un sistema automatizado para la selección de la granada. Se han llevado a cabo distintas investigaciones que detectan e identifican enfermedades en varios tipos de frutos, tales como: fresas, uvas, café, manzanas, cítricos, entre otros. En la granada se presentan varias enfermedades, entre las cuales se encuentran: la Botrytis cinérea, Alternaria, Cercospora y Anthracnose. En la literatura se reportan pocas investigaciones dirigidas al fruto de la granada. Es por ello que

se realizó la identificación de la enfermedad de la Botrytis cinérea, la cual beneficie a las empresas que requieren de este servicio. GEGEMA, S.P.R. DE R.L. es una de las empresas que se dedica a la producción de jugo, agua, mermelada y licor de granada. Los procedimientos utilizados para la producción de sus productos se realizan de manera manual, en la primer etapa de corte de coronilla se empieza a realizar la pre-selección de las granadas para procesar, donde solo se descartan granadas podridas que son fácil de detectar, y dependiendo de la capacitación de las personas no siempre se separa toda la fruta mala. Enseguida se pasa a la mesa de corte, donde se realiza la selección final de la granada, el inconveniente de esta etapa es que al realizar un corte a las granadas podridas sueltan jugo infectado con Botrytis, y se puede contaminar el área de corte, donde es posible que el producto bueno también se contamine. El personal con experiencia puede identificar en su mayoría granadas enfermas, pero no todo el personal es experto. Con la finalidad de aportar una herramienta innovadora que apoye a las empresas en la selección de la granada en el procesamiento, se propone llevar a cabo la identificación de la enfermedad de la Botrytis cinérea en la granada a través de procesamiento de imágenes. La eficiencia en la selección de la granada se puede medir obteniendo el total del peso en kilogramos de las granadas con la enfermedad de Botrytis que llegan a la mesa de corte. Dando respuesta a la siguiente pregunta. ¿Se podrá identificar la Botrytis cinérea en la granada mediante el procesamiento de imágenes?

### 1.3. Justificación

Con el presente proyecto de investigación, los procesos de selección se realizarán de manera rápida y eficiente, la fruta seleccionada será la de mejor calidad y se desperdiciará menos ya que las granadas con Botrytis cinérea serán descartadas antes de ser procesadas, evitando la contaminación del producto. La elaboración del proyecto de tesis será una innovación que aportará beneficios a la empresa GEGEMA, S.P.R. DE R.L. debido a que la empresa no cuenta con un sistema computacional auxiliar en la administración de la empresa y en el procesamiento de la granada. Así mismo en la región sur de Jalisco no se ha encontrado en el mercado un sistema administrativo o de producción para el sector agropecuario dedicado a los derivados de la granada, que cuente con selección pos cosecha de granada por medio del reconocimiento de imágenes, por lo que este proyecto de tesis será pionero para el sector de productores de los derivados de la granada. Actualmente el empleado debe revisar de manera rápida que no vayan granadas con Botrytis cinérea, se separan en un contenedor las que aparentan estar en buen estado y en otro las que están podridas, enseguida, se le

realizan cortes a la granada, y se revisa que no esté podrida (que no contenga Botrytis ni Alternaria). A la mesa de corte llega mucha granada podrida, exponiendo a la fruta sana a la contaminación de los hongos. La identificación de la Botrytis cinérea en la granada, se llevó a cabo con la finalidad de aportar a la ciencia una herramienta innovadora, que permita disminuir la cantidad de fruta podrida que se procesa en las empresas debido a los errores causados por la subjetividad por parte del empleado o cambios de criterios presentados por realizar de manera rápida el trabajo, así como mejorar los resultados de la producción, obteniendo menos pérdidas en los derivados de la granada.

## 1.4. Objetivos

### 1.4.1. Objetivo general

Identificar la enfermedad de Botrytis cinérea en la granada antes de llegar a la mesa de corte, a través del procesamiento de imágenes, para llevar a cabo una buena selección de la fruta en la producción de los derivados.

### 1.4.2. Objetivos específicos

Para poder llevar a cabo el objetivo general, se plantean los siguientes objetivos específicos:

- Revisar el estado del arte de la identificación de la enfermedad de la Botrytis cinérea en las frutas.
- Definir y realizar la obtención de la imagen.
- Diseñar y programar el pre-procesamiento de imagen.
- Realizar la segmentación de las imágenes.
- Obtener la extracción de las características de las granadas.
- Elegir una red neuronal para el entrenamiento y desarrollar el clasificador para la identificación de la Botrytis en las granadas.
- Realizar pruebas experimentales del proyecto.



## 1.5. Hipótesis

Con el desarrollo de la aplicación para la identificación de la enfermedad de Botrytis cinérea en la granada por medio del procesamiento de imágenes la empresa llevará el control de selección con mayor eficiencia.

---

---

## CAPÍTULO 2

---

# Estado del arte

### Contenido

---

<b>2.1. Introducción</b>	<b>6</b>
<b>2.2. Antecedentes</b>	<b>7</b>
2.2.1. Granada	7
2.2.2. Fresas	9
2.2.3. Uvas	9
2.2.4. Cítricos	10
2.2.5. Varios (uva, manzana, granada)	11
2.2.6. Manzana	12
2.2.7. Café	13

---

### 2.1. Introducción

En los últimos años se han llevado a cabo investigaciones de métodos no destructivos (métodos que no tienen contacto directo con las frutas), con la finalidad de identificar enfermedades y otras características, como hongos, bacterias y maduración en diversas frutas, a través del procesamiento de imágenes y el uso de visión artificial. Estas técnicas son utilizadas para la evaluación de la calidad en los frutos y han tomado mucha importancia debido a que son rápidas, consistentes y objetivas.

## 2.2. Antecedentes

### 2.2.1. Granada

En [Deshpande T., 2014] se propone una metodología para identificar la bacteria Blight en hojas, tallo y fruto de la planta de la granada. Para ello se utilizaron técnicas de pre-procesamiento de imágenes como la erosión y dilatación para remover las sombras. La segmentación se realizó con K-means clustering con un  $k=2$ , una vez obtenida la enfermedad se calcula el porcentaje del área afectada respecto al tamaño de la hoja o fruto analizado. En la investigación de [Pawar, 2012], se detectan defectos de la piel en granadas utilizando las características del color de las texturas y la DTW (Discrete Wavelet Transform / Transformada Discreta de Wavelets). Esta transformada se utiliza para calcular las características estadísticas, 3 características se calcularon para R, G y B. Las mejores características se utilizaron como entrada para el clasificador de la Máquina de Vector de Soporte (SVM) y se realizaron pruebas para identificar el mejor modelo de clasificación. Las características que mostraron resultados óptimos fueron media (99 %), varianza (99,80 %), sombra del grupo (99,88 %), prominencia del grupo (99,88 %), intensidad media (99,81 %). El modelo HSI (Hue, Saturation, Intensity / Matiz, Saturación, Intensidad) se utiliza para obtener características de textura de color y transformación de onda discreta. Se obtuvieron muestras de piel infectada y no infectados. Para cada muestra se recorta la imagen, se obtiene el modelo HSI a partir de la cual se calculan 12 características de textura. Simultáneamente, cada muestra se transforma en componentes R, G, B para obtener DWT de cada capa y se calculan tres características estadísticas: media, varianza y desviación estandar. Las muestras consisten en imágenes de quemaduras solares, Alternarías y enfermedades de Cercospora. Se realiza una base de datos de imágenes con piel infectada y otra de imágenes de piel sana, todas las imágenes fueron convertidas a RGB (Red, Green, Blue / Rojo, Verde, Azul) para obtener DWT de cada capa y obtener características estadísticas. Una vez realizado esto se procede a crear una SVM para la clasificación de los datos, la detección de defectos en la piel de las granadas tiene un radio de aceptación arriba del 99 %.

En la india se realizó un sistema de reconocimiento de imágenes, para la identificación de enfermedades en el fruto de la granada [Mundokar N., 2017], donde se utiliza análisis de color y OpenCv con E-NOSE (Electronic nose / Nariz electrónica), con esto las imágenes se clasifican por medio del: color, textura, morfología, estructura del agujero y el olor, esta última se detecta por medio de un sensor llamado E-NOSE.

[Dhakate M., 2015] propone un procesamiento de imagen y métodos de red neuronal para la detección de enfermedades y la clasificación de la fruta de granada, así como las hojas ya que se ven afectadas por diversas enfermedades causadas por hongos, bacterias y las condiciones climáticas. Las imágenes en color se pre-procesan y se someten a la segmentación con K-means. Las características de textura se extraen utilizando el método GLCM (Gray-Level Co-Occurrence Matrix / Matriz de co-Ocurrencia de Nivel de gris), y se introducen a la red neuronal artificial de back propagation. Se clasifican y detectan principalmente cuatro enfermedades de la planta de granada: tizón bacteriano, la mancha de la fruta, la podredumbre de la fruta y la mancha foliar. Los resultados experimentales de la red neuronal muestran que casi todas las muestras dan la mejor discriminación entre las categorías. La fruta y la hoja buenas dan un resultado del 100 %, la mancha de la hoja da 87.50 %, el tizón bacteriano da 85.71 % mientras que la mancha de la fruta y la podredumbre de la fruta dan el resultado de 83.33 %. El trabajo propuesto proporciona resultados precisos y satisfactorios y da una precisión del 90 %.

En [Kulkarni A., 2012], se propone un sistema de detección de enfermedades en las plantas, enfocado en el granado, donde se estudia la detección de algunas enfermedades: alternaria, bacteria blight, anthracnose en fruto y hojas. En el pre-procesamiento de la imagen se quitan brillos, para la segmentación se utiliza el espacio de color CIELab y para extraer las características el filtro de Gabor. Se entrena una red neuronal para el clasificador la cual obtuvo mejores resultados y una tasa de reconocimiento de hasta el 91 %.

Otra investigación [Palanivel N., 2017] tiene como objetivo principal clasificar las hojas enfermas de las sanas, basándose en la identificación de la textura, la forma y las propiedades del color. Para ello se procesan las imágenes usando el filtro paso bajo para minimizar el ruido, con el histograma se mejora el contraste, la calidad y la ecualización de la imagen. El Gray-Level Co-Occurrence Matrix (GLCM) y los sistemas de Patrones Binarios Locales (LBP) se introducen para la extracción de características. El análisis de rendimiento muestra que el sistema propuesto ofrece mayor precisión, sensibilidad y especificidad que la técnica de SVM existente.

En las investigaciones anteriores, identifican y clasifican las granadas de acuerdo a enfermedades, defectos en la piel, quemaduras solares y olor. Estos estudios se realizan en su mayoría en el fruto y en las hojas, y una investigación considera el tallo. Los objetivos varían unos son en la cosecha y otros en pos cosecha. Los métodos que se emplean para el pre-procesamiento son filtros de Gabor y paso bajos. Operaciones morfológicas de erosión y dilatación. Conversión de espacios de color de HIS, HSV y CIELa. En los métodos de

segmentación y clasificación se utiliza K-means, openCV, GLCM, histograma, LBP, redes neuronales backpropagation. El aporte de esta investigación será implementar métodos para la identificación de la Botrytis en las granadas en la pos cosecha, estos son, el método de Otsu, una red neuronal convolutiva de TensorFlow, crecimiento de regiones, segmentación manual, base de datos de píxeles enfermos, clasificador de máquina de vectores de soporte y Random Forest.

### 2.2.2. Fresas

Las fresas son unas de las frutas en las que se han centrado ciertas investigaciones para la detección de enfermedades. [Hurtado D. A., 2015] propone un algoritmo que analiza los frutos de las fresas donde se identifica la Botrytis y Sphaerotheca sp. Se obtuvieron las imágenes de manera directa en los cultivos. Éstas fueron procesadas usando el lenguaje CSharp con la librería Emug CV. El espacio RGB se pasa a HSV (Hue, Saturation, Value / Matiz, Saturación, Valor) usando el algoritmo de OpenCv Miscellaneous Image Transformations, enseguida se realiza una umbralización la cual retorna una imagen binaria, se obtuvo la media de la imagen binaria para determinar los niveles de intensidad de los tonos rojos para poder distinguir el fruto, si la media es superior a 6.0 la imagen es candidata para ser analizada. Enseguida se aplican filtros de suavizado y dilatación, para aumentar las regiones de brillo. A la imagen original se le aplica un umbralizado en formato HSV, se saca la media y se extraen los contornos, se aplica una máscara AND que define si las regiones resaltadas de la enfermedad se encuentran en la zona de la máscara principal. Descartando así resultados erróneos por contornos de otros objetos. Esta comparativa se realiza con patrones ya predefinidos con el método MatchShapes. Consideran al algoritmo propuesto una buena herramienta para biólogos, tecnólogos en horticultura y productores, que necesiten una aproximación de monitoreo de incidencia de anomalías en las plantas de fresa.

### 2.2.3. Uvas

Las uvas también han sido objetivo de los métodos de detección de enfermedades por medio de sistemas no destructivos. En [Narvekar P., 2014], se capturan imágenes de las hojas para determinar el estado de salud de la planta, se busca encontrar puntos negros o cafés, moho polvoriento y moho suave. Se pre-procesan las imágenes, convirtiendo del espacio de color RGB a HSI, enseguida se realiza con el método de Otsu un enmascaramiento del

color verde, para poder segmentar el área de la enfermedad, se realizaron parches de 32 x 32 píxeles. Para extraer las características de las enfermedades se utiliza el método de co-ocurrencia de color, este método toma en cuenta la textura y el color de la imagen, y se desarrolló mediante el uso de Matrices de Dependencia de Nivel de Gris Espacial (SGDM). Los resultados experimentales indican que el enfoque propuesto es un enfoque valioso, que puede apoyar significativamente una detección precisa de enfermedades de la hoja en un pequeño esfuerzo computacional. En [Padol P. B., 2016] se realiza un clasificador SVM de enfermedades en las hojas de las uvas, se centró en el reconocimiento y la clasificación de las enfermedades fúngicas como el mildiu y el mildiu pulverulento mediante el procesamiento de imágenes. Donde para obtener las enfermedades se pre-procesa realizando redimensionamiento de las imágenes, se aplica el umbral y el filtro gaussiano. Para segmentar las imágenes se utiliza K-means, y para la extracción de características se toma en cuenta la textura y el color. Finalmente, la técnica de clasificación SVM se utiliza para detectar el tipo de enfermedad foliar. En el experimento se consideraron dos clases de hojas de uva: el Mildeo Puro y el Mildeo Powdery. El sistema proporciona un 88.89% de exactitud promedio para Downey y enfermedad de la hoja de la uva de Powdery.

#### 2.2.4. Cítricos

De igual manera se han realizado trabajos en los cítricos de métodos no destructivos. Para [Gavhale K., 2014] el objetivo es implementar técnicas de análisis de imágenes y clasificación para la extracción y clasificación de enfermedades foliares. La imagen de la hoja se captura y luego se procesa para determinar el estado de cada planta. El marco propuesto es un modelo en cuatro partes de pre-procesamiento de imágenes incluyendo RGB a diferentes conversiones de espacio de color, tales como Lab y Ycber para la mejora de la imagen. Se segmentó la región de interés utilizando agrupación K-means. Para la extracción de características de textura se utilizó Gray-Level Co-occurrence Matrix (GLCM) estadístico y característica de color mediante valores medios. Por último, la clasificación lograda mediante la Máquina de Vectores de Soporte (SVM). Esta técnica garantizó que los productos químicos sólo se apliquen cuando las hojas de las plantas se hayan detectado con la enfermedad.

### 2.2.5. Varios (uva, manzana, granada)

Algunas investigaciones analizan varios frutos con el mismo proyecto, por ejemplo en [Awate A., 2015] muestra un sistema que funciona para tres frutas y sus enfermedades, las cuales son:

- Uva: se buscan las enfermedades de Black Rot, Moho polvoriento y Moho suave.
- Manzana: infección de la costra de manzana, pudrición y manzanas manchadas.
- Granada: bacteria blight, Aspergillus Fruit Rot y Botrytis.

Para el desarrollo de la aplicación se utilizó K-means para la segmentación de las imágenes. Se tomaron en cuenta cuatro características: color, textura, morfología y la forma del hoyo de la fruta. El algoritmo usado para extraer las características es SURF (Speed Up Robust Feature). Se realizó un análisis Blob, que permite extraer el área de interés de las diferentes propiedades de una imagen, como la comparación de color y brillo con sus regiones cercanas. Este análisis consiste en tres etapas: extracción con la técnica de umbralización, limpiar la calidad de la imagen, y el análisis donde se evalúa y se calcula el resultado. Una vez obtenidas las imágenes se entrena la red neuronal para el clasificador.

En [Jhuria M., 2013] se analizan las hojas de uvas y manzanas, donde se identifican las enfermedades de pudrición negra y Mildew polvoriento en la uva, costra y putrefacción en la manzana. Se extraen tres características para la base de datos de aprendizaje: color, morfología y textura. El color es el más importante para discriminar objetos. El procesamiento de la imagen se dividió en 3 áreas: la transformación del color de RGB a HSI, procesamiento del plano de color individual y procesamiento del vector de color. En la segunda se utiliza la morfología para extraer características de las enfermedades. Al igual que el color, el espacio RGB se convierte en HSI con planos cuantificados **H**, **S** e **I**, a **30**, **30**, **20** niveles respectivamente. Y la tercera es la textura la cual describe las características visuales y se utiliza la descomposición de Daubechies 2-D Wavelet. Después de la extracción de las características se realiza la clasificación con una red neuronal de tipo back propagation, la cual es entrenada y después probada.

### 2.2.6. Manzana

En [Dubey S., 2012] se utiliza segmentación y SVM para hacer un clasificador automatizado de enfermedades en las manzanas las cuales son: mancha de manzana, podredumbre de manzana, costra de manzana. Esto utilizando segmentación en las imágenes con K-means para separar la parte sana y de la enfermedad. Para hacer la clasificación se siguen los siguientes pasos:

1. Leer imagen.
2. Transformar imagen de RGB a  $L^*a^*B$ .
3. Clasificar colores usando clustering de K-means en  $a^*b^*$ .
4. Etiquetar cada pixel de la imagen del resultado de K-means.
5. Generar imágenes segmentadas por color.
6. Seleccionar el segmento que contiene la enfermedad.

La precisión de clasificación reportada en este trabajo es del 89.88 %, 90.71 %, 96.66 % y 99.33 % para mancha de manzana, podredumbre de manzana, costra de manzana y manzana normal respectivamente, resultando una precisión promedio de 93.14 %. Cuando se entrena la red neuronal se hace con 50 imágenes por clase. Una investigación similar a la anterior pero con un proceso más elaborado es la de [Samajpati B., 2016], en el cual se realiza lo siguiente:

1. Cargar imágenes de entrenamiento.
2. Leer imágenes de prueba.
3. Realizar la extracción de características usando el color, histograma global del color, vector de coherencia del color, características de textura, patrón binario local, patrón binario completo local, patrón ternario local, características Gabor.
4. Realiza fusión a nivel de característica usando color y características de textura. Global Color Histogram (GCH), Color Coherence Vector(CCV), Local Binary Pattern(LBP) y Complete Local Binary Pattern(CLBP).
  - GCH + Gabor



- GCH + LTP
  - Gabor + GCH + LBP
  - Gabor + LBP
  - Gabor + CLBP
  - Gabor + LTP
  - Gabor + CLBP + LTP
5. Aplicar el clasificador de Random Forest en las imágenes segmentadas.
  6. Si la fruta está infectada por alguna de las enfermedades, ir al paso 7 de lo contrario, ir al paso 8.
  7. Aplicar el agrupamiento K-means para la segmentación de imágenes usando la distancia euclidiana para la agrupación de K-Means.
  8. La fruta es fruta normal u ordinaria.
  9. Resultados de las enfermedades clasificadas de la manzana.

En otras investigaciones sobre las enfermedades en las manzanas se utilizan técnicas similares como el K-means, aunque se complementen con métodos diferentes, como en [Dubey S., 2014], donde propone un algoritmo que identifique la enfermedad de la sarna, costra, podredumbre y mancha en la manzana. Para ello se realizó un pre-procesamiento, donde se transformó de RGB al espacio de  $L^*a^*b^*$ , se usó K-means para segmentar las imágenes en el espacio  $a^*$  y  $b^*$ , se generaron imágenes de la segmentación por color y se seleccionaron las que contenían las enfermedades. En este experimento, se utilizó Multi-class Support Vector Machine (MSVM) como un conjunto de Máquinas Binarias de Soporte Vectorial (SVM) para el entrenamiento y la clasificación.

### 2.2.7. Café

Las necesidades de poder identificar las características de los frutos por medio de la ayuda de las computadoras ha sido tal, que ha llegado hasta los granos de café, los cuales son contaminados por una plaga llamada broca. En [Ortiz M., 2015] y en [Herrera J. C., 2016] se realizan dos algoritmos, uno en el cual se identifica si el café está verde o maduro y el otro algoritmo determina si el café está infectado por la broca (es una plaga presente en el

café). La diferencia entre estas dos investigaciones es que, en la primera se utiliza Matlab con una red neuronal de base radial (Neural Network Toolbox), y en la segunda investigación se utiliza Python 2.7.8 con la librería OpenCV 2.4.9.

Debido a la gran necesidad de analizar los frutos se han realizado una cantidad de algoritmos de identificación de enfermedades, por lo que en [Raut S., 2017] se hace una recopilación de técnicas de identificación y clasificación de enfermedades realizadas por varios autores. Las técnicas que se estudiaron son K-means, SVM, Redes neuronales, conversión de RGB a HSI, algoritmos de back-propagation y SURF (Surface plot/ Superficie), GLCM, conversión de RGB a HSV, clasificador Random forest, fusión de color y características de texturas, espacios de color tales como YCbCr, RGB, CIELa\*b, HSV, clasificador de distancia mínima, filtro de Gabor. Dando como resultado que la técnica más usada es K-means para la segmentación de imágenes a color y las redes neuronales para la clasificación.

### Otros

Gracias a la visión artificial se han podido identificar objetos de manera rápida e inteligente, por lo que en [Camargo A., 2009] se realizó un sistema de visión artificial para identificar síntomas visuales en las plantas de algodón. Donde se utiliza el procesamiento de imágenes y la clasificación de patrones. Con este sistema se identifican los síntomas visuales de: chinche verde del sur (*Nezara viridula*), Bacteriano angular (*Xanthomonas campestris*) y Tizon de ascochyta (*Ascochyta gossypii*). Donde se utilizaron los siguientes métodos:

- Extracción de características convirtiendo a I3a, I3b, H. Se calcularon las características en los canales RGB y HSV. De los cuales se discriminaron las características de: Shape, textura, dimensión fractal, lacunaruty, dispersión, niveles de grises, discriminación del histograma gris, Fourier.
- Matriz de co-ocurrencia: usada para calcular la textura de las imágenes. Este método mide la ocurrencia de niveles de gris entre una posición específica en la imagen y un pixel vecino de acuerdo a una distancia y dirección dada. Se estimaron las siguientes mediciones: Energía, inercia, entropía, homogeneidad y correlación.
- Dimensión Fractal: usada para medir las dimensiones de un objeto con el algoritmo de conteo de cajas.
- Lacunarity: método para medir la textura asociada con un patrón de dispersión espacial. Se usó el algoritmo Gliding Box.

- Evaluación de modelos de diagnósticos, para ello se utilizó una red neuronal para reconocimiento de patrones. Se evaluaron varios modelos de clasificación mediante validación cruzada. Con el SMV (Multi-Class Support Vector Machine / Máquina de Vectores de Soporte) se usó el núcleo de Función de Base Radial (RBF) y el método de uno contra uno (one-against-one).
- Clasificador de clases múltiples: se requirió identificar más de un grupo de características.
- Selección de características: para evaluar la selección de características se usó el seven-fold cross-validation.

En [Gila D. M., 2015], se propone un sistema de inspección automática basado en la visualización computacional para clasificar automáticamente diferentes tipos de residuos para la producción de aceite cuando se inicia el proceso de molienda. La clasificación se basa en la diferenciación en línea, entre las aceitunas de tierra y de árbol. Para este propósito, las muestras de aceitunas se han obtenido recolectando bayas directamente del árbol o del suelo. El vector característico de las muestras se ha obtenido a partir de los histogramas de imagen de aceituna. Además, se han empleado diferentes algoritmos de procesamiento de imágenes y técnicas de clasificación lineal tales como Fisher Discriminant Analysis. En este trabajo se presenta un sistema automático de clasificación de las aceitunas antes de ser procesado para obtener aceite, se han resuelto diferentes problemas como el brillo producido por la piel de olivo o la configuración del hardware. El clasificador, que diferencia entre el suelo y las aceitunas de árbol para la producción de aceite de oliva, utiliza información como la arruga de su piel y su color. Los resultados muestran un éxito del 97 % con aceitunas provenientes del suelo y 100 % con aceitunas procedentes del árbol.

La contribución de esta investigación [Beaulieu P., 2014] es la reducción y determinación del tamaño de píxeles de características de imagen seleccionando los píxeles de características importantes y significativos, así como organizándolos de acuerdo con una métrica de criterio de rendimiento propuesta. Se propone estudiar dos métricas de criterio de rendimiento: el coeficiente de variación, y la relación entre la media y la varianza de dos clases separadas. Se estudian los métodos de Análisis de Componentes Principales, Análisis de Discriminación Lineal de Fisher y Correlación. En este artículo se presenta un estudio sobre el uso de la prueba  $t$  de 2 muestras como método para seleccionar las características significativas de la imagen para la clasificación de tres tipos diferentes de fruta. Produce excelentes resultados con una alta precisión de reconocimiento para todos los algoritmos de clasificación bajo ciertas condiciones de normalización y enmascaramiento de la varianza.

Como se ha visto en esta sección ya se han realizado varias investigaciones en la detección de las enfermedades en los frutos. En [Mahendran R., 2012] se desarrolló una aplicación con visión artificial en la cual se detectan diferentes características de las frutas incluyendo las enfermedades, utilizando el lenguaje de programación C, C++ y MATLAB. El software se dividió en dos módulos: una aplicación para la formación del sistema; y otra para ordenar la adquisición, procesar las imágenes y proporcionar los parámetros de calidad estimados de cada fruta. En [Al-Bashish D., 2011] el aporte de la investigación es una herramienta de procesamiento de imágenes para la detección y clasificación de hojas de plantas enfermas, utilizando técnicas de K-means y redes neuronales. Un trabajo interesante es el de [Sannakki S., 2011], donde se realiza la clasificación de la enfermedad foliar por medio de la visión artificial combinada con la lógica difusa. Se pre-procesa la imagen cambiando su tamaño, aplicando el filtro gaussiano y se convierte la imagen de RGB a CIELab. En la segmentación se utiliza K-means y después de esta fase, se saca el área total y el área enferma para promediar los píxeles enfermos. Una vez que se cuenta con esta información, se utiliza la clasificación de enfermedades por Fuzzy Logic (sistema de inferencia difusa para la clasificación, la cual necesita una variable de entrada: el porcentaje de infección).

Los proyectos antes mencionados, indagan sobre la detección de enfermedades de las frutas, la maduración, el peso y manchas, a través del procesamiento de imágenes, utilizando herramientas tales como Matlab, Python y librerías como OpenCV, Neural Network Toolbox, Statistics Toolbox Functions, entre otras. Analizan las frutas con procesos y métodos diferentes, algunos como el uso de K-means, SVM, Random Forest, método de Otsu, características estadísticas, características de dominio de la frecuencia con DTW y la conversión de espacios de color, serán replicados en la investigación que se plantea en esta tesis, en la cual se pretende detectar la enfermedad de la Botrytis cinérea en la granada, extrayendo las características del color de la enfermedad. Se estudiarán algunos métodos que no se mencionan en los antecedentes, tales como: crecimiento de regiones, binarización, método matemáticas de punto interior a un triángulo, base de datos de píxeles enfermos, segmentación manual y Tensorflow.

---

---

## CAPÍTULO 3

---

# Fundamento Teórico

### Contenido

---

<b>3.1. Introducción</b>	<b>18</b>
<b>3.2. Botrytis cinérea</b>	<b>18</b>
<b>3.3. Inteligencia artificial</b>	<b>20</b>
<b>3.4. Visión artificial</b>	<b>21</b>
3.4.1. Procesamiento de imágenes	21
3.4.2. Filtros y operaciones morfológicas	23
3.4.3. Imágenes a color	27
3.4.4. Métodos de segmentación	30
3.4.5. Extracción de características estadísticas	36
3.4.6. Transformación del dominio Wavelets	38
3.4.7. Reconocimiento de patrones	40
<b>3.5. Redes neuronales</b>	<b>41</b>
3.5.1. Neurona biológica	41
3.5.2. Neurona artificial	43
3.5.3. Tipos de función de activación	44
3.5.4. Arquitecturas fundamentales	46
3.5.5. Red neuronal convolucional	50
<b>3.6. TensorFlow Network de Google</b>	<b>52</b>
<b>3.7. Máquina de Vectores de Soporte (SVM)</b>	<b>54</b>
<b>3.8. Clasificador Random Forest</b>	<b>55</b>
<b>3.9. Herramientas de programación</b>	<b>57</b>
3.9.1. Python	57
3.9.2. Matlab	57
3.9.3. Weka	57

---

## 3.1. Introducción

El procesamiento de imágenes ha sido un avance significativo junto con la Inteligencia artificial, permitiendo el desarrollo de sistemas no invasivos para el análisis de frutos, las tecnologías estudiadas son: reconocimiento de patrones, procesamiento de imágenes, Inteligencia artificial y redes neuronales. Las redes neuronales y el Machine Learning, ha permitido que las computadoras sean capaces de aprender y clasificar diferentes características, permitiendo reconocer los diferentes objetos que componen una imagen.

En este capítulo se presenta la teoría de los métodos, técnicas y herramientas, utilizadas para el procesamiento de las imágenes con la finalidad de identificar la enfermedad de la Botrytis en las granadas.

## 3.2. Botrytis cinérea

Botrytis cinérea es conocida también como podredumbre del fruto, es la enfermedad más importante que afecta la producción de granada. Es causada por el hongo Botrytis cinérea el cual entra por el pistilo y va extendiéndose dentro del fruto hasta podrir los granos completamente, e inclusive la corteza. El desarrollo del hongo es favorecido por las lluvias, y los frutos que están al ras del suelo son los más susceptibles. Atacan también a frutos en el almacén [Mondragón J., 2008]. Existen tres tipos de granados que se cultivan:

- Granado común de frutos dulces.
- Granado agrio.
- Granado de frutos sin pepita.

En la empresa GEGEMA, S.P.R de R.L ubicada en el sur de Jalisco, se procesan dos tipos de granada: dulces y agrias. La granada dulce en su mayoría es de fruto grande, de un color externo rojo brillante, las semillas son pequeñas y medianamente duras. Otras en su minoría son verdes, amarillentas o rojizas, de corteza más delgada que la roja. Las granadas agrias son de tamaño mediano, de colores verdes, amarillos y rosados. Estas presentan menor porcentaje de granadas enfermas en la pos cosecha, ya que se descartan en cosecha las enfermas, esto

hace que solo se busque identificar que no lleguen granadas demasiado pequeñas, con Botrytis o verdes.

La Botrytis cinérea se detecta en todos los tipos de granadas, y sus características son:

- Puede tener piel rugosa.
- Puede tener piel en buen estado.
- Presenta mancha café externa, ver Figura 3.1.
- En la zona de la mancha, la granada está blanda y débil.
- En algunas condiciones no presenta mancha café externa, pero puede estar podrida en el centro, por ejemplo, ver Figura 3.2.
- En otros casos lo arilos no presentan putrefacción, solo la corteza está contaminada y aún así la granada debe ser descartada, ver Figura 3.3.
- Grietas con el hongo, puede ser que la piel o la corteza estén contaminadas y el arilo sano, aun así se descarta o en otros casos solo los arilos se observan contaminados, ver Figura 3.4.



Figura 3.1: Granada agria con hongo.



Figura 3.2: Granada visiblemente sana, pero tiene Botrytis cinérea por dentro.



Figura 3.3: Granada dulce con grieta, piel con corteza contaminada y arilo sano.



Figura 3.4: Granada dulce con grieta, arilos con hongo.

**Condiciones que causan confusión en la detección:** la granada se golpea y magulla, en ocasiones se llegan a agrietar o pinchar un poco, debido a la humedad la piel y la corteza se empiezan hacer café, pero esto no es motivo de descarte, ya que realmente la granada se encuentra en buen estado.

### 3.3. Inteligencia artificial

La Inteligencia artificial (IA), es el término que se da cuando una máquina pretende imitar el proceso mental, el razonamiento y la conducta del ser humano, esas son basadas en la experiencia, el conocimiento continuo del ambiente y resolver problemas, siendo las máquinas cada vez más capaces de realizar dichas actividades [Payrumani, 2008]. La IA debe apoyarse en tres pilares básicos:

- Algoritmos genéticos (análogo al proceso de evolución de las cadenas de ADN).
- Redes neuronales artificiales (análogo al funcionamiento físico del cerebro de animales y humanos).
- Razonamiento mediante una lógica formal (análogo al pensamiento abstracto humano).



## 3.4. Visión artificial

Uno de los sentidos más importantes de los seres humanos es la visión. Ésta es empleada para obtener la información visual del entorno físico. La visión humana es el sentido más desarrollado y el que menos se conoce debido a su gran complejidad. Es una actividad inconsciente y difícil de saber cómo se produce. La vista tiene más de dos millones de terminaciones nerviosas. La radiación exterior recibida por el ojo debe ser transformada en señales que sean procesadas por el cerebro. El ojo es el elemento transductor mientras que el cerebro es el que procesa dicha información.

La visión artificial o también llamada visión por computadora, pretende capturar la información visual del entorno físico para extraer características relevantes visuales, utilizando procedimientos automáticos, que se producen a partir de imágenes del mundo exterior. Esta pretende construir algoritmos que sean capaces de simular hasta cierto grado el proceso de visión biológico [Azuela, 2013], [Cuevas E., 2010]. Resumiendo, las principales ventajas de la visión humana respecto a la artificial y viceversa, son:

- Sistema humano:
  - Mejor reconocimiento de objetos.
  - Mejor adaptación a situaciones imprevistas.
  - Utilización de conocimiento previo.
  - Mejor en tareas de alto nivel de proceso.
- Sistema artificial:
  - Mejor midiendo magnitudes físicas.
  - Mejor para la realización de tareas rutinarias.
  - Mejor en tareas de bajo nivel de proceso.

### 3.4.1. Procesamiento de imágenes

El procesamiento digital de imágenes es el conjunto de técnicas, englobadas en el pre-procesamiento de imágenes, tales como, segmentación, extracción de características, realce y

detección de bordes, realce de contrastes, etc., los cuales se aplican a una imagen origen para obtener una imagen final cuyo resultado sea más adecuado para mejorar ciertas características y así poder mejorar la calidad o facilitar la búsqueda de información.

Para lograr esto la imagen debe pasar por algunas etapas de análisis, las cuales de acuerdo al paradigma de [González, 1992], son:

- Captación de la imagen: en esta etapa hay que tener en cuenta el problema de la resolución espacial y niveles de grises. La calidad del resultado obtenido de las siguientes etapas de procesamiento está ligada a la calidad y fidelidad de las imágenes originales.
- Pre-procesamiento: esta etapa no siempre se lleva a cabo, pero en algunas aplicaciones tiene un peso importante, ya que consiste en la atenuación del brillo presente en la imagen, el mejoramiento del contraste, filtrados entre otros. La salida de esta etapa es una imagen en niveles de grises.
- Segmentación: es la partición de una imagen en un conjunto de regiones no solapadas y homogéneas con respecto a algún criterio, cuya unión cubre la imagen entera. La imagen de salida es una imagen etiquetada o simbólica de donde se ha pasado de una imagen digital bruta, a una imagen simplificada donde se espera que los objetos aparezcan nítidamente distinguidos entre sí.
- Representación: el objetivo de esta etapa es llevar los datos de cada uno de los objetos o regiones segmentadas a formas en que la computadora pueda trabajar con ellos de manera más apropiada.
- Descripción: el objetivo es capturar las diferencias esenciales entre objetos pertenecientes a clases diferentes.
- Reconocimiento, análisis o interpretación de la imagen: la detección es el procedimiento de decir si una instancia de un objeto dado se encuentra presente en una imagen. El reconocimiento asigna una etiqueta a un objeto a partir de la información proporcionada por sus descriptores. La interpretación es la que determina el significado de un conjunto de objetos previamente reconocidos [Rodríguez R., 2011].

La profundización de las etapas depende de:

- Calidad de la imagen original.

- El hardware empleado en la adquisición.
- Los métodos utilizados en el procesamiento.
- El objetivo de la aplicación.

### 3.4.2. Filtros y operaciones morfológicas

En la etapa de pre-procesamiento, se aplican operaciones en una imagen origen con el objetivo de obtener una imagen final mejorada para los objetivos del proyecto. Los objetivos principales de los filtros son suavizar la imagen, eliminar ruido, realzar borde o detectar bordes. Los filtros se aplican a los píxeles de una imagen para optimizarla, enfatizar cierta información o mostrar un efecto especial en la imagen. A continuación se describen algunos de ellos:

#### Filtros espaciales lineales

Los filtros espaciales se implementan mediante un proceso llamado convolución espacial. Este proceso espacial se mueve a través de la imagen de entrada, píxel por píxel, ubicando los píxeles resultantes en la imagen de salida. El valor digital de brillo de cada píxel en la imagen de salida depende de un grupo de píxeles de entrada que rodean al píxel que se está procesando. Al utilizar la información del brillo de los píxeles vecinos a un píxel central, la convolución espacial calcula la actividad de frecuencia espacial en esa área, y por lo tanto, es capaz de filtrar en base al contenido de frecuencia espacial existente. Existen 3 tipos, filtro paso bajo, paso alto y realce de bordes por desplazamiento y diferencia [Aldalur B., 2002].

1. Filtros paso bajo: tienen por objeto suavizar los contrastes espaciales presentes en una imagen. Deja pasar o mantener intocables los componentes de baja frecuencia espacial de una imagen. Los componentes de alta frecuencia están atenuados o virtualmente ausentes en la imagen de salida [Aldalur B., 2002]). Unos ejemplos de estos son:
  - Filtro de la media: se visita cada píxel de la imagen y se reemplaza por la media de los píxeles vecinos. Se puede operar mediante convolución con una máscara de convolución que tiene dimensión 3 x 3 y sus nueve coeficientes son iguales a

1/9. Este filtro puede crear nuevas intensidades de grises que no aparecían en la imagen [Aldalur B., 2002].

- Filtro Gaussiano: permite suavizar las regiones en donde los valores de intensidad son homogéneos sin diluir los bordes de la imagen [Cuevas E., 2010]. Este filtro corresponde a una máscara modelizada con la función gaussiana:

$$G_{\sigma}(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.1)$$

2. Filtros paso alto: intensifica los detalles, bordes y cambios de alta frecuencia, mientras que atenúa las zonas de tonalidad uniforme. Esto permite una mejor identificación posterior de los objetos que se encuentren en la imagen, puesto que el brillo se hace mayor en las zonas con frecuencias más altas, al mismo tiempo que se oscurecen las zonas de frecuencias bajas. Es común la aparición de ruido tras el proceso.

3. Realce de bordes por desplazamiento y diferencia: permite realzar información de bordes verticales u horizontales. Desplazando una imagen a la izquierda en un píxel y restando esta nueva imagen de la original, se resaltarán los bordes verticales. Esto se debe a que el valor de brillo de cada píxel de entrada es restado de su vecino horizontal, dando un valor de la diferencia de brillo entre ambos. Si dos píxeles adyacentes tienen brillos muy diferentes, la diferencia resultará grande. Por el contrario, si dos píxeles adyacentes tienen valores similares de brillo la diferencia de brillos resultante será pequeña. Los realces de bordes son implementados a través de filtros espaciales. Los más utilizados son:

- Filtros Laplacianos: es una operación omnidireccional que resalta todos los bordes en una imagen, independientemente de sus orientaciones. Esta operación está basada en la tasa de cambio de la pendiente del brillo dentro de un núcleo de píxeles de dimensión 3 x 3. La máscara Laplaciana está definida como:

$$\begin{bmatrix} -1 & -1 & 1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (3.2)$$

- Prewitt: determina un realce de borde direccional. Este filtro está definido por la ecuación 3.3. La cual es aplicada sobre los diferentes vecinos del píxel en cuestión [Cuevas E., 2010].

$$H_x^p = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad y \quad H_y^p = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (3.3)$$

- Sobel: este filtro es similar a prewit con la diferencia de que en este filtro se le da un mayor peso al renglón o columna central del filtro [Cuevas E., 2010]. La matriz de coeficiente para este operador es definida como:

$$H_x^s = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad y \quad H_y^s = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (3.4)$$

- Roberts: el filtro es pequeño utilizando una matriz de coeficientes de tan solo 2x2, para la determinación del gradiente en sus dos diferentes direcciones a lo largo de las diagonales [Cuevas E., 2010]. El operador está definido por:

$$H_1^r = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad y \quad H_2^r = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad (3.5)$$

### Filtros espaciales no lineales

Existen otros filtros espaciales que no son función lineal de los valores de brillo de los píxeles de una imagen. Es decir, no se calculan como una suma lineal de elementos (brillo de los píxeles) multiplicados por pesos constantes (coeficientes de la máscara). Estos filtros se conocen como filtros espaciales no lineales. Constituyen también técnicas de procesamiento por grupo de píxeles, operando sobre un núcleo de píxeles de entrada que rodean a un píxel central. La diferencia es que en lugar de utilizar un promedio ponderado, emplean otras técnicas que combinan los valores de brillo del grupo de píxeles de entrada [Aldalur B., 2002]. Un ejemplo de este tipo de filtros es:

- La Mediana: se basa en sustituir el valor de brillo del píxel central del núcleo por la mediana de todos los valores de brillo de los píxeles que forman dicho núcleo. Permite eliminar objetos no deseados en la imagen productos de ruidos aleatorios en la adquisición de los datos, sin afectar significativamente a los bordes [Cuevas E., 2010].

### Operaciones morfológicas

A través de la morfología matemática pueden cuantificarse rigurosamente muchos aspectos de las estructuras geométricas contenidas en una imagen, de manera que concuerden con la

intuición y percepción humana. Estos filtros morfológicos se usan sobre imágenes binarias, donde los píxeles solo tienen dos posibles valores 0 y 1, blanco y negro. A continuación se describen las operaciones morfológicas fundamentales:

**Histograma:** el ajuste de histograma tiene como objetivo obtener para la imagen de salida un histograma con una forma específica. Una de las formas más conocidas de ajuste de histograma es la ecualización y la liberalización, la cual es una forma específica de transformación que tiende a lograr una aproximación muy próxima a una distribución uniforme de los valores de la escala de los niveles de grises. La ecualización del histograma tiene una ventaja, la cual viene dada por el hecho de que la imagen resultante, después de la transformación, implícitamente tiene máxima entropía. Una imagen con máxima entropía es aquella que tiene mayor homogeneidad y mayor contraste. De igual manera tiene sus desventajas la primera es que puede crear estructuras que no se encontraban previamente en la imagen original. La segunda es que puede originar una degradación de los niveles de grises alrededor de los bordes de la imagen. La tercera es que la ecualización del histograma puede producir una sobresaturación en las partes homogéneas de la imagen [Rodríguez R., 2011].

**Binarización:** esta operación convierte el valor de los píxeles de una imagen a valores 0 y 1, convirtiendo la imagen en colores de blanco y negro.

**Dilatación:** es una operación morfológica correspondiente a la idea intuitiva del crecimiento, añadiendo una capa de píxeles a la estructura de la imagen, este crecimiento es controlado por medio de una estructura de referencia. Esta operación está definida en la notación de conjuntos de puntos como:

$$I \oplus H = \{(x', y') = (x + i, y + j) | (x', y') \in P_I, (i, j) \in P_H\} \quad (3.6)$$

el conjunto de puntos que constituyen la dilatación de una imagen  $I$  y la estructura de referencia  $H$ , queda definido por todas las posibles combinaciones de los pares de coordenadas de los conjuntos de puntos  $P_I$  y  $P_H$ .

**Erosión:** es la operación cuasi inversa a la dilatación, la cual se expresa en la siguiente notación de conjuntos de puntos.

$$I \ominus H = \{(x', y') | (x' + i, y' + j) \in P_I, \forall (i, j) \in P_H\} \quad (3.7)$$

Para cada punto  $(x', y')$  de la imagen, el resultado se compone por los puntos  $(x', y')$  para los cuales todos los posibles valores  $(x' + i, y' + j)$  se encuentran en  $I$ .

Las operaciones de dilatación y erosión no pueden ser consideradas inversas, ya que no es posible reconstruir completamente una imagen erosionada mediante la aplicación sucesiva de dilataciones. Lo mismo aplica a la inversa. Sin embargo, presentan una relación dual, ya que una dilatación de los píxeles con valor 1 puede llevarse a cabo realizando una erosión del fondo o de los píxeles en 0 de una imagen [Cuevas E., 2017].

**Erosión – Dilatación:** es una combinación de apertura definida como una erosión seguida de una dilatación utilizando la misma estructura de referencia para ambas operaciones, ver ecuación 3.8.

$$I \circ H = (I \ominus H) \oplus H \quad (3.8)$$

Una apertura produce que todos los píxeles de la imagen en 1 más pequeños que la estructura de referencia, sean eliminados. Las estructuras que permanezcan en la imagen, serán, a través de la dilatación, suavizadas y aumentadas a su tamaño original.

**Dilatación – Erosión:** a la dilatación seguida de una erosión, usando la misma estructura de referencia, se denomina cierre, la operación se define como:

$$I \cdot H = (I \oplus H) \ominus H \quad (3.9)$$

Con la operación de cierre se rellenan los hoyos encontrados dentro de los objetos de la imagen, los cuales son más pequeños que la estructura de referencia [Cuevas E., 2017].

### 3.4.3. Imágenes a color

Las imágenes a color son de gran importancia en la vida cotidiana, así como en la representación de imágenes en el mundo digital, tales como la televisión, la fotografía o bien la impresión. La percepción del color es un complicado fenómeno sobre el cual se han encargado desde hace décadas más de una rama de investigadores, como psicólogos, filósofos, científicos y artistas.

#### Modelo de color RGB

El modelo de color RGB, se basa en la combinación de los colores primarios Red (R), Green (G), Blue (B). La mayoría de los programas para el procesamiento de imágenes y representación gráfica utilizan este modelo para la representación interna del color. En el

modelo RGB la combinación de colores es la adición de los componentes individuales siendo la base el color negro. La intensidad de los componentes de color determina tanto el tono como la iluminación del color resultante. El modelo RGB forma un cubo, cuyos ejes de coordenadas corresponden a los 3 colores primarios R, G, y B. Estos valores son positivos y se encuentran restringidos al intervalo de  $[0, V_{max}]$ , donde  $V_{max} = 255$ . Cada posible color  $C_i$  corresponde a un punto dentro del cubo RGB, con el componente:

$$C_i = (R_i, G_i, B_i) \quad (3.10)$$

donde  $0 \leq R_i, G_i, B_i \leq V_{max}$ . Normalmente el intervalo de valores de los componentes de color son normalizados al intervalo  $[0, 1]$ , de tal que el espacio de colores queda representado por un cubo unitario mostrado en la Figura 3.5. El punto  $N = (0, 0, 0)$  corresponde al negro,  $W = (1, 1, 1)$  al blanco y todos los puntos entre  $S$  y  $W$  son las tonalidades a escalas de grises donde  $R, G$  y  $B$  son iguales.

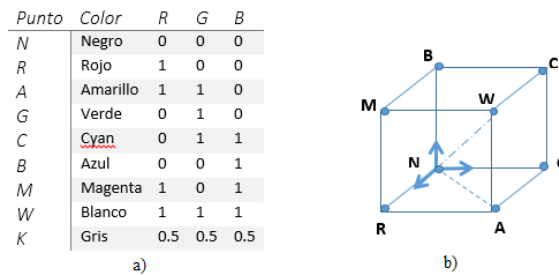


Figura 3.5: Representación del espacio de color RGB como cubo unitario.

Los colores primarios  $R, G$  y  $B$  de la Figura 3.5 representan los ejes coordenados. Los colores puros  $R, G, B, C, M$  y  $A$  mostrados en la tabla a) se encuentran ubicados en las esquinas del cubo de color. Todos los valores de grises, tales como  $K$  se ubican sobre la diagonal trazada de  $N$  a  $W$ .

Una imagen en RGB es un arreglo de tamaño  $M \times N \times 3$ , donde  $M \times N$  define las dimensiones de los planos, mientras que la dimensión 3 define a cada uno de  $R, G$  y  $B$ . Por lo anterior puede considerarse un arreglo de 3 imágenes a escalas de grises [Cuevas E., 2010].

### Histograma de una imagen RGB

Los histogramas son distribuciones que describen la frecuencia con la que se presentan los valores contenidos en la imagen. Cada valor del histograma es definido como  $h(i) =$  el número



de píxeles de la imagen con el valor definido  $i$  para todos los valores  $0 \leq i < K$ , siendo  $K$  el valor máximo permisible para describir los datos presentes en la imagen. Expresado en la siguiente ecuación:

$$h(i) = \text{card}(u, v) | I(u, v) = i_{11} \quad (3.11)$$

donde  $h(0)$  es el número de píxeles con el valor 0,  $h(1)$  el número de píxeles que tienen el valor de 1 así sucesivamente hasta que  $h(255)$  presenta el número de píxeles blancos de la imagen. El histograma de cada plano  $R$ ,  $G$  y  $B$  dan información adicional de la distribución de color en la imagen. Y cada plano es considerado como una imagen a escala de grises independiente y es desplegada de igual manera, ver Figura 3.6 [Cuevas E., 2010].

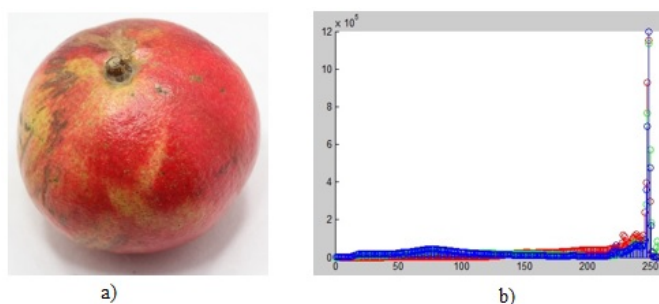


Figura 3.6: Histograma para RGB. a) Imagen original, b) Histogramas concatenados de los planos RGB.

### Modelo de color HSV

En el modelo de color HSV se representa la información del color a través de 3 componentes: Tonalidad (Hue), Saturación (Saturation) y Valor (Value). Este modelo es representado por una pirámide invertida, ver Figura 3.7, donde el pico de la pirámide son los tonos negros y la parte central de la base de la pirámide representa los tonos blancos. El eje principal representa el valor  $V$ , la distancia horizontal tomando como referencia el eje  $V$  es la saturación y el ángulo que se establece tomando como punto la rotación del eje  $V$  define la tonalidad. Los colores rojo, verde y azul se encuentran distribuidos en la base de la pirámide [Cuevas E., 2010].

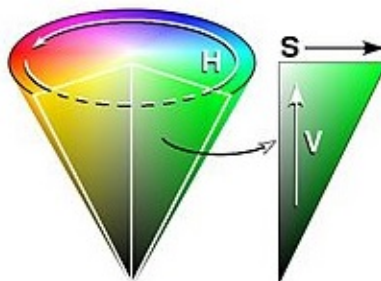


Figura 3.7: Modelo de color HSV.

### Modelo de color CIE $L^*a^*b^*$

El modelo CIE  $L^*a^*b^*$  fue desarrollado para linealizar la relación cambio de posición contra cambio de tonalidad, mejorando la forma en como el humano percibe estos cambios, haciendo el modelo más intuitivo. En este modelo el espacio de color se encuentra representado por 3 variables,  $L^*$  que representa el brillo,  $a^*$  y  $b^*$  la tonalidad. El valor de  $a^*$  define la distancia a lo largo del eje rojo-verde, mientras que el valor de  $b^*$  representa la distancia a lo largo del eje azul-amarillo. Los valores de  $L^*$  son siempre positivos y se encuentran dentro del intervalo de  $[0, 100]$ . Los valores de  $a^*$  y  $b^*$  se encuentran dentro del intervalo de  $[-127, 127]$ . Este modelo produce un cambio coherente de color por cambio de posición. La linealidad permite mediante la diferencia vectorial, encontrar un criterio que permita evaluar la diferencia en el plano de color [Cuevas E., 2010]. La diferencia entre dos colores  $Col_1$  y  $Col_2$  se define como la distancia euclideana entre ambos y esta se define como:

$$Dist(Col_1, Col_2) = ||Col_1 - Col_2|| = \sqrt{(L_1^* - L_2^*)^2 + (a_1^* - a_2^*)^2 + (b_1^* - b_2^*)^2} \quad (3.12)$$

#### 3.4.4. Métodos de segmentación

La segmentación de imágenes es considerada un campo activo de investigación en el área de visión artificial, con una innumerable cantidad de aplicaciones. Y se define como el proceso de separar una imagen en diferentes regiones, de modo que los píxeles de cada región cumplan ciertas características y sean diferentes entre las regiones. Para la segmentación de una imagen se pueden utilizar diversos métodos entre los cuales se encuentran:

## Binarización

La binarización de una imagen consiste en un proceso de reducción de la información de la misma, en la que sólo persisten dos valores: verdadero y falso. En una imagen digital, estos valores, verdadero y falso, pueden representarse por los valores 0 y 1 o, más frecuentemente, por los colores negro (valor de gris 0) y blanco (valor de gris 255).

En el proceso y análisis de imagen, la binarización se emplea para separar las regiones u objetos de interés en una imagen del resto. Las imágenes binarias se usan en operaciones booleanas o lógicas para identificar individualmente objetos de interés o para crear máscaras sobre regiones.

Para poder binarizar una imagen digital es mediante el histograma de dicha imagen. A través del histograma se obtiene una gráfica donde se muestra el número de píxeles por cada nivel de gris que aparecen en la imagen. Para binarizar la imagen, se deberá elegir un valor adecuado dentro de los niveles de grises (umbral), de tal forma que el histograma forme un valle en ese nivel. Todos los niveles de grises menores al umbral calculado se convertirán en negro y todos los mayores en blanco. En otros casos, una imagen binaria es simplemente el resultado de una selección interactiva de regiones de interés, las cuales se utilizarán como máscaras de comparación o referencia [Cuevas E., 2010] y [Cuevas E., 2017].

## Método de Otsu

El método de Otsu, llamado así en honor a Nobuyuki Otsu que lo inventó en 1979, utiliza técnicas estadísticas, para resolver el problema. En concreto, se utiliza la varianza, que es una medida de la dispersión de valores, en este caso se trata de la dispersión de los niveles de gris. El objetivo es calcular el valor umbral de forma que la dispersión dentro de cada clase sea lo más pequeña posible, pero al mismo tiempo la dispersión sea lo más alta posible entre clases diferentes [Rodríguez R., 2011].

El método considera una imagen  $I$  de dimensiones  $M \times N$  con  $L - 1$  escala de grises. El histograma presenta dos distribuciones traslapadas: la primera corresponde al objeto A, que involucra los valores de intensidad 0 al valor  $k$ , mientras que la segunda integra los niveles de gris de  $k + 1$  hasta  $L - 1$ .

El método de Otsu inicia calculando las probabilidades de cada objeto o clase, ver ecuación 3.13.

$$P(A) = \sum_{i=0}^k h(i) = W_A(k) = W(k)$$

$$P(B) = \sum_{i=k+1}^{L-1} h(i) = W_B(k) = 1 - W(k)$$
(3.13)

donde  $h(i)$  representa el número de píxeles de intensidad  $i$  contenidos en la imagen. Bajo estas condiciones, los promedios de las intensidades de cada objeto pueden ser calculadas de la siguiente manera (3.14):

$$\mu_A(k) = \frac{1}{w_A(k)} \sum_{i=0}^k i \cdot h(i)$$

$$\mu_B(k) = \frac{1}{w_B(k)} \sum_{i=k+1}^{L-1} i \cdot h(i)$$
(3.14)

apartir de estos valores, la varianza puede ser obtenida de la siguiente manera (3.15):

$$\sigma_A^2(k) = \frac{1}{w_A(k)} \sum_{i=0}^k (i - \mu_A(k))^2 \cdot h(i)$$

$$\sigma_B^2(k) = \frac{1}{w_B(k)} \sum_{i=k+1}^{L-1} (i - \mu_B(k))^2 \cdot h(i)$$
(3.15)

la varianza dentro de las clases puede definirse en terminos de la siguiente ecuación 3.16:

$$\sigma_D^2 = w_A(k) \cdot \sigma_A^2(k) + w_B(k) \cdot \sigma_B^2(k)$$
(3.16)

El método de Otsu busca el nivel de intensidad  $k$  que minimiza la varianza existente dentro de las clases. La idea es hacer la varianza de cada objeto lo más pequeña posible para que pueda minimizar su traslape. La varianza total de la distribución calcula la varianza del histograma completo. Se calcula de la siguiente manera:

$$\sigma_i^2 = \sum_{i=0}^{L-1} (i - \mu_T)^2 * h(i)$$
(3.17)

donde  $\mu_T$  es la intensidad promedio del histograma que representa a la distribución:

$$\mu_T = \sum_{i=0}^{L-1} i * h(i)$$
(3.18)

Finalmente, la varianza entre clases  $\sigma_E^2$  puede plantearse como la resta entre la varianza total  $\sigma_t^2$  y la varianza dentro de las clases  $\sigma_D^2$ . Quedando definida por la ecuación 3.19.

$$\begin{aligned}\sigma_E^2 &= \sigma_t^2 - \sigma_D^2 \\ &= w(k) \cdot (\mu_A(k) - \mu_T)^2 - (1 - w(k)) \cdot (\mu_B(k) - \mu_T)^2 \\ &= w(k) \cdot (1 - w(k)) \cdot (\mu_A(k) - \mu_B(k))^2\end{aligned}\quad (3.19)$$

Debido a que la varianza total es constante e independiente del nivel de intensidad  $k$ , el efecto de calcular el umbral óptimo recae sol en las varianzas  $\sigma_D^2$  y  $\sigma_E^2$ . Este método de optimización es posible emplearlo son problemas debido a que la función  $\sigma_E^2$  siempre es suave y unimodal [Cuevas E., 2017].

### K-means clustering

La segmentación de imágenes se refiere al conjunto de regiones no solapadas y homogéneas que cumplen con algún criterio, la unión de estas partes cubre la imagen original. La imagen de salida es una imagen simplificada donde se pretende que los objetos aparezcan distinguidos entre sí. La selección del cual es la partición que contiene las características necesarias depende del problema a resolver y ésta debe detenerse cuando estas características han sido aisladas [Cuevas E., 2010]. El objetivo en este trabajo es segmentar la imagen hasta identificar las regiones que califica como la enfermedad de la Botrytis. Hay varias técnicas para segmentar una imagen, en este proyecto se utiliza el método de agrupamiento K-means. Este método de agrupamiento realiza la partición de una colección de datos de  $k$  grupos de observaciones, los cuales están representados por un centroide, y cada observación pertenece al grupo que tiene el valor medio más cercano al centroide. El centroide es el vector formado por las medias de cada uno de los componentes de los elementos pertenecientes al clúster. K-means consta de dos fases separadas. La primera se calcula el centroide  $k$  y la segunda lleva cada punto al clúster del centroide más cercano. Para definir la distancia del centroide más cercano se aplican métodos de métrica de distancia, entre los cuales se encuentran:

- El método de la distancia euclidiana cuadrada. Cada centroide es la media de los puntos de ese clúster, la fórmula está dada por la ecuación 3.20.

$$d(x, c) = (x - c)(x - c)' \quad (3.20)$$

- El método de la suma de diferencias absolutas. Cada centroide es la mediana de los componentes del clúster. La fórmula se da en la ecuación 3.21.

$$d(x, c) = \sum_{j=1}^p |x_j - c_j| \quad (3.21)$$

- Método del coseno. Cada centroide es la media de los componentes de ese grupo, después de centrar y normalizar esos puntos a media cero y desviación estandar de la unidad, como se puede ver en la ecuación 3.22.

$$d(x, c) = 1 - \frac{xc'}{\sqrt{(xx)' - (cc)'}} \quad (3.22)$$

- Método de correlación. Cada centroide es la media de los componentes de ese grupo, después de centrar y normalizar esos puntos en media cero y desviación estándar de la unidad, ecuación 3.23.

$$d(x, c) = 1 - \frac{(x - \vec{x})(c - \vec{c})'}{\sqrt{(x - \vec{x})(x - \vec{x})'}\sqrt{(c - \vec{c})(c - \vec{c})'}} \quad (3.23)$$

Donde:

$$\vec{x} = \frac{1}{p} \left( \sum_{i=1}^p x_y \right) \vec{1}_p \quad (3.24)$$

$$\vec{c} = \frac{1}{p} \left( \sum_{i=1}^p c_y \right) \vec{1}_p \quad (3.25)$$

A continuación se muestra el algoritmo básico de K-means (Dhanachandra, N., et al., 2015):

1. Seleccionar centroides, y el número de k clúster deseado.
2. Asignar cada punto al centroide más cercano y cada colección de puntos asignados a un centroide es un clúster (Región de Voronoi), ecuación 3.26.

$$d = \|p(x, y) - c_k\| \quad (3.26)$$

3. Asignar los píxeles cercanos al centroides de cada clúster, basados en la distancia euclidiana.

4. Después de asignar todos los píxeles, re calcular nueva posición del centroide usando la siguiente relación 3.27.

$$c_k = \frac{1}{k} \sum_{y \in c_k} \sum_{y \in c_k} p(x, y) \quad (3.27)$$

5. Se vuelve a formar los píxeles del clúster en la imagen original.

En este trabajo, la parte de la enfermedad es extraída con K-means y enseguida se analizan sus características. Si se encuentran manchas cafés, entonces se ha identificado la enfermedad de la Botrytis cinérea en la granada.

### Crecimiento de regiones

Es un procedimiento que agrupa los píxeles o subregiones de la imagen en regiones mayores basándose en un criterio prefijado. Normalmente se empieza con unos puntos “semillas” para formar una determinada región, añadiendo aquellos píxeles vecinos que cumplan la propiedad especificada (por ejemplo, que estén en un rango de nivel de gris determinado).

La propiedad considerada en el crecimiento de regiones debe tener en cuenta la información sobre conectividad o adyacencia de la imagen. Otro factor importante es la condición de parada.

Sea  $f(x, y)$  la imagen de entrada;  $S(x, y)$  la matriz semilla conteniendo unos en las localizaciones de los píxeles semilla y ceros en el resto; y  $Q$  un predicado para ser aplicado en cada posición  $(x, y)$ . Asumimos que  $f$  y  $S$  con matrices del mismo tamaño [Rodríguez R., 2011] y [Cuevas E., 2017]. Un algoritmo de crecimiento de regiones basado en la 8-conectividad consiste en:

1. Encuentra todas las componentes conexas en  $S(x, y)$  y erosiona cada componente conexa a un píxel; etiqueta todos los píxeles encontrados con unos y el resto de píxeles con ceros.
2. Forma una imagen  $fQ$  tal que,  $fQ(x, y) = 1$  si la imagen  $f$  satisface el predicado en la posición  $(x, y)$ ; en caso contrario  $fQ(x, y) = 0$ .

### 3.4.5. Extracción de características estadísticas

La generación de características estadísticas es un procedimiento que calcula nuevas variables que de una manera u otra se origina a partir de los valores almacenados de la matriz de imágenes  $I(m, n)$ . El objetivo es generar características que exhiban propiedades de empaquetado de información, desde el punto de vista de separabilidad de clase. Ya que no se pueden usar los datos brutos  $I(m, n)$  directamente, las características deberían codificar de manera eficiente la información relevante que reside en los datos originales.

La textura de una región de imagen está determinada por la forma en que se distribuyen los niveles de gris sobre los píxeles en esta región. Aunque no hay una definición clara de "Textura", se describe una imagen por el aspecto, como fina o grueso, liso o irregular, homogéneo o no homogéneo, etc. El objetivo es cuantificar estas propiedades de una región de imagen. Estas características surgirán al explotar el espacio de relaciones subyacentes a la distribución de nivel de grises [Theodoridis S., 2010].

#### Características estadísticas de primer orden

Se permite que  $\mathbf{I}$  sea la variable aleatoria que representa los niveles grises en la región de interés. El histograma de primer orden  $\mathbf{P}(\mathbf{I})$  se define como:

$$P(I) = \frac{\text{número de píxeles con niveles de gris } I}{\text{Total de números de píxeles en la región}} \quad (3.28)$$

$P(I)$  es la fracción de los píxeles con niveles de gris  $I$ .  $N_g$  es la cantidad de niveles de gris posibles. En base a la ecuación 3.28, se definen las siguientes variables: Momentos:

$$m_t = E[I^t] = \sum_{I=0}^{N_g-1} I^t P(I), \quad i = 1, 2, \dots \quad (3.29)$$

Donde  $m_0 = 1$  y  $m_1 = E[I]$ , el valor medio de  $I$ . Momentos centrales:

$$\mu_i = E[(I - E[I])^i] = \sum_{I=0}^{N_g-1} (I - m_1)^i P(I) \quad (3.30)$$



Los momentos centrales más utilizados son  $\mu_2$ ,  $\mu_3$ , y  $\mu_4$ .  $\mu_2 = \sigma^2$  es la varianza, y  $\mu_3$  se conoce como la asimetría y  $\mu_4$  como la curtosis del histograma. La varianza es una medida del ancho del histograma, es decir, una medida de cuánto difieren los niveles grises del medio. La asimetría es una medida del grado de simetría del histograma alrededor de la media, y  $\mu_4$  es la medida de la nitidez del histograma [Theodoridis S., 2010].

### Características estadísticas de segundo orden

Las características resultantes de las estadísticas de primer orden proporcionan información relacionada con la distribución de nivel de grises de la imagen, pero no proporcionan ninguna información sobre las posiciones relativas de los distintos niveles de gris dentro de la imagen. Esta información se puede extraer de los histogramas de segundo orden, donde los píxeles se consideran por pares. Tomando en cuenta dos parámetros, la distancia relativa entre los píxeles y su orientación relativa. Sea  $d$  la distancia relativa medida en números de píxeles. La orientación  $\Phi$  se cuantifica en cuatro direcciones: horizontal, diagonal, vertical y antidiagonal (0°, 45°, 90°, 135°) [Theodoridis S., 2010]. Para cada combinación de  $d$  y  $\Phi$  se define un histograma bidimensional:

$$0^\circ : P(I(m, n)) = I_1, I(m \pm d, n) = I_2$$

$$45^\circ : P(I(m, n)) = I_1, I(m \pm d, n \mp d) = I_2$$

$$90^\circ : P(I(m, n)) = I_1, I(m, n \pm d) = I_2$$

$$135^\circ : P(I(m, n)) = I_1, I(m \pm d, n \pm d) = I_2$$

A continuación se muestran algunas funciones para la extracción de características de primer y segundo orden.

**Media:** es la medida que indica la intensidad promedio de una imagen.

$$m = \sum_{i=0}^{L-1} P(Z_i) \quad (3.31)$$

**Desviación estándar:** mide cuanto es la distancia de los datos, sacando el promedio del contraste.

$$\sigma = \sqrt{\sum_{i=0}^{L-1} (Z_i - m)^2 P(Z_i)} \quad (3.32)$$

**Smoothness:** mide la suavidad relativa de la intensidad en una región.

$$R = \frac{1 - 1}{(1 + \sigma^2)} \quad (3.33)$$

**Asimetría:** es la medida que indica la simetría de la distribución de una variable respecto a la media aritmética, sin necesidad de hacer la representación gráfica. Los coeficientes de asimetría indican si hay el mismo número de elementos a la izquierda y derecha de la media.

$$\mu_3 = \sum_{i=0}^{L-1} (Z_i - m)^3 P(Z_i) \quad (3.34)$$

**Energía:** esta medida da valores altos cuando en la matriz de co-ocurrencia tiene pocas entradas de gran magnitud, y es baja cuando todas las entradas son similares. Si todos los píxeles son iguales la energía es mínima. Es una medida de la homogeneidad local.

$$C_3 = \sum_{i,j} -C_{i,j} * \log_2[C_{i,j}] \quad (3.35)$$

**Entropía:** la entropía es una medida de aleatoriedad y toma valores bajos para imágenes suaves.

$$H_{xy} = - \sum_{f=0}^{N_g-1} \sum_{f=0}^{N_g-1} P(i,j) \log_2 P(i,j) \quad (3.36)$$

### 3.4.6. Transformación del dominio Wavelets

Las Wavelets son una forma de onda de duración limitada que tiene un valor medio cero. Que consiste en descomponer la señal  $F$  en versiones escaladas y trasladadas de la wavelet original. Resulta imposible realizar los cálculos de los coeficientes para todas las escalas y posiciones, es por ello que se emplea la versión discreta denominada Transformada Discreta de Wavelets (TDW), donde se elige un subconjunto de escalas y posiciones. El filtrado de

baja frecuencia suaviza la imagen y esta sigue siendo visible, y el de alta frecuencia extrae los bordes y la imagen no es visible. Es por esto que se habla de descomposición de la imagen de entrada en dos componentes:

1. Aproximación de baja frecuencia.
2. Detalle de la señal.

El resultado de descomponer la señal de entrada en versiones paso bajo y paso alto se conoce como sub-bandas, que se pueden ir descomponiendo por el mismo procedimiento. Por lo que se dice que la TDW descompone una señal de entrada en un cierto número de bandas de frecuencia.

La TDW descompone la imagen original en cuatro imágenes submuestreadas o diezmadas. El submuestreo se realiza cada dos píxeles y se obtiene una imagen que ha sido filtrada mediante un filtro paso alto en las direcciones horizontal y vertical, una imagen filtrada en paso alto en la vertical y en paso bajo en la horizontal, una imagen filtrada en paso bajo en vertical y en paso alto en horizontal, y una filtrada en paso bajo en ambas direcciones [Pajares Martinsanz G., 2008].

Se han propuestos varios filtros para implementar las TDW, y los más utilizados son Haar y Daubechies. Los vectores base se muestran en la Tabla 3.1.

	Haar		Daubechies				
Paso Bajo	$\frac{1}{\sqrt{2}}$	[1, 1]	$\frac{1}{4\sqrt{2}}$	$1 - \sqrt{3}$	$3 - \sqrt{3}$	$3 + \sqrt{3}$	$1 + \sqrt{3}$
Paso ALto	$\frac{1}{\sqrt{2}}$	[-1, 1]	$\frac{1}{4\sqrt{2}}$	$-1 - \sqrt{3}$	$3 + \sqrt{3}$	$-3 + \sqrt{3}$	$1 - \sqrt{3}$

Tabla 3.1: Ejemplos de vectores base Haar y Daubechies para filtrado Paso Bajo y Paso Alto.

Una vez elegidos los vectores base para las operaciones Paso Alto o Bajo, los pasos para obtener la TDW son:

1. Realizar la convolución de las filas con el filtro paso bajo y guardar los resultados.
2. Realizar la convolución de las columnas con el filtro paso bajo, a partir de los resultados del paso 1.

3. Realizar la convolución del resultado del paso 1 con el filtro paso alto en las columnas.
4. Realizar la convolución de la imagen original con el filtro paso alto en las filas y guardar el resultado.
5. Realizar la convolución del resultado del paso 4 con el filtro paso bajo en las columnas.
6. Realizar la convolución de las columnas del resultado del paso 4 con el filtro paso alto.

Los resultados de la TDW, como imagen, se muestran en la Tabla 3.2.

Paso Bajo / Paso Bajo	Paso Bajo / Paso Alto
Paso Alto / Paso Bajo	Paso Alto / Paso Alto

Tabla 3.2: Bandas de frecuencia en una TDW con cuatro bandas.

La Figura 3.8 [Pajares Martinsanz G., 2008] muestra un esquema del algoritmo anterior, donde PBD significa Filtro Paso Bajo Directo y PAD Filtro Paso Alto Directo, se entiende por directos los filtros de la Tabla 3.1 .

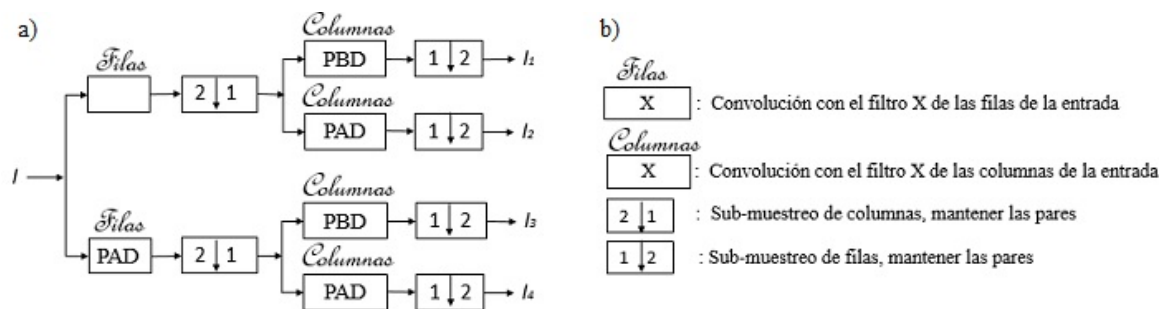


Figura 3.8: a) esquema del algoritmo de filtrado por filas y columnas para la descomposición de la imagen original en cuatro subimágenes; b) simbología utilizada.

### 3.4.7. Reconocimiento de patrones

La capacidad de reconocer patrones ha sido considerada siempre como una de las características fundamentales de la inteligencia humana, en ese sentido, ha sido también una de las áreas de estudio de prácticamente todas las disciplinas que conforman la ciencia cognitiva. El

reconocimiento de patrones es una disciplina que estudia la asignación automática de objetos a clases. Estas clases pueden estar predeterminadas o no. Es por ello que la estructura para el aprendizaje supervisado de una red neuronal es fundamentalmente asociativa. Se tiene un conjunto de datos de entrada, esto es vectores de estado, para el entrenamiento, por lo que se sabe cuál es la salida para una entrada dada. La salida podría ser otro vector de estado de cualquier forma, la cual corresponde a una clasificación, que es lo que se requiere para un reconocimiento de patrones [Anderson, 2007].

El número de aplicaciones prácticas es muy extenso: lectura automática de caracteres manuscritos e impresos; reconocimiento de voz e imágenes, caras, huellas digitales, firmas; identificación de partículas elementales; detección de fraude; segmentación de mercados, etc.

## 3.5. Redes neuronales

Las redes neuronales son una forma de simular ciertas características propias del ser humano, tales como la capacidad de memorizar y de asociar hechos. Algunos problemas no siempre se pueden solucionar con un algoritmo, las soluciones a esos problemas se basan en la experiencia, y esta es una capacidad del ser humano, por lo que se quiere reproducir esta capacidad humana con las redes neuronales, se dice que estas son un modelo artificial y simplificado del cerebro humano, inspirada en la célula fundamental del sistema nervioso humano: "la neurona".

### 3.5.1. Neurona biológica

Es una célula del sistema nervioso especializado en captar estímulos provenientes del ambiente y de transmitir impulsos nerviosos. El cerebro humano contiene alrededor de 100 mil millones de neuronas. Cada neurona puede estar conectada hasta a 10.000 neuronas, transmitiéndose señales entre sí a través de hasta 1.000 billones de conexiones sinápticas. Las neuronas se dividen en tres partes principales, ver Figura 3.9 [Camperos E. N. S., 2006]:

- Las dendritas: son el elemento receptor; son fibras que cargan de señales eléctricas al cuerpo de la célula.
- El soma o cuerpo de la célula: realiza la suma de esas señales.

- El axón es una fibra larga que lleva las señales desde el cuerpo de la célula hacia otras.

La conexión entre dos neuronas se llama sinapsis. Esta se origina entre el botón terminal de un axón y las dendritas iniciales de otra neurona [Camperos E. N. S., 2006].

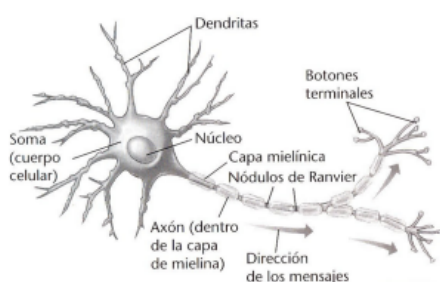


Figura 3.9: Representación de una neurona biológica.

## Sinapsis

La sinapsis neuronal es la zona de transmisión de impulsos nerviosos eléctricos entre dos neuronas para comunicarse entre sí. El término sinapsis significa ‘conexión’ y fue introducido por Charles Sherrington en 1897. Esta se trata de un pequeño espacio que separa las neuronas y para que se produzca la comunicación entre las neuronas, la membrana de una neurona genera un impulso eléctrico que debe viajar por un axón hasta la terminal sináptica. La comunicación entre las neuronas tiene lugar cuando los impulsos eléctricos llegan al botón sináptico, lo que provoca la liberación de neurotransmisores por parte de la célula pre sináptica y la absorción de estos por la célula pos sináptica, lo que tiene lugar en la hendidura sináptica. Convirtiendo una señal eléctrica de la neurona pre sináptica. En la Figura 3.10 se puede observar la interconexión de dos neuronas biológicas [Camperos E. N. S., 2006].

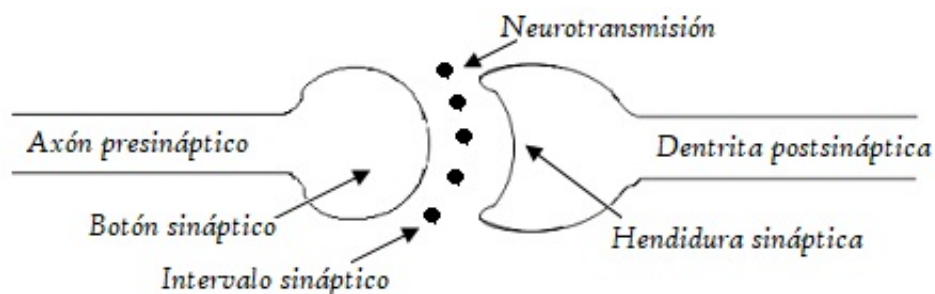


Figura 3.10: Sinapsis biológica.

### 3.5.2. Neurona artificial

La neurona es la unidad de proceso de información fundamental en una red neuronal [Haykin, 1999]. En la Figura 3.11 [Haykin, 1999] se muestra la estructura una red neuronal artificial.

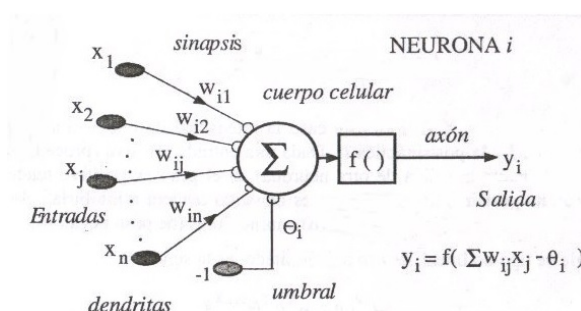


Figura 3.11: Representación de una neurona artificial tipo McCulloch-Pitts.

La neurona artificial procesa un vector de entradas y producen una respuesta o salida única. La neurona se compone de:

- Enlaces de conexión: los parámetros son los pesos sinápticos  $w_{nj}$ . Donde el primer subíndice corresponde a la neurona receptora y el segundo a la neurona emisora. Si  $w_{jn} > 0$ , entonces la conexión es excitadora; así mismo, si  $w_{jn} < 0$ , la conexión es inhibitoria.
- Sumador ( $\Sigma$ ): suma los componentes de las señales de entrada multiplicadas por  $w_{jn}$ .

- Función de activación: el valor obtenido con la regla de propagación, se filtra a través de la función de activación y es la que nos da la salida de la neurona.
- Umbral: es la condición que debe presentar un estímulo para poder alcanzar el impulso nervioso o potencial de acción. El valor de salida de la función de activación puede ser comparada con el umbral. Si la suma de las salidas es mayor al umbral, la neurona genera una señal, si es menor no se generara ninguna señal.

### 3.5.3. Tipos de función de activación

Las funciones de activación denotadas por  $f(v)$ , definen la salida de la neurona en función del potencial de activación  $v$ . A continuación se describen tres tipos básicos de activación [Camperos, E. N. S., et al., 2006]:

1. Función escalón o umbral. Esta función presentada en la Figura 3.12, se tiene la ecuación 3.37:

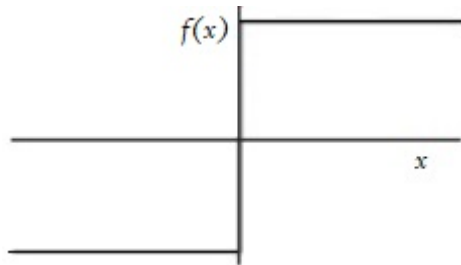


Figura 3.12: Función escalón o umbral

$$f(v) = \begin{cases} 1 & \text{si } v \geq 0 \\ 0 & \text{si } v < 0 \end{cases} \quad (3.37)$$

correspondiente a la salida de la neurona  $n$  aplicando la función escalón queda expresada como:

$$y_n = \begin{cases} 1 & \text{si } v_n \geq 0 \\ 0 & \text{si } v_n < 0 \end{cases} \quad (3.38)$$

con  $v_n$  dada por la función 3.39

$$v_n = \sum_{j=0}^m w_{nj}x_j \quad (3.39)$$



2. Función lineal a tramos: para este tipo de función, representada por la Figura 3.13, se tiene la función 3.40:

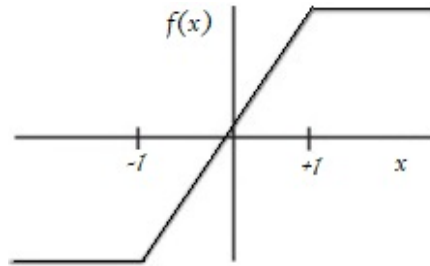


Figura 3.13: Función lineal a tramos

$$f(v) = \begin{cases} 1 & v \geq +\frac{1}{2} \\ v & +\frac{1}{2} > v > -\frac{1}{2} \\ 0 & v \leq -\frac{1}{2} \end{cases} \quad (3.40)$$

donde el factor de amplitud dentro de la región de operación se supone igual a la unidad.

3. Función sigmoidea: esta es la función más utilizada en las redes neuronales artificiales. Es estrictamente creciente, con un comportamiento asintótico. Un ejemplo de esta función es la función logística:

$$f(v) = \frac{1}{1 + e^{-av}} \quad (3.41)$$

donde  $a$  es el parámetro que determina la pendiente de la función sigmoidea Figura 3.14.

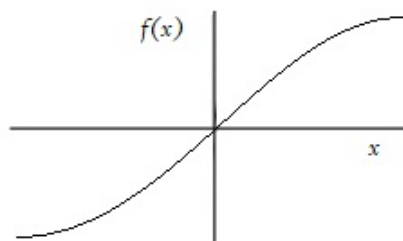


Figura 3.14: Función sigmoidea

Las funciones de activación hasta aquí mencionadas toman valores en el intervalo cerrado de  $[0, 1]$ . Sin embargo, también pueden tomar valores en el intervalo cerrado  $[-1, 1]$ ; en ese caso

la función escalón queda definida como:

$$f(v) = \begin{cases} 1 & \text{si } v > 0 \\ 0 & \text{si } v = 0 \\ -1 & \text{si } v < 0 \end{cases} \quad (3.42)$$

la cual es comúnmente llamada función signo. Para la forma correspondiente de la función sigmoidea se puede usar la función tangente hiperbólica, definida como:

$$f(v) = \tanh(v) \quad (3.43)$$

#### 3.5.4. Arquitecturas fundamentales

Más de dos neuronas pueden ser acomodadas en una capa y varias capas pueden conformar la organización y disposición de las neuronas en la red formando capas o agrupaciones de neuronas. Los parámetros fundamentales de la red son:

- Número de capas.
- Número de neuronas por capa.
- Grado de conectividad.
- Tipo de conexión entre neuronas.

Al hacer una clasificación topológica de las Redes Neuronales Artificiales (RNAs) se suelen distinguir:

- Redes unicapa: se establecen conexiones laterales entre las neuronas que pertenecen a la única capa que constituye la red. Las redes mono capa se utilizan típicamente en tareas relacionadas con lo que se conoce como auto asociación; por ejemplo, para regenerar informaciones de entrada que se presenta como incompleta o distorsionada [Olabe, X. B. 1998].

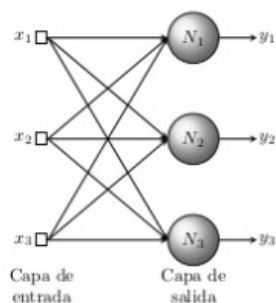


Figura 3.15: Red neuronal unicapa

- Redes multicapa: se forman con un grupo de capas simples en cascada. La salida de una capa es la entrada de la siguiente capa. Se ha demostrado que las redes multicapa presentan cualidades y aspectos por encima de las redes de una capa simple, disponen las neuronas agrupadas en varios niveles. Dado que este tipo de redes disponen de varias capas, las conexiones entre neuronas pueden ser del tipo feedforward (conexión hacia adelante) o del tipo feedback (conexión hacia atrás) [Olabe, X. B. 1998].

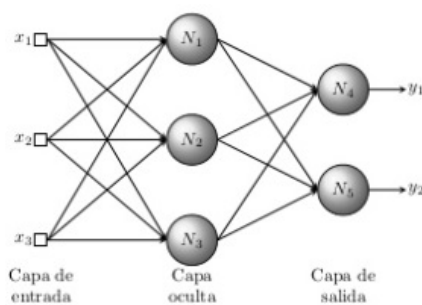


Figura 3.16: Red neuronal multicapa

### Mecanismos de aprendizaje

El aprendizaje es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada. Los cambios que se producen durante la etapa de aprendizaje se reducen a la destrucción, modificación y creación de conexiones entre las neuronas. Se considera que el proceso de aprendizaje ha terminado cuando los valores de los pesos permanecen estables  $dwj/dt = 0$ . De forma general existen dos tipos de aprendizajes:

**Aprendizaje supervisado:** el proceso de aprendizaje se realiza mediante un entrenamiento controlado por un agente externo (supervisor, maestro) que determina la respuesta que de-

bería generar la red a partir de una entrada determinada.

El aprendizaje se puede realizar de tres formas:

- Aprendizaje por corrección de error: ajusta los pesos de las conexiones de la red en función de la diferencia entre los valores deseados y los obtenidos en la salida. La fórmula para la corrección de los pesos podría ser la siguiente:  $(w_{ji}) = \beta y_i (d_j - y_j)$  Siendo:  
*Incr*( $w_{ji}$ ): variación en el peso de la conexión entre las neuronas  $i$  y  $j$ .  
 $y_i$ : salida de la neurona  $i$ .  
 $d_j$ : valor de salida deseado para la neurona  $j$ .  
 $y_j$ : valor de salida obtenido en la neurona  $j$ .  
 $\beta$ : factor de aprendizaje ( $0 < \beta \leq 1$ ) que regula la velocidad del aprendizaje.
- Aprendizaje por refuerzo: este tipo de aprendizaje es más lento que el anterior y se basa en la idea de no indicar durante el entrenamiento la salida exacta que se desea que proporcione la red ante una determinada entrada. Aquí la función del supervisor se reduce a indicar mediante una señal de refuerzo si la salida obtenida en la red se ajusta a la deseada (*éxito* = +1 *fracaso* = -1) y en función de ello se ajustan los pesos basándose en un mecanismo de probabilidades.
- Aprendizaje estocástico: consiste básicamente en realizar cambios aleatorios en los valores de los pesos y evaluar su efecto a partir del objetivo deseado y de distribuciones de probabilidad.

**Aprendizaje no supervisado:** examinan los datos de entrada y organizan los datos de manera interesante y apropiada, pero tiene poca información inicial ya que no reciben información por parte del entorno que le indique si la salida generada en respuesta a una determinada entrada es o no correcta; por lo que estos algoritmos pueden ser difíciles de construir y utilizar. Estas redes deben encontrar las características, regularidades, correlaciones o categorías que se pueden establecer entre los datos de la entrada [Azuela, 2013]. Las posibles interpretaciones de las salidas pueden ser:

- La salida representa el grado de familiaridad o similitud entre la información de entrada y las informaciones mostradas con anterioridad.
- Clusterización o establecimiento de categorías, indicando la red a la salida a qué categoría pertenece la información de entrada, siendo la propia red la que debe establecer las correlaciones oportunas.

- Codificación de los datos de entrada, generando a la salida una versión codificada con menos bits, pero manteniendo la información relevante de los datos.
- Mapeo de características, obteniéndose una disposición geométrica que representa un mapa topográfico de las características de los datos de entrada.

Los algoritmos de aprendizaje no supervisado suelen ser de dos tipos:

- Aprendizaje hebbiano: pretende medir la familiaridad o extraer características de los datos de entrada. Este aprendizaje consiste básicamente en el ajuste de los pesos de las conexiones de acuerdo con la correlación de los valores de activación de las dos neuronas conectadas:

$$\text{Incr}(w_{ji}) = y_i y_j \quad (3.44)$$

Si las dos unidades son salidas positivas, se produce un reforzamiento de la conexión. Si por el contrario, si la salida es negativa, se produce un debilitamiento de la conexión. Por tanto, la modificación de los pesos se realiza en función de los estados de las neuronas, obtenidos tras la presentación de cierto estímulo, sin tener en cuenta si se deseaba obtener o no esos estados de activación.

- Aprendizaje competitivo y cooperativo: las neuronas compiten unas con otras con el fin de llevar a cabo una tarea dada. Las neuronas compiten por activarse, quedando las perdedoras a sus valores de respuesta mínimos. La conexión entre neuronas se realiza en todas las capas de la red, existiendo en estas neuronas conexiones recurrentes de autoexcitación y conexiones de inhibición por parte de neuronas vecinas.

La diferencia entre ambos tipos de aprendizaje estriba en la existencia o no de un agente externo que controle todo el proceso. Así como en considerar si la red puede aprender durante su funcionamiento o requiere de una fase previa de entrenamiento. En este último debe existir un conjunto de datos de entrenamiento y un conjunto de datos de prueba; igualmente los pesos de las conexiones no se modifican después de terminar la etapa de entrenamiento de la red. En la red no supervisada los pesos varían dinámicamente cada vez que se presente una nueva información al sistema [Azuela, 2013].

### 3.5.5. Red neuronal convolucional

Una red neuronal convolucional es un tipo de red neuronal artificial donde las neuronas corresponden a campos receptivos de una manera muy similar a las neuronas en la corteza visual primaria de un cerebro biológico. Este tipo de red es una variación de un perceptrón multicapa, pero su funcionamiento las hace mucho más efectivas para tareas de visión artificial, especialmente en la clasificación de imágenes. Los fundamentos de las redes neuronales convolucionales se basan en el Neocognitron (Red neuronal creada para el reconocimiento de caracteres escritos a mano, basado en el estudio de la corteza visual del ojo humano), introducido por Kunihiko Fukushima en 1980. Este modelo fue más tarde mejorado por [Yann LeCun et al., 1982] al introducir un método de aprendizaje basado en Backpropagation para poder entrenar el sistema correctamente. En el año 2012, fueron refinadas por Dan Ciresan et al., y fueron implementadas en un GPU consiguiendo así resultados impresionantes.

#### Arquitectura

Una red neuronal convolucional es una red multicapa que están contruidas con una estructura que contiene tres tipos de capas diferentes, ver Figura 3.17 [Rodríguez Carmona J., 2016]:

1. Una capa convolucional, que es la que le da el nombre a la red.
2. Una capa de reducción o de pooling, la cual va a reducir la cantidad de parámetros al quedarse con las características más comunes.
3. Una capa clasificadora totalmente conectada, la cual va a dar el resultado final de la red.

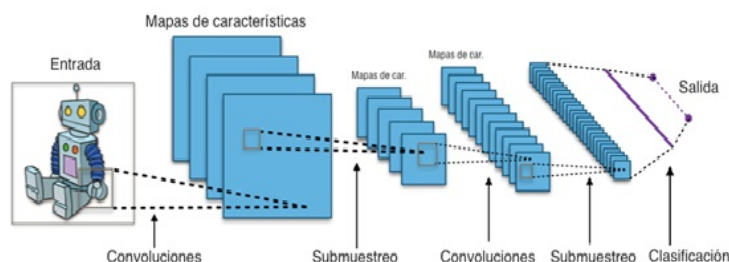


Figura 3.17: Red neuronal convolutiva

**Capa convolucional:** es una operación de productos y sumas entre la imagen de entrada y un filtro que genera un mapa de rasgos. Los rasgos extraídos corresponden a cada posible ubicación del filtro en la imagen original. La ventaja es que el mismo filtro sirve para extraer el mismo rasgo en cualquier parte de la imagen. Esto reduce el número de conexiones y el número de parámetros a entrenar en comparación con una red multicapa de conexión total, ver Figura 3.18 [Rodríguez Carmona J., 2016].

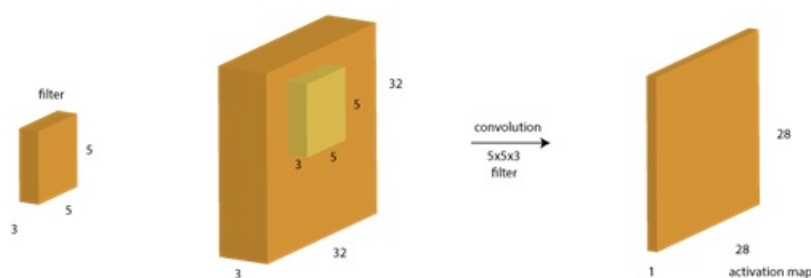


Figura 3.18: Convolución

**Capa de Pooling:** es una capa de reducción, la tarea principal es la disminución de los parámetros (ancho, alto) de la imagen, para ser usada por la siguiente capa convolucional, una de las principales desventajas de aplicar esta capa es de que cada vez que se aplica se pierde información por lo que también se le conoce como reducción de muestreo. La operación más usada en esta capa es la de max-poolin, la imagen original se divide en rectángulos y de cada uno se va obteniendo el valor más grande entre ellos, los valores seleccionados son usados para generar la siguiente capa, en la Figura 3.19 [Rodríguez Carmona J., 2016].

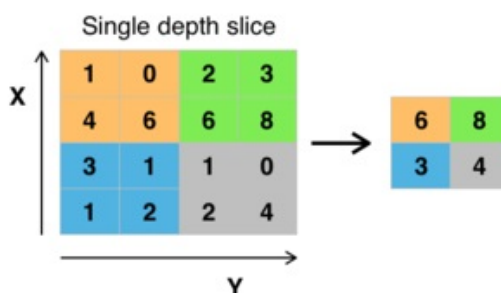


Figura 3.19: Max pooling

Otra operación que se puede utilizar es min-poolin que al igual que el max-poolin se divide la imagen en rectángulos, solo que aquí se tomará el valor más pequeño de cada uno de los rectángulos.

**Capa de clasificación (totalmente conectada):** es la parte final de la red neuronal, todas las neuronas de la capa final están conectadas con todas las neuronas de la capa anterior. Por lo tanto es aquí donde se calculan las salidas para las diferentes clasificaciones de las imágenes, ver Figura 3.20 [Rodríguez Carmona J., 2016].

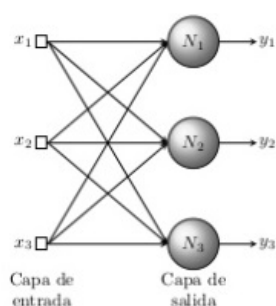


Figura 3.20: Conexión de las neuronas

En la visión artificial se usan las redes neuronales convolutivas, para poder aprender a clasificar todo tipo de datos donde estos estén distribuidos de una forma continua a lo largo del mapa de entrada, y a su vez sean estadísticamente similares en cualquier lugar del mapa de entrada. Por esta razón, son especialmente eficaces para clasificar imágenes [Rodríguez Carmona J., 2016].

### 3.6. TensorFlow Network de Google

TensorFlow es una biblioteca de código abierto desarrollada por Google que permite realizar cálculos numéricos usando diagramas de flujo de datos, que describen cálculos matemáticos con un grafo de nodos y arcos. Los nodos representan operaciones matemáticas y se asignan a los dispositivos computacionales, donde se ejecutan de forma asincrónica y en paralelo una vez que todos los tensores en los arcos de entrada están disponibles. Los nodos al realizar algún cálculo producen cero o más tensores. Mientras que los arcos o aristas describen las relaciones de entrada o salida entre los nodos y están representados por los arreglos de



datos multidimensionales o tensores. El flujo de los tensores a través del grafo es el que le da el nombre de tensorflow.

Un grafo de TensorFlow es una descripción de cálculos. Para calcular cualquier cosa dentro de TensorFlow, el grafo debe ser lanzado dentro de una sesión. La sesión coloca las operaciones del grafo en los diferentes dispositivos, tales como CPU o GPU, y proporciona métodos para ejecutarlas. Esta arquitectura flexible permite realizar los cálculos en más de un CPU o GPU utilizando la misma API.

Tensorflow utiliza una red neuronal convolucional inception V3, la cual fue utilizada en el desafío de reconocimiento visual ImageNet en el 2012. Donde el objetivo principal era clasificar imágenes completas en 1000 categorías diferentes. Esta red neuronal es gratuita y está desarrollada en Python. Inception V3 consta de dos partes:

1. Extracción de características con una red neuronal convolucional.
2. Clasificación con capas totalmente conectadas y softmax.

El modelo extrae las características generales de las imágenes de entrada en la primera parte y las clasifica según las características de la segunda parte. En el aprendizaje de transferencia, cuando construye un nuevo modelo para clasificar el conjunto de datos originales, reutiliza la parte de extracción de características y vuelve a entrenar la parte de clasificación con el conjunto de datos. Debido a que no tiene que entrenar la parte de extracción de características (que es la parte más compleja del modelo), puede entrenar el modelo con menos recursos de computación y tiempo de capacitación. En la figura 3.21, se muestra la arquitectura de inception V3.

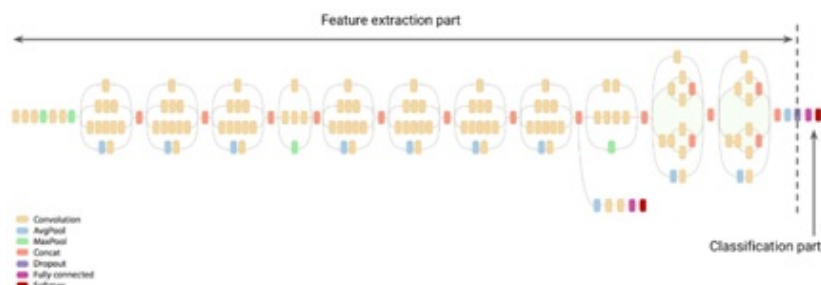


Figura 3.21: Estructura de Inception V3

### 3.7. Máquina de Vectores de Soporte (SVM)

SMV es un sistema de aprendizaje automático el cual ayuda en la búsqueda de soluciones en la clasificación y regresión, el cual está basado en la teoría de aprendizaje estadístico. Su eficiencia se debe a su fundamentación matemática solida, a que está basada en el concepto de la minimización del riesgo estructural y a disponer de potentes herramientas y algoritmos para encontrar la solución óptima.

Una SMV transforma los datos de entrada por medio de una función kernel, que emplea el producto escalar, a un espacio de características de mayor dimensión, donde es posible encontrar el hiperplano óptimo que separe las clases y maximice el margen  $m$  entre ellas, como se puede observar en la figura 3.22 [Riaño Melo O., 2016].

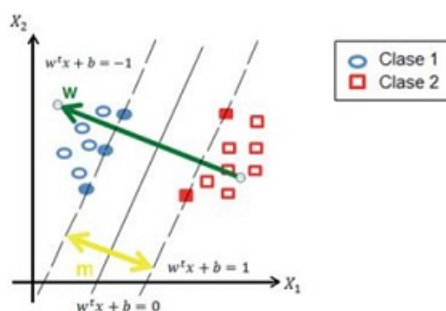


Figura 3.22: Hiperplano óptimo de separación entre dos clases en el espacio transformado.

Maximizar el margen  $m$  es un problema de programación cuadrática ( $QP$ ) y puede ser resuelto por su problema dual introduciendo multiplicadores de Lagrange. Sin ningún conocimiento del mapeo, la SVM encuentra el hiperplano óptimo utilizando el producto punto con funciones en el espacio de características que son llamadas kernels. La solución del hiperplano óptimo puede ser escrita como la combinación de unos pocos puntos de entrada que son llamados vectores de soporte.

La función kernel calcula el producto escalar de los puntos de entrada en el espacio de características, realizando el cálculo en el espacio de entrada cuyo grado de complejidad es menor. La función es la siguiente:

$$K : R^n \times R^n \rightarrow R \quad (3.45)$$

definida por

$$K(X_i, X_j) = \phi(X_j) \cdot \phi(X_j) \quad (3.46)$$

Los tipos de funciones kernel más utilizadas son la polinomial – homogéneo, perceptrón, función de base radial y sigmoidea [Riaño Melo O., 2016].

### 3.8. Clasificador Random Forest

Random forest es un método que combina una cantidad grande de árboles de decisión independientes probados sobre conjuntos de datos aleatorios con igual distribución. Cada uno de los árboles de clasificación se construye utilizando una muestra de arranque de los datos, y en cada división el conjunto de variables candidatas, es un subconjunto aleatorio de las variables.

Por lo tanto, random forest utiliza ensacado (agregación de bootstrap), un enfoque exitoso para combinar aprendizajes inestables y selección aleatoria de variables, para la construcción de árboles. Cada árbol no se poda (se cultiva completamente), para obtener árboles de bajo sesgo; al mismo tiempo, el empaquetamiento y la selección aleatoria de variables dan como resultado una baja correlación de los árboles individuales. El algoritmo produce un conjunto que puede lograr tanto un sesgo bajo como una baja varianza.

La fase de aprendizaje consiste en crear muchos árboles de decisión independientes, construyéndolos a partir de datos de entrada ligeramente distintos. Se altera, por tanto, el conjunto inicial de partida, haciendo lo siguiente:

- Se selecciona aleatoriamente con reemplazamiento un porcentaje de datos de la muestra total.

Es habitual incluir un segundo nivel de aleatoriedad, esta vez afectando los atributos:

- En cada nodo, al seleccionar la partición óptima, se tiene en cuenta sólo una porción de los atributos, elegidos al azar en cada ocasión.

Una vez que se tienen muchos árboles, la fase de clasificación se lleva a cabo de la siguiente forma:

- Cada árbol se evalúa de forma independiente y la predicción del bosque será la media de los árboles totales que se generaron. La proporción de árboles que toman una misma respuesta se interpreta como la probabilidad de la misma.

Sus principales ventajas son:

- Es uno de los algoritmos de aprendizaje más certeros que hay disponible. Para un set de datos lo suficientemente grande produce un clasificador muy certero.
- Corre eficientemente en bases de datos grandes.
- Puede manejar cientos de variables de entrada sin excluir ninguna.
- Da estimados de qué variables son importantes en la clasificación.
- Tiene un método eficaz para estimar datos perdidos y mantener la exactitud cuando una gran proporción de los datos está perdida.
- Computa los prototipos que dan información sobre la relación entre las variables y la clasificación.
- Computa las proximidades entre los pares de casos que pueden usarse en los grupos, localizando valores atípicos, o (ascendiendo) dando vistas interesantes de los datos.
- Ofrece un método experimental para detectar las interacciones de las variables.

En los algoritmos basados en Random forest, un límite superior se pueden derivar para el error de generalización en términos de dos parámetros que son medidas de la precisión de los clasificadores individuales y de la dependencia entre ellos. La interacción entre estos dos da la base para la comprensión del funcionamiento de los bosques al azar [Díaz-Uriarte R., 2006].

## 3.9. Herramientas de programación

### 3.9.1. Python

Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas [Marzal A., 2002].

### 3.9.2. Matlab

Es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio llamado M. Es empleado para diferentes funciones en distintas disciplinas científicas y se ha convertido prácticamente en un estándar de programación y desarrollo rápido de aplicaciones. La unidad de procesamiento de Matlab es la matriz y por ello es muy útil en el procesamiento de imágenes [Cuevas E., 2010].

### 3.9.3. Weka

WEKA es un software que tiene como objetivo proporcionar una colección completa de algoritmos de aprendizaje automático y preprocesamiento de datos, es una herramienta para investigadores y profesionales por igual. Su arquitectura modular y extensible permite sofisticados procesos de minería de datos para ser construidos de la amplia colección de algoritmos de aprendizaje básico y herramientas proporcionadas. El banco de trabajo incluye algoritmos para regresión, clasificación, agrupamiento, extracción de reglas de asociación y selección de atributos [Hall M., 2009].

---

---

## CAPÍTULO 4

---

# Metodología

### Contenido

---

<b>4.1. Introducción</b>	<b>59</b>
<b>4.2. Tipo de Investigación</b>	<b>60</b>
<b>4.3. Universo, población o unidades de análisis</b>	<b>60</b>
<b>4.4. Adquisición de la imagen</b>	<b>61</b>
<b>4.5. Pre-procesamiento de la imagen</b>	<b>62</b>
4.5.1. Obtener área de interés	62
4.5.2. Filtros y operaciones morfológicas	62
4.5.3. Realce de contrastes	64
<b>4.6. Segmentación</b>	<b>65</b>
4.6.1. Segmentación manual	65
4.6.2. Binarización	65
4.6.3. Método de Otsu	66
4.6.4. Crecimiento de regiones	67
4.6.5. Método matemático de punto interior a un triángulo	68
4.6.6. Generación de base de datos de pixeles enfermos	72
4.6.7. K-means clustering	73
<b>4.7. Extracción de características estadísticas</b>	<b>74</b>
<b>4.8. Extracción de características de frecuencia de dominio con transformadas wavelets</b>	<b>75</b>
<b>4.9. TensorFlow Network de Google</b>	<b>75</b>
4.9.1. Instalar TensorFlow	76
4.9.2. Entrenamiento y clasificación con TensorFlow	76
<b>4.10. Clasificador SVM y Random Forest</b>	<b>77</b>

---

## 4.1. Introducción

En este capítulo se muestra la metodología propuesta para la identificación de la *Botrytis cinérea* en la granada por medio del procesamiento de imágenes. La metodología propuesta se muestra en la Figura 4.1. La cual se divide en varias etapas: adquisición de imágenes, pre-procesamiento, segmentación y clasificación. Para cada etapa se analizaron diferentes métodos para poder identificar cual de ellas brinda mejores resultados. Para la implementación de los métodos se utilizó Matlab R2014a, Photoshop CS6, Python 2.7 y Tensorflow. Cada una de las etapas se describe en el desarrollo de este capítulo.

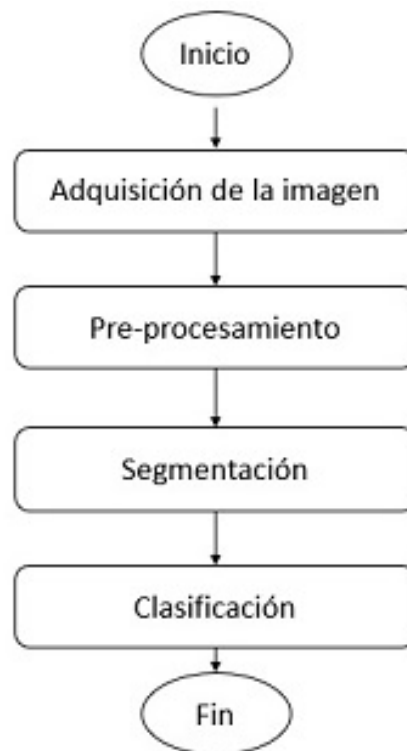


Figura 4.1: Metodología utilizada para la identificación de la *Botrytis* en la granada.

## 4.2. Tipo de Investigación

Para llevar a cabo la elaboración del proyecto se realizó una investigación experimental. Ya que se realizará observación, control y medición del procesamiento de las imágenes para detectar la enfermedad en las granadas. Los resultados son cuantitativos y cualitativos los cuales ayudarán en la calidad de la selección de la granada para la producción de sus derivados.

## 4.3. Universo, población o unidades de análisis

La investigación del proyecto apoyará a las empresas que se dedican a la cosecha y producción de los derivados de la granada en la región sur de Jalisco. Se tomará como muestra la empresa GEGEMA, S.P.R. DE R.L ubicada en el Rancho la Herradura, en las Rancherías de Cd. Guzmán, Jalisco. Siendo esta empresa interesada y pionera en la producción de derivados de la granada, por lo cual el muestreo será no probabilístico. Las unidades de análisis serán las granadas cosechadas en el Rancho la Herradura y granadas wonderfull de proveedores de la región, de las cuales se obtendrá la imagen para procesarla y adquirir los datos.

Se realizó una estadía profesional en la empresa GEGEMA S.P.R de R.L., ubicada en el Rancho la Herradura del municipio de Zapotlán el Grande. Empezando el 23 de Agosto y culminando el 30 de Septiembre del 2017.

Durante la estadía en la empresa, se realizó el análisis de los requerimientos y el estudio de las enfermedades de las granadas que afectan a la producción de los derivados de esta. La empresa requiere que se le ayude con la selección de la fruta pos cosecha, su objetivo principal es que lleguen a la mesa de corte solo granadas sin Botrytis para la producción. Descartando las granadas con Botrytis. Las actividades realizadas en el estudio de campo fueron:

- Entrenamiento para la selección de la granada.
- Corte de la coronilla de las granadas. Desechando las granadas que se observan con algún estado inaceptable para el procesamiento, como las granadas con Botrytis, corazón negro, abiertas con los arilos putrefactos, etc. Muchas de las granadas enfermas no son detectadas en ese momento ya que no todo el personal cuenta con toda la experiencia ni la capacitación para desechar la fruta.



- Pasar las granadas a corte, es aquí donde realmente se realiza la selección de la granada, ya que al realizar los cortes se observa el estado real de la granada.
- Realizar cortes en las granadas para la separación de los arilos.
- Recopilación de fotografías para la base de datos de granadas.
- Al seleccionar las granadas se realizaron las siguientes observaciones:
  - No todas las granadas con la piel rugosa y dura son granadas en mal estado, algunas de ellas son granadas que están verdes.
  - Las granadas con Botrytis, la cual es la más fácil de detectar, aún siguen estando presente en la mesa de corte.
  - Las granadas abiertas o con grietas son aceptadas para procesar siempre y cuando todas las partes de la granada se encuentren sanas.
  - Las granadas duras tienen algo en común, y es la forma de la fruta, estas frutas no tienen una forma muy redondeada, son más como una forma hexagonal, y la piel de la granada es un poco más opaca, pero esto no es exclusivo de las granadas enfermas, también se presentan estas condiciones en las frutas que están verdes.
  - Algunas granadas empiezan a tener Botrytis por dentro sin que se note algún síntoma en la piel. Se realizan los dos primeros cortes y la granada se observa en buen estado, al partir a la mitad la granada, se observa la putrefacción en el centro contaminando arilos y la corteza, en otros casos solo se ve la corteza, podrida o con manchas cafés, sin afectación en los arilos.
  - La forma de la granada no influye en las enfermedades, ya que se procesaron granadas no muy frescas, las cuales presentaban formas no esféricas e irregulares, las cuales estaban sanas.

#### 4.4. Adquisición de la imagen

En primera instancia se adquieren imágenes de granadas enfermas. Para obtener dichas imágenes se visitó la empresa GEGEMA, S.P.R. DE R.L, donde se generó una base de datos de 1093 imágenes, las cuales se comprenden de imágenes de granadas enfermas con Botrytis, Alternaria y granadas sin enfermedades. Todas las imágenes fueron guardadas en formato JPG. Las imágenes fueron tomadas con una cámara Canon EOS t16 de 18 Mp, en una

cabina fotográfica con luz blanca, ver Figura 4.2. Esta cabina se implementó con la finalidad de que las imágenes tuvieran las mismas condiciones de luz independientemente del horario y las condiciones climáticas.

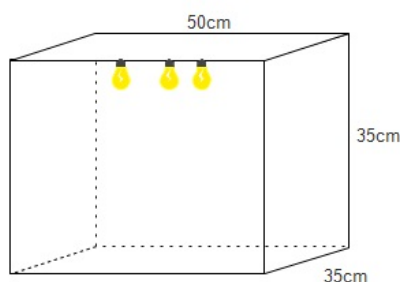


Figura 4.2: Cabina fotográfica de luz blanca

## 4.5. Pre-procesamiento de la imagen

### 4.5.1. Obtener área de interés

Las imágenes capturadas tienen fondo blanco, una etiqueta y la granada. Las imágenes son de 5184 x 3456 píxeles, por lo que se requirió la reducción de la imagen seleccionando solo la granada, ya que es la zona que se requiere trabajar para la identificación de la enfermedad. La reducción de la imagen se llevó a cabo con la finalidad de reducir la carga computacional para el procesamiento posterior. Esta fase se realizó con el software Photoshop CS6, donde se selecciona el área de interés y la imagen obtenida se guarda en formato JPG.

### 4.5.2. Filtros y operaciones morfológicas

Una vez obtenida el área de interés, las imágenes se pasaron por filtros para eliminar el posible ruido presentado en las imágenes, generando una imagen filtrada y otra en blanco y negro, estos filtros son: Gauss, mediana, media, Prewit, Sobel y Laplaciant. Este proceso se realizó utilizando Matlab R2014a y los pasos se especifican en el Algoritmo 1:

---

**Algoritmo 1** Aplicación de filtros

---

- 1: Se lee la imagen de entrada.
  - 2: Para obtener las imágenes en blanco y negro se convierte a escala de grises.
  - 3: Crear el filtro dimensional con  $f_{special}(filtro)$  donde  $filtro$  puede ser Gaussian, Sobel, Laplacian, average (media), Prewit. En el caso de la mediana este paso no es necesario ya que se aplica directamente la instrucción  $medfilt2(Imagen)$ .
  - 4: Se aplica el filtro multidimensional.
  - 5: Se guardan las imágenes filtradas a color y escala de grises.
- 

Una vez aplicados los filtros estos son evaluados con PSNR y MAE para definir que filtro proporciona imágenes más limpias. Los pasos se especifican en el Algoritmo 2.

---

**Algoritmo 2** Evaluación PSNR y MAE

---

- 1: Leer la imagen filtrada.
  - 2: Para  $MAE$  aplicar la función  $mae(ImagenFiltrada, ImagenOriginal)$ .
  - 3: Para  $PSNR$  la ecuación  $mse = \frac{sum((imagenOriginal(:)-ImagenFiltrada(:))^2)}{numel(imagenOriginal)}$ ;  $psnr = 10 * \log_{10}(255 * 255 / mse)$ .
  - 4: Guardar los resultados en un archivo de texto.
- 

Una vez definido el filtro que se va a utilizar, se aplican las siguientes operaciones morfológicas:

1. Binarización.
2. Ecuilibrado del histograma.
3. Dilatación.
4. Erosión.
5. Erosión - Dilatación.
6. Dilatación - Erosión.

La finalidad de generar estas imágenes es poder evaluar imágenes con diferentes características y poder evaluarlas en la etapa de clasificación en el entrenamiento de la red neuronal. Los pasos se especifican en el Algoritmo 3.

---

**Algoritmo 3** Aplicación de operaciones morfológicas

---

- 1: Leer la imagen de entrada.
  - 2: Aplicar la función correspondiente:
    - Binarización.
    - Ecuilibrado del histograma.
    - Dilatación.
    - Erosión.
    - Erosión – Dilatación.
    - Dilatación - Erosión. Se crea un elemento de estructuración en forma de disco con un radio de 10 píxeles.
  - 3: Guardar la imagen resultante.
- 

#### 4.5.3. Realce de contrastes

Los colores que componen la granada son tonos rojos, amarillos y cafés. La enfermedad en su mayoría es de color café, marrón y tonos tintos, y la parte sana se compone de colores amarillos, rojos, tintos, marrones. En algunas granadas los colores de la enfermedad y la parte sana a nivel pixel, no tiene mucha diferencia. Es por ello que se requiere realizar un realce de contrastes, para poder acentuar más la parte enferma y poder segmentarla. Para este caso se manipuló el brillo y se mezclaron los canales en el espacio de color RGB. Los pasos se especifican en el Algoritmo 4.

---

**Algoritmo 4** Realce de contraste

---

- 1: Leer imagen de entrada.
  - 2: Convertir la imagen en tipo double.
  - 3: Extraer cada plano de color  $R$ ,  $G$  y  $B$  y aumentar el brillo en 50.
  - 4: En el canal  $G$ , mezclar los colores multiplicando  $R * .34, G * 686, B * .268$ .
  - 5: En el canal  $B$ , mezclar los colores multiplicando  $R * 0, G * .634, B * .131$ .
  - 6: Crear la imagen de salida uint8, concatenando las matrices en una sola dimensión con  $cat(3, R, G, B)$ .
  - 7: Guardar la imagen resultante.
-

## 4.6. Segmentación

La segmentación de imágenes se refiere al proceso de partición de una imagen digital en regiones u objetos que la constituyen para obtener una imagen final más significativa y fácil de analizar. En este caso el objetivo de la segmentación es identificar las regiones de la imagen que contengan la enfermedad de la Botrytis. Hay varias técnicas para la segmentación de imágenes las cuales se describen en esta sección.

### 4.6.1. Segmentación manual

Se extrajo la enfermedad manualmente, realizando recortes del área contaminada de un tamaño de 657x657 pixeles, con el software Photoshop. Así mismo, se realizaron recortes de partes no contaminadas del mismo tamaño.

### 4.6.2. Binarización

Por medio de la binarización se quiere obtener el área identificada como Botrytis, reduciendo la información de la imagen a 0 y 1. Este proceso se realiza sobre el canal  $R$  del espacio de color RGB. Para ello los pasos se especifican en el Algoritmo 5.

---

**Algoritmo 5** Binarización

---

- 1: Leer imagen de entrada.
  - 2: Obtener el plano  $R$  de la imagen.
  - 3: Calcular el umbral automático de la imagen.
  - 4: Binarizar la imagen.
  - 5: Eliminar picos en la imagen con procesamiento morfológico de apertura (Erosión - Dilatación).
  - 6: Eliminar huecos en la imagen con procesamientos morfológico de cierre (Dilatación - Erosión).
  - 7: Etiquetar los objetos de la imagen resultante.
  - 8: Calcular el área de los objetos.
  - 9: Hacer una máscara con el área mayor y convertirla a `double`.
-

- 
- 10: Aplicar la máscara a cada uno de los planos del espacio de color RGB de la imagen original.
  - 11: Convertir a uint8 los planos RGB.
  - 12: Guardar la imagen final.
- 

### 4.6.3. Método de Otsu

Se realizaron dos algoritmos de Otsu diferentes, las imágenes de entrada para ambos, son las imágenes originales y las imágenes con el realce de contrastes, cada algoritmo se prueba con variables de nivel de intensidad diferentes, para probar cual proporcionaba mejores resultados. El primer algoritmo es el Algoritmo 6.

---

**Algoritmo 6** Primer método de Otsu

---

- 1: Leer imagen de entrada.
  - 2: Convertir la imagen RGB a escala de grises.
  - 3: Se calcula el histograma.
  - 4: Se calcula el promedio total del histograma.
  - 5: **Para** nivel de intensidad 1:256 **hacer**
  - 6:     La ecuación (3.13).
  - 7:     Si el resultado es igual a 0, continua.
  - 8:     Se calcula la ecuación (3.14).
  - 9:     Si el resultado es diferente de 0 continua, de lo contrario termina.
  - 10:    Se generan dos clases de niveles de intensidad A y B.
  - 11:    Se calcula la media de cada una de las clases.
  - 12:    Se calcula la varianza entre las clases con la ecuación (3.19).
  - 13:    Si los valores son los máximos, se va generando la imagen final.
  - 14: Guardar imagen final.
- 

El segundo algoritmo es el Algoritmo 7.

---

**Algoritmo 7** Segundo método de Otsu

---

- 1: Leer imagen de entrada.
  - 2: Separar los planos del espacio de color RGB, para obtener R.
  - 3: Calcular el umbral promedio.
  - 4: **Para** Para cada pixel del plano de color R **hacer**
-

---

```
5:   Si pixel  $\leq$  umbral promedio entonces
6:     imagen final = 0
7:   si no
8:     imagen final = 255
9: Guardar imagen.
```

---

#### 4.6.4. Crecimiento de regiones

Para realizar la segmentación por medio de la similitud de los píxeles enfermos se utiliza el método de crecimiento de regiones. Este permite elegir un píxel como punto de partida, lo que permite elegir un píxel enfermo para la muestra y de allí realizar la búsqueda de los píxeles similares, los pasos se especifican en el Algoritmo 8.

---

**Algoritmo 8** Crecimiento de regiones

---

```
1: Leer imagen de entrada.
2: Convertir imagen a escala de grises.
3: Convertir la imagen a double y mostrar imagen.
4: Indicar la semilla inicial, haciendo clic en el área de la enfermedad.
5: Guardar el valor de la semilla.
6: Obtener valores de las coordenadas  $[x, y]$ .
7: Definir la disimilitud permisible.
8: Generar matriz inicializada con ceros, del mismo tamaño de la imagen de entrada.
9: Número de píxeles en la región igual a 1.
10: Reservar memoria para la operación.
11: Generar una lista.
12: Definir matriz para la búsqueda local.
13: Inicializar la similitud inicial en 0.
14: Mientras la similitud sea  $<$  a la disimilitud permitida and el número de píxeles en la
    región sea  $<$  al número de elementos en la imagen hacer
15:   Realizar la búsqueda local.
16:   Calcular la nueva posición.
17:   Verificar si el píxel está dentro de la imagen.
18:   Verificar si el píxel se encuentra en la lista, si no ésta se agrega.
19:   Verificar la similitud del píxel con la lista.
20:   Calcular nueva similitud de la región segmentada.
21:   El píxel encontrado es el nuevo píxel.
```

---

---

22: Se eliminan características de la lista.

23: Guardar imagen final.

---

#### 4.6.5. Método matemático de punto interior a un triángulo

Para la implementación de este método matemático se tomaron 60 de las imágenes segmentadas manualmente. Para el espacio de color RGB, se graficó  $R$  y  $G$ ,  $R$  y  $B$ ,  $G$  y  $B$ , los resultados se muestran en las Figuras 4.3, 4.4, 4.5 respectivamente.

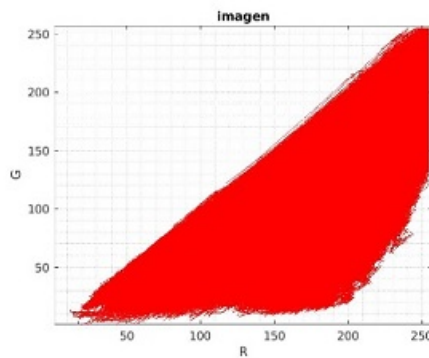


Figura 4.3: Gráficas de los píxeles de los planos  $R$  y  $G$ .

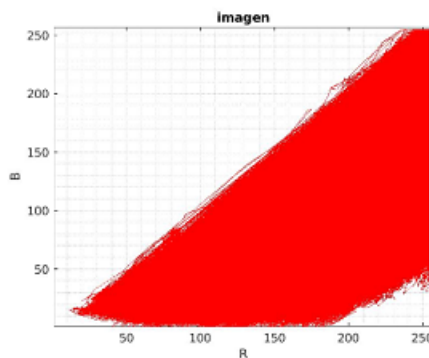


Figura 4.4: Gráficas de los píxeles de los planos  $R$  y  $B$ .



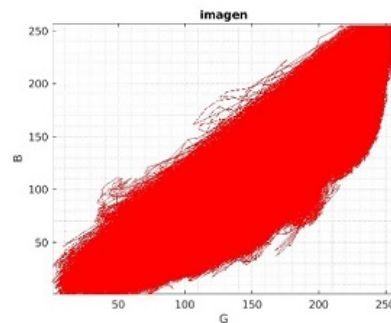


Figura 4.5: Gráficas de los pixeles de los planos  $G$  y  $B$ .

Para el espacio de color HSV se graficaron  $S$  y  $V$  Figura 4.6.

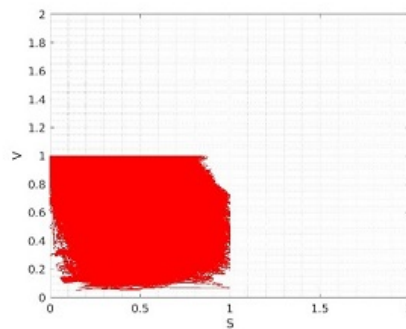


Figura 4.6: Gráficas de los pixeles de los planos  $S$  y  $V$ .

En el espacio de color CIELab se graficaron los pixeles de  $a$  y  $b$  Figura 4.7.

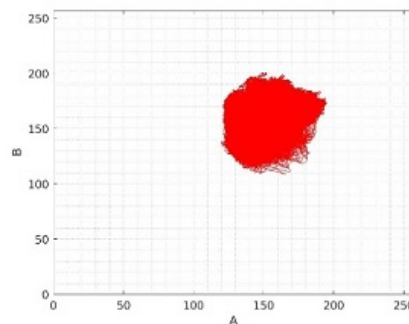


Figura 4.7: Gráficas de los pixeles de los planos  $a^*$  y  $b^*$ .

Una vez obtenidas las gráficas, se trazan rectángulos o cuadrados para los espacios de color RGB, HSV y CIELab, dentro del área graficada. Las Figuras 4.8, 4.9, 4.10, 4.11, 4.12, muestran los resultados respectivamente.

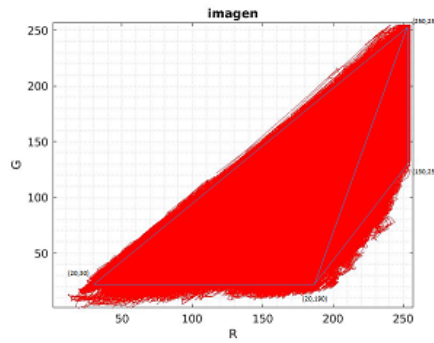


Figura 4.8: Trazo de rectángulos en el espacio de color RGB en los planos R y G.

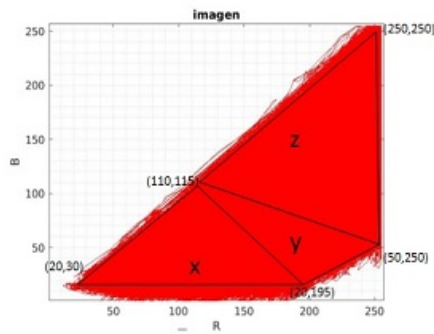


Figura 4.9: Trazo de rectángulos en el espacio de color RGB en los planos R y B.

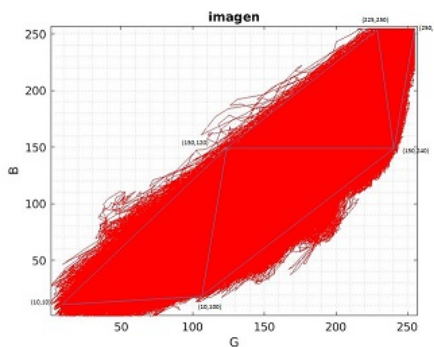


Figura 4.10: Trazo de rectángulos en el espacio de color RGB en los planos G y B.

Para el espacio de color HSV se graficaron  $S$  y  $V$  Figura 4.6.

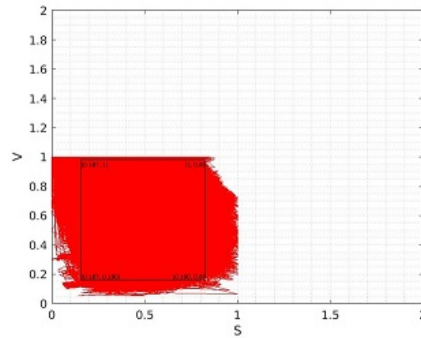


Figura 4.11: Trazo de cuadrados en el espacio de color HSV en los planos  $S$  y  $V$ .

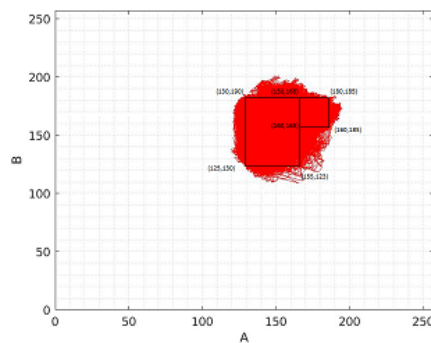


Figura 4.12: Trazo de cuadrados en el espacio de color CIELab en los planos  $a^*$  y  $b^*$ .

Se toman los valores de las esquinas de los triángulos o cuadrados, los cuales son utilizados en los algoritmos programados, los pasos se especifican en el Algoritmo 9.

---

**Algoritmo 9** Método matemático de punto interior a un triángulo

---

- 1: Leer imagen de entrada.
  - 2: Según sea la gráfica se deben obtener los planos del espacio de color graficados:
    - Gráfica R y G: obtener espacios R ( $I(:, :, 1)$ ) y G ( $I(:, :, 2)$ ) de la imagen.
    - Gráfica R y B: obtener espacios R ( $I(:, :, 1)$ ) y B ( $I(:, :, 3)$ ) de la imagen.
    - Gráfica B y G: obtener espacios B ( $I(:, :, 3)$ ) y G ( $I(:, :, 2)$ ) de la imagen.
    - Gráfica H y S: obtener los espacios H ( $I(:, :, 2)$ ) y V ( $I(:, :, 3)$ ) de la imagen.
    - Gráfica a y b: obtener los espacios a ( $I(:, :, 2)$ ) y b ( $I(:, :, 3)$ ) de la imagen.
-

- 3: Se toma cada uno de los pixeles de cada plano y se busca dentro de los triángulos/cuadrados según corresponda, los cuales conforman el área graficada, esto se hace por medio de una función llamada *pointIntoATriangle*, esta función recibe los pixeles a buscar y retorna un 1, si el dato fue encontrado o un 0 si el dato no fue encontrado. Esta función se manda llamar por cada triangulo/cuadrado generado en la gráfica, y los pixeles se buscan en cada uno de los triángulos/cuadrados, ver Figura 4.13.

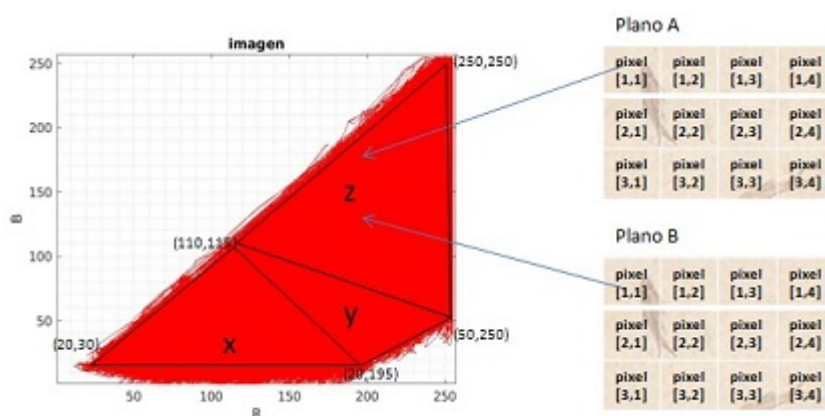


Figura 4.13: Punto interior a un triángulo.

Los pixeles se buscan en los tres triángulos enviando los datos de la siguiente manera:

$x = \text{pointIntoATriangle}(A(i,j), B(i,j), 20, 30, 195, 20, 110, 115);$

$y = \text{pointIntoATriangle}(A(i,j), B(i,j), 20, 195, 110, 115, 20, 250);$

$z = \text{pointIntoATriangle}(A(i,j), B(i,j), 110, 115, 50, 250, 250, 250);$

- 4: Si el pixel es encontrado dentro de algún triángulo, entonces el pixel de la imagen de entrada se deja como esta, de lo contrario su valor es igual a 255.
- 5: Guardar imagen final.

#### 4.6.6. Generación de base de datos de pixeles enfermos

Para este método se utilizaron 60 imágenes de la segmentación manual, para la generación de una base de datos de los valores de los pixeles del espacio de color RGB. Esta base de datos se generó como se muestra en el Algoritmo 10.

---

**Algoritmo 10** Generar base de datos

---

- 1: Leer imagen de entrada.
  - 2: Concatenar los valores de los planos de color de R (200), G (176), B (59), para obtener valores únicos, por ejemplo 20017659.
  - 3: Los valores fueron almacenando en un archivo de texto.
- 

Una vez obtenida la base de datos, se programa el algoritmo de búsqueda, los pasos se muestran en el Algoritmo 11.

---

**Algoritmo 11** Búsqueda de pixeles

---

- 1: Leer imagen de entrada.
  - 2: Se abre el archivo de la base de datos.
  - 3: Se almacenan los datos en un arreglo.
  - 4: **Para** Para cada pixel en la imagen de entrada **hacer**
  - 5:     Se extrae los pixeles de los planos de RGB.
  - 6:     Se concatenan los valores.
  - 7:     El valor generado se busca en la base de datos. Si se encuentra se deja el pixel de esa posición de la imagen original en la imagen de salida. De lo contrario el pixel es igual a 255 (blanco).
  - 8: Se guarda la imagen final.
- 

#### 4.6.7. K-means clustering

Para este método se utilizaron los espacios de color RGB, CIElab y HSV, y cuatro métricas de distancia: suma de diferencia absoluta, distancia euclidiana, coseno y correlación. El agrupamiento de K-means es un  $K=3$ . Los pasos se muestran en el Algoritmo 12.

---

**Algoritmo 12** K-means clustering

---

- 1: Leer imagen de entrada.
  - 2: Para el caso de CIElab y HSV, realizar la conversión de espacio de color.
  - 3: Para CIElab tomar y agrupar las capas a y b, para HSV las capas S y V.
  - 4: Sacar el tamaño de las dimensiones de la matriz, para generar la matriz final.
  - 5: Etiquetar cada pixel en la imagen usando la función *kmeans()* de Matlab, con un  $K = 3$  y las métricas de distancia, suma de diferencia absoluta, distancia euclidiana, coseno y correlación.
-

- 
- 6: Crear imágenes de la segmentación, separando los objetos etiquetados por color, en este caso serán 3 imágenes.
  - 7: Aplicar la operación morfológica de erosión a las segmentaciones.
  - 8: Guardar imágenes finales.
- 

## 4.7. Extracción de características estadísticas

Otra manera de clasificar una imagen es extraer las características de las texturas, esto se realiza con un algoritmo que se desarrolló en Matlab, que da como resultado un archivo con la información que se utiliza en los clasificadores Random Forest y SMV. Los pasos se especifican en el Algoritmo 13.

---

**Algoritmo 13** Extracción de características estadísticas

---

- 1: Leer imagen de entrada.
  - 2: Dividir la imagen en bloques de  $3 \times 3$ .
  - 3: **Para** En cada bloque **hacer**
  - 4:     Convertir la matriz en un vector.
  - 5:     Calcular el histograma del vector  $n$  con un umbral de 256.
  - 6:     Sumar la frecuencia del histograma  $n$ .
  - 7:     **Para** Para cada valor del histograma, desde  $x=0$  **hacer**
  - 8:         Obtener  $p$  dividiendo la frecuencia en la posición  $x$  entre  $n$ .
  - 9:         Si  $p$  es diferente de 0: calcular la media, energía, y entropía.
  - 10: Calcular asimetría.
  - 11: Calcular desviación estándar.
  - 12: Calcular la suavidad.
  - 13: Generar archivo con extensión CSV.
-

## 4.8. Extracción de características de frecuencia de dominio con transformadas wavelets

Extraer las características de dominio de frecuencia con un algoritmo que se desarrollo en Matlab, para generar el archivo que se utiliza en los clasificadores Random Forest y SMV. El Algoritmo 14 se describe a continuación:

---

**Algoritmo 14** Extracción de características de frecuencia de TWD

---

- 1: Leer imagen de entrada.
  - 2: Realizar la descomposición Wavelet de la matriz en el nivel  $N$ , obteniendo el vector de descomposición  $C$  y la matriz de contabilidad  $S$ .
  - 3: Extraer los componentes de los coeficientes horizontales  $H$ , verticales  $V$  y diagonales  $D$ .
  - 4: Calcular la media de  $x$ .
  - 5: Calcular la desviación estándar de  $x$ ,  $D$ ,  $H$  y  $V$ .
  - 6: Generar archivo *CSV*.
- 

## 4.9. TensorFlow Network de Google

Para poder utilizar tensorflow se requiere contar con las siguientes características en hardware y software:

1. Computadora con procesador de 64-bits.
2. Sistema operativo Ubuntu 16.04 o mayor.
3. Python.
4. Librería Pillow de python, para permitir la edición de las imágenes. Para su instalación en la terminal se ejecuta el comando `pip install Pillow`.

Tensorflow se instaló en una laptop, HP Pavilion, Intel Core i5-6200U, CPU de 2.30Hz, 8GB de memoria RAM, Sistema Operativo de 64bits, procesador x64 y 1 Tera de Disco Duro. Por lo que se procedió a instalar la versión para CPU con Python 2.7.

### 4.9.1. Instalar TensorFlow

Ya instalados los prerequisites en Linux, instalar tensorflow desde la terminal invocando el comando : `pip install tensorflow`.

### 4.9.2. Entrenamiento y clasificación con TensorFlow

Debido a que Inception V3 es una red neuronal de gran tamaño y capaz de clasificar entre 1000 categorías, solo se re-entrena la capa final, ya que ésta es la que se encarga de la obtención de los resultados para las categorías en las que se van a clasificar las imágenes de entrada. Para poder re-entrenar la red neuronal Inception V3, se realizó lo siguiente:

1. Se descargó el archivo *retrain.py* de la página oficial de tensorflow.org.
2. Se generó una carpeta llamada “src” donde se copió el archivo *retrain.py*, dentro de ésta se generó una carpeta llamada “GRANADAS” la cual contiene dos carpetas más una para el entrenamiento y la otra pruebas, éstas a su vez contienen dos carpetas llamadas “Botrytis”, donde se colocaron todas las imágenes relacionadas a la enfermedad y “SinBotrytis”, se colocaron las imágenes que no contienen la enfermedad.
3. Se generó un archivo llamado *train.sh*, en el cual se guardan los siguientes parámetros:

```
python src/retrain.py
--bottleneck_dir = src/bottlenecks
--how_many_training_steps 500
--model_dir = src/inception
--output_graph = src/retrained_graph.pb
--output_labels = src/retrained_labels.txt
--image_dir src/GRANADAS/Entrenamiento
```

4. En la terminal de Ubuntu, se ejecutan los siguientes comandos:

```
chmod +x train.sh
./train.sh
```



los cuales generaron dos archivos, uno es el archivo "*retrained\_labels.txt*". En este archivo se registran las categorías en las que se va a clasificar la imagen, en este caso se ingresaron las categorías de Botrytis y SinBotrytis. Y el segundo archivo llamado "*retrained\_graph.pb*", proporciona los datos resultantes del entrenamiento, los cuales serán utilizados en el clasificador que se describe más adelante.

Para el clasificador de la enfermedad se generó una red neuronal, que se describe a continuación en el Algoritmo 15.

---

**Algoritmo 15** Clasificador TensorFlow

---

- 1: Importación de la librería TensorFlow.
  - 2: Importación de la librería pillow.
  - 3: Lectura de la imagen de entrada.
  - 4: Extracción de las etiquetas de las categorías del archivo *retrained\_labels.txt* generado en el entrenamiento.
  - 5: Definición del umbral en 0.9.
  - 6: Lectura del archivo *retrained\_graph.pb*.
  - 7: Inicio de sesión en tensorflow.
  - 8: Ingreso de la imagen de entrada en el graph para obtener las predicciones.
  - 9: Se revisa el umbral, si no lo supera no tienen categoría.
  - 10: Impresión de los resultados de predicción.
- 

## 4.10. Clasificador SVM y Random Forest

Para la implementación de los algoritmos de SVM y Random Forest para la clasificación de las imágenes con la enfermedad de la Botrytis cinérea. Se utilizó el software llamado Weka. Con este software se hará uso de los archivos generados en la extracción de características estadísticas y de la extracción de características de dominio de la frecuencia con transformadas Wavelets. Para ello se realizó lo siguiente:

### Entrenamiento de los clasificadores SVM y Random Forest

1. Para los archivos "statisticalFeature.csv" y "waveletsFeatures.csv" cambiarlos a formato *.arff* de la siguiente manera:

1. Abrir el archivo en un block de notas.
2. En ambos archivos se agregó una columna al final, indicando la categoría 1 para los datos de las imágenes con botrytis y 2 para las imágenes sin Botrytis.
3. Para el archivo "waveletsFeature.csv" se agregaron las siguientes líneas en el encabezado del documento:

```
@RELATION waveletsFeaturesok
@ATTRIBUTE X1 REAL
@ATTRIBUTE X2 REAL
@ATTRIBUTE X3 REAL
@ATTRIBUTE X4 REAL
@ATTRIBUTE X5 REAL
@ATTRIBUTE class 1,2
@DATA
```

4. Para el documento de "statisticalFeature.csv", se agregaron las siguientes líneas: @RE-

```
LATION statisticalFeatureok
@ATTRIBUTE X1 REAL
@ATTRIBUTE X2 REAL
@ATTRIBUTE X3 REAL
@ATTRIBUTE X4 REAL
@ATTRIBUTE X5 REAL
@ATTRIBUTE X6 REAL
@ATTRIBUTE X7 REAL
@ATTRIBUTE X8 REAL
@ATTRIBUTE X9 REAL
@ATTRIBUTE X10 REAL
@ATTRIBUTE X11 REAL
@ATTRIBUTE X12 REAL
@ATTRIBUTE X13 REAL
@ATTRIBUTE X14 REAL
```

@ATTRIBUTE X15 REAL  
@ATTRIBUTE X16 REAL  
@ATTRIBUTE X17 REAL  
@ATTRIBUTE X18 REAL  
@ATTRIBUTE X19 REAL  
@ATTRIBUTE X20 REAL  
@ATTRIBUTE X21 REAL  
@ATTRIBUTE X22 REAL  
@ATTRIBUTE X23 REAL  
@ATTRIBUTE X24 REAL  
@ATTRIBUTE X25 REAL  
@ATTRIBUTE X26 REAL  
@ATTRIBUTE X27 REAL  
@ATTRIBUTE X28 REAL  
@ATTRIBUTE X29 REAL  
@ATTRIBUTE X30 REAL  
@ATTRIBUTE X31 REAL  
@ATTRIBUTE X32 REAL  
@ATTRIBUTE X33 REAL  
@ATTRIBUTE X34 REAL  
@ATTRIBUTE X35 REAL  
@ATTRIBUTE X36 REAL  
@ATTRIBUTE X37 REAL  
@ATTRIBUTE X38 REAL  
@ATTRIBUTE X39 REAL  
@ATTRIBUTE X40 REAL  
@ATTRIBUTE X41 REAL  
@ATTRIBUTE X42 REAL  
@ATTRIBUTE X43 REAL  
@ATTRIBUTE X44 REAL  
@ATTRIBUTE X45 REAL  
@ATTRIBUTE X46 REAL  
@ATTRIBUTE X47 REAL  
@ATTRIBUTE X48 REAL  
@ATTRIBUTE X49 REAL  
@ATTRIBUTE X50 REAL

```
@ATTRIBUTE X51 REAL
@ATTRIBUTE X52 REAL
@ATTRIBUTE X53 REAL
@ATTRIBUTE X54 REAL
@ATTRIBUTE class 1,2
@DATA
```

5. Una vez preparados los archivos abrir Weka.
6. Se selecciona *Explorer*.
7. Se selecciona el archivo "statisticalFeature.csv" o "waveletsFeatures.csv" según sea el caso, haciendo clic en *OpenFile*, ver Figura 4.14-1.
8. Se seleccionan todos los atributos, en *Atributes* dando clic en el boton *ALL*, ver Figura 4.14-2.
9. Ir a la parte del clasificador haciendo clic en *Classify* de la barra del menú, ver Figura 4.14-3.

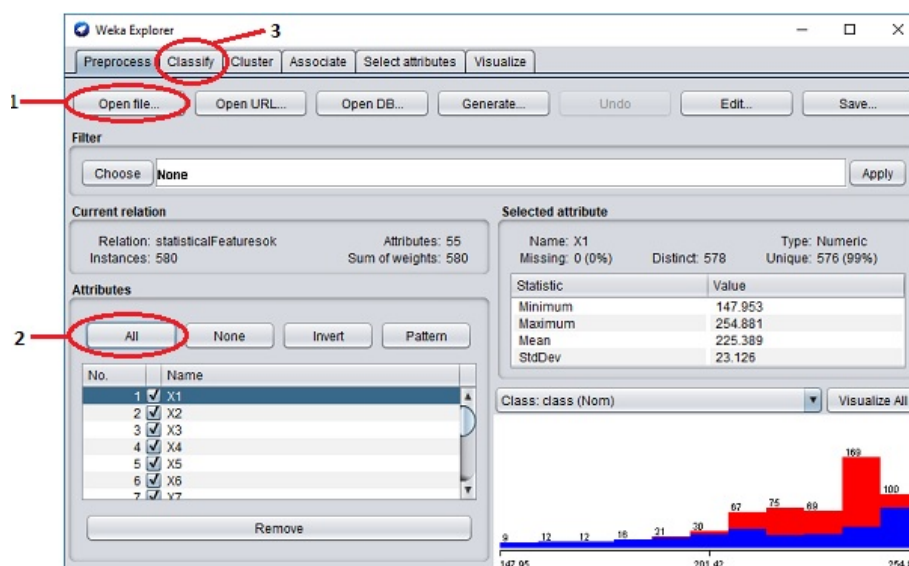


Figura 4.14: Clasificación con Weka.

10. Se selecciona el clasificador SMV, haciendo clic en Choose → Classifiers → Functions → SMO, ver Figura 4.15.

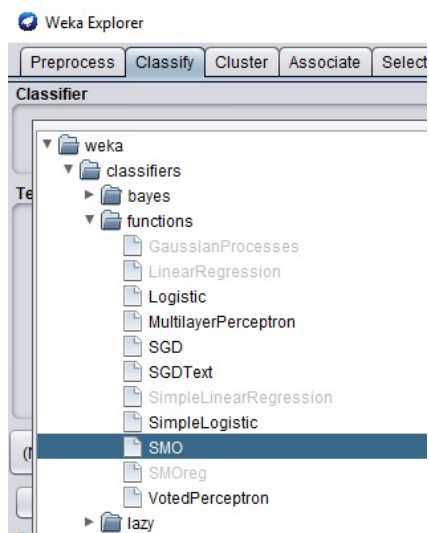


Figura 4.15: Seleccionar SVM.

11. Se selecciona el clasificador Random Forest, haciendo clic en Chosse → Classifiers → trees → RandomForest, ver Figura 4.16.

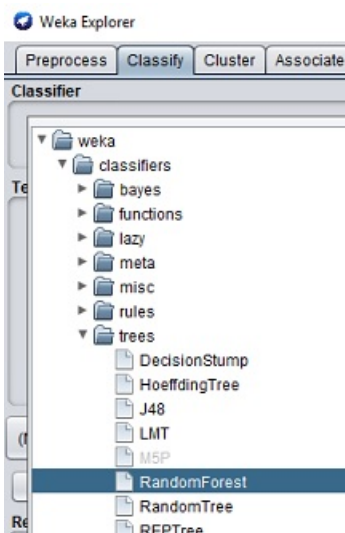


Figura 4.16: Seleccionar Random Forest

12. Al hacer clic en Start, se muestran los resultados en la ventana de *classifier output*, que se encuentra del lado derecho de la pantalla, ver Figura 4.17.

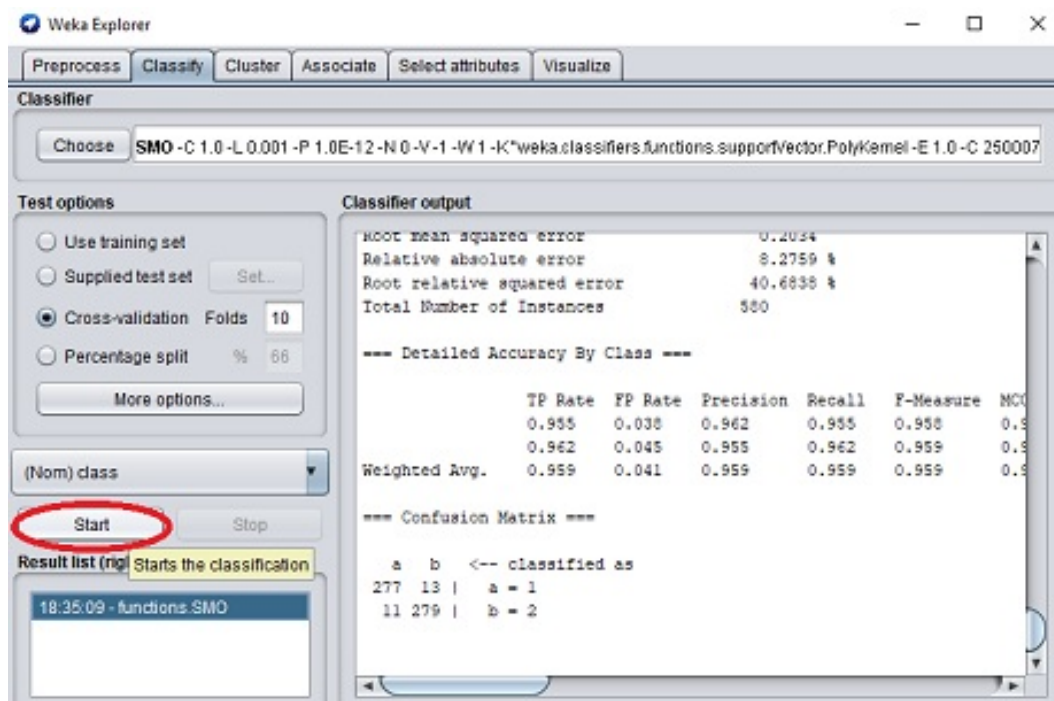


Figura 4.17: Clasificación con Weka.

---

---

## CAPÍTULO 5

---

# Resultados

### Contenido

---

<b>5.1. Introducción</b>	<b>83</b>
<b>5.2. Adquisición de la imagen</b>	<b>84</b>
<b>5.3. Pre-procesamiento de la imagen</b>	<b>84</b>
5.3.1. Obtener área de interés	84
5.3.2. Filtros y operaciones morfológicas	84
5.3.3. Realce de contrastes	87
<b>5.4. Segmentación</b>	<b>87</b>
5.4.1. Segmentación manual	87
5.4.2. Binarización	88
5.4.3. Método de Otsu	89
5.4.4. Crecimiento de regiones	91
5.4.5. Método matemático de punto interior a un triángulo	91
5.4.6. Generación de base de datos de pixeles enfermos	92
5.4.7. K-means clustering	92
<b>5.5. Extracción de características estadísticas</b>	<b>101</b>
<b>5.6. Extracción de características en el dominio de la frecuencia con transformadas Wavelets</b>	<b>102</b>
<b>5.7. Clasificación de la Botrytis cinérea</b>	<b>102</b>
<b>5.8. Clasificación con SVM y Random Forest</b>	<b>103</b>

---

### 5.1. Introducción

En este capítulo se muestran los resultados obtenidos en la adquisición de las imágenes, en el pre-procesamiento, los diferentes métodos de segmentación: binarización, crecimiento de

regiones, Otsu, K-means, características estadísticas y características de dominio de frecuencia Wavelets, y el entrenamiento de los clasificadores de TensorFlow, SVM y Random Forest.

## 5.2. Adquisición de la imagen

Se obtuvo un total de 1093 imágenes en formato JPG, de 5184 x 3456 píxeles con buena calidad. De las cuales 300 son con Botrytis cinérea, 207 con Alternaria y 586 sin enfermedad.

## 5.3. Pre-procesamiento de la imagen

### 5.3.1. Obtener área de interés

Las imágenes originales con Botrytis y sin Botrytis de 5184 x 3456 píxeles, se recortaron a un tamaño de 2353 x 2353 píxeles, ver Figura 5.1, con la finalidad de obtener solo el área de interés y reducir la carga computacional.

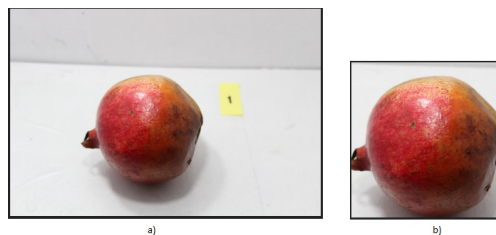


Figura 5.1: Obtener área de interés. a) imagen de granada con Botrytis, b) reducción al área de interés.

### 5.3.2. Filtros y operaciones morfológicas

Una vez obtenida el área de interés, se aplicaron los filtros de gauss, mediana, media, Prewit, Sobel y Laplaciant, dando como salida dos imágenes, una filtrada y otra en escala de



grises, con excepción de la mediana que solo proporciona una imagen en escala de grises, las cuales se pueden observar en las Figuras 5.2, 5.3, 5.4:



Figura 5.2: Aplicación de filtros. a) imagen original, b) Gauss, c) media, d) mediana.

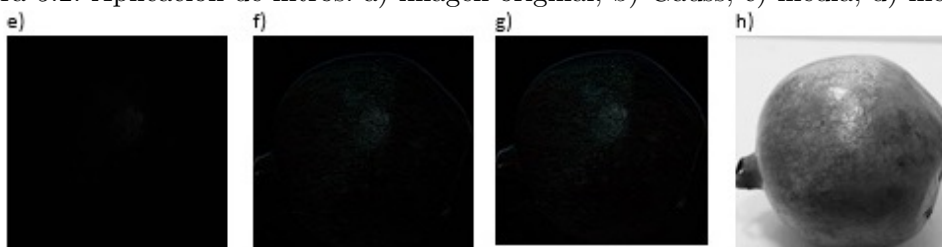


Figura 5.3: Aplicación de filtros. e) Laplaciant, f) Prewitt, g) Sobel, h) Gauss en escala de grises.

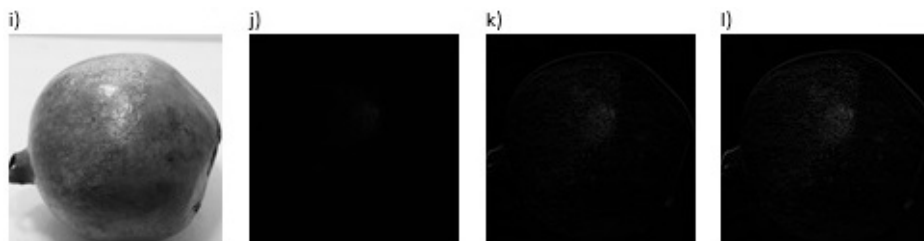


Figura 5.4: Aplicación de filtros. i) media en escala de grises, j) Laplaciant en escala de grises, k) Prewitt en escala de grises y l) Sobel en escala de grises.

Dados los resultados de la aplicación de los filtros presentados en las Figuras 5.2, 5.3 y 5.4, se descartan Prewitt, Sobel y Laplaciant. A los filtros de Gauss, media y mediana se les realizó una evaluación con MAE y PSNR, los resultados se muestran en las Figuras 5.5, 5.6. Se puede observar que los valores menores de MAE se presentan con el filtro de Gauss al igual que los valores mayores del PSNR, por lo que es el filtro elegido para trabajar en este proyecto.

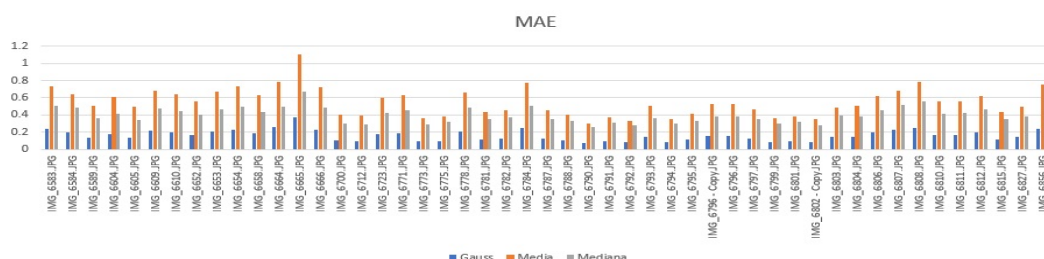


Figura 5.5: Evaluación MAE para los filtros de Gauss, Media y Mediana.

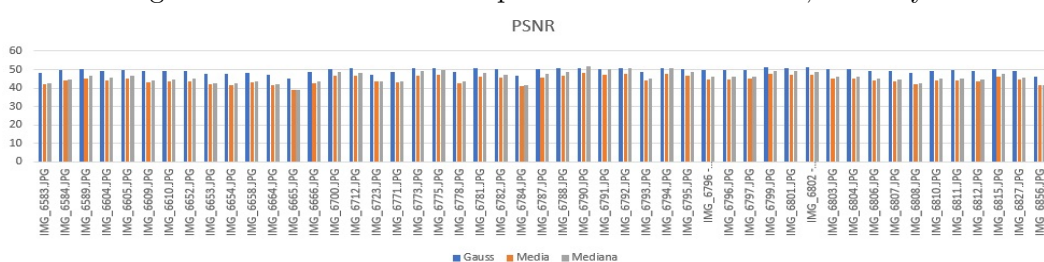


Figura 5.6: Evaluación PSNR para los filtros de Gauss, Media y Mediana.

Una vez elegido el filtro Gaussiano, se procede a aplicar las operaciones morfológicas, con la finalidad de obtener imágenes con diferentes características para el proceso posterior. Las operaciones morfológicas aplicadas y las imágenes resultantes se observan en la Figura 5.7. Como puede visualizarse, se descarta el uso de dilatación – erosión, ya que las imágenes resultantes son completamente negras.

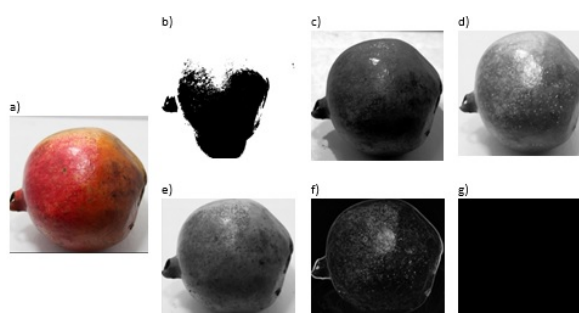


Figura 5.7: Operaciones morfológicas. a) imagen original, b) binarización, c) ecualización del histograma, d) dilatación, e) erosión, f) dilatación – erosión, g) erosión – dilatación.

### 5.3.3. Realce de contrastes

Se realiza el realce de contraste incrementando el brillo y mezclando los colores en los planos de  $G$  y  $B$ . El proceso anterior se llevó a cabo con la finalidad de disminuir los colores rojos y acentuar los colores cafés, como se muestra en la Figura 5.8.

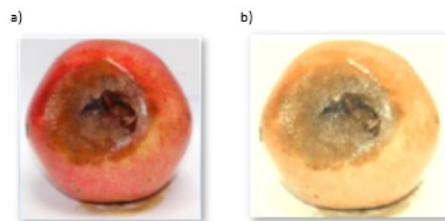


Figura 5.8: Realce de contraste.

## 5.4. Segmentación

En esta sección se muestran los resultados obtenidos de cada uno de los métodos de segmentación analizados.

### 5.4.1. Segmentación manual

Se realizó el recorte de un total de 514 imágenes de un tamaño de 657 x 657 píxeles, de las cuales 257 imágenes son con Botrytis y 257 sin Botrytis. Dando una imagen de salida como se muestra en la Figura 5.9.

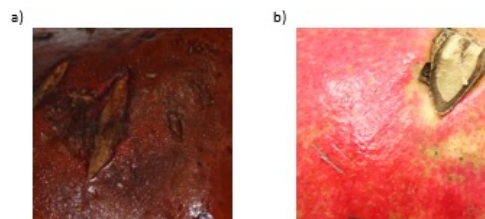


Figura 5.9: Segmentación manual de la granada. a) Enfermedad de la Botrytis, b) Parte sin Botrytis.

### 5.4.2. Binarización

Se utilizaron un total de 250 imágenes para la segmentación por medio de la binarización, de las cuales el 71.15 % de ellas no se obtuvieron los resultados esperados debido a que se observan imágenes con granadas completas, otras imágenes sin enfermedad y otras con ambos casos. El 28.25 % dió como resultado imágenes segmentadas con la enfermedad, algunas de estas imágenes aún presentan parte de la imagen sin Botrytis, como se puede observar en la Figura 5.10.

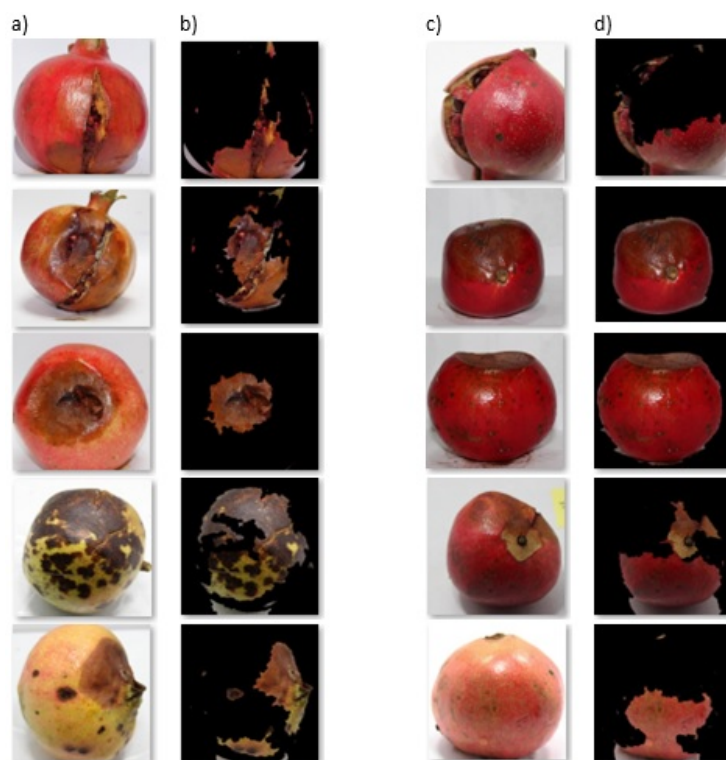


Figura 5.10: Segmentación por binarización. a) imágenes originales, b) binarización aceptable, c) imágenes originales, d) binarización rechazada.

En la Figura 5.10, se observa que la columna b) muestra los mejores resultados de la segmentación mientras que d) muestran algunos de los peores resultados ya que no se obtiene el objetivo esperado.

### 5.4.3. Método de Otsu

Se utilizaron un total de 250 imágenes para la segmentación con el método de Otsu, para el *Algoritmo 6* presentado en el capítulo anterior, los resultados muestran que del total de imágenes procesadas el 23% presentó solo la enfermedad en la imagen resultante y el 73% no logró separar la enfermedad de la Botrytis del resto de la imagen, esto resulta ser un resultado negativo para el objetivo del proyecto, ver Figura 5.11.

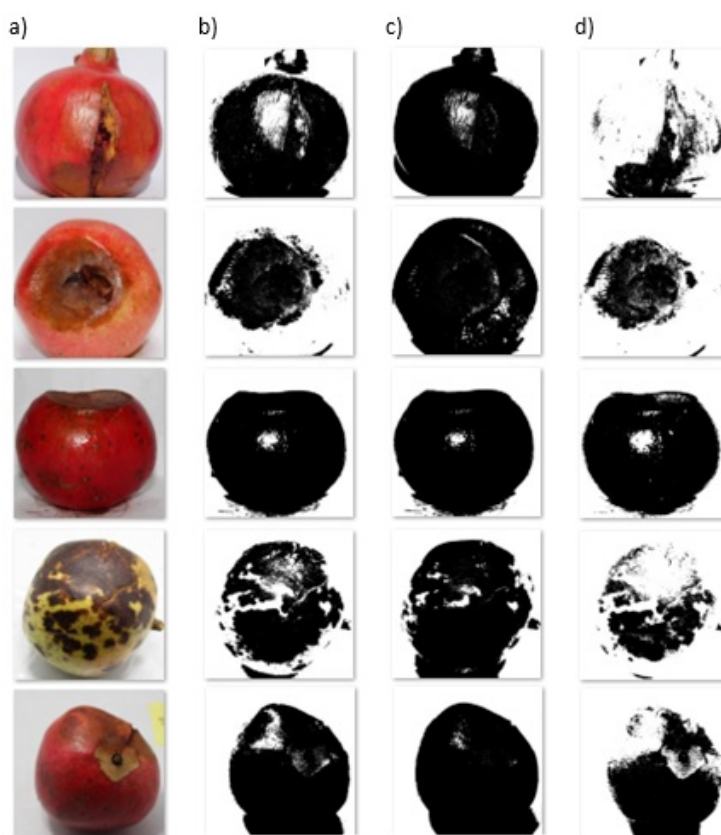


Figura 5.11: Resultado del primer algoritmo del método de Otsu.

En la Figura 5.11 se observa en la columna a) las imágenes originales, en b) los resultados del método de Otsu con un umbral de 150. Este umbral se definió sacando el umbral promedio de 60 imágenes. En la columna c) se muestra el método de Otsu con umbral automático y d) muestra el resultado de las imágenes de entrada pre-procesadas con realce de contraste.

Para el *Algoritmo 7* presentado en el capítulo anterior, se obtuvo un 34 % de resultados favorables, donde se extrae la enfermedad de la Botrytis del resto de la imagen y un 66 % de resultados que no lograron extraer la Botrytis, obteniendo imágenes con partes sin Botrytis o con ambos casos, con y sin Botrytis, ver Figura 5.12.

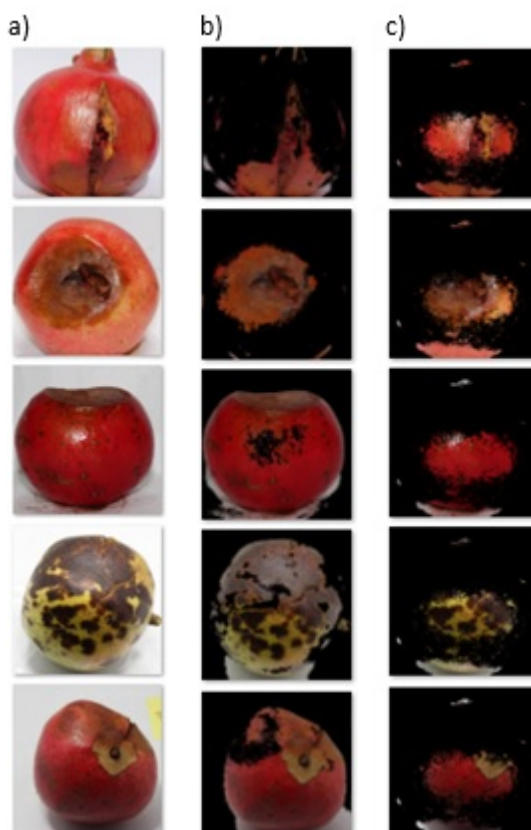


Figura 5.12: Resultados del segundo algoritmo del método de Otsu.

En la Figura 5.12, se muestra en la columna a) las imágenes originales, en b) los resultados del método de Otsu con imágenes originales, el cual logró extraer la mayor parte de la enfermedad y en c) el método de Otsu con imágenes con realce de contraste el cual no logró extraer de manera exitosa la enfermedad, como se puede observar en la Figura, la imágenes resultantes muestran la granada con y sin Botrytis .

#### 5.4.4. Crecimiento de regiones

Otro de los métodos que se probaron para la segmentación de la Botrytis, fue el crecimiento por regiones. Debido a las características de los colores de la imagen, este método no logró segmentar las imágenes, ya que al seleccionar la semilla, ésta no detecta los pixeles vecinos similares a ella. Los resultados se pueden visualizar en la Figura 5.13.

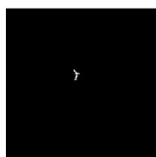


Figura 5.13: Crecimiento de regiones.

#### 5.4.5. Método matemático de punto interior a un triángulo

Para este método se hicieron pruebas con 10 imágenes, pero los resultados obtenidos no fueron favorables, ya que no logra segmentar la enfermedad y el tiempo de ejecución es de 10 minutos por imagen, ver figura 5.14.

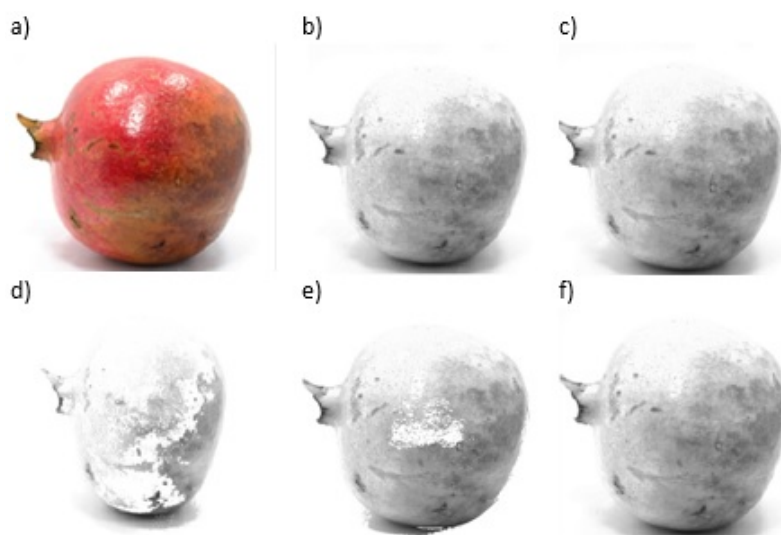


Figura 5.14: Resultados del método de punto interior a un triángulo. a) imagen original, b) planos R y G, c) planos R y B, d) planos G y B de RGB, e) planos H y S del espacio de color HSV y f) planos  $a^*$  y  $b^*$  del espacio de color CIELab.



En la Figura 5.14, se puede observar en a) la imagen original, en b) el resultado de la gráfica de los canales  $R$  y  $G$ , en c) el resultado de los canales  $R$  y  $B$ , en estos dos casos las imágenes obtenidas no muestran cambios significativos respecto a la original, ya que no se logró segmentar la enfermedad, en d) se observan los resultados de los canales  $G$  y  $B$ , estos canales fueron los que lograron extraer la enfermedad, sin embargo no es una imagen completamente limpia, ya que se muestran sombras y residuos de partes sin Botrytis en gran parte del a granada. En e) se observan los canales  $H$  y  $S$  del espacio de color HSV, y en f) los canales  $a^*$  y  $b^*$  del espacio de color CIELab, donde se puede observar que no se logró segmentar la enfermedad de la Botrytis.

#### 5.4.6. Generación de base de datos de pixeles enfermos

Con este método se obtuvo una base de datos almacenada en un archivo de texto no estructurado con formato .txt, el cual contiene 19,289,340 datos, no se hace uso de gestores ni manejadores de bases de datos. Los datos almacenados se usaron para segmentar la imagen por medio de una búsqueda de píxeles. El resultado de la segmentación fue mejor que los métodos anteriores, como se puede observar en la Figura 5.15. Sin embargo, la duración del procesamiento del algoritmo es significativo, con una duración de 2 días para procesar una sola imagen y no resulta un método eficaz para nuestro objetivo.

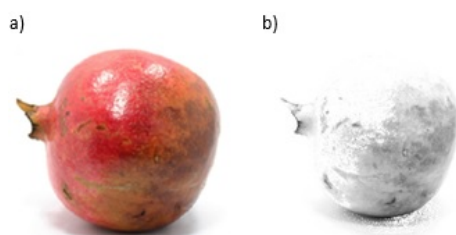


Figura 5.15: Búsqueda en base de datos de pixeles. a) imagen original, b) imagen segmentada.

#### 5.4.7. K-means clustering

Para la segmentación de la imagen a color con K-means, se trabajaron 250 imágenes y se estableció un  $k=3$ . Para poder tomar este valor se hicieron pruebas previas con  $k=2$ ,  $k=3$ ,  $k=4$  y  $k=5$ , donde se observó que el  $k=3$  era la mejor opción ya que agrupaba la mayor parte



de la enfermedad en un clúster. Los resultados para los diferentes espacios de color y métricas de distancia, se muestran a continuación:

#### En las imágenes originales

En el espacio de color RGB se aplicaron las operaciones morfológicas: diferencia absoluta, distancia euclidiana y coseno. Los resultados se observan en las Figuras 5.16, 5.17 y 5.18 respectivamente. Cada imagen muestra los tres clusters resultantes de la segmentación con K-means.

La Figura 5.16, muestra en el *Cluster 1* la granada sin Botrytis. En el *Cluster 2* se encuentra la enfermedad de la Botrytis, se puede observar que en el borde de la enfermedad se encuentran fragmentos de granada no contaminada y en el *Cluster 3* se encuentra la segmentación del fondo.



Figura 5.16: Resultado de la operación morfológica de la suma de diferencia absoluta.

La Figura 5.17, muestra en el *Cluster 1* la enfermedad de la Botrytis, así mismo, se puede observar que en el borde de la enfermedad se encuentran fragmentos de granada no contaminada. En el *Cluster 2* se encuentra la segmentación del fondo, y en el *Cluster 3* se observa la granada sin Botrytis.



Figura 5.17: Resultado de la operación morfológica de distancia euclidean.

La Figura 5.18, muestra en el *Cluster 1* la granada con fragmentos de la enfermedad aunque predomina la parte sin Botrytis, en el *Cluster 2*, se encuentra predominando la enfermedad de la Botrytis y al rededor se observa granada sin enfermedad y en el *Cluster 3* se encuentra la segmentación del fondo.



Figura 5.18: Resultado de la operación morfológica de coseno.

En el espacio de color RGB de las imágenes originales, la distancia euclideana es la que muestra una mejor segmentación de la enfermedad.

En el espacio de color CIELab se aplicaron las operaciones morfológicas: diferencia absoluta, distancia euclidiana y coseno. Los resultados de los tres clusters se observan en las Figuras 5.19, 5.20 y 5.21 respectivamente.

La Figura 5.19, muestra en el *Cluster 1* la granada con y sin Botrytis predominando la parte sin Botrytis. En el *Cluster 2* se presenta la enfermedad con bastante residuo de granada sin Botrytis, y en el *Cluster 3* se encuentra el fondo con partículas de la enfermedad.



Figura 5.19: Resultado de la operación morfológica de la suma de diferencia absoluta.

La Figura 5.20, muestra los resultados similares a la imagen anterior, en el *Cluster 1* se observa la granada con y sin Botrytis predominando la parte sin Botrytis. En el *Cluster 2* se presenta la enfermedad con bastante residuo de granada sin Botrytis, y en el *Cluster 3* se encuentra el fondo con partículas de la enfermedad.



Figura 5.20: Resultado de la operación morfológica de distancia euclideana.

La Figura 5.21, muestra que los resultados no son mejores a las Figuras anteriores, en el *Cluster 1* se observan pigmentos de la enfermedad y de la parte sin Botrytis. En el *Cluster 2* se encuentra el fondo y partes de la granada con y sin Botrytis, y en el *Cluster 3* se puede observar gran parte de la granada sin la enfermedad.



Figura 5.21: Resultado de la operación morfológica de coseno.

Como se puede observar en las figuras anteriores, la segmentación de la imagen original en el espacio de color CIELab, no aportó buenos resultados, ya que no se logra obtener solo la enfermedad de la Botrytis en ninguno de los tres clusters.

En el espacio de color HSV se aplicaron las operaciones morfológicas: diferencia absoluta, distancia euclidiana, coseno y correlación. Los resultados de los tres clusters se observan en las Figuras 5.22, 5.23, 5.24 y 5.25 respectivamente.

La Figura 5.22, muestra en el *Cluster 1* la enfermedad con algunos pigmentos pequeños de granada sin Botrytis. En el *Cluster 2* se muestra el fondo, y en el *Cluster 3* se puede ver la granada sin Botrytis y pigmentos de la enfermedad.



Figura 5.22: Resultado de la operación morfológica de la suma de diferencia absoluta.

La Figura 5.23, muestra en el *Cluster 1* el fondo. En el *Cluster 2* se tienen la enfermedad con menos pigmentos pequeños de granada sin Botrytis, y en el *Cluster 3* se puede ver la granada sin Botrytis y pigmentos de la enfermedad.



Figura 5.23: Resultado de la operación morfológica de distancia euclídeana.

La Figura 5.24, muestra que los resultados no varían mucho en comparación con las imágenes de la Figura anterior. En el *Cluster 1* se tienen la enfermedad con pocos pigmentos pequeños de granada sin Botrytis. En el *Cluster 2* se muestra el fondo, y en el *Cluster 3* se puede ver la granada sin Botrytis y pigmentos de la enfermedad.



Figura 5.24: Resultado de la operación morfológica de coseno.

La Figura 5.25, muestra en el *Cluster 1* el fondo. En el *Cluster 2* se puede ver la granada sin Botrytis y pigmentos de la enfermedad, y en el *Cluster 3* se tienen la enfermedad con menos pigmentos pequeños de granada sin Botrytis.



Figura 5.25: Resultado de la operación morfológica de correlación.

Para el espacio de color HSV con las imágenes originales, todas las operaciones morfológicas logran segmentar la enfermedad de la granada, sin embargo la distancia euclídeana y el coseno dan mejores resultados, dejando la enfermedad más limpia.

#### En imágenes con realce de contraste

En el espacio de color RGB se aplicaron las operaciones morfológicas: diferencia absoluta, distancia euclídeana y coseno. Los resultados se observan en los clusters generados por K-means, en las Figuras 5.26, 5.27 y 5.28 respectivamente.

La Figura 5.26, presenta en el *Cluster 1* la enfermedad. En el *Cluster 2* se encuentra la segmentación del fondo, y en el *Cluster 3* se observa la granada sin Botrytis.



Figura 5.26: Resultado de la operación morfológica de la suma de diferencia absoluta.

La Figura 5.27, se presentan tres imágenes con muy poca diferencia en los resultados de la Figura 5.26, en el *Cluster 1* se encuentra la segmentación del fondo. En el *Cluster 2* se presenta la enfermedad, y en el *Cluster 3* se observa la granada sin Botrytis.



Figura 5.27: Resultado de la operación morfológica de distancia euclideana.

La Figura 5.28, presenta en el *Cluster 1* la enfermedad y granada sin enfermedad. En el *Cluster 2* se encuentra la segmentación del fondo con enfermedad, y en el *Cluster 3* se obtiene la imagen con y sin Botrytis, predominando la parte sin Botrytis



Figura 5.28: Resultado de la operación morfológica de coseno.

Para el espacio de color RGB con las imágenes con realce de contraste, la diferencia absoluta y la distancia euclideana, logran segmentar la enfermedad en las granadas.

En el espacio de color CIELab se aplicaron las operaciones morfológicas: diferencia absoluta, distancia euclidiana y coseno. Los resultados de los tres clusters resultantes de K-means se observan en las Figuras 5.29, 5.30 y 5.31 respectivamente.

La Figura 5.29, presenta en el *Cluster 1* la granada sin Botrytis con pigmentos de la enfermedad. En el *Cluster 2* se encuentra la enfermedad, y en el *Cluster 3* se encuentra la segmentación del fondo.



Figura 5.29: Resultado de la operación morfológica de la suma de diferencia absoluta.

La Figura 5.30, presenta en el *Cluster 1* se encuentra el fondo. En el *Cluster 2* se tiene la granada sin Botrytis con pigmentos de Botrytis, y en el *Cluster 3* se observa la Botrytis.



Figura 5.30: Resultado de la operación morfológica de distancia euclidean.

La Figura 5.31, presenta en el *Cluster 1* la segmentación del fondo. En el *Cluster 1* se tiene la Botrytis, y en el *Cluster 1* se observa la granada sin Botrytis.



Figura 5.31: Resultado de la operación morfológica de coseno.

Para el espacio de color CIELab con las imágenes con realce de contraste, las tres operaciones morfológicas logran segmentar la enfermedad en las granadas.



En el espacio de color CIELab se aplicaron las operaciones morfológicas: diferencia absoluta, distancia euclidiana y coseno. Los resultados de los tres clusters se observan en las Figuras 5.32, 5.33 y 5.34 respectivamente.

La Figura 5.32, presenta en el *Cluster 1* la granada sin Botrytis. En el *Cluster 2* se encuentra la Botrytis, y en el *Cluster 3* se observa el fondo de la imagen.



Figura 5.32: Resultado de la operación morfológica de la suma de diferencia absoluta.

La Figura 5.33, presenta en el *Cluster 1* se presenta el fondo de la imagen. En el *Cluster 2* se encuentra la enfermedad, y en el *Cluster 3* se observa la granada sin Botrytis.



Figura 5.33: Resultado de la operación morfológica de distancia euclidiana.

La Figura 5.34, presenta en el *Cluster 1* el fondo de la imagen. En el *Cluster 2* se visualiza la granada casi completa, es mínima la extracción de la enfermedad, y en el *Cluster 3* se observa una pequeña parte de la Botrytis.



Figura 5.34: Resultado de la operación morfológica de coseno.



La Figura 5.35, se presenta en el *Cluster 1* se tiene una imagen con la enfermedad y partes de la granada sin Botrytis. En el *Cluster 2* se encuentra la segmentación del fondo, y en el *Cluster 3* se observa la granada completa.



Figura 5.35: Resultado de la operación morfológica de correlación.

Al observar los resultados de las diferentes variables en el algoritmo de K-means, se concluye que, la métrica de suma de diferencia absoluta con el espacio de color HSV y las imágenes pre-procesadas, son las variables que proporciona mejores resultados ya que se obtiene gran parte de la enfermedad, al momento de segmentarlas.

## 5.5. Extracción de características estadísticas

La ejecución del algoritmo mencionado en el capítulo anterior para la extracción de las características de la caracterización de las texturas da como resultado un archivo llamado "statisticalFeatures" con extensión .csv, ver Figura 5.36. Este archivo contiene un total de 54 columnas, con los datos de: media, desviación estándar, suavizado, entropía, energía y asimetría. Estos datos resultan de la división de la imagen en bloques de 3 x 3. El archivo será utilizado para el entrenamiento y pruebas de los clasificadores Random Forest y SMV.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	240.1404	15.71695	0.192131	3.646023	-9911.02	0.995968	231.5403	43.53516	0.195388	3.864784	-215099	0.999473	227.4022	47.74119	0.204661	3.846448	-260
2	239.0094	14.55783	0.156554	4.006287	-6176.92	0.995304	216.793	60.88631	0.165641	4.282815	-346622	0.99973	211.4029	61.89701	0.146092	4.523092	-321
3	240.4346	16.31491	0.175796	3.710358	-12156.7	0.996257	230.4155	45.3592	0.170004	3.862335	-215042	0.999514	225.4024	50.87057	0.168282	3.982732	-276
4	238.0589	12.96249	0.075462	4.545187	-2701.53	0.994084	231.2254	22.64191	0.07238	4.918448	-18726.1	0.998053	223.8483	38.25447	0.078617	4.905694	-116
5	242.4943	11.65749	0.155318	3.895881	-4000.59	0.992695	228.8393	46.92561	0.15122	4.275116	-252827	0.999546	226.4236	46.52778	0.149938	4.290108	-233

Figura 5.36: Extracción de características estadísticas.

## 5.6. Extracción de características en el dominio de la frecuencia con transformadas Wavelets

La ejecución del algoritmo mencionado en el capítulo anterior para la extracción de características en el dominio de la frecuencia con Wavelets, da como resultado un archivo llamado "waveletFeatures.csv" con extensión .csv, ver Figura 5.37. El archivo contiene las características de las imágenes, con datos de la media y desviación estándar de los coeficientes de aproximación. El archivo será utilizado para el entrenamiento y pruebas de los clasificadores Random Forest y SMV.

	A	B	C	D	E
1	1344.987	595.194	8.948078	22.64397	19.67784
2	1196.91	651.3405	3.173628	13.11421	10.3898
3	1184.687	673.339	11.78909	25.98077	21.8737
4	1080.356	621.9126	17.41416	29.20657	25.44136
5	933.719	488.8934	14.24444	27.2282	26.87691
6	992.186	598.602	18.2657	33.71348	33.78391
7	1038.083	542.611	18.80323	34.27846	35.80711
8	1060.141	604.2777	15.6451	29.4204	26.42206
9	924.1235	625.9613	21.09572	33.59376	38.40908
10	1010.341	654.8876	17.37735	27.78819	29.10905
11	1071.983	620.7845	23.03097	40.22263	38.46045
12	1123.148	550.2965	19.9823	35.72542	40.66829
13	1188.72	608.3344	12.35083	25.894	25.32808
14	1133.979	580.2005	18.60877	33.47442	31.82119
15	1216.215	599.2083	6.940899	16.1763	17.465
16	1130.112	597.0762	19.14927	30.62643	37.33822
17	1216.311	615.5454	22.97072	39.95885	38.06383

Figura 5.37: Extracción de características en el dominio de la frecuencia con transformadas Wavelets.

## 5.7. Clasificación de la Botrytis cinérea

La red neuronal de TensorFlow fue entrenada con ocho tipos de imágenes de entrada distintas. Para cada grupo se utilizaron un total de 580 imágenes, de las cuales 290 fueron

imágenes con Botrytis y 290 sin Botrytis. Se utilizó el 60 % de las imágenes para entrenamiento y el 40 % para pruebas. Para la etapa del entrenamiento, las imágenes con Botrytis se deben guardar en "scr\GRANADAS\Entrenamiento\Botrytis", y las imágenes sin Botrytis en "scr\ GRANADAS \Entrenamiento\ sinBotrytis", puesto que éstas son las rutas de las cuales el TensorFlow toma las imágenes para el entrenamiento. En la clasificación las imágenes con Botrytis se deben guardar en "scr\GRANADAS\Pruebas\Botrytis", y las imágenes sin Botrytis en "scr\ GRANADAS \ Pruebas\sinBotrytis". En la sección 4.9.2 se explica como se realizó el entrenamiento y la clasificación con TensorFlow. Como resultado del entrenamiento se obtiene un archivo "retrained\_graph.pb" por cada conjunto de datos de entrada, este archivo contiene las características numéricas (pesos) resultantes de las imágenes procesadas por TensorFlow. Los resultados de la clasificación se muestran en la tabla 5.1.

	Clasificación		
	Correctas	Erroneas	No identificadas
Segmentación manual	96.93 %	1.32 %	1.75 %
Filtro Gaussiano	81.58 %	1.75 %	16.67 %
Segmentación con K-means	74.58 %	25.42 %	0 %
Dilatación	44.74 %	30.26 %	25.00 %
Histograma	41.23 %	19.30 %	39.47 %
Dilatación-Erosión	37.28 %	35.09 %	27.63 %
Erosión	25.00 %	4.82 %	70.18 %
Binarización	12.72 %	20.18 %	67.11 %

Tabla 5.1: Resultados con TensorFlow.

Como se puede observar en la Tabla 5.1, los mejores resultados fueron para la segmentación manual, seguida de las imágenes completas con la aplicación del filtro Gaussiano, el peor de los casos se presentó con la binarización de las imágenes.

## 5.8. Clasificación con SVM y Random Forest

Se realizó la extracción de las características estadísticas y de dominio de la frecuencia (Transformada Wavelet), en 580 de las imágenes con el filtro Gaussiano, de las cuales 290

fueron imágenes con Botrytis y 290 sin Botrytis. Se utilizó el 60 % de las imágenes para entrenamiento y el 40 % para pruebas. Los resultados se muestran en la tabla 5.2.

Clasificador	Correctas	Erróneas
Características Estadísticas		
Random Forest	94.08 %	6.89 %
SVM	93.1 %	5.91 %
Transformada Wavelet		
Random Forest	78.81 %	71.18 %
SVM	82.26 %	17.73 %

Tabla 5.2: Resultados SVM y Random Forest.

La extracción de las características estadísticas muestra los resultados con mayor aciertos en los clasificadores de Random Forest y SVM, con un porcentaje mayor al 90 %. Las características extraídas con las Transformadas Wavelets proporcionan resultados con un porcentaje menor, con el 78.81 % para Random Forest y el 82.26 % para SVM. Aún con estos resultados ambos métodos de extracción son funcionales para la clasificación de la enfermedad de la Botrytis en las granadas.

---

## CAPÍTULO 6

---

# Conclusiones

En el presente trabajo de tesis de maestría se realizó el estudio de diferentes métodos de segmentación y extracción de características de las imágenes de las granadas: binarización, crecimiento de regiones, Otsu, K-means clustering, modelo matemático de punto interior a un triángulo, generación de base de datos de píxeles, extracción de características estadísticas y de características del dominio de la frecuencia con transformadas Wavelets, con el fin de obtener los datos para el entrenamiento de la red neuronal y lograr el objetivo principal del proyecto.

Las primeras pruebas realizadas en la red neuronal, fueron imágenes pre-procesadas solo con las operaciones morfológicas, con lo cual no se obtuvieron los resultados esperados, todos dieron resultados asertivos menores del 50%. Por lo que se indagó en la búsqueda de otras opciones, empezando por el uso de la segmentación de la enfermedad con K-means, al principio los resultados fueron malos, ya que no se lograba separar la enfermedad de la imagen, después de varios intentos, se optó por probar otros métodos, como: el método de Otsu y crecimiento de regiones, estos métodos no arrojaron los resultados que se esperaban, pero con esto se pudo observar que la diferencia de los colores entre algunos tonos de la enfermedad y la parte sana a nivel pixel era mínima, por lo que a nivel computacional no se podían diferenciar. Enseguida se procedió a la búsqueda de la obtención de los rangos de colores que conformaban la enfermedad. Se realizó la segmentación manual de la enfermedad, para poder obtener la base de datos de los píxeles enfermos y poder obtener los rangos de color, también se procedía realizar el método del punto interior a un triángulo, con estos métodos no se obtuvieron los resultados esperados. Con los métodos que no dieron resultados favorables se llegó a la conclusión que primero se debía realizar un realce de contraste, el cual pudiera marcar la diferencia entre la enfermedad y el resto de la granada; una vez obtenido este objetivo, se regresó al estudio de K-means y del método de Otsu, siendo el K-means quien arrojó resultados favorables.

Se realizaron varios entrenamientos de la red neuronal convolucional, usando diferentes tipos de datos de entrada: operaciones morfológicas, segmentación manual, la segmentación

con K-means y las imágenes originales solo con el filtro de Gauss. Se pudo observar que los mejores resultados fueron con la segmentación manual (96.93 %) y el filtro de Gauss (81.58 %). Al obtener los resultados de la red neuronal, se planteó la opción de realizar una comparativa con los clasificadores de SMV y Random Forest, con dos tipos de características: estadísticas y de dominio de la frecuencia con transformadas Wavelet.

Las pruebas arrojaron que los mejores resultados se obtuvieron con la segmentación manual con un 96.93 % en la RNN, características estadísticas: Random Forest de 94.08 %, SMV con 96.1 %, transformadas Wavelets en SVM con el 82.26 % y el Filtro Gaussiano con el 81.58 % en la RNN. Los resultados obtenidos muestran porcentajes aceptables de asertividad, lo que indica que se podrá apoyar a la empresa GEGEMA, S.P.R de R.L, en el control de selección del fruto al momento de la producción de los derivados de la granada. Se considera que llega a la mesa de corte de un 13 % - 20 % de granada que contiene Botrytis. Como trabajo futuro se pretende realizar pruebas de la aplicación derivada de esta tesis cuando se implemente en la línea de producción automatizada, de esta manera se podrá reducir el porcentaje de selección de granada con Botrytis.

Durante el desarrollo del proyecto, se adquirieron conocimientos sobre el procesamiento de imágenes y librerías dedicadas a la visión artificial, con esto se obtuvo la actualización de conocimientos sobre las tecnologías que se utilizan actualmente, para poder emprender proyectos propios o realizar aportaciones en los proyectos que ya existen. El uso de Matlab facilitó el procesamiento de las imágenes, ya que cuenta con gran variedad de funciones y métodos dedicados a esto. Respecto a la red neuronal de TensorFlow se comprobó que es una herramienta poderosa, con gran capacidad de análisis para la identificación de objetos de imágenes de gran tamaño y apoyó de manera especial a cumplir el objetivo planteado en la tesis.

---

---

## APÉNDICE A

---

# Productos obtenidos

### A.1. Estadía profesional

**Nombre de la empresa:** GEGEMA, S.P.R. DE R.L. (Ariños, derivados de granada).

**Nombre del encargado:** Gerardo Álvarez González.

**Domicilio:** Rancho la Herradura del municipio de Zapotlán el Grande.

**Periodo de estadía:** del 23 de Agosto al 30 de Septiembre.

**Actividades:**

- Se conoció el área de trabajo y el proceso que se realiza durante la producción de los derivados de la Granada.
- Se separó manualmente granadas enfermas y sanas.
- Se obtuvieron características de las granadas con Botrytis.
- Se obtuvieron características de las granadas con Alternaria sp.
- Se creó la base de datos de imágenes de granadas con Botrytis y sin Botrytis.

## A.2. Ponencia de un artículo en la Academia Journals Morelia 2018

El artículo se podrá consultar en las memorias de la Academia Journals Morelia 2018, en el tomo 14, página 2270. En el link <http://www.academiajournals.com/pub-morelia/>.

Memorias del Congreso Internacional  
de Investigación Academia Journals  
Morelia 2018

© Academia Journals 2018

Morelia, Michoacán, México  
16 al 18 de mayo de 2018

### IDENTIFICACIÓN DE LA ENFERMEDAD BOTRYTIS CINÉREA EN LA GRANADA UTILIZANDO K - MEANS

ISC. Veronica Esmeralda Miramontes Varo<sup>1</sup>, Dra. María Guadalupe Sánchez Cervantes<sup>2</sup> y  
Dr. Sergio Sandoval Pérez<sup>3</sup>

**Resumen**— Existen diferentes enfermedades que se presentan en las granadas, la Botrytis Cinérea es una de ellas. Esta enfermedad afecta la producción de los derivados de la granada, ya que puede llegar a contaminar el producto final y por consiguiente acelerar su putrefacción. Debido a lo anterior, el objetivo del presente artículo es identificar la enfermedad de la Botrytis en la etapa de selección previa a la producción a través del reconocimiento de patrones utilizando el método de agrupamiento K-means. Para llevar a cabo lo anterior, se pre-procesa la imagen de entrada, la cual es convertida del formato RGB a HSV para obtener la información de los colores. Una vez adquiridos los canales, se aplica el K-means con un  $k=3$ . Como resultado se obtiene que la segmentación de salida de una granada enferma con la Botrytis se encuentra en el Clúster 3.

**Palabras clave**—Botrytis, K-means clustering, segmentación de imágenes a color, granada.

#### Introducción

La granada es un fruto que aporta muchos beneficios para la salud, su masa está compuesto por 80% de agua y es una fruta baja en calorías, que contiene potasio, vitamina A, vitamina C y vitamina B9. El tamaño de una granada puede oscilar entre el de una naranja a una toronja. Las granadas también están expuestas a enfermedades, tales como: la podredumbre del fruto que puede ser causada por dos tipos de hongos Botrytis cinérea y Alternaria spp. Otra enfermedad es el Cribado (Clasterosporium carpophilum) la cual se manifiesta con manchas negras en la piel del fruto. La podredumbre del fruto es la enfermedad más importante que afecta la producción de granada. La Botrytis cinérea es uno de los hongos que la causa, el cual entra por el pistilo y se extiende dentro del fruto hasta pudrir los granos completamente e incluso la corteza. La Alternaria spp. Es otro hongo que padre la granada, un síntoma es la presencia de micelio negro al interior de la fruta atacando los arillos, comenzando por el calyx, mientras que la corteza es dura y correa pero con apariencia saludable. Ambas enfermedades son favorecidas por las lluvias (Mondragón, J. C., et al, 2008).

Actualmente en una empresa de la región sur de Jalisco dedicada a la producción de los derivados de la granada, la tarea de seleccionar la fruta en la etapa de pos cosecha en un estado óptimo para ser procesada y sacar el mayor provecho es ardua, debido a que se elabora de manera manual y a criterio del ser humano. Esto lleva el interés de investigar métodos no invasivos que permitan realizar una selección de frutos sin dañarlos, es por ello que el objetivo de este proyecto es identificar la enfermedad de la Botrytis por medio del reconocimiento de patrones en las imágenes. Algunos investigadores han propuesto metodologías para identificar las enfermedades en las frutas, en (Deshpande, T., et al, 2014) se propuso un algoritmo que identifica y determina el tamaño de la invasión de la bacteria Blith en hojas, tallos y el fruto de la planta de la granada, utilizando técnicas de pre-procesamiento de imágenes y K-means clustering con  $K=2$ , al extraer la enfermedad se le calculo el porcentaje del área enferma respecto al tamaño de la hoja o fruto.

Algunos estudios como en (Pawar, M. M., et al, 2012), se enfocan en los defectos de la piel tales como, quemaduras solares, alternaria y enfermedades de Cercospora. Se realizó una base de datos de piel infectada y no infectada de las granadas utilizando características de color de las texturas de la piel, usando la transformada wavelet (DWT) se calcularon características estadísticas, éstas sobre los canales R, G y B, las mejores se usaron como entradas para un clasificador de la Máquina de Vector de Soporte (SMV).

En la india se realizó un sistema de reconocimiento de imágenes para la identificación de enfermedades en las granadas (Mundokar, N., et al, 2017) donde se utiliza análisis de color y OpenCv (ANN) con E-NOSE, con esto

<sup>1</sup> ISC. Veronica Esmeralda Miramontes Varo es estudiante de Maestría en Ciencias de la Computación en el Instituto Tecnológico de Cd. Guzmán, Jalisco [verovaro@amari.com](mailto:verovaro@amari.com) (autor correspondiente)

<sup>2</sup> La Dra. María Guadalupe Sánchez Cervantes es Profesora investigadora y Coordinadora de la Maestría en Ciencias de la Computación en el Instituto Tecnológico de Cd. Guzmán, Jalisco [msanchez@icg.edu.mx](mailto:msanchez@icg.edu.mx)

<sup>3</sup> Dr. Sergio Sandoval Pérez es Profesor y Coordinador de la Maestría en Electrónica en el Instituto Tecnológico de Cd. Guzmán, Jalisco [sandoval@icg.edu.mx](mailto:sandoval@icg.edu.mx)

Figura A.1: Memorias Academia JOurnals Morelia 2018



---

---

## APÉNDICE B

---

### Actividades de apoyo

#### **B.1. Sistema Administrativo para la empresa GEGEMA S.P.R. DE R.L.**

La empresa GEGEMA S.P.R. DE R.L. dedicada a producir derivados de la granada, solicitó al Instituto Tecnológico de Cd. Guzmán la colaboración de estudiantes de licenciatura y maestría para realizar algunos proyectos. El objetivo de estos proyectos es poder automatizar los procesos que se realizan de manera manual. Uno de esos proyectos es un sistema administrativo que permita llevar un mejor control de la empresa, abarcando, materias primas, empleados, ventas, producción, entre otros. El segundo es una aplicación para la identificación de la Botrytis en las granadas. Es por ello que se apoyó a los estudiantes de licenciatura en la obtención de requerimientos, acompañando a las estudiantes residentes encargadas del sistema administrativo a las instalaciones de la empresa, para realizar entrevistas sobre la administración, objetivos y procesos, para poder identificar las necesidades que se tienen. Una vez analizados los requerimientos se realizó un escrito con el planteamiento de la propuesta a desarrollar, el cual fue aprobado por el empresario. Se apoyó en el análisis y diseño de la base de datos, realizando reuniones en el aula de posgrado, en donde se revisó y se analizó la base de datos del sistema administrativo por medio de la elaboración del diagrama E-R y diagramas de clases, estos diagramas fueron normalizados aplicando 1FN, 2FN, 3FN y FNB.

El sistema administrativo podrá registrar altas, bajas y modificaciones de materias primas, productos terminados, empleados, mermas, clientes, proveedores, sucursales, compras, ventas, mantenimiento, insumos, refacciones, producción, abonos, devoluciones, control de inventarios, almacén de productos terminados y almacén general donde se almacenara jugo y pulpa de los cuales se derivarán los productos terminados. Contará con las sesiones de: administrador, responsable de inventario, responsable de almacén, compras, ventas, mantenimiento, devoluciones y producción.

---

---

## APÉNDICE C

---

### Glosario

**Algoritmos genéticos:** son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza de acorde con los principios de la selección natural y la supervivencia de los más fuertes. Por imitación de este proceso, los Algoritmos Genéticos son capaces de ir creando soluciones para problemas del mundo real.

**Análisis de Discriminación Lineal:** es un método utilizado en estadística, reconocimiento de patrones y aprendizaje de máquinas para encontrar una combinación lineal de rasgos que caracterizan o separan dos o más clases de objetos o eventos.

**Anthractnose:** es un síntoma de enfermedad de las plantas de zonas calurosas y húmedas, causada por un hongo el cual causa manchas hundidas de diversos colores en las hojas y necrosis en los nervios de estas, tallos, frutos o flores, que muchas veces derivan en el marchitamiento y muerte de los tejidos.

**Arilos:** es una cobertura carnosa de ciertas semillas formado a partir de la expansión del funículo (filamento de unión de la semilla al ovario) o del hilo (punto de inserción del anterior).

**Aspergillus Fruit Rot:** es un hongo patógeno que es común en las frutas fisiológicamente maduras.

**Backpropagations:** es un método de cálculo del gradiente utilizado en algoritmos de aprendizaje supervisado para entrenar redes neuronales artificiales. El método emplea un ciclo propagación – adaptación de dos fases. Se aplica un patrón a la entrada de la red como estímulo, este se propaga desde la primera capa a través de las capas siguientes de la red, hasta generar una salida. La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas. Las salidas de error se propagan hacia atrás,

partiendo de la capa de salida, hacia todas las neuronas de la capa oculta que contribuyen directamente a la salida. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido una señal de error que describa su contribución relativa al error total.

**Bacteria Blight:** es una enfermedad de las plantas causada por el patógeno bacteriano *Xanthomonas campestris*.

**Black Rot:** conocida como roña negra de la vid, es una enfermedad de las plantas de origen americano, esta enfermedad afecta a las plantas verdes, los síntomas son manchas de color pardo que van aumentando de tamaño y, progresivamente, adquieren un color rojizo rodeado de una banda oscura.

**Calyx:** verticilo floral formado por los sépalos, estos suelen ser verdosos y tienen una función protectora.

**Centroide:** punto donde se produce la intersección de las medianas que forman parte de un objeto. Cabe recordar que una mediana es una recta que se traza desde un vértice del objeto hasta el punto que está en el medio del lado opuesto.

**Cercospora:** enfermedad producida en las hojas de las por el hongo parásito microscópico conocido científicamente como *Cercospora beticola* Sacc.

**Emug CV:** es un contenedor .Net de plataforma cruzada para la biblioteca de procesamiento de imágenes OpenCV, que permite que las funciones de OpenCV se invoquen desde lenguajes compatibles con .NET.

**Enfermedad foliar:** se trata de las enfermedades que se dan en las hojas que puede ser causada por diversos hongos o bacterias presentes en los cultivos.

**E-NOSE:** es un sistema electrónico con capacidad analítica para detectar olores, este sistema imita el principio del sistema olfativo de los mamíferos.

**Filtro de Gabor:** es un filtro lineal cuya respuesta de impulso es una función sinusoidal (esta es la curva que representa a la función seno) multiplicada por una función gaussiana. Son funciones paso baja.

**Fisher Discriminant Analysis (FDA):** Análisis Discriminante Lineal (ADL) es una generalización del discriminante lineal de Fisher, un método utilizado en estadística, recono-

cimiento de patrones y aprendizaje de máquinas para encontrar una combinación lineal de rasgos que caracterizan o separan dos o más clases de objetos o eventos.

**GLCM (Gray-Level Co-occurrence Matrices):** es una matriz que se define sobre una imagen para que sea la distribución de los valores de píxel concurrentes (valores de escala de grises, o colores) en una compensación dada.

**Gliding box:** es un algoritmo denominado 'caja de deslizamiento' dirigido al análisis de lacunaridad y se ha extendido al modelado multifractal. Este método se puede usar para implementar varias técnicas multifractales, que incluyen, los métodos de momento y multiplicador.

**Gray mold:** también conocida como Botrytis cinérea, es un hongo necrotrófico que afecta a muchas especies de plantas.

**LBP:** Local Binary Pattern (LBP) es un operador de textura simple pero muy eficiente que etiqueta los píxeles de una imagen mediante el umbral de la vecindad de cada píxel y considera el resultado como un número binario.

**Lógica formal:** es la parte de la lógica que se dedica al estudio de la inferencia mediante la construcción de lenguajes formales, sistemas deductivos y semánticas formales.

**Machine learning:** es el sub campo de las ciencias de la computación y una rama de la inteligencia artificial cuyo objetivo es desarrollar técnicas que permitan a las computadoras aprender.

**Max-pooling:** es una operación que encuentra el valor máximo entre una ventana de muestra y pasa este valor como resumen de características sobre esa área. Como resultado, el tamaño de los datos se reduce por un factor igual al tamaño de la ventana de muestra sobre la cual se opera.

**Micellaneous image transformations:** es una función que aplica un umbral adaptativo a una matriz. La función transforma una imagen en escala de grises en una imagen binaria.

**Mildeo powdery:** es una enfermedad fúngica que afecta a una gran variedad de plantas. Esta enfermedad crece en ambientes con alta humedad y temperaturas moderadas.

**Mildeo puro:** es un hongo de desarrollo interno, por lo que su tratamiento es más complicado que otro tipo de enfermedades. Las esporas, al situarse sobre la hoja, emiten un tubo germinativo que penetra a través del estoma y coloniza el interior del cultivo.

**Mildew (moho):** es una forma de hongo. Se distingue por su color blanco. Aparece como un crecimiento delgado y superficial que consiste en diminutas hifas ( filamentos fúngicos ) producidas especialmente en plantas vivas o materia orgánica como la madera, el papel o el cuero.

**Min-pooling:** es una operación que encuentra el valor mínimo entre una ventana de muestra y pasa este valor como resumen de características sobre esa área.

**Moho polvoriento (powdery):** es una enfermedad fúngica común de las angiospermas (básicamente, cualquier planta, excepto las coníferas). El hongo en crecimiento se ve blanco grisáceo y polvoriento. Es causada por una infección de una de las muchas especies de hongos de la familia Erysiphaceae. Los mohos polvorientos viven, fuera de su anfitrión, no dentro de él.

**PCA:** el Análisis de Componentes Principales (en inglés, PCA) es una técnica utilizada para describir un set de datos en términos de nuevas variables no correlacionadas. Los componentes se ordenan por la cantidad de varianza original, por lo que la técnica es útil para reducir la dimensionalidad de un conjunto de datos.

**Pooling:** es una operación de consulta constante, para realizar una actividad sincrónica sin el uso de interrupciones.

**Pudrición negra:** es un hongo cuyo principal objetivo es la marchitez de las plantas. Es considerada como una de las enfermedades más importantes de las plantas a nivel mundial.

**Seven fold cross validation:** es una técnica utilizada para evaluar los resultados de un análisis estadístico y garantizar que son independientes de la partición entre datos de entrenamiento y prueba. Consiste en repetir y calcular la media aritmética obtenida de las medidas de evaluación sobre diferentes particiones. Se utiliza en entornos donde el objetivo principal es la predicción y se quiere estimar la precisión de un modelo que se llevará a cabo a la práctica.

**SURF (Speed Up Robust Feature):** es un algoritmo que forma parte de la inteligencia artificial, en el área de visión, es capaz de obtener una representación visual de una imagen y extraer una información detallada y específica del contenido. Esta información es tratada para realizar operaciones como por ejemplo la localización y reconocimiento de determinados objetos, personas o caras, realización de escenas 3D, seguimiento de objetos y extracción de puntos de interés.

**Tizón de ascochyta (*Ascochyta gossypii*)** es un género de hongos ascomycotas que contiene varios patógenos de plantas.

**Umbral:** son un grupo de algoritmos cuya finalidad de separar los objetos de una imagen que nos interesen del resto, transforma la imagen en ceros y unos (blanco y negro), los píxeles blancos representan los píxeles de la imagen cuyo valor está en el rango del umbral, y los negros el valor que está fuera de este rango.

---

## Bibliografía

- [Al-Bashish D., 2011] Al-Bashish D., Braik M., B.-A. S. (2011). Detection and classification of leaf diseases using k-means-based segmentation. *Information Technology Journal*, 10(2):267–275.
- [Aldalur B., 2002] Aldalur B., S. M. (2002). Realce de imágenes: filtrado espacial. *Revista de teledetección*, pages 17, 31–42.
- [Anderson, 2007] Anderson, J. (2007). *Redes neuronales*. Alfaomega.
- [Awate A., 2015] Awate A., Deshmankar D., A. G. (2015). Fruit disease detection using color, texture analysis and ann. *In Green Computing and Internet of Things (ICGCIoT), 2015 International Conference. IEEE*, pages 970–975.
- [Azuela, 2013] Azuela, J. H. S. (2013). *Visión artificial: rasgos descriptores para el reconocimiento de objetos*. Ra-Ma, firsts edition.
- [Beaulieu P., 2014] Beaulieu P., M. D. B. (2014). A study of the effect of feature reduction via statistically significant pixel selection on fruit object representation, classification, and machine learning prediction. *In Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), 2014 IEEE International Conference. IEEE*, pages 82–87.
- [Camargo A., 2009] Camargo A., S. J. S. (2009). Image pattern classification for the identification of disease causing agents in plants. *Computers and Electronics in Agriculture*, 66(2):121–125.
- [Camperos E. N. S., 2006] Camperos E. N. S., G. A. (2006). *Redes neuronales: conceptos fundamentales y aplicaciones a control automático*. Pearson Educación, first edition.
- [Cuevas E., 2017] Cuevas E., Díaz C. M., C. M. (2017). *Tratamiento de imágenes con MATLAB*. Alfaomega, first edition.
- [Cuevas E., 2010] Cuevas E., Zaldívar D., P. M. (2010). *Procesamiento digital de imágenes con MATLAB y Simulink*. Alfaomega Ra-Ma, first edition.
- [Deshpande T., 2014] Deshpande T., Sengupta S., R. K. (2014). Grading and identification of disease in pomegranate leaf and fruit. *International Journal of Computer Science and Information Technologies*, 5(3):pages = 4638–4645.

- [Dhakate M., 2015] Dhakate M., I. A. B. (2015). Diagnosis of pomegranate plant diseases using neural network. in computer vision, pattern recognition, image processing and graphics (NCVPRIPG). *2015 Fifth National Conference IEEE*, pages 1–4.
- [Dubey S., 2012] Dubey S., J. A. (2012). Detection and classification of apple fruit diseases using complete local binary patterns. *In Computer and Communication Technology (ICCCT), 2012 Third International Conference. IEEE*, pages 346–351.
- [Dubey S., 2014] Dubey S., J. A. (2014). Adapted approach for fruit disease identification using images. *arXiv preprint arXiv*, page 1405.4930.
- [Díaz-Uriarte R., 2006] Díaz-Uriarte R., D.-A. S. (2006). Gene selection and classification of microarray data using random forest. *bmc bioinformatics. BMC bioinformatics*, 7(1):3.
- [Gavhale K., 2014] Gavhale K., Gawande U., H. K. (2014). Unhealthy region of citrus leaf detection using image processing techniques. *In Convergence of Technology (I2CT), 2014 International Conference. IEEE.*, pages 1–6.
- [Gila D. M., 2015] Gila D. M., Puerto D. A., G. J. G. (2015). Automatic classification of olives for oil production using computer vision. *In Industrial Technology (ICIT), 2015 IEEE International Conference. IEEE*, pages 1651–1656.
- [Hall M., 2009] Hall M., Frank E., H. G. (2009). The weka data mining software an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.
- [Herrera J. C., 2016] Herrera J. C., Medina S. M., B. K. (2016). Diseño de un sistema automático de selección de frutos de café mediante técnicas de visión artificial. *Revista UIS Ingenierías*, 15(1):7–14.
- [Hurtado D. A., 2015] Hurtado D. A., M. J. (2015). Método preliminar de detección de patógenos biológicos en cultivos de fresa por medio del procesamiento digital de imágenes. *Revista de Investigación Agraria y Ambiental*, 6(1):111–122.
- [Jhuria M., 2013] Jhuria M., Kumar A., B. R. (2013). Image processing for smart farming: Detection of disease and fruit grading. *In Image Information Processing (ICIIP), 2013 IEEE Second International Conference. IEEE*, pages 521–526.
- [Kulkarni A., 2012] Kulkarni A., P. A. (2012). *Applying image processing technique to detect plant diseases*, volume 2(5).
- [Mahendran R., 2012] Mahendran R., Jayashree G., A. K. (2012). Application of computer vision technique on sorting and grading of fruits and vegetables. *J Food Process Technol S1-001. doi*, 10:2157–7110.



- [Marzal A., 2002] Marzal A., L. I. (2002). Introducción a la programación con python y c. *Publicacions de la Universitat Jaume I*.
- [Mondragón J., 2008] Mondragón J., J. C. (2008). La granada roja; guía para su producción en guanajuato. *Campo Experimental Bajío. INIFAP. Celaya, Guanajuato, México. Folleto técnico*, 2.
- [Mundokar N., 2017] Mundokar N., Kale P., B. M. (2017). Fruit disease detection using color analysis and ann with e-nose. *Imperial Journal of Interdisciplinary Research*, 3(5).
- [Narvekar P., 2014] Narvekar P., Kumbhar M., P. S. (2014). Grape leaf diseases detection analysis using sgdm matrix method. *International Journal of Innovative Research in Computer and Communication Engineering*, 2(3):3365–72.
- [Ortiz M., 2015] Ortiz M., Manuel S., H. P. J. (2015). Diseño de un sistema automático de selección de frutos de café mediante técnicas de visión artificial.
- [Padol P. B., 2016] Padol P. B., Y. A. (2016). Svm classifier based grape leaf disease detection. *In Advances in Signal Processing (CASP), Conference. IEEE.*, pages 175–179.
- [Pajares Martinsanz G., 2008] Pajares Martinsanz G., G. C. (2008). Ejercicios resueltos de visión por computador (no. sirsi). *i9789701513729*.
- [Palanivel N., 2017] Palanivel N., Lavanya S., D. E. (2017). N. pca and rf: an automatic plant leaf disease detection using texture, shape and color features. *International Journal of Engineering Applied Sciences and Technology*, 2:120–125.
- [Pawar, 2012] Pawar, M. (2012). Skin defect detection of pomegranates using color texture features and dwt. *In Computing and Communication Systems (NCCCS), 2012 National Conference IEEE*, pages 1–5.
- [Payrumani, 2008] Payrumani, I. R. (2008). ¿qué es inteligencia artificial? *Revista de Información, Tecnología y Sociedad*.
- [Raut S., 2017] Raut S., F. A. (2017). Review on fruit disease detection using image processing techniques. *Int. J. Innov. Emerg. Res. Eng*, 4(4).
- [Riaño Melo O., 2016] Riaño Melo O., Acosta Medina C., L. P. R. (2016). Clasificación de una imagen satelital empleando máquinas de soporte vectorial para cuantificar el área de pinus patula en una plantación. *Revista Electrónica Redes de Ingeniería*.
- [Rodríguez Carmona J., 2016] Rodríguez Carmona J., P. (2016). Detection using convolutional neural networks: An exploratory study.

- 
- [Rodríguez R., 2011] Rodríguez R., S. J. (2011). *Procesamiento y análisis digital de imágenes*. Ra-Ma, first edition.
- [Samajpati B., 2016] Samajpati B., D. S. (2016). Hybrid approach for apple fruit diseases detection and classification using random forest classifier. *In Communication and Signal Processing (ICCSP), 2016 International Conference. IEE*, pages 1015–1019.
- [Sannakki S., 2011] Sannakki S., Rajpurohit V., N. V. (2011). Leaf disease grading by machine vision and fuzzy logic. *Arun Kumar R et al, Int. J. Comp. Tech. Appl*, 2(5):1709–1716.
- [Theodoridis S., 2010] Theodoridis S., Pikrakis A., K. K. (2010). Introduction to pattern recognition: a matlab approach. *Academic Press*.