

UNIVERSIDAD PERUANA UNIÓN

FACULTAD DE INGENIERÍA Y ARQUITECTURA

Escuela Profesional de Ingeniería de Sistemas



Una Institución Adventista

Diagnóstico automatizado para la clasificación de café trillado con broca mediante procesamiento de imágenes con Deep Learning

Por:

Nilda Espinoza Villafuerte

Asesor:

Ing. Angel Rosendo Condori Coaquira

Juliaca, mayo de 2019

**DECLARACION JURADA
DE AUTORIA DEL INFORME DE TESIS**

Ing. Ángel Rosendo Condori Coaquira, de la Facultad de Ingeniería y Arquitectura, Escuela Profesional de Ingeniería de Sistemas, de la Universidad Peruana Unión.

DECLARO:

Que el presente informe de investigación titulado: **"DIAGNÓSTICO AUTOMATIZADO PARA LA CLASIFICACIÓN DE CAFÉ TRILLADO CON BROCA MEDIANTE PROCESAMIENTO DE IMÁGENES CON DEEP LEARNING"** constituye la memoria que presenta la bachiller Nilda Espinoza Villafuerte para aspirar al título Profesional de Ingeniero de Sistemas ha sido realizada en la Universidad Peruana Unión bajo mi dirección.

Las opiniones y declaraciones en este informe son de entera responsabilidad del autor, sin comprometer a la institución.

Y estando de acuerdo, firmo la presente declaración en Juliaca a los catorce días del mes de mayo del año dos mil diecinueve.



Ing. Angel Rosendo Condori Coaquira

Diagnóstico automatizado para la clasificación de café trillado con
broca mediante procesamiento de imágenes con Deep Learning

TESIS

Presentada para optar el título profesional de Ingeniero de Sistemas

JURADO CALIFICADOR



Mg. Henry Lennin Centurión Julca
Presidente



Dr. Jorge Alejandro Sánchez Garcés
Secretario



Mg. Nilton Omar Santillan Aching
Vocal



Jorge Eddy Otazu Luque
Vocal



Ing. Ángel Rosendo Condori Coaquira
Asesor

Juliaca, 14 de mayo de 2019

DEDICATORIA

Quiero dedicar este informe de tesis en especial para mi padre Antonio Espinoza Pando y mi madre Grimaldina Villafuerte Leo y mis hermanas (os); con quienes estoy muy agradecido porque siempre tuvieron un apoyo incondicional que me brindaron, gracias a ellos y a la ayuda de Dios pude lograr mis sueños trazados.

AGRADECIMIENTO

Agradezco a Dios todo poderoso autor de mi vida y sabiduría; por estar siempre conmigo en cada paso que doy, por hacer cada vez más fuerte mis sentimientos e iluminar mi mente y crecer espiritualmente y formarme como una nueva persona de servicio a la sociedad y la iglesia.

A la casa de estudios “Universidad Peruana Unión” especialmente a los docentes de la Escuela Académico profesional de Ingeniería de Sistemas quienes me impartieron sus conocimientos en el transcurso de mi formación profesional.

Al Ingeniero Ángel Rosendo Condori Coaquira y al Doctor Jorge Alejandro Sánchez Garcés quienes fueron las personas que siempre me impulsaron y estuvieron en todo mi proceso de tesis en cual estoy muy agradecida con ellos y que dios le bendiga infinitamente.

ÍNDICE GENERAL

DEDICATORIA.....	iv
AGRADECIMIENTO	v
ÍNDICE DE TABLAS.....	viii
ÍNDICE DE FIGURAS	ix
ÍNDICE DE ANEXOS	x
SÍMBOLOS USADOS.....	xi
RESUMEN.....	xii
ABSTRACT.....	xiii
CAPÍTULO I EL PROBLEMA	14
1.1 Identificación del Problema.....	14
1.2 Justificación.....	15
1.3 Presunción filosófica	17
1.4 Objetivos de la investigación.....	17
1.4.1 Objetivo general	17
CAPÍTULO II BASES TEÓRICOS.....	18
2.1 Revisión de la literatura.....	18
2.2 Marco teórico.....	20
2.2.1 El café.....	20
2.2.2 Inteligencia artificial.....	23
2.2.3 Visión computacional.....	27
2.2.4 Etapas del procesamiento de la imagen digital.....	29
2.2.5 ¿Por qué el procesamiento de imágenes?	34
2.2.6 Aprendizaje de la maquina	34
CAPÍTULO III MATERIALES Y MÉTODOS	38
3.1 Descripción del lugar de ejecución.....	38
3.2 Metodología de la investigación.....	38
3.2.1 Tipo de investigación	38
3.2.2 Arquitectura de solución.....	38
3.3 Herramientas.....	39
3.3.1 Opencv (librería).....	39
3.3.2 Scikit-Learn: Machine Learning in Python Library	40
3.3.3 Keras.....	40
3.3.4 Anaconda.....	41
3.4 Métodos y procedimientos.....	42

3.4.1	Descriptor de textura con patrones	42
3.4.2	Descriptor de textura de haralick	42
3.4.3	Máquinas de vectores de soporte (SVM)	45
3.4.4	K means	46
3.4.5	Thresholding:.....	46
3.4.6	Método otsu	47
3.5	Procedimiento para la clasificación de broca	48
3.5.1	Estructura del modelo algorítmico	48
3.5.2	Principales módulos.....	48
3.6	Detalles de los módulos del modelo algorítmico.....	49
3.6.1	Caracterización de regiones sanas e infectadas	49
3.6.2	Clasificación automática de regiones sanas e infectadas.....	53
3.6.3	Muestra del proceso de segmentación de las manchas de broca en el grano de café	56
3.7	Caracterización de regiones sanas e infectadas	56
3.7.1	Sobre el conjunto de imágenes a utilizarse.....	56
3.7.2	Sobre la resolución y estandarización de las imágenes	57
3.7.3	Sobre la resolución y estandarización de las imágenes	57
3.7.4	Implementación del descriptor de textura de Haralick.....	57
3.7.5	Generación de los vectores de características.....	60
3.7.6	Extracción de características mediante el uso de histogramas RGB	60
3.7.7	Implementación del operador LBP.....	62
CAPÍTULO IV RESULTADOS Y DISCUSIÓN		65
4.1	Resultado N° 1	65
4.2	Resultado N° 2.....	65
4.3	Resultado N° 3.....	66
CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES		69
5.1	Conclusiones.....	69
5.2	Recomendaciones	69
REFERENCIAS BIBLIOGRÁFICAS		71
ANEXOS.....		75

ÍNDICE DE TABLAS

Tabla 1. Gráfico de la broca del grano de café, <i>Hypothenemus hampei</i> Ferrari	21
Tabla 2. Tabla comparativa de herramientas	41
Tabla 3. Numero de muestras tomadas para las resoluciones.	57
Tabla 4. Resultados de errores según cada modelo.	66

ÍNDICE DE FIGURAS

Figura 1. Estado físico del grano de café.....	15
Figura 2. El proceso del café.	21
Figura 3. Imagen de la broca en café seco.....	22
Figura 4. La broca en el café verde y maduro.	23
Figura 5. Imagen de café trillado con broca.	23
Figura 6. Imagen representativa del funcionamiento del Deep learning para la clasificación	25
Figura 7. Arquitectura de redes neuronales convulsiónales.	26
Figura 8. Arquitectura.....	29
Figura 9. Conversión a RGB la hoja de café en roya amarilla.	30
Figura 10. Segmentación de frutos.	32
Figura 11. Modelo de matriz de confusión.....	37
Figura 12. Arquitectura de solución.	38
Figura 13. El resultado final de tasa de errores en modelos de clasificación.....	45
Figura 14. Imagen de la umbralizacion.	47
Figura 15. Modelo del proceso del algoritmo para el diagnóstico.	48
Figura 16. Caracterización de granos de café con resolución 96.	49
Figura 17. Esquema de carpetas.	49
Figura 18. Generación de los vectores de característica.....	50
Figura 19. Histograma LBP.....	53
Figura 20. Clasificación automática de regiones sanas e infectadas por broca.	54
Figura 21. Imágenes de prueba.....	54
Figura 22. Determinación automática de la clasificación.....	55
Figura 23. Segmentación de manchas de broca.....	56
Figura 24. Fotografía tomada de café bueno	56
Figura 25. Fotografía tomada de café con broca.....	56
Figura 26. Histograma del café bueno.....	61
Figura 27. Histogramas en RGB del café con broca.	62
Figura 28. Matriz de pixeles.	64
Figura 29. Banco de imágenes tomada.	65
Figura 30. Muestra de entrenamiento del algoritmo.	67
Figura 31. Resultado de la clasificación de café broca.....	67
Figura 32. Resultado de clasificación de café bueno.....	68

ÍNDICE DE ANEXOS

Anexo A. Resultado obtenidos según SVN Y CNN	75
Anexo B. CNN redes convulcionales	77

SÍMBOLOS USADOS

- SVM: Máquinas de vectores de soporte
- RGB : (red, Green y blue)
- HSI Espacio de color Intensidad tono saturación
- CIE: El espacio de color de Comisión Internacional de Iluminación
- CNN: Redes neuronales convolucionales
- INEI: Instituto Nacional de Estadística
- LBP: Patrones Binarios Locales

RESUMEN

En la actualidad en el Distrito de Ocobamba la enfermedad de la broca (*Hypothenemus hampei Ferrari*) afecta mucho a la producción agrícola del café, puesto que la selección de los granos se realiza manualmente y esto ocasiona pérdida de tiempo y por ende baja la producción en toda la agricultura de café. El cual realizar la selección es difícil manualmente pues ocasiona pérdida de tiempo y con esta demora de tiempo en la selección afecta para su proceso de exportación y también afecta económicamente a los agricultores. Frente a este problema de selección del grano de café se plantea diagnosticar automáticamente la clasificación en los granos de café trillado que están infectados por la enfermedad de la broca, esto se realizó con el proceso de toma de imágenes de los granos de café con broca y bueno para formar un labels de imágenes. Después de crear los labels de imágenes se procedió a realizar el entrenamiento de los algoritmos con modelos de máquinas de vectores de soporte para su reconocimiento automático, para distinguir rápidamente los granos de café según el estado que tienen cada imagen tomada. En la construcción del algoritmo se utilizó el método de Haralick y LBP las cuales están formadas matemáticamente y estas utilizan matrices las cuales sufren un proceso de conversión en algoritmos para realizar la clasificación automática, por ende realizaron entrenamientos con el labels de imágenes que tenían dos estados que son granos de café con y sin broca. Se obtuvo el resultado en el que se verificó que el algoritmo funciona correctamente en la detección de imágenes de granos de café en 50 imágenes de prueba, en las que el 96% tenía una clasificación óptima y un margen de error del 04%.

Palabras clave: broca, café, imágenes, algoritmos, clasificación y automático.

ABSTRACT

Currently in the District of Ocobamba disease drill (*Hypothenemus Hampei Ferrari*) greatly affects agricultural production of coffee, since the selection of the beans is done manually and this causes losses of time and lower production in all coffee agriculture. Which to makes the selection is difficult manually because it causes waste of time and this delay of time in the selection affects for its export process and also economically affects to the farmers. About, this problem of coffee bean selection, it is proposed to automatically diagnose for the classification in the threshed coffee beans that are infected by the disease of the coffee berry borer, this was done with the process of taking pictures of coffee beans with and without of the “broca” plague for forming an image labels. After creating the image labels, we proceeded to train the algorithms with models of Super Vector Machine (SVM) for automatic recognition, to quickly distinguish the coffee beans according to the state of each image. In the construction of the algorithm were used the Haralick method and Local Binary Pattern (LBP) methods, which are mathematically formed and they use matrices which undergo a conversion process in algorithms to realize the automatic classification, therefore training was done with the labels of images that had two states that are of coffee beans with and without of the “broca” plague. The result was obtained in which it was verified that the algorithm works correctly in the detection of images of coffee beans in 50 test images, in which 96 % was with an optimal classification and a margin of error 04%.

Keywords: broca, plague, coffee, images, algorithms, automatic and classification.

CAPÍTULO I

EL PROBLEMA

1.1 Identificación del Problema

Los agricultores del Distrito de Ocobamba Provincia La Convención Departamento de Cusco, se dedican a la producción de café lo cual difícil de realizar la selección, porque el grano de café es afectado por la enfermedad de broca y después de realizar el proceso del trillado al café seco, se realiza la selección del grano de café manualmente con el personal de trabajo. En la selección de café se requiere mucho tiempo para obtener que todos los granos en un buen estado (sin broca). Según la opinión del Gerente de la cooperativa José Olaya Ltda. N° 252 de Quellouno que es acopiador en el Distrito de Ocobamba quien menciona de la compra de 100 kilos de café en grano sin trillar el descarte del café malogrado es un aproximado de 20 kilos a más. En este Distrito no existe ninguna herramienta óptima para la selección del grano de café la cual es un problema para realizar la selección rápidamente.

Según las estadísticas de la INEI (2017) los departamentos con mayor producción registrada son Junín (62.7%), Cajamarca (3.8%) y Amazonas (1.3%) que en conjunto concentraron el 52.6% del total nacional. También, creció en Lambayeque (235.5%), Pasco (82.8%), Piura (41.1%), Huánuco (18.8%), La Libertad (0.8%) y Puno (0.7%). Por el contrario, disminuyó la producción de este grano en Ayacucho (-30.1%), Ucayali (-7.8%), San Martín (-7.1%) y Cusco (-2.6%) la causa de la disminución en producción del café fue por los hongos de la roya del café esto ataca a las hojas del café y esto hace que la producción del café disminuya.

Según el autor Camilo (2014) se observó un efecto de la enfermedad de la broca en el peso del grano de café pilado. En promedio el peso de 100 granos sanos pesó 16.20 gramos ± 4.40 . Los frutos clasificados defectuosos por broca severa presentaron el menor peso (9.8 gr ± 3.57), seguido de los granos con defecto de broca moderada y basado en estos hallazgos se observó una significativa reducción en el peso de los granos de café pilados, poco más del 40%, para el caso de los frutos calificados con ataque severo de broca. También se

observó una reducción, aunque en menor proporción, para el caso del peso del grano calificado como moderado y levemente brocado y esta investigación se realizó en áreas cafetaleras de la zona de Jarabacoa de la República Dominicana.

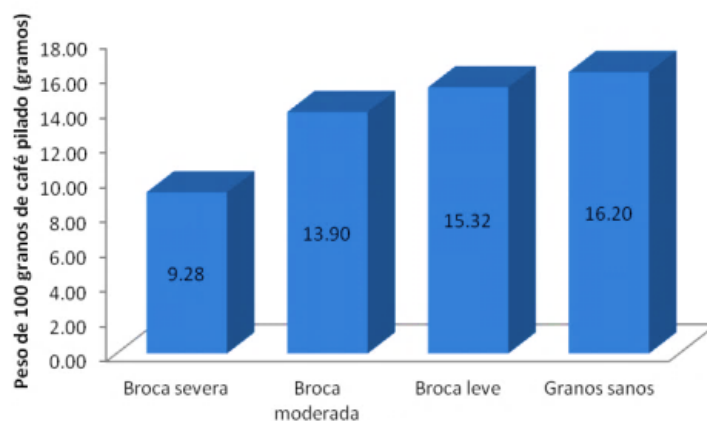


Figura 1. Estado físico del grano de café.
Fuente: Impacto económico de la broca del café (*Hypothenemushampei*) en la caficultura de Jarabacoa (Camilo, 2014).

Según el investigador Ramirez (2017) realizó una investigación sobre la visión artificial en café verde arábica con procesamiento de imágenes, logró la detección de imágenes de granos de café con visión sin defectos físicos. También logró obtener las características del café utilizando histogramas de intensidad, la cual el investigador utilizó varios métodos y también influye en ahí lo que es la Central de Cooperativas Agrarias Cafetaleras de los Valles de Sandía (CECOVASA), Perú.

Como se puede apreciar el problema la enfermedad de broca se presenta en los países cafetaleros como Colombia, Brasil, República Dominicana y también en Perú, en nuestro país se realizaron algunas investigaciones a nivel de cooperativas agrarias especialmente en la cooperativa CECOVASA, de los valles de Sandía, pero aún es un problema latente especialmente para los agricultores del Distrito de Ocobamba, Cusco, con los cuales se realizó esta investigación.

1.2 Justificación

La siguiente investigación se realiza porque en la actualidad en el Distrito de Ocobamba no existe ninguna herramienta óptima para la selección de granos de café en especial cuando

ataca la enfermedad de la broca, esta enfermedad afecta al café desde su estado de maduración por la cual en el café grano seco no es visible la enfermedad de la broca que afecto, es visible cuando el grano de café es trillado es por ello se realiza el diagnostico automatizado mediante toma de fotos para la clasificación de café bueno y broca y de esa manera se agilice el proceso.

Según la entrevista realizada a la Cooperativa Agraria José Olaya nos dio la información que para realizar la exportación del café se necesita que este en buen precio económicamente para que favorezca a la población y a la cooperativa. El café debe tener una buena calidad, porque en la actualidad la selección se realiza manualmente esto demora mucho tiempo, la cual las ultimas acopiaciones que se realiza la cooperativa de los agricultores para la exportación es en donde se demora mucho tiempo en la selección y ya no se alcanza al mismo precio de las primeras exportaciones que se realizó y eso no favorece ni a la cooperativa también a los agricultores.

Según la INEI (2014) el Departamento de cusco tiene una producción en sus cultivos de café en el año 2013 la producción fue en toneladas 38 545 en el ámbito de café la cual nos quiere decir que la mayoría de las personas a nivel del departamento de cusco se dedican a la agricultura de café la cual va ser muy importante para realizar el algoritmo de procesamiento de imágenes.

Según la tesis de Arias y Sierra (2016) llegó a la conclusión con su algoritmo utilizado que es con el Matlab en detección de imágenes cumplió con su objetivo principal, la forma de visualizar datos y manipular variables es fácil gracias al workspace, por último las herramientas disponibles en MATLAB facilitan la manipulación de imágenes de manera interactiva y didáctica. En la cual utilizo una herramienta libre y también el lenguaje de programación c++ y que es compatible con Opencv lo cual facilita la implementación a futuro.

Esta investigación se realizó gracias a la clasificación automática de café con broca y bueno, con lo cual se logra la calidad final del café para su proceso de exportación se mejore y con esto los agricultores de café se mejoren económicamente.

Por otro lado el uso de tecnologías de la información en los procesos que realizan los agricultores apoya en sus estilos de vida y trabajo con lo cual disponen de mayor tiempo para realizar más actividades y la cual también disminuye el tiempo para que se realice la exportación ya que con el uso de las tecnologías se disminuye el tiempo de selección.

1.3 Presunción filosófica

Según el texto bíblico **Marcos 10:45** “Porque el Hijo del Hombre no vino para ser servido, sino para servir, y para dar su vida en rescate por muchos.” Nos dice que nosotros somos para el servicio a la comunidad tal como Dios nos enseñó, también estar con el prójimo por que dios nos enseñó siempre estar para cualquier servicio y enseñanzas dar al pueblo, para de esa manera los agricultores mejoren económicamente.

El texto bíblico **Hebreos 13:16** dice Y no os olvidéis de hacer el bien y de la ayuda mutua, porque de tales sacrificios se agrada Dios; con esto nos quiere decir cualquier sacrificio que ágamos tal como dios nos enseñó y hacer el bien para el servicio del pueblo y realizar una ayuda mutua para el agrado de Dios, será recompensado por nuestro dios según mi investigación será un apoyo para la comunidad para de esa manera se mejore en el sector de agrícola ya que la población es el recurso que necesitan para salir adelante.

1.4 Objetivos de la investigación

1.4.1 Objetivo general

Diagnosticar automáticamente la clasificación de café trillado con broca mediante procesamiento de imágenes con Deep Learning.

1.4.1.1 Objetivos específicos

- Crear los grupos de imágenes según labels (bueno y con broca)
- Entrenar los algoritmos clasificadores que permitan detectar el café bueno y con broca y validar los clasificadores entrenados.
- Automatizar los algoritmos desarrollados para los granos de café trillado.

CAPÍTULO II

BASES TEÓRICOS

2.1 Revisión de la literatura

En la universidad Pontificia Católica del Perú Barriga y Arrasco (2018) realizó la investigación para el “diagnóstico automático de roya amarilla en hojas de café aplicando técnicas de procesamiento de imágenes y aprendizaje de máquina, el objetivo fue desarrollar un modelo algorítmico que permita automatizar” el proceso de diagnóstico de roya amarilla en hojas de café, de tal forma que detecte cuando una hoja está o no infectada, y en caso lo estuviese, calcular el grado de severidad del hongo utilizando el método fue Patrones Binarios Locales (LBP), Descriptor de textura de Haralick, Máquinas de Vectores de Soporte (SVM), Thresholding y Método Otsu y llegando a un resultado. Se lograron conseguir todos los objetivos planteados, desde la extracción de características de una región sana o infectada hasta la cuantificación del grado de severidad y la implementación de todo el modelo algorítmico en una herramienta grafica funcional.

Según Rios (2018) investigó sobre la aplicación de técnicas de “aprendizaje profundo para la clasificación y reconocimiento de objetos en imágenes los objetivos era desarrollar un sistema automático de análisis de imágenes que permita el reconocimiento” de un conjunto limitado de objetos utilizando la metodología de clasificación basada en la similitud y contexto de los objetos. A lo largo del desarrollo del proyecto se pudo concluir que la inclusión de la red pre entrenada de Alex Net permitiría generalizar todos los objetos que conforman la base de datos creada con la obtención de sus resultados. Las pruebas realizadas para poner a prueba el funcionamiento del proyecto se han constituido en tres etapas. Inicialmente se debe evaluar el sistema de inteligencia desarrollado en Matlab, evidenciando los resultados que arrojan las matrices de confusión y las gráficas de superficie a la hora de clasificar imágenes.

En la tesis realizada por Ramirez (2017) en la universidad Nacional San Agustín de Arequipa consistió en reconocimiento de granos de café verde arábica sin defectos físicos en muestras usando visión artificial; el café de la especie arábica es un producto de alta

demanda en el mercado internacional. Dicho reconocimiento es propenso a errores, debido a que actualmente es realizado en su mayoría de forma manual y subjetiva. Los principales trabajos que tratan este problema basado en visión artificial requieren de prototipos de adquisición, que toman cada imagen desde un ángulo completamente vertical con respecto a la superficie que contiene a la muestra de granos de café con el objetivo de proponer un modelo para el reconocimiento automático de granos de café verde arábica sin defectos físicos en imágenes de muestras capturadas manualmente con un Angulo de toma diagonal usando visión artificial y basándose para el reconocimiento en el estándar de la SCAA y utilizando el método de la adquisición de imágenes, mejora de imágenes, segmentación, extracción de características, clasificación y comparación con otros trabajos y al final se llegó al resultado de que siguientes métodos : Se pudo ver que el método de segmentación tuvo algunos errores debido a tres casos los cuales son: separación no visible, sombra y conclusión. Además el método de selección de parámetros y validación, muestra como a mayor cantidad los resultados casi no varían mucho entre valores del 94% al 100% de precisión general. Finalmente en la comparación con otros trabajos se vio que la propuesta de este trabajo es superior a los otros con los que se comparó.

En la universidad Pontificia Javeriana de Colombia Arias y Sierra (2016) realizaron el estudio en procesamiento de imágenes para la clasificación de café verde por que este país se caracteriza e la producción de café la cuales consta con 5 partes de procesos la cuales son recolección, despulpado , secado, trillado y tostado . Los procesos de clasificación normalmente son: selección manual o selección a nivel industrial, los cuales ofrecen desventajas como el agotamiento del trabajador llevando a errores en la selección o el alto costo de la maquinaria industrial, lo que le dificulta al pequeño productor ofrecer un producto de alta calidad. El objetivo de los investigadores fue desarrollar un algoritmo para la detección de defectos en café verde mediante el procesamiento de imágenes utilizando el método de procesamiento basados en la adquisición digita, a diferencia de la análoga se concentra en el desarrollo de algoritmos computacionales para la obtención de información o el uso de diferentes herramientas diseñadas y llegando a un resultado donde se observan resaltados en color verde, los falsos rechazos del sistema, , de 90 granos buenos 15 son el falso rechazo ,de estos cinco fueron por solo color, uno solo broca y nueve por color y broca.

El investigador Viera (2017) de la Universidad de Piura realizó en procesamiento de imágenes usando OpenCV aplicado en Raspberry Pi para la clasificación del cacao con el propósito de desarrollar de un sistema de visión artificial para la clasificación de granos de cacao según características externas como el tamaño en la fase final del secado; las cuales se obtuvo un resultado finalmente, en la clasificación resultante, el acierto obtenido por la clasificación realizada por los trabajadores de la asociación es de aproximadamente 78% . Y acierto obtenido por la clasificación realizada por el programa es de aproximadamente 89%.

2.2 Marco teórico

2.2.1 El café

Es uno de los recursos naturales más consumidos tanto en el mundo como en el Perú, lo que garantiza es que es orgánico y son granos de calidad y es considerada como el producto más importante en la agrícola en el comercio internacional (Barriga y Arrasco, 2018)

2.2.1.1 El proceso del café

El proceso de elaboración del café ya desde un punto de un fruto maduro son cinco pasos son de la siguiente manera: recolección, el despulpado es un proceso se retira la cascara más gruesa del café, luego realizar el secado y para realizar el trillado y el tostado(Arias y Sierra, 2016)

- a) **Recolección:** se recolectan los cafés rojos o amarillos que está en la rama de la planta para que la cosecha no dure más de 2 semanas, evitando que la semilla se seque en el árbol.
- b) **Despulpado:** el despulpado de café se realiza después del recojo del café
- c) **Secado:** el secado del café se tiene que realizar sobre una tarima para que tenga una buena calidad.
- d) **Trillado:** se realiza para su respectiva exportación
- e) **Tostado:** Esto se realiza para el procesamiento del café.

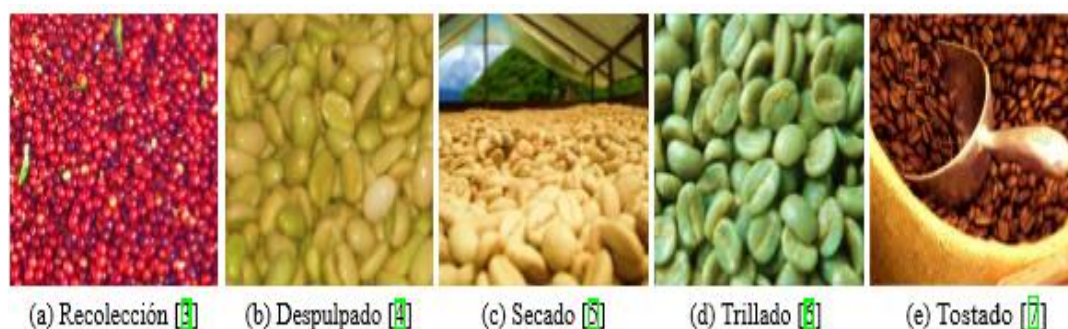


Figura 2. El proceso del café.

Fuente: Procesamiento de imágenes para la clasificación de café verde (Arias y Sierra, 2016).

2.2.1.2 La broca en el café

La broca del café (*Hypothenemus hampei*) es dañada por unos hongos en algunos cafés desde el estado en que están en cerezo o verde, la broca se nota en la elección de café y cuando ya es café arábico (trillado) en el procesamiento y esto hace que el café trillado tenga defectos físicos y afecta a la exportación, es como una plaga que afecta al café desde el estado de maduración (Camilo, 2014).

2.2.1.3 Clasificación taxonómica y hospederos de broca

En la clasificación de la taxonomía de la broca son los siguientes que se detalla a continuación:

Tabla 1.

Gráfico de la broca del grano de café, *Hypothenemus hampei* Ferrari

Reino:	Animalia
Phylum:	Arthropoda
Subphylum:	Uniramia
Clase:	Insecta
Orden:	Coleoptera
Suborden:	Polyphaga
Familia:	Curculionidae
Género:	Hypothenemus
Especie:	hampei Ferrari

Fuente: Evaluación de la factibilidad para la aplicación de un plan de manejo integrado de plagas para el control de broca (*hypothenemus hampei*) en cafetales de la isla santa cruz (Torres, 2018 p.24).

2.2.1.4 Daños que ocasiona

La broca provoca la caída de frutos verdes y maduros esta enfermedad puede perjudicar la producción el 30 % y 80% de la cosecha, porque al afectar esta enfermedad bajara la calidad del grano de café y afecta principalmente el rendimiento del beneficio, porque los granos que están perforados por las brocas la cuales afectan mucho para la exportación y por ende causan perdidas en el mercado nacional e internacional (Acosta, n.d.).

2.2.1.5 Metodología de evaluación y diagnóstico

La broca es atraída por los cafetales por el olor que tienen según la literatura, se indica que la broca es un insecto muy abundante bajo condiciones de sombra, pero esto no significa que sus poblaciones sean mayores en este hábitat; Los períodos prolongados de sequía en los cafetales causan una maduración acelerada de los frutos, por lo que la almendra es más pequeña y deformada por la falta de agua, lo que da como resultado café de calidad inferior. También, se puede producir la caída de frutos y si éstos están infestados con la broca, ésta continúa reproduciéndose en su interior. La broca, como se dijo antes, no emerge de los frutos durante los períodos secos, solamente inicia su salida cuando comienzan las lluvias debido a que por efecto de las precipitaciones la broca no permanece mucho tiempo en los frutos y su reproducción es menor (Bustillo, 2015).



Figura 3. Imagen de la broca en café seco.

Fuente: Impacto económico de la broca del café (*Hypothenemus hampei*) en la caficultura de Jarabacoa (Camilo ,2014).



Figura 4. La broca en el café verde y maduro.
Fuente: Plagas del café en nuestra región (Agroecotur, 2016).



Figura 5. Imagen de café trillado con broca.
Fuente: Procesamiento de imágenes para la clasificación de café verde (Arias Y Sierra, 2016).

2.2.2 Inteligencia artificial

La inteligencia artificial es un componente biológico que siendo esto un elemento con diversas entradas que tiene ramificaciones de las células que estén presentes en las neuronas y con variable de salida que es la neurita, las entradas son multiplicadas por un vector de pesos para luego ser de esa manera comparadas o analizadas por una función que determinara la salida, esto realiza un énfasis a un aprendizaje automático es como estaría unas aplicaciones basadas en algoritmos y técnicas evolutivas (Rios, 2018).

2.2.2.1 Machine learning

El aprendizaje automático es hacer que las computadoras se programen a sí mismas. Si la programación es una automatización, entonces el aprendizaje está automatizando para el proceso. El software de escritura en el cuello de botella, no tenemos suficientes buenos desarrolladores. Deja que los datos hagan el trabajo en el lugar de las personas. El aprendizaje automático es la manera de hacer que la programación se escalable (Jason Brownlee, 2016).

Programación tradicional: los datos y el programa se ejecutan en la computadora para producir la salida.

Aprendizaje automático: los datos y la salida se ejecutan en la computadora para crear un programa. Este programa puede ser utilizado en la programación tradicional.

El aprendizaje automático es como la agricultura o la jardinería. Las semillas son los algoritmos, los nutrientes son los datos, el jardinero es usted y las plantas son los programas

2.2.2.1.1 Deep learning

Es un aprendizaje exhaustivo del subconjunto de Machine Learning que tiene varias bibliotecas, librerías, algoritmos y el desarrollador es que el aprendizaje profundo está inspirado en el cerebro mismo para crear redes neuronales con subconjuntos de niveles y esto nos permite obtener patrones o características simples de voces complejas, en un ejemplo de un caso de detección de rostro, las primeras capas son responsables de reconocer la intensidad de la luz y así sucesivamente hasta alcanzar la capa final que ya es responsable de la detección de la cara (Rios, 2018).

Esta es una forma de aprendizaje automático que permite a las computadoras aprender de la experiencia y comprender el mundo de las jerarquías de conceptos. Debido a que las computadoras acumulan conocimiento sobre la experiencia, esta jerarquía permite que las computadoras tengan la facilidad de aprender conceptos complejos al construirlos a partir de conceptos más simples (Goodfellow, Bengio & Courville, 2016).

“Es un trasfondo matemático donde nos abarcan muchos temas que son probabilidades, computación numérica y aprendizaje automático y se describe este modelo de secuencia como reconocimiento de voz, visión artificial y recomendación lineal” (Lecun, 2015).

El Deep learning, es llamado redes neuronales artificiales, puede estimar o aproximar funciones que puedan ser una cantidad de un gran número de entradas desconocidas, y hoy en día puede ejecutar pequeños modelos de aprendizaje en su computadora portátil (Guerrero, 2016).

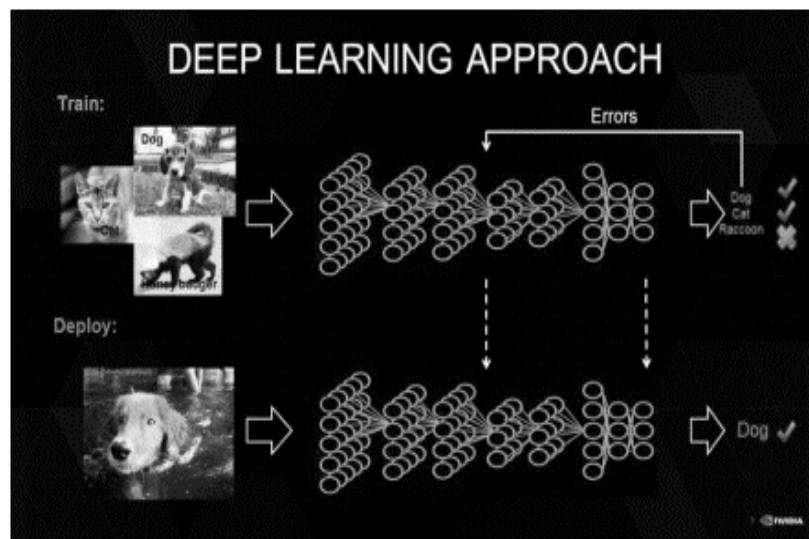


Figura 6. Imagen representativa del funcionamiento del Deep learning para la clasificación de animales.

Fuente: Aplicación de técnicas de aprendizaje profundo para la clasificación y Reconocimiento de objetos en imágenes (Rios, 2018).

Librerías

Deep learning tiene muchas librerías entre ellos tenemos algunos en las cuales nos sirven para el procesamiento de imágenes.

- a) Tensorflow: Es una librería libre para machine intelligenci soporta el Python y c++ es para cálculos numéricos mediante diagrama de flujos de datos y nodos que representan diagramas de operaciones matemáticas. Tiene una flexible arquitectura y se necesita solo una API para desplegar y su función es para el aprendizaje automático (Tensorflow, 2018)
- b) Theano: Es una librería de código abierto más veterana y estable permite evaluar las expresiones matemáticas en matrices multidimensionales e involucran grandes datos

es un estándar de la industria para la investigación de aprendizaje profundo y desarrollo. Fue diseñada exclusivamente para la utilización en los tipos de computación que son requeridos en grandes empresas (Python, 1996)

- c) Keras: Esta es una biblioteca de Python fue desarrollada para permitir la experimentación rápida y es para reducir la carga cognitiva, puede entenderse sólo como una secuencia o como un gráfico, proporciona lo justo para lograr un resultado, sin florituras y maximizando la legibilidad (Python, 1996).

2.2.2.1.2 Redes neuronales convulsiónales

Las redes neuronales son parecidas a las redes neuronales multicanal la principal ventaja es que cada red entrena para realizar una tarea, por lo cual el entrenamiento es más rápido y presenta invariancia a la traslación de los patrones. Las redes neuronales convulsiónales son muy buenos en lo que es análisis de imágenes asta son capaces de detectar características simples, lo que son detección de bordes, líneas y entre otros y esta fue entrenada a partir de las imágenes utilizando solo pixeles y etiquetas de enfermedades como entradas; estas constan de múltiples capas diseñadas (Calvo, 2017).

- La arquitectura

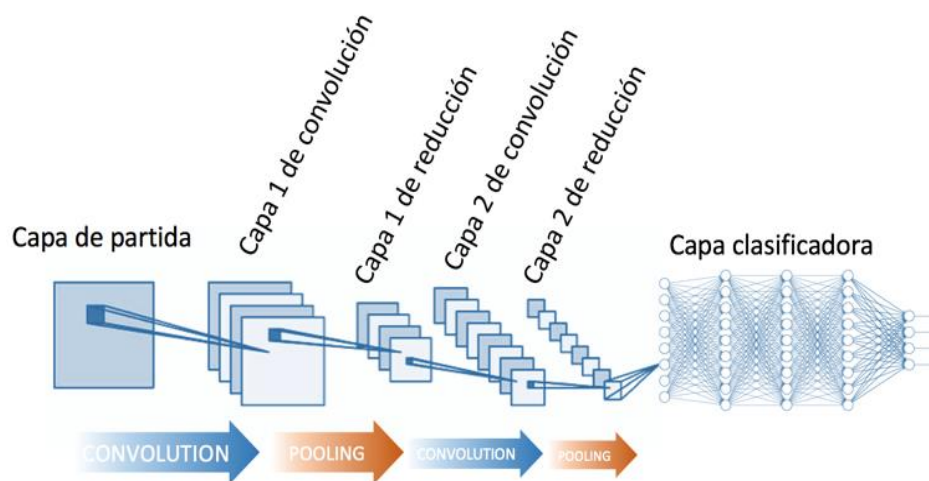


Figura 7. Arquitectura de redes neuronales convulsiónales.
Fuente: Red Neuronal Convulsional CNN (Diego Calvo, 2017).

En La arquitectura de una CNN mostrada según la Figura 7 Arquitectura de redes neuronales convulsiónales, consiste en que tiene una capa de partida, capas ocultas y una

capa clasificadora la que es de salida es activación pooling convolution, el convoltion es el consiste en el conjunto de filtros cada filtro es como un cuadrado pequeño durante el filtro se realiza el ancho y altura del volumen de entrada (Hamza Bendemra, 2018).

2.2.3 Visión computacional

En visión computacional está conectada con el aprendizaje de máquina, es como visión por computador extracción automática características útiles de las imágenes. La visión computación es basada en una visión humana y esta trabaja con un proceso modelado usando un software (Comstor, 2017).

Ahora bien, debemos considerar que la visión computacional y el procesamiento de imágenes son cuestiones diferentes. El procesamiento de imagen trata sobre cómo mejorar una imagen para su interpretación por una persona mientras que la visión computacional trata de interpretar las imágenes por la computadora (Loyo, n.d.).

La visión computacional se puede dividir en 3 grandes etapas:

- Procesamiento de nivel bajo. Se extraen propiedades como orillas, gradiente, profundidad, textura, color, etcétera.
- Procesamiento de nivel intermedio. Agrupa elementos de nivel bajo para obtener contornos y regiones con el propósito de segmentar.
- Procesamiento de alto nivel. Consiste en la interpretación utilizando modelos y/o conocimiento del dominio del problema.

2.2.3.1 Librerías

La visión computación tiene muchas librerías con las cuales se puede realizar detección de imágenes en tiempo real o una extracción automática asta estas pueden ser utilizadas en cámaras de video vigilancia para rastreos de algún robo y aquí tenemos las siguientes librerías.

- a) DL4J: Esta librería es para java esta flexibilidad le permite combinar codificadores automáticos variaciones, codificadores automáticos secuencia a secuencia, redes convolucionales o redes recurrentes según sea necesario (Guerrero, 2016) .

- b) Opencv: Es una librería creada para el aprendizaje por visión del computador, se ha utilizado en infinidad de aplicaciones, al ser una fuente abierta nos facilita la biblioteca cuenta con más de 2500 algoritmos optimizados que son de visión artificial y un aprendizaje automático (opencv, 2018).
- c) Convolucionales: Fueron desarrolladas para detección de objetos, están logrando resultados de última generación en tareas de visión por computador; redes neuronales convolucionales para el aprendizaje profundo, esta es la razón por la que la red es tan útil para el reconocimiento de objetos en las fotografías, la selección de dígitos, caras, objetos y así sucesivamente con orientación variable (Python, 1996).
- d) Skimage: con esta librería es una colección de algoritmos que podría ser utilizado para procesamiento de imágenes esta librería es forma gratuita o tiene ninguna restricción funciona con la herramienta de pyhon, y con la matriz numpy expone exactamente los datos en pixeles (Schönberger, Nunez, Boulogne, Warner, Yager, Gouillart, 2014).

Se llegó a una conclusión donde la librería de opencv es para el aprendizaje por visión del computador, al ser una fuente abierta nos facilita la biblioteca cuenta con más de 2500 algoritmos optimizados que son de visión artificial y un aprendizaje automático es por la cual se optó por esta librería

2.2.3.2 Algoritmos

- a) Canny: El algoritmo es para la realización de detección de bordes de una imagen esto de suma importancia esto nos facilita el reconocimiento de objetos y segmentaciones es uno de los mejores algoritmos en detección de bordes es un algoritmo de múltiples capas (Python, 1996)
- b) Umbralización: En la sección anterior, usamos un valor global como valor de umbral pero puede que no sea bueno en todos los casos donde las imágenes difieren en condiciones de luz en diferentes áreas. En ese caso, utilizamos umbrales adaptativos. En esto, el algoritmo calcula el umbral para una pequeña región de la imagen. Así obtenemos diferentes umbrales para diferentes regiones de la misma imagen. Y nos da mejores resultados para imágenes con iluminación variante (Python, 1996).

- c) Segmentación: Es para realizar escala de grises y realizar una binarización en la imagen con valores de 0 a 255 para binarizar es muy importantes que los pixeles de las imágenes estén umbralizados es en una escala gris (Raizel, 2014).

2.2.3.3 Arquitectura

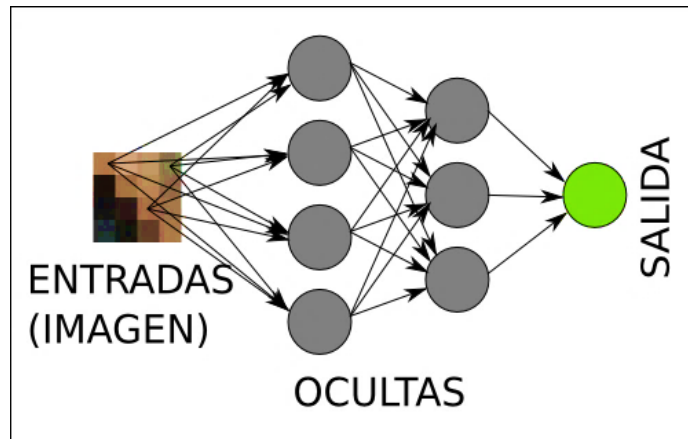


Figura 8. Arquitectura.

Fuente: <https://rubenlopezg.wordpress.com/2014/05/07/que-es-y-como-funciona-deep-learning/>

2.2.4 Etapas del procesamiento de la imagen digital

En esta etapa están los procesos que se siguen para el procesamiento de imágenes, para que sea entendible la investigación.

2.2.4.1 Adquisición

Es una imagen digitalizada es la primera etapa de un sistema de visión artificial y consistió en recabar las imágenes digitales en la cual las imágenes lo representaron como funciones continuas para realizar un análisis simple de las imágenes obtenidas, si se detectan fallas o posibles mejoras, se trata de corregir el sistema de adquisición y adquirir un nuevo conjunto de imágenes (Viera, 2017).

2.2.4.2 Imagen digitalizada

Las imágenes digitalizadas es cuando se realizó el procesamiento de imagen en café verde se realizó un cambio de color a la imagen y la distribución del color de la imagen lo realiza en histogramas, en imágenes digitales lo represento con pixeles que tienen una determinada intensidad; el procesamiento de imágenes hace referencia al uso de la computadora para la obtención de características, valores numéricos que permiten describir la imagen, teniendo

en cuenta que una imagen digital es una colección finita de elementos con una posición particular, estos elementos normalmente son llamados píxeles, para la obtención de características con valores numéricos que permiten describir la imagen teniendo en cuenta que una imagen digital (Arias y Sierra, 2016, p. 3).

2.2.4.3 Espacios de color

Los espacios del color nos sirven mucho para realizar clasificación según los colores o conversión de colores de la imagen los colores están representados por los canales R, G y B, que definen respectivamente las intensidades con que los colores primarios rojo, amarillo y azul están presentes. Por lo tanto, es un modelo aditivo, ya que el color está representado por la suma de los valores de sus canales (Ramírez, 2017).

2.2.4.3.1 RGB (red, Green y blue)

El modelo de RGB se representa con tres vectores que son 3 colores primarios. Los modelos se basan en un sistema cartesiano en el que los colores que son rojo, verde y azul representan diferentes coordenadas que son x, y, z, que se aplicó en la tesis para detectar la enfermedad en las hojas de café donde se realizó la transformación estructural de la imagen RGB al espacio LUV, luego se enmascaran y eliminan los píxeles verdes mediante un umbral que sigue un proceso de segmentación (Barriga y Arrasco, 2018).

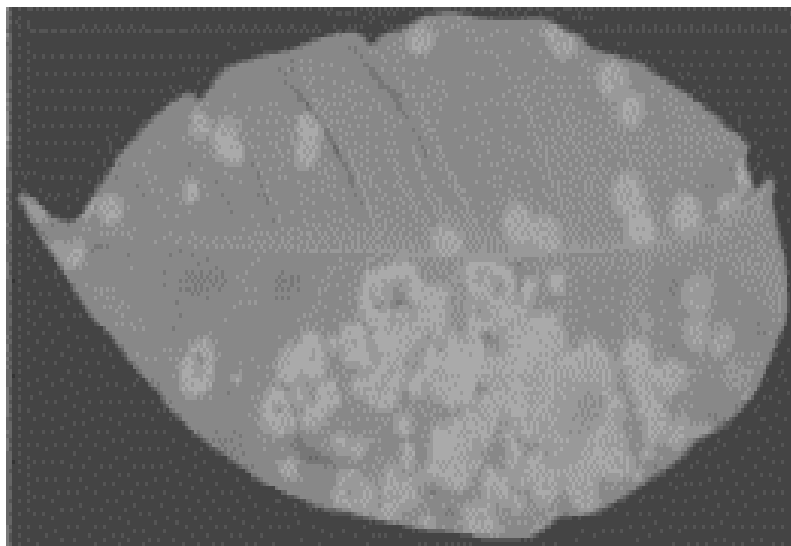


Figura 9. Conversión a RGB la hoja de café en roya amarilla.

Fuente: Diagnóstico automático de roya amarilla en hojas de cafeto aplicando técnicas de procesamiento de imágenes y aprendizaje de máquina (Barriga y Arrasco, 2018).

2.2.4.3.2 Espacio de color Intensidad tono saturación (HSI)

El espacio de color HSI se basa en cómo percibimos los colores que tienen los humanos, este sistema caracteriza el color en términos del tono que los componentes mostrados le permiten ser favorable cuando el tono de color es importante para la segmentación. Las transformaciones matemáticas que permiten que este espacio de color es que es posible modificar la luminosidad de la imagen (Torres, 2018).

Sin embargo, al usar el espacio HSI, podemos segmentar los colores según el tono, una luz verde y una roja se distinguen por su diferente tonalidad que representa la longitud de onda de ese color, sin necesidad de recurrir a la intensidad de la luz. Además, algunos colores que se encuentran en la naturaleza (púrpura, por ejemplo) no se pueden obtener de la descomposición de la luz blanca, pero se obtienen de la mezcla de dos colores (azul y rojo), por lo tanto, no se pueden representar con el RGB Modelo, aunque en este caso se utilizan aproximaciones. El componente de saturación permite distinguir la blancura de un color (el color rojo muestra la saturación máxima y el color rosado mínimo) (Alegre, Pajares, & Junio, 2016)

2.2.4.3.3 El espacio de color de Comisión Internacional de Iluminación (CIE)

Este espacio de color se definió un modelo que permite representar todos los colores mediante síntesis aditiva de tres fuentes primarias con una combinación de estos tres colores pueden corresponderse con cualquier fuente monocromática de luz. Uno de los primeros espacios de color definidos por CIE fue el CIE XYZ donde cada una de sus componentes primarias de color fue definida experimentalmente, basados directamente en mediciones de cómo el ojo humano percibe los colores, este espacio sirvió de base para la construcción de otros espacios de color (Torres, 2018, p. 46).

2.2.4.3.4 Los espacios XYZ, Luv, Lab.

Con las representaciones anteriores del espacio del color aparecen coeficientes negativos para colores de determinadas longitudes de onda, lo cual supone una fuente de error importante en los cálculos colorimétricos (comparación de un color monocromático con el blanco de referencia). La CIE (Commission International de Eclairage) estableció un nuevo espacio de color (XYZ) con la finalidad de evitar los coeficientes negativos. En dicho

espacio la ordenada Y (luminosidad) es perpendicular al plano definido por XZ (coloridad) de luminancia nula. Este espacio de color se caracteriza por ser independiente del dispositivo que se esté usando (Alegre et al, 2016).

2.2.4.4 Segmentaciones

El proceso de segmentación según el que el espacio y los algoritmos de segmentación más conocidos se basan en propiedades básicas de la imagen, como discontinuidad y similitud, el proceso de segmentación, solo se utilizaron granos segmentados correctamente, pero debido a que la clase de defecto físico contenía una cantidad irregular para cada defecto, procedió a tener la misma cantidad de granos por defecto, esta cantidad es de 29 granos por defecto porque esta cantidad era la misma cantidad que el defecto con menos número de granos segmentados correctamente, por lo que finalmente la clase de defecto físico Tiene un total de 348 granos (Ramírez, 2017).

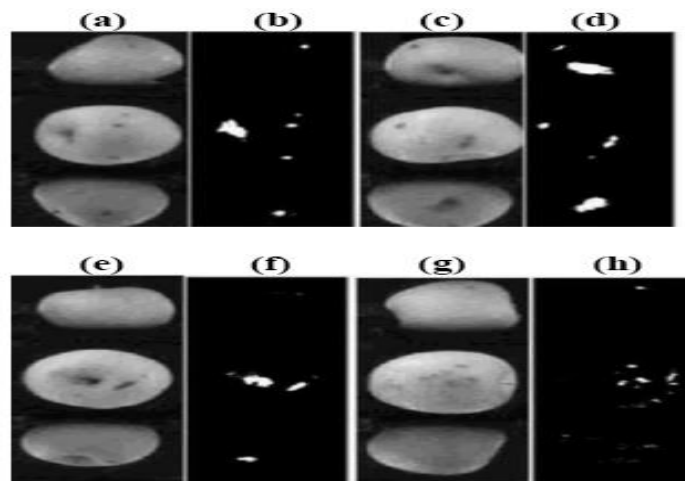


Figura 10. Segmentación de frutos.

Fuente: Reconocimiento de granos de café verde arábica sin defectos físicos en muestras usando visión artificial (Ramírez, 2017).

2.2.4.4.1 Segmentaciones basadas en regiones

Esta es basada en las técnicas de segmentación que se definen en regiones esto es directamente un análisis de características de los píxeles las cuales siempre toda imagen está compuesta, este proceso realiza los valores mismos de la imagen que tiene sin ninguna necesidad de que la imagen sea transformada, La prueba de la homogeneidad es completamente dependiente de la aplicación en si se sugiere que se tiene que definir

predicados basados en características que resulten similares , una de las características básica, la varianza se define como base en los tonos de gris de los pixeles de la imagen esta fuera mejor que un umbral (Chacon, Sandoval & Vega, 2015).

2.2.4.5 Descriptores de imágenes

Los descriptores de imágenes son específicamente para extraer las características por visión del computador es una distribución de probalidades de la forma que un objeto de va considerando distorsiones espaciales por el tamaño; también con el objetivo que el clasificador es analizar los datos obtenidos y de esa manera distinguir entre dos objetos; es decir que todo esto es con el fin de extraer las características necesarias para hacer uso de esa información ene l proceso de reconocer nuevas imágenes (Cuenca, 2008).

2.2.4.6 Operaciones morfológicas

La morfología matemática para extraer elementos de la imagen que son útiles en la representación y descripción de la forma de una región y la morfología binaria son las operaciones de deslizamiento de un elemento sobre una imagen en su forma y tamaño y depende de las características seleccionadas de los objetos analizados. Existen dos tipos de procesamiento morfológico que es por dilatación y erosión; la dilatación consiste en es incorporación de un objeto todos los puntos que se conectan al objeto que esta pueda ser utilizada para diferencia entre dos objetos, mientras en el otro que es la erosión es eliminar todos los puntos del contorno es como los pixeles de los objetos esta puede ser utilizada para eliminar de una imagen los detalles demasiados pequeños (Viera, 2017 p. 65).

- **Erosión:** Esto consiste en la disminución de pixeles blancos de una imagen binaria, esto tiene como base la morfología estructurante de un elemento, dicho elemento se representa a través de una matriz cuadrada de 1'a y 0's que matemática es expresada de la siguiente manera (Navarrete, 2018).

$$A \ominus B = \{X / (B)_x \subseteq A\}$$

Donde A representa a la imagen binaria, y B al elemento estructurante.

- **Dilatación:** Esta consiste en el aumento de pixeles blanco de la imagen binaria esto es tomado como una base morfología de un estructurante que representa a

través por una matriz cuadrada conformada por 1's y 0's. la matemática de la operación de la dilatación se expresa de este modo (Navarrete, 2018).

$$A \oplus B = \{x \mid [(\widehat{B})_x \cap A] \subseteq B\}$$

Donde A representa a la imagen binaria, y B al elemento estructurante.

2.2.5 ¿Por qué el procesamiento de imágenes?

La clasificación o el descarte de la broca en el café trillado se realiza manualmente por una inspección visual la cual tiene una relación con procesamiento de imágenes, se ve la posibilidad de utilizar cámaras no especializadas y de uso común por ejemplo la cámara de un teléfono móvil (Arias y Sierra, 2016).

2.2.6 Aprendizaje de la maquina

Es una rama de la inteligencia artificial que tiene como objetivo desarrollar métodos para que las computadoras modifiquen o adapten sus acciones para que estas acciones sean precisas y aprendan y elijan las acciones más correctas. El aprendizaje automático generalmente se divide en dos tipos: aprendizaje supervisado y aprendizaje no supervisado (Barriga y Arrasco, 2018).

2.2.6.1 Aprendizaje supervisado

Este aprendizaje se basa en la experiencia que conforma el conjunto de algoritmos y técnicas capaces de aprender de obtener los datos de entrenamiento y obtener respuestas correctas. Estos algoritmos deberían tener la capacidad de generalizar obtener respuestas para cualquier entrada que no esté en los datos de entrenamiento (Barriga y Arrasco, 2018 p. 25).

2.2.6.2 Aprendizaje no supervisado

El aprendizaje no supervisado es encontrar conjuntos con datos de entrada eso el objetivo principal, los subgrupos con características similares tener la necesidad y tener la información previa y los datos de entrenamiento y hay una diferencia del aprendizaje supervisado el algoritmo consiste en que debe hallar similitudes por sí mismo y clasificar los elementos segunda las características que se logran se identifican (Barriga y Arrasco, 2018).

2.2.6.3 Aprendizaje por refuerzo

Se basan en modelos y funcionalidades enfocadas en aprovechar al máximo las recompensas. Para ello, debe basarse en acciones y tener en cuenta el tipo de entorno en el que se desarrollará el robot. Este es el algoritmo que más se aproxima al comportamiento de los humanos, ya que su modelo se basa en la acción-recompensa. Con ello, se intenta que el algoritmo se adhiera a la recompensa que otorga el entorno y que las acciones que emprenderá estén sujetas a esas recompensas. Está claro que es el más adecuado para aprender robots. Es importante tener en cuenta que ninguno de estos sistemas es más importante que los otros. Su calificación se mide de acuerdo con lo que se logrará en el aprendizaje automático. Sólo en ese caso se elegirá el algoritmo (Briega, 2015).

2.2.6.4 Reconocimiento de patrones y clasificaciones

Es un conjunto de características y en las imágenes son conocidas como descriptores en el aprendizaje automático hace la referencia al problema donde tiene que tener una automatización en la detección de hojas y esto se aplico es hojas de café que tienes la enfermedad de roya y hojas sanas (Barriga y Arrasco, 2018, p. 25).

2.2.6.5 Técnicas de reconocimiento para la validación de modelos de clasificación

- **validación cruzada** : Es la predicción a partir de imágenes o señales cerebrales de la cual requiere una evaluación empírica de su poder predictivo. En esa evaluación se logra también se utiliza para ajustar los hiper-parámetros de los decodificadores la validación cruzada para la decodificación en neuroimagen los aspectos prácticos se resaltan con un extenso estudio empírico de los decodificadores comunes en las predicciones internas y transversales de los sujetos; la validación cruzada anidada puede ajustar los parámetros de los decodificadores mientras evita el sesgo de circularidad. Sin embargo, encontramos que puede ser favorable utilizar valores predeterminados sanos, en particular para decodificadores no dispersos (Varoquaux et al, 2017).
- **Leave One Out:** Este procedimiento es atractivo por dos razones. Primero, la mayor cantidad posible de datos utilizados para el entrenamiento, en cada caso, aumenta la probabilidad de que el clasificador sea más preciso. Segundo, el procedimiento es

determinista: es decir, no tiene sentido repetir el procedimiento 10 veces, o repetirlo en absoluto: ya que se obtendrá el mismo resultado cada vez. Sin embargo, dejar uno fuera parece ofrecer la oportunidad de aprovechar al máximo un pequeño conjunto de datos y llegar lo más preciso posible. (Barriga y Arrasco, 2018).

2.2.6.6 Métricas de evaluación del rendimiento de un clasificador

En este contexto, es esencial conocer en detalle la precisión de las diferentes pruebas de diagnóstico, es decir, la capacidad de clasificar correctamente a los pacientes en categorías o estados en relación con la enfermedad, generalmente dos, por ejemplo: si el café tiene un tamaño pequeño o no distribuidos a una escala continua, por lo que es necesario elegir un punto de corte apropiado o un valor límite que permita resumir los resultados de dos categorías: positivo y negativo. En la presente investigación, se utilizan las eficiencias de eficiencia: sensibilidad, productividad específica, positiva y negativa, así como la clasificación correcta para evaluar el desempeño en la clasificación. Para un sistema de clasificación binario como el umbral de la decisión varía. La luz de decisión es la que decide si un caso, en el vector de producción de la clasificación, pertenece o no a cada una de las clases. Los indicadores de eficiencia utilizados en esta investigación para evaluar las distancias en la clasificación son áreas bajo la curva (Rodríguez, Martínez, 2016).

$$F - measure = \frac{2 \times (Precision \times S_e)}{Precision + S_e}$$

$$H - mean = \frac{2 \times (S_p \times S_e)}{S_p + S_e}$$

2.2.6.6.1 Matriz de confusión

En una matriz de confusión son resultados de las predicciones acerca de un problema de clasificaciones en esta realiza las muestras en que el modelo de clasificación se confunde cuando realiza las predicciones, brinda información sobre los errores cometido, esta es utilizada con un clasificador binario que so por asignaciones predeterminadas y determinadas para ver los errores que está ocurriendo (Brownlee, 2016).

		Predicción	
		Positivos	Negativos
Observación	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Figura 11. Modelo de matriz de confusión.
Fuente: <https://rpubs.com/chzelada/275494>.

Descripción del modelo de la matriz de confusión

- **VP** es la cantidad de positivos que fueron clasificados correctamente como positivos por el modelo.
- **VN** es la cantidad de negativos que fueron clasificados correctamente como negativos por el modelo.
- **FN** es la cantidad de positivos que fueron clasificados incorrectamente como negativos.
- **FP** es la cantidad de negativos que fueron clasificados incorrectamente como positivos

CAPÍTULO III MATERIALES Y MÉTODOS

3.1 Descripción del lugar de ejecución

El lugar donde se ejecuto fue en el Distrito de Ocobamba Provincia La convención Departamento de Cusco para los agricultores de café.

3.2 Metodología de la investigación

3.2.1 Tipo de investigación

La investigación se realizó en el Distrito Ocobamba con los agricultores de café la cual la investigación es propositiva y aplicada

3.2.2 Arquitectura de solución

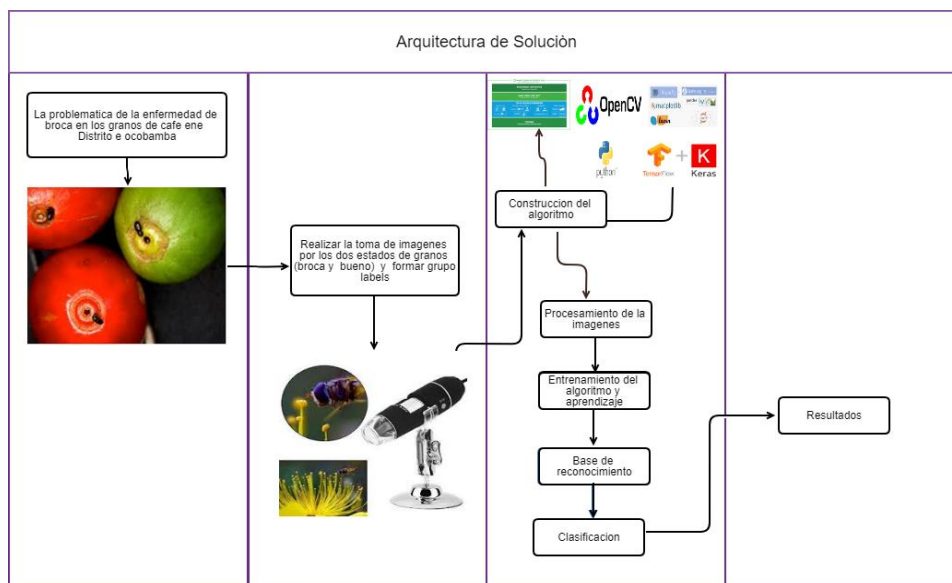


Figura 12. Arquitectura de solución.
Fuente: Elaboración propia.

3.2.2.1 Fases de la arquitectura de solución

- 1) Los agricultores que se dedica a la producción del café las cual afecta la enfermedad de la broca en los granos de café y esta se realiza la selección manualmente.

- 2) Toma de fotografías a variedades granos de café con broca y bueno con una cámara microscópica para realizar la digitalización de las imágenes.
- 3) Construcción del algoritmo utilizando las librerías opencv, Python, anaconda y algoritmos.
- 4) Procesar con las imágenes digitalizadas.
- 5) Entrenamiento y aprendizaje del Algoritmo
- 6) Base de reconocimiento de las imágenes de prueba
- 7) Clasificación de la imágenes de prueba
- 8) Posteriormente aprenda la máquina y nos den resultados automáticamente en la detección de imágenes de café.

3.3 Herramientas

A continuación detallare las herramientas que tuvieron más relevancias en la ejecución del proyecto.

3.3.1 Opencv (librería)

OpenCV es una librería o paquete de código abierto, la cual alberga cientos de algoritmos de visión computacional. Fue diseñada para ser muy eficiente, ya que está escrita en C/C++ optimizado y toma ventaja del multiprocesamiento. Actualmente, cuenta con una comunidad de usuarios de más de 47 mil personas, ha sido adoptada y ampliamente utilizada alrededor del mundo, ya que soporta plataformas Linux, Windows, Mac OS, IOS, Android; además, brinda interfaces para Python, C y C++; la cual me permitirá en mi investigación realizar con Python la detección de imágenes digitales de café con broca y sin broca (Barriga y Arrasco, 2018, p. 13).

La librería Opencv se optó por utilizar por que como se menciona en el anterior párrafo es una librería de código abierto, la cual alberga con cientos algoritmos de visión computacional y es muy eficiente, también brinda interfaces para Python en detección de imágenes y como también mencionados por los anteriores autores, es por ello que se optó por utilizar esta librería.

3.3.2 Scikit-Learn: Machine Learning in Python Library

Scikit-learn se ha convertido en uno de los eventos abiertos más populares, a cuenta con bibliotecas de aprendizaje de código fuente para Python. scikit-learn proporciona algoritmos para realizar tareas de aprendizaje automático que incluyen clasificación, regresión, reducción de dimensionalidad y agrupamiento. También proporciona módulos para extraer características, procesar datos. Bibliotecas de Python NumPy y matplotlib, NumPy extiende Python para soportar eficientemente operaciones en matrices grandes y matrices multidimensionales, matplotlib proporciona herramientas de visualización, y SciPy proporciona módulos para computación científica (Hackeling, 2014).

Scikit-learn es popular para la investigación académica porque tiene un bien documentado, la cual utiliza el API fácil de usar y versátil. Los desarrolladores pueden usar scikit-learn para experimentar con muchos diferentes algoritmos cambiando solo unas pocas líneas del código. scikit-learn wraps en algunas implementaciones populares de algoritmos de aprendizaje automático, como LIBSVM y LIBLINEAR. Otras bibliotecas de Python, incluyendo NLTK, incluyen envoltorios scikit-learn también incluye una variedad de conjuntos de datos, lo que permite a los desarrolladores centrarse en los algoritmos en lugar de obtener y limpiar los datos. Esta cuenta con licencia permisiva, scikit-learn puede usarse en comerciales y aplicaciones sin restricciones. Muchos de los algoritmos de scikit-learn son rápidos y finalmente, scikit-learn se destaca por su confiabilidad; gran parte de la biblioteca está cubierta por pruebas automatizadas (Hackeling, 2014)

Esta es una herramienta utilizada para Python y es gratuito y tiene una compatibilidad con Numpy y matplotlib. Esta es una biblioteca de aprendizaje es por la cual se optó por esta herramienta.

3.3.3 Keras

Es una librería de Python que proporciona, de una manera sencilla, la creación de una gran gama de modelos de Deep Learning usando como backend otras librerías como TensorFlow. El código ha sido liberado bajo la licencia permisiva la cual permite expresar redes neuronales de una manera muy modular, considerando un modelo como una secuencia.

Keras está actualmente conectado a Tensorflow, pero también se puede usar como una biblioteca de Python. Al principio, creo que la segunda opción es la más adecuada y, para

esto, mi sugerencia será usar Jupyter, porque es un entorno de desarrollo muy extendido y muy fácil de usar, y luego explica por qué uso el jupyter.

Keras es una librería de alto nivel la cual nos permite un aprendizaje profundo con unos mínimos códigos y mínimos conocimientos esta es muy utilizada para temas de clasificación y la cual se optó por esta librería.

Tabla 2.
Tabla comparativa de herramientas

Herramienta	características
Spider	Su entorno es Python es similar al Matlab y no tiene acceso a muchas bibliotecas y tienes que explorar archivos
Jupyter	Es un entorno de Python y R donde tiene varios accesos a bibliotecas científicos y numéricos, es juntamente se instala con anaconda y al iniciar la sesión se abrirá en un navegador y cada vez al editar el código se guarda automáticamente.

Fuente: Anaconda, Jupyter, Spyder (Vellayaraj, 2017).

Se llegó la conclusión según la tabla comparativa que se realizó que utilizare el jupyter porque tienes el acceso a muchas bibliotecas y sea más factible realizarlo.

3.3.4 Anaconda

Es una colección poderosa de paquetes para Python que permite la gestión a gran escala de datos, análisis y visualización para la Inteligencia de Negocios, Análisis científico e Ingeniería de aprendizaje automático, también provee un entorno de desarrollo (IDE) de distribución libre para Python con el cual se ha desarrollado la programación en este proyecto. Este IDE contiene más de 195 paquetes (librerías) para este lenguaje de programación (Barriga y Arrasco, 2018).

Anaconda provee instaladores para versiones de Python 2.7, 3.4, 3.5 y 3.7 la cual se optó por la versión 3.7 y ya viene con algunas librerías incluidas es por esta que se utilizó este lenguaje de programación ya que viene agrupado en el anaconda el python.

3.4 Métodos y procedimientos

Se describen los métodos y procedimientos utilizados en el proyecto.

3.4.1 Descriptor de textura con patrones

El operador LBP puede acceder, además de los modelos estadísticos y modelos estructurales del análisis de textura. El modelo de dispositivo LBP descrita por primera vez en términos Micro-primitivas (Textones) y sus reglas estadísticas; opcionalmente, las primitivas pueden ser utilizadas como una medida complementaria del contraste local de la imagen, lo cual mide la fuerza de las primitivas. Al principio el operador LBP de textura fue introducido como una medida complementaria para medir el contraste local de una imagen. En esta primera versión el LBP trabaja con los 8 vecinos más cercanos de un píxel y maneja los conceptos de umbral y pesos (Ojala, 1996).

3.4.2 Descriptor de textura de haralick

Permiten características cuantitativas, como la diferencia en los diferentes ángulos de los diferentes ángulos, la diferencia, la diferencia según la cobertura del objeto caracterizar la textura se ha calculado para cada objeto la matriz de co-ocurrencias de niveles de gris, derivando de ella los descriptores definidos por Haralick. Esta información se ha complementado con cálculos de coeficientes de curtosis y asimétricos caracterizados por la distribución del histograma, y os valores de media desviación estándar de la intensidad de bordes, se incluyen las características derivadas de la transformada wavelet. (Recio, Fdez. Sarría, Ruiz & Hermosilla, 2008).

A continuación mostramos las siguientes ecuaciones que definen las características.

$P(i,j)$ es la entrada (i,j) en una dependencia espacial tono-gris con una matriz normalizada $= p(i,j)/R$ que es la constante de normalización.

- Segundo momento angular

$$f_1 = \sum_i \sum_j \{p(i,j)\}^2$$

- Contraste: donde $i-j$ es igual a n

$$f_2 = \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) \right\}$$

- Correlación : Donde: u_x, u_y, σ_x y σ_y significan la desviación estándar de p_x y p_y

$$f_3 = \frac{\sum_i \sum_j (ij) p(i,j) - u_x u_y}{\sigma_x \sigma_y}$$

- Suma de cuadrados: varianza

$$f_4 = \sum_i \sum_j (i - u)^2 p(i,j)$$

- Momento de diferencia inversa

$$f_5 = \sum_i \sum_j \frac{1}{1 + (i - j)^2} p(i,j)$$

- Suma promedio:

$$f_6 = \sum_{i=2}^{2N_g} i p_{x+y}(i)$$

- Suma de varianza

$$f_7 = \sum_{i=2}^{2N_g} (i - f_6)^2 p_{x+y}(i)$$

- Suma de entropía

$$f_8 = \sum_{i=2}^{2N_g} p_{x+y}(i) \log\{p_{x+y}(i)\}$$

- Entropía

$$f_9 = \sum_i \sum_j p(i,j) \log(p(i,j))$$

➤ Diferencia de varianza

$$f_{10} = \text{varianza}(p_{x-y})$$

➤ Diferencia de entropía:

$$f_{11} = \sum_{i=0}^{N_{g-1}} p_{x+y}(i) \log\{p_{x-y}(i)\}$$

➤ Medidas de información de correlación:

$$f_{12} = \frac{HXY - HXY1}{\max\{hx, hy\}}$$

$$f_{13} = (1 - \exp[-2.0(HXY2 - HXY)]^{1/2}$$

Donde:

- $HXY = \sum_i \sum_j p(i,j) \log(p(i,j))$
- HX y HY son entropías de p_x y p_y
- $HXY1 = - \sum_i \sum_j p(i,j) \log\{p_x(i)p_y(j)\}$
- $HXY2 = - \sum_i \sum_j p_x(i)p_y(j) \log\{p_x(i)p_y(j)\}$

➤ Coeficiente de máxima correlación:

$$f_{14} = (\text{segundo mayor valor propio de } Q)^{1/2}$$

Dónde:

- $(i,j) = \sum_k \frac{p(i,j)p(j,k)}{p_x(i)p_y(k)}$

3.4.3 Máquinas de vectores de soporte (SVM)

Las máquinas de vectores soporte (SVM, del inglés Support Vector Machines) están diseñados para resolver problemas de clasificación binarios, que actualmente se utilizan para resolver otros tipos de problemas (regresión, agrupación, clasificación múltiple). También hay varias áreas en las que se han utilizado con éxito, como la visión artificial, el reconocimiento de caracteres, las categorías de texto e hipertexto, la clasificación de proteínas, el procesamiento del lenguaje natural y el análisis de series de tiempo. De hecho, desde su introducción, han sido reconocidos adecuadamente por sus sólidos fundamentos teóricos; optimización margen geométrica representa un problema de optimización cuadrática con restricciones lineales que puede ser resuelto mediante técnicas de programación estándar cuadrados necesaria para resolver una característica curva garantiza una solución única, en comparación con mayo de unicidad de la solución producida por una red neuronal artificial entrenada (Suárez, 2014).

Las cuales se optó por el SVM es por los resultados de una investigación que a continuación se detalla:

METODO	Entrenamiento	Prueba	Diferencia
	% error (d.e)	% error (d.e)	
Naïve Bayes	75,3 (5,58)	68,9 (3,07)	6,3
Bayes Informativo	63,8 (6,15)	84,7 (2,50)	-21,0
k-vecinos	91,6 (1,84)	54,4 (2,84)	37,2
Arbol de decision	91,9 (2,78)	53,7 (1,89)	38,2
Red Neuronal	97,3 (1,84)	68,5 (3,17)	28,8
Maquina de Soporte de Vectores	84,9 (1,49)	66,4 (3,63)	18,5

Figura 13. El resultado final de tasa de errores en modelos de clasificación.

Fuente: Comparación de modelos de clasificación automática de patrones texturales de minerales presentes en los carbones colombianos (Carvajal, 2004).

En la Figura 13. El resultado final de tasa de errores en modelos *de* muestra los resultados donde el error de entrenamiento de diferentes métodos como se puede observar, en las cuales está el método el SVM data de un 84,9 % del error de entrenamiento, en cambio observamos que la red neuronal data de un 97,3 por ciento la cual es muy alto es un índice

de error muy alto en la red neuronal, en el error de prueba que se realiza en el SVM data de un 66,4 % y la red neuronal tiene un 68,5 % por el cual se optó por utilizar el método de SVM.

3.4.4 K means

El K -means es un algoritmo de agrupamiento utiliza el refinamiento iterativo para producir un resultado final. Las entradas del algoritmo son el número de grupos K y el conjunto de datos. El conjunto de datos es una colección de características para cada punto de datos. Los algoritmos se inicia con estimaciones iniciales para los *kappa* centroides, que o bien se pueden generar aleatoriamente o se selecciona entre el conjunto de datos al azar (Trevino, 2016).

3.4.5 Thresholding:

Es también conocida como un método umbral la cual utilizan como límites que permiten decidir qué conjunto de tonos de gris pertenece a una determinada región u objeto, también define un rango de valor de intensidad de una imagen original. En una segmentación se puede escoger uno o más umbrales, consiguiendo, en el primer caso, una imagen binaria y, en el segundo, una imagen clasificada. Las principales ventajas de esta técnica son que se trata de una técnica de segmentación rápida, tiene un coste computacional bajo y puede desarrollarse durante la captura de la imagen (Viera, 2017).

La imagen segmentada (x ,) por umbralización viene dada por:

$$g(x, y) = \begin{cases} 1, & \text{if } f(x, y) > T \\ 0, & \text{if } f(x, y) \leq T \end{cases}$$

Cuando la variable T se puede aplicar a una toda una imagen, la variable se conoce como umbral global, pero cuando el valor de T cambia según áreas en la imagen, la variable se conoce como umbral variable.

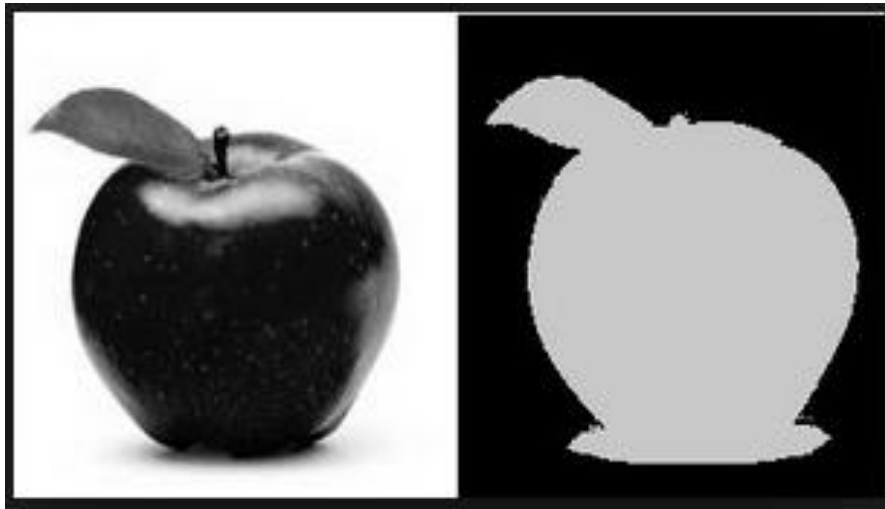


Figura 14. Imagen de la umbralizacion.
Fuente: Obtenida de Opencv documentación(opencv, 2018).

3.4.6 Método otsu

Es un método muy importante debido a que se debe conocer el umbral de cada imagen para ellos es necesario las siguientes formulas

Umbral optimo $T = \max(\sigma^2)$

T es igual a máximos de sigma al cuadrado

Donde Varianza es $\sigma^2 = W_B(\mu_B - \mu)^2 + W_F(W_F - U)^2$

Sigma al cuadrado es igual w sub B por μ sub B – μ todo al cuadrado más W sub por W sub F menos μ todo al cuadrado

Probabilidad acumulada $W_K = \sum_{i=0}^K P_i$

Sumatoria de la p_i donde i baria de 0 a k

Media acumulada $\mu_K = \sum_{i=0}^K i \cdot P_i$

Sumatoria de los i por p_i donde i varía desde 0 hasta k

3.5 Procedimiento para la clasificación de broca

3.5.1 Estructura del modelo algorítmico

A continuación describiremos la evaluación y parametrización de métodos para la elaboración del modelo del algoritmo final

3.5.2 Principales módulos

El algoritmo desarrollado está compuesto por métodos y funciones adecuadamente parametrizadas en la experimentación realizada de tal forma solucione el problema planteado

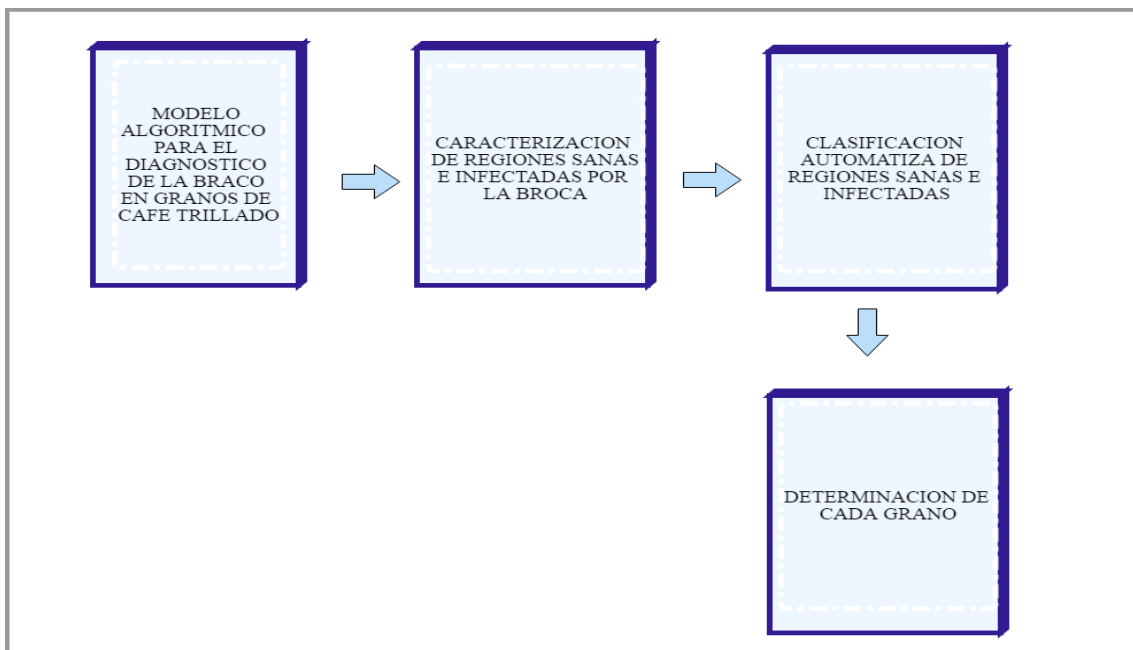


Figura 15. Modelo del proceso del algoritmo para el diagnóstico.
Fuente: Elaboración propia.

En la Figura 15. Modelo del proceso del algoritmo para el diagnóstico nos muestra una estructura del proceso del modelo del algoritmo la cual funcionara las cuales se optó por realizar en 4 fases para realizar el proceso del diagnóstico, es en donde consiste en la caracterización para llegar al final a la determinación de cada grano que si está o no infectado por la broca.

3.6 Detalles de los módulos del modelo algorítmico

3.6.1 Caracterización de regiones sanas e infectadas

En este módulo se realiza con dos objetivos principales que son conformar banco de imágenes e implementar los descriptores que sean necesarios para extraer caracteres de este banco de imágenes que se toman.

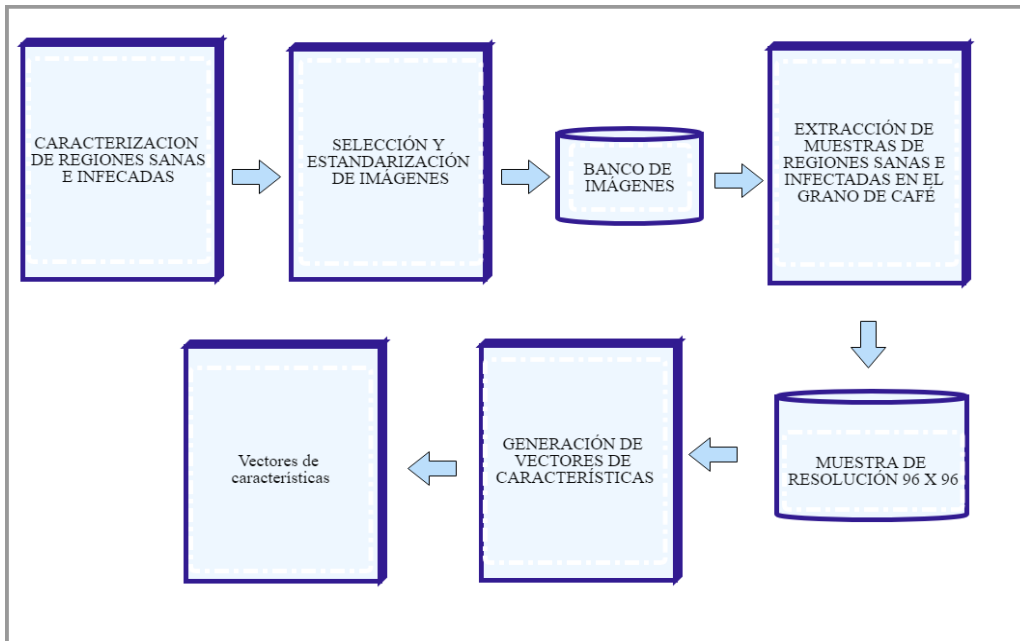


Figura 16. Caracterización de granos de café con resolución 96.

Fuente: Elaboración propia.

Según la Figura 16. Caracterización de cafés de las muestras de detalla sobre la selección de estandarización las cuales son realizadas con el banco de imágenes que fueron tomadas, las imágenes tuvieron una resolución de 96 x 96 la cual culmina con la caracterización de vectores.

Las cuales las imágenes fueron guardadas con el siguiente esquema de carpetas para un mejor entrenamiento de las imágenes la cual fue de la siguiente manera como se muestra en la Figura 17.

Este equipo > Nuevo vol (E:) > proyectotesisfinal > tesisHaralick > dataset				
	Nombre	Fecha de modifica...	Tipo	Tamaño
	test	8/04/2019 14:52	Carpeta de archivos	
	train	2/04/2019 23:16	Carpeta de archivos	

Figura 17. Esquema de carpetas.

Fuente: Elaboración propia.

A continuación observamos el proceso de generación de los vectores y sus características, la cual se implementaron diferentes descriptores que son métodos encontrados en literatura, uno es basado en textura y el otro netamente en el color de la imagen.

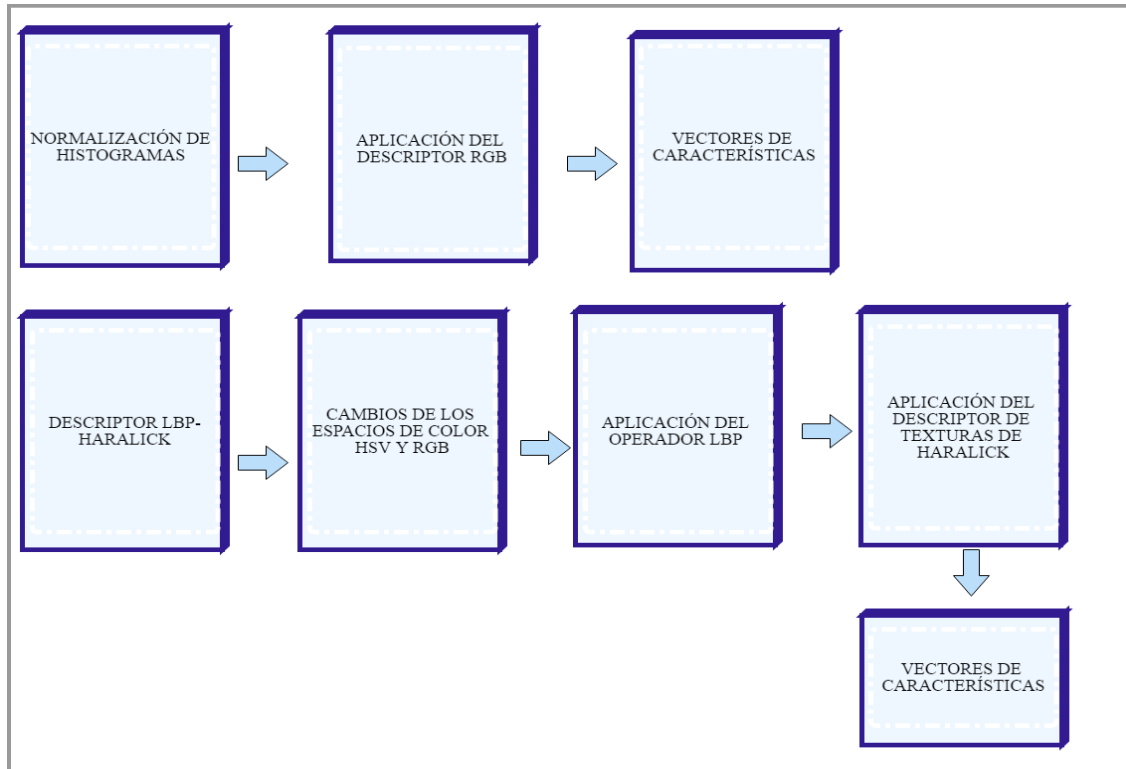


Figura 18. Generación de los vectores de característica
Fuente: Elaboración propia

En la Figura 18. Generación de los vectores de característica se realiza el esquema de caracterización de texturas de las imágenes con el modelo de LBP, la cual con las histogramas se realización la aplicación del descriptor RGB las cuales se obtienen resultados después de aplicar los modelos de LBP y la textura de haralick se obtienen vectores de característica.

- **La generación del algoritmo lbp para los histogramas:** la cual se importa la librería cv2, también el numpy la cual es utilizado para matrices se construye una biblioteca de matemáticas de alto nivel para matrices y operaciones la cual es una librería; también se importa el matplotlib está sirve para generar los gráficos es decir donde nos muestra el histograma. Tal como se muestra en la siguiente línea de código.

```

import cv2
import numpy as np
from matplotlib import pyplot as plt

def get_pixel(img, center, x, y):
    new_value = 0
    try:
        if img[x][y] >= center:
            new_value = 1
    except:
        pass
    return new_value

def lbp_calculated_pixel(img, x, y):
    """
    1 | 2 | 4
    -----
    8 | 0 | 16
    -----
    32 | 64 | 128
    """
    center = img[x][y]
    val_ar = []
    val_ar.append(get_pixel(img, center, x-1, y+1)) # top_right
    val_ar.append(get_pixel(img, center, x, y+1)) # right
    val_ar.append(get_pixel(img, center, x+1, y+1)) #
    bottom_right
    val_ar.append(get_pixel(img, center, x+1, y)) # bottom
    val_ar.append(get_pixel(img, center, x+1, y-1)) #
    bottom_left
    val_ar.append(get_pixel(img, center, x, y-1)) # left
    val_ar.append(get_pixel(img, center, x-1, y-1)) # top_left
    val_ar.append(get_pixel(img, center, x-1, y)) # top

```

```

    power_val = [1, 16, 128, 64,
32, 8, 1, 2]
    val = 0
    for i in range(len(val_ar)):
        val += val_ar[i] *
    power_val[i]
    return val

def show_output(output_list):
    output_list_len =
len(output_list)
    figure = plt.figure()
    for i in range(output_list_len):
        current_dict = output_list[i]
        current_img =
current_dict["img"]
        current_xlabel =
current_dict["xlabel"]
        current_ylabel =
current_dict["ylabel"]
        current_xtick =
current_dict["xtick"]
        current_ytick =
current_dict["ytick"]
        current_title =
current_dict["title"]
        current_type =
current_dict["type"]
        current_plot =
figure.add_subplot(1,
output_list_len, i+1)
        if current_type == "gray":

current_plot.imshow(current_img
, cmap = plt.get_cmap('gray'))

current_plot.set_title(current_title
)

current_plot.set_xticks(current_x
tick)

current_plot.set_yticks(current_y
tick)

```

```

current_plot.set_xlabel(current_x
label)

current_plot.set_ylabel(current_y
label)
    elif current_type ==
"histogram":

current_plot.plot(current_img,
color = "black")

current_plot.set_xlim([0,260])

current_plot.set_title(current_title
)

current_plot.set_xlabel(current_x
label)

current_plot.set_ylabel(current_y
label)
    ytick_list = [int(i) for i in
current_plot.get_yticks()]

current_plot.set_yticklabels(ytick
_list,rotation = 90)

plt.show()

def main():
    image_file = 'broca.jpg'
    img_bgr =
cv2.imread(image_file)
    height, width, channel =
img_bgr.shape
    img_gray =
cv2.cvtColor(img_bgr,
cv2.COLOR_BGR2GRAY)

    img_lbp = np.zeros((height,
width,3), np.uint8)
    for i in range(0, height):
        for j in range(0, width):

            img_lbp[i, j] =
lbp_calculated_pixel(img_gray, i,
j)
            hist_lbp =
cv2.calcHist([img_lbp], [0],
None, [256], [0, 256])
            output_list = []
            output_list.append({
                "img": img_gray,
                "xlabel": "",
                "ylabel": "",
                "xtick": [],
                "ytick": [],
                "title": "Gray Image",
                "type": "gray"
            })
            output_list.append({
                "img": img_lbp,
                "xlabel": "",
                "ylabel": "",
                "xtick": [],
                "ytick": [],
                "title": "LBP Image",
                "type": "gray"
            })
            output_list.append({
                "img": hist_lbp,
                "xlabel": "Bins",
                "ylabel": "Number of
pixels",
                "xtick": None,
                "ytick": None,
                "title": "Histogram(LBP)",
                "type": "histogram"
            })

            show_output(output_list)

            cv2.waitKey(0)
            cv2.destroyAllWindows()
            print("LBP Program is
finished")

if __name__ == '__main__':
    main()

```

- También se muestra la imagen del resultado del lbp

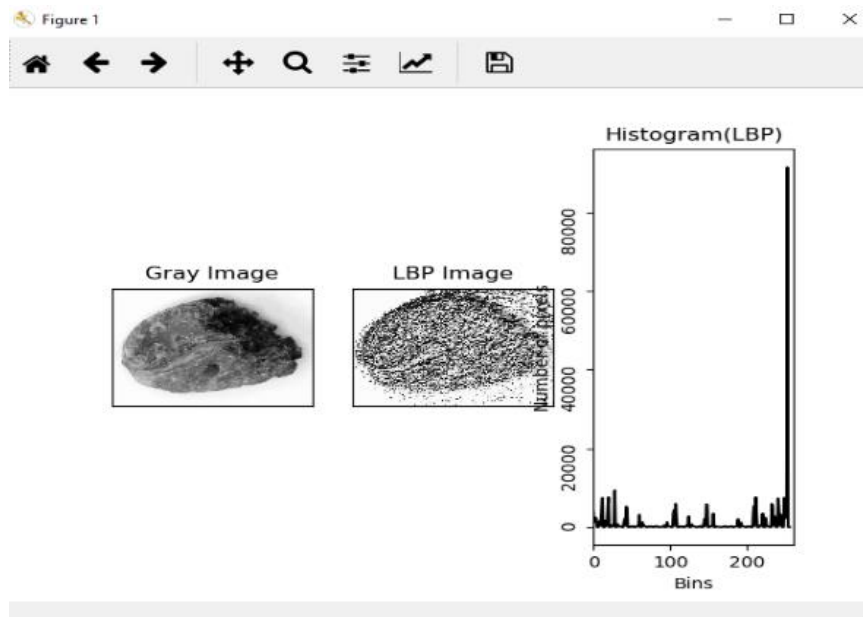


Figura 19. Histograma LBP.

Fuente: Elaboración propia.

En la Tabla 19 nos muestra el resultado del histograma Según LBP de una imagen con broca, para realizar el histograma primeramente se realizó la importación de la librería numpy, la biblioteca Matplotlib se importó el módulo pyplot después se realizó el pixel de la imagen en las ejes x , y desde el centro de la imagen, para luego calcular los pixeles en una matriz de 3 x 3 seguidamente se tuvo que dar la ruta de la imagen la cual la imagen estaba con el nombre de broca.jpg donde después se dice que i es altura y j es ancho de la imagen y las respuestas nos muestren en la escala gris, después nos dio un resultado que nos muestra en la imagen las cual nos muestra que llega hasta un numero de pixeles a más de 80000 es el pixel que realiza el algoritmo y también nos muestra el número de contenedores la cual fue el límite en el eje x fue de 260 y es respecto a los bordes de las imágenes.

3.6.2 Clasificación automática de regiones sanas e infectadas

En este proceso se muestra la construcción de diferentes modelos de clasificación es construido a partir del algoritmo de aprendizaje de la máquina que es más conocido como SVM.

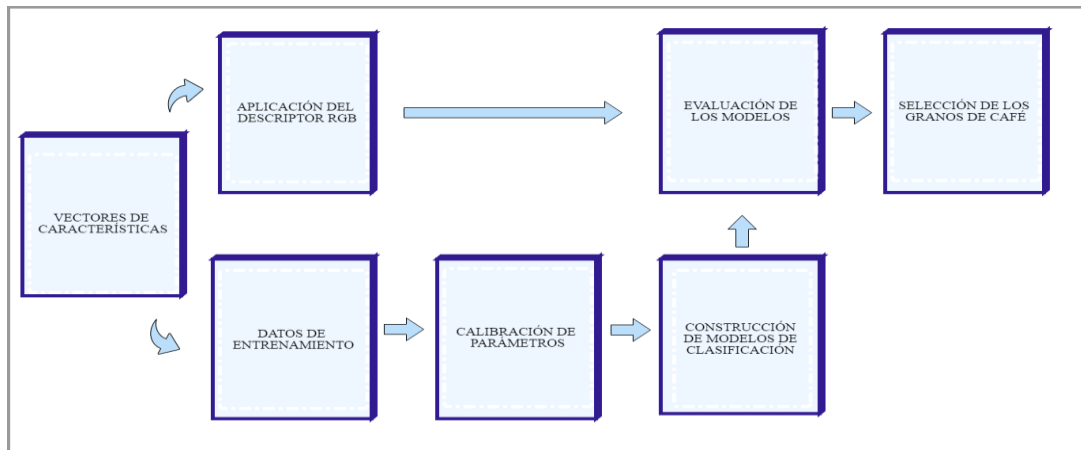


Figura 20. Clasificación automática de regiones sanas e infectadas por broca.

Fuente. Elaboración propia.

La Figura 20 Clasificación automática de regiones sanas e infectadas por broca nos muestra los pasos que después de haber realizado la obtención de los vectores de características se realiza los datos de entrenamiento con el algoritmo y la calibración de los parámetros y luego se construye el modelo de clasificación de SVM la cual se realiza una evaluación de los modelos que fueron realizados y se realiza la selección de la selección de cafés para continuar con el proceso en donde a continuación de muestra una imágenes de las fotos de prueba.

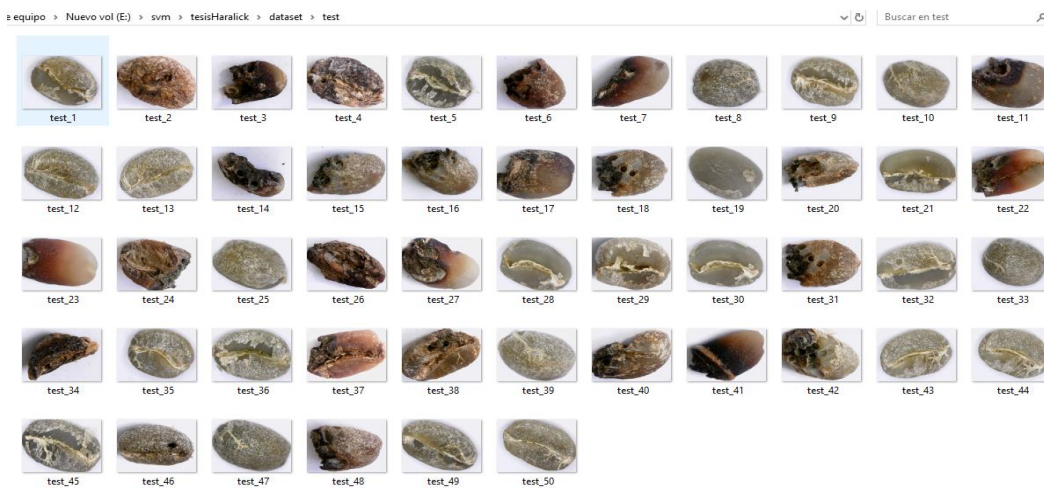


Figura 21. Imágenes de prueba.

Fuente: Elaboración propia.

En la Figura 21 Imágenes de prueba, se muestra las fotos que fueron tomadas para realizar las pruebas de los algoritmos para que de esta manera se obtengan unos resultados más

óptimas ya que las imágenes de pruebas serán muy aparte de las base de datos de entrenamiento que se tiene.

Este proceso a continuación nos dará a conocer según el algoritmo de haralick la clasificación de las imágenes de los granos de café las cuales se tienen que aplicar los métodos que son segmentaciones en el grano de café y comparación de color.

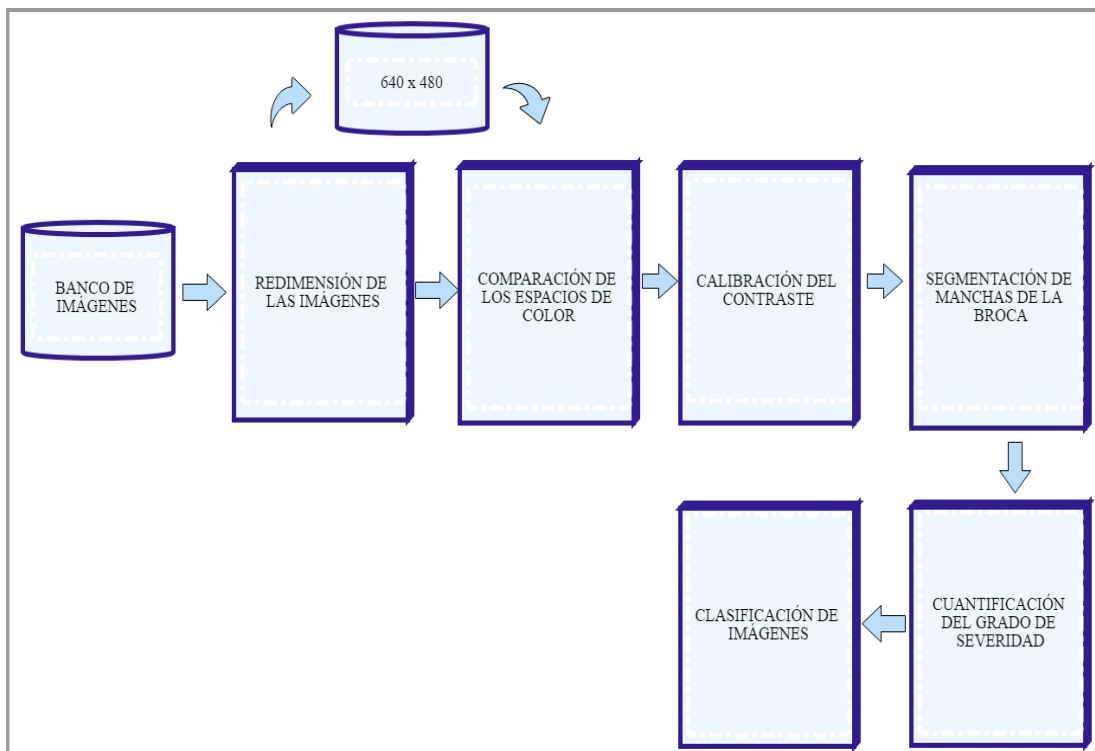


Figura 22. Determinación automática de la clasificación.

Fuente: Elaboración propia.

En la Figura 22, la clasificación muestra el proceso continuo de la anterior figura la cual se decide la utilización svm la cual aquí toma el banco de imágenes tomadas con las cámara microscópica la cual cuentan con 3000 imágenes por muestra luego de esto de realiza el cambio que son los espacios de color y luego la segmentación de la broca, para luego obtener la precisión de los resultados según el pixel de 480 por 640.

3.6.3 Muestra del proceso de segmentación de las manchas de broca en el grano de café

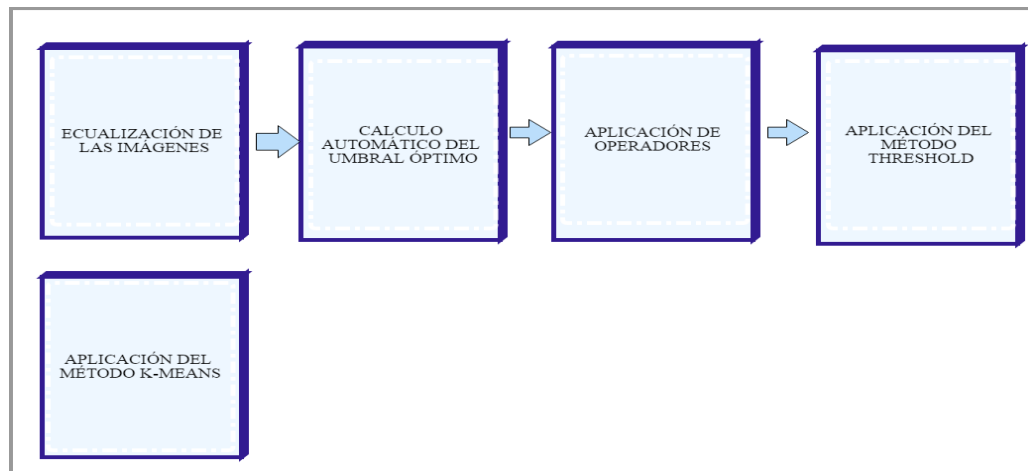


Figura 23. Segmentación de manchas de broca.
Fuente: Elaboración propia.

En Figura 23. Segmentación de manchas de broca se muestra después de haber realizado la ecualización de imágenes a todo el banco que se tenía se realizó un cálculo del umbral óptimo que se tiene para luego aplicar los operadores morfológicos y también la aplicación de threshold y como también se tiene la aplicación del método k-means.

3.7 Caracterización de regiones sanas e infectadas

3.7.1 Sobre el conjunto de imágenes a utilizarse

Las imágenes fueron tomadas en un ambiente no controlada se tomó con una cámara microscópica que son de 2MP (mega-pixeles) con una ampliación de 50x a 500x, la cual se consiguió tomar 3000 fotografías de cada muestra con un formato jpg con un pixel de 480 x 640.



Figura 24. Fotografía tomada de café bueno
Fuente: Elaboración propia

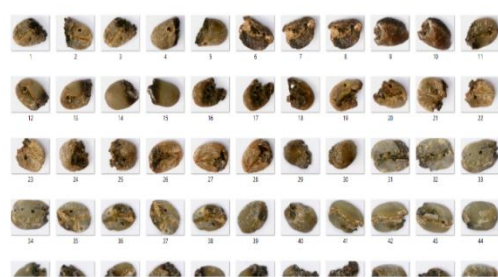


Figura 25. Fotografía tomada de café con broca
Fuente: Elaboración propia

3.7.2 Sobre la resolución y estandarización de las imágenes

Para este aspecto para la fase experimental de pruebas experimentales se realizó redimensionamiento de imágenes las cuales constaron de 2 grupos que son de 480 pixeles la cual cada uno está compuesto por 3000 imágenes de granos de imágenes.

3.7.3 Sobre la resolución y estandarización de las imágenes

Del conjunto de imágenes que se tomaron se separaron en dos grupos las cuales fueron llamadas café bueno y café con broca esto se realizó con la finalidad para que en el proceso de clasificación por esto bueno contar con una cantidad buena de tomas de imágenes para un buen entrenamiento y su posterior clasificación. La cuales las imágenes cuantas con una resolución de 96 x 96 las cuales son por cada grupo que son café bueno y café con broca.

Tabla 3.

Numero de muestras tomadas para las resoluciones.

RESOLUCION	BUENO	BROCA	TOTAL
96 X 96	3000	3000	6000

Fuente: Elaboración propia.

3.7.4 Implementación del descriptor de textura de Haralick

En esta implementación se permite cuantificar la frecuencia con el cual de diferentes combinaciones de niveles de gris ocurren en las imágenes, las cuales las imágenes fueron procesadas teniendo un nivel de 640 en escala gris que esta se realiza de 0 a 639 la cual la matriz es de la siguiente forma.

- Matriz $GLCMfreq[x, y]_{480 \times 640}$ En esta matriz se almacena el número de ocurrencias de cada pixel con un pixel vecino que esta la cual se detalla de la siguiente manera.

	Nivel 0	Nivel 1	...	Nivel 639
Nivel 0	$F_{\theta}^{\delta}(0,0)$	$F_{\theta}^{\delta}(0,1)$		$F_{\theta}^{\delta}(0,639)$
Nivel 1		$F_{\theta}^{\delta}(1,1)$		
.....				
Nivel 479				$F_{\theta}^{\delta}(639,479)$

En esta tabla nos indica que se tiene $F_{\theta}^{\delta}(0,0)$ que esto nos quiere decir que F es frecuencia de pixeles de las x, y con una dirección θ con una separación de δ . este método recibe como parámetros de la dirección en θ y la separación entre los pixeles δ , las cuales toma solo las direcciones que son $\theta=45^{\circ}$, $\theta = 90$ y $\theta 135$.

- Matriz GLCM $_{pij}[x,y]_{480 \times 640}$ en esta matriz se define para el almacenamiento de la frecuencia relativa o probabilidad de coocurrencia desde un pixel de referencia que sea un pixel vecino.

	Nivel 0	Nivel 1	...	Nivel 639
Nivel 0	$P_{\theta}^{\delta}(0,0)$	$P_{\theta}^{\delta}(0,1)$		$P_{\theta}^{\delta}(0,639)$
Nivel 1		$P_{\theta}^{\delta}(1,1)$		
.....				
Nivel 479				$P_{\theta}^{\delta}(639,479)$

Esta tabla se define como $P_{\theta}^{\delta}(x,y)=F_{\theta}^{\delta}(x,y)/S$ esta es igual a la frecuencia relativa o probabilidad de ocurrencia de pixeles de x, y donde tienen una dirección θ con una separación de δ ; en donde la S es la suma total de ocurrencia. Las características extraídas por cada matriz generada fueron las siguientes:

- Contraste: donde: $|i - j| = n$

$$f_1 = \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) \right\}$$

- Desimilaridad: donde $|i - j| = n$

$$f_2 = \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) \right\}$$

- Homogeneidad:

$$f_3 = \sum_i \sum_j \frac{1}{1 + (i - j)^2} p(i,j)$$

- Segundo momento angular:

$$f_4 = \sum_i \sum_j \{p(i,j)\}^2$$

- Energía:

$$f_5 = \sqrt{\sum_i \sum_j \{p(i,j)\}^2}$$

- Máxima probabilidad:

$$f_6 = \max_prob(p(i,j))$$

- Entropía:

$$f_7 = - \sum_i \sum_j p(i,j) \log(p(i,j))$$

- Suma de promedio:

$$f_8 = \sum_{i=2}^{2N_g} ip_{x+y}(i)$$

- Suma de cuadrados: varianza

$$f_9 = \sum_i \sum_j (i - u)^2 p(i, j)$$

- Desviación estándar

$$f_{10} = \sqrt{\sum_i \sum_j (i - u)^2 p(i, j)}$$

- Correlación:

$$f_{11} = \frac{\sum_i \sum_j (ij)p(i, j) - u_x u_y}{\sigma_x \sigma_y}$$

3.7.5 Generación de los vectores de características

3.7.5.1 En esta generación se consideraron dos propiedades las cuales son:

- El espacio de color sobre lo que extraerían las características de las imágenes las cuales son el RGB la cual se de determinar dos espacios de color las cuales se obtienen mejor para ver la región que tiene broca en el café.
- La distancia entre el pixel de diferencia y pixeles vecinos para calcular la matriz de co-ocurrencia en la cual se tomó una decisión de disminuir el número de distancias finalmente lo que se obtiene son cuatro vectores

3.7.6 Extracción de características mediante el uso de histogramas RGB

Para la realización de histogramas es que las imágenes sean convertidas en escalas grises, esto proporciona una información acerca del origen de los pixeles que resulta desde el proceso de ecualización del histograma básicamente es la redistribución de los niveles de gris de la imagen por la reasignación de los valores de brillo de los píxeles, sin embargo el ruido surge como un problema para la ecualización, debido a que este aumenta el contraste

y distorsiona los distintos objetos que forman la imagen La suma de todos los componentes de un histograma normalizado es igual a 1 y la dispersión del histograma resulta en un mayor rango dentro del intervalo $[0, L - 1]$. Donde L es el número total de posibles niveles de gris en la imagen (Viera, 2017). Tal como se muestra en la siguiente línea de código.

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('test_2.jpg')
cv2.imshow('test_2.jpg', img)

color = ('b','g','r')

for i, c in enumerate(color):
    hist = cv2.calcHist([img], [i], None, [256], [0, 256])
    plt.plot(hist, color = c)
    plt.xlim([0,256])

plt.show()

cv2.destroyAllWindows()
```

- Resultado de histograma de RGB.

Extracción de histogramas según RGB: para esto se instaló opencv y matplotlib se importó posteriormente se pone la ruta de imagen se convierte en escala gris y se muestra en rgb la imagen en histogramas y el rango está siendo de 0 a 256.

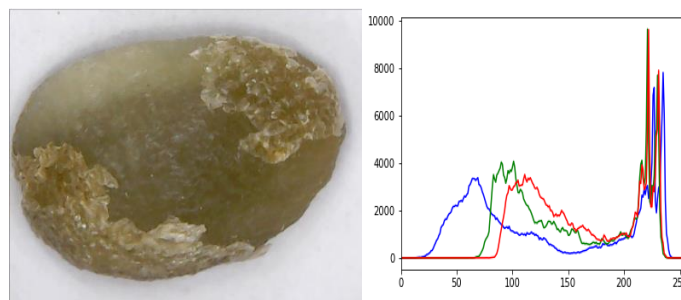


Figura 26. Histograma del café bueno.

Fuente: Elaboración propia.

En la Figura 26 nos muestra el histograma en RGB del café bueno en el histograma según la imagen se observa que el café bueno está el rango de 4000.

- El eje x esta los valores de la luminosidad de que se es claro u oscuro que desde el 0 al 256 el cero representa el negro absoluto al 255 que es el blanco absoluto.
- El eje y son los porcentajes donde nos muestra los valores repetidos de forma relativa y la cantidad de pixeles.

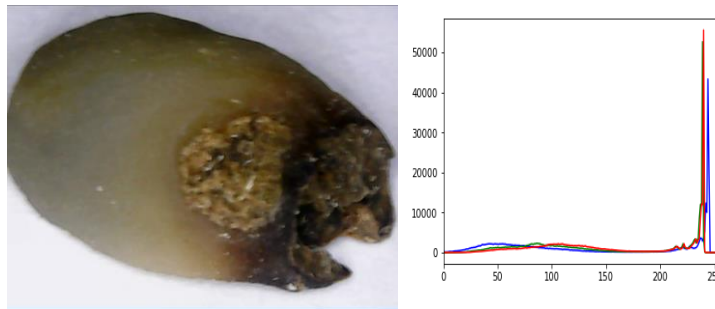


Figura 27. Histogramas en RGB del café con broca.
Fuente: Elaboración propia.

La Figura 27 nos muestra el histograma en RGB del café broca en el histograma según la imagen se observa que el café con broca alcanza hasta el rango de 0 inicia hasta un 2500 inicia desde cero y es casi el mismo nivel por el café tiene mucho la parte oscura o sea la broca es oscura es por ese motivo que se mantiene casi en el mismo nivel y hasta llegar en un nivel 50000 en pixeles de la imagen.

3.7.7 Implementación del operador LBP

Esta fase se implementó el operador LBP la cual nos permite tomar en cuenta variaciones locales de color para las imágenes, la cual se utilizaron tres estructuras; la primera fase es una matriz la cual consiste en mascara de pesos que son del 1 al 128; la segunda fase la matriz es donde se almacenan los valores que se obtienen a partir de la umbralizacion y la tercera matriz almacena los valores finales obtenidos de la codificación de LBP.

- Weightmask de una matriz de 3 x 3.

1	2	4
8		16
32	64	128

- IbpMatriz de 3 x 3 esta matriz es la que la almacena los valores obtenidos luego de la umbralización.

0	1	0
0		1
0	0	0

- ImageLbp de una m x m la cual esta matriz almacena la intensidad de los pixeles luego aplica la matriz sobre la imagen.

	P1	P2	P3	
	P4	P5	P6	
	P7	P8	P9	

- En esta imagen que a continuación se muestra de que manera realiza el pixel matemáticamente en el procesamiento de imágenes.

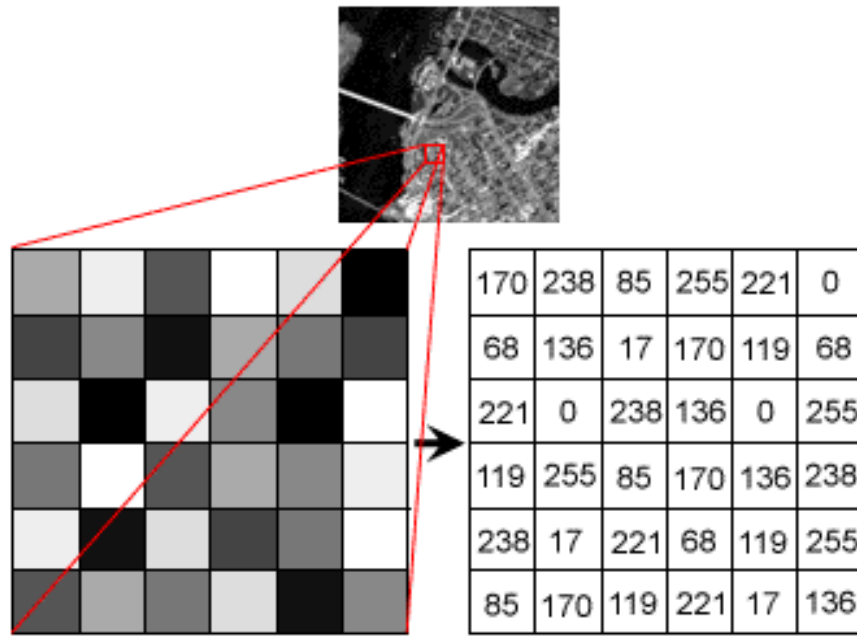


Figura 28. Matriz de pixeles.

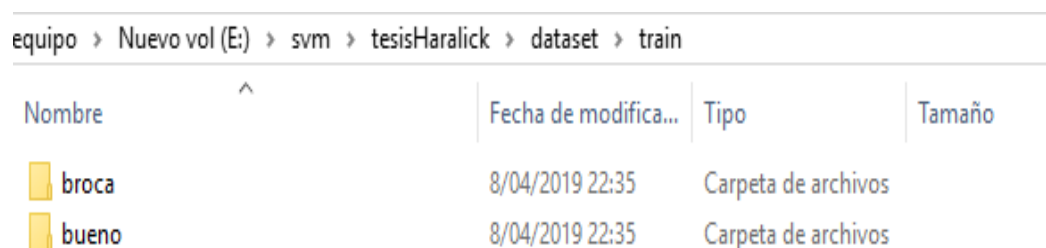
Fuente: Blog de divulgación científica de la Asociación Con/Ciencia (Yanguas, 2017).

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1 Resultado N° 1

En esta parte se tomó fotografía en un lugar no controlado a los granos de café en estado trillado en sus dos condiciones (bueno y broca) con una cámara microscópica de 2Mp y se logró formar el banco de imágenes.



Nombre	Fecha de modifica...	Tipo	Tamaño
broca	8/04/2019 22:35	Carpeta de archivos	
bueno	8/04/2019 22:35	Carpeta de archivos	

Figura 29. Banco de imágenes tomada.

Fuente: Elaboración propia.

Según la Figura 29, los labels que se muestran en sus dos estados de granos de café (bueno y broca) las cuales cuentan en cada carpeta 3000 imágenes con pixeles de 480 x 640, al formar esos labels de imágenes fueron muy eficientes en el entrenamiento y un aprendizaje del algoritmo al 96 % óptimo por cada grupo de imágenes.

4.2 Resultado N° 2

Se realizó que los algoritmos validen la toma de fotos, se entrenó con diferentes parámetros mejorando el modelo para no tener errores posteriores. Se probó con imágenes de pixeles de 480 x 640, en las cuales se realizó los entrenamientos del algoritmo CNN, para ello se necesitaron de redes neuronales convulsiónales las cual importan las librerías y paquetes de keras, luego se preparó los datos de entrenamiento y pruebas, en seguida se inició con el cálculo y al final se realizó las predicciones tal como se muestra en la siguiente tabla.

Tabla 4.
Resultados de errores según cada modelo.

Modelo	Datos de entrenamiento	Datos de prueba	Resultados óptimos	Error
SVM	6000	50	48	2
CNN 50 pasos	6000	50	45	5
CNN 100 pasos	6000	50	45	5

Fuente: Elaboración Propia.

Según la Tabla 4 se obtuvo los resultados según los datos de prueba el primer resultado según el modelo de SVM de las 50 fotos de prueba hubo 48 fotos optimas y 2 fotos de error la cual 2 fotos equivale al 4% de error de la data que hubo esto fue con el modelo de SVM y la cual también se tuvo resultados con el CNN con 50 pasos de validación se realizó con prueba de 50 imágenes y tuvo 45 imágenes con un resultado óptimo y 5 imágenes de error y este error equivale a un 10% de la data y también el CNN con 100 pasos las cuales es el mismo margen de error que los de 50 pasos. La diferencia con el modelo de SVM es de un 6 % según se detalla en la tabla.

4.3 Resultado N° 3.

En esta fase se logró realizar de acuerdo a los labels de las imágenes la automatización del algoritmo la cual está con un error de 04 % en los resultados ya con imágenes que no están dentro de los bancos de imágenes entrenadas.

Este algoritmo importa el cv2 y numpy que son librerías de Opencv. Después de haber realizado la importación se procede con la función de extraer texturas haralick de las imágenes, luego calcular la textura de los 4 adyacentes, es decir de los 4 lados, y empieza a cargar los 3000 datos por cada muestra y a entrenar el algoritmo. Internamente convierte la imagen en escala gris, agrega al vector características y etiquetas, crea un clasificador, evalúa

el modelo, predice la etiqueta, y al final muestra la imagen de salida es decir las imágenes clasificadas.

```
cmd: C:\Windows\system32\cmd.exe - python train_test. cmd: C:\Windows\system32\cmd.exe - python train_t
Processing Image - 2183 in broca Processing Image - 2455 in bueno
Processing Image - 2184 in broca Processing Image - 2456 in bueno
Processing Image - 2185 in broca Processing Image - 2457 in bueno
Processing Image - 2186 in broca Processing Image - 2458 in bueno
Processing Image - 2187 in broca Processing Image - 2459 in bueno
Processing Image - 2188 in broca Processing Image - 2460 in bueno
Processing Image - 2189 in broca Processing Image - 2461 in bueno
Processing Image - 2190 in broca Processing Image - 2462 in bueno
Processing Image - 2191 in broca Processing Image - 2463 in bueno
Processing Image - 2192 in broca Processing Image - 2464 in bueno
Processing Image - 2193 in broca Processing Image - 2465 in bueno
Processing Image - 2194 in broca Processing Image - 2466 in bueno
Processing Image - 2195 in broca Processing Image - 2467 in bueno
Processing Image - 2196 in broca Processing Image - 2468 in bueno
Processing Image - 2197 in broca Processing Image - 2469 in bueno
Processing Image - 2198 in broca Processing Image - 2470 in bueno
Processing Image - 2199 in broca Processing Image - 2471 in bueno
Processing Image - 2200 in broca Processing Image - 2472 in bueno
Processing Image - 2201 in broca Processing Image - 2473 in bueno
Processing Image - 2202 in broca Processing Image - 2474 in bueno
```

Figura 30. Muestra de entrenamiento del algoritmo.

Fuente: Elaboración Propia.

En la Figura 30, muestra de entrenamiento del algoritmo es donde empieza a procesar todas las imágenes para su aprendizaje automático y de esa manera aprende el algoritmo para que a continuación ya clasifique o reconozca las imágenes de prueba que son 50 imágenes que son muy aparte de las imágenes de entrenamiento. El entrenamiento tarda 11 minutos en el entrenamiento de las 6000 imágenes.

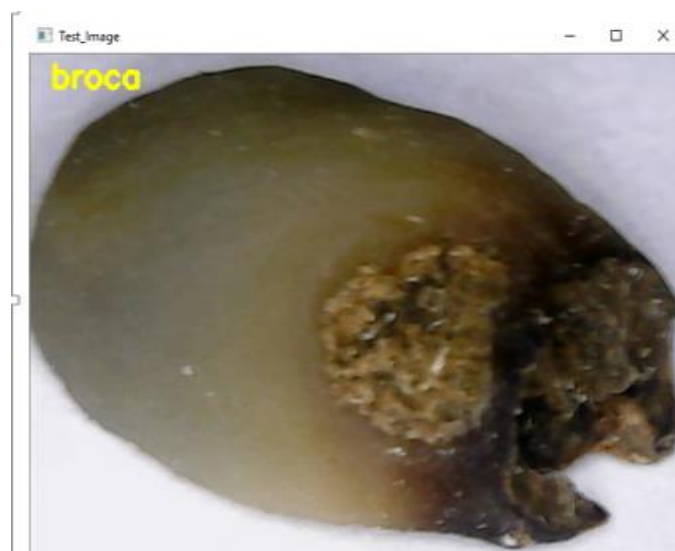


Figura 31. Resultado de la clasificación de café broca.

Fuente: Elaboración Propia.

En la Figura 31 en donde se tiene ya la clasificación por imagen es donde nos muestra el resultado con el nombre que es broca esta fue realizado según el modelo SVM la cual es un resultado óptimo esta se realizó con 50 imágenes de muestra la cual tuvo 4 % margen de error.

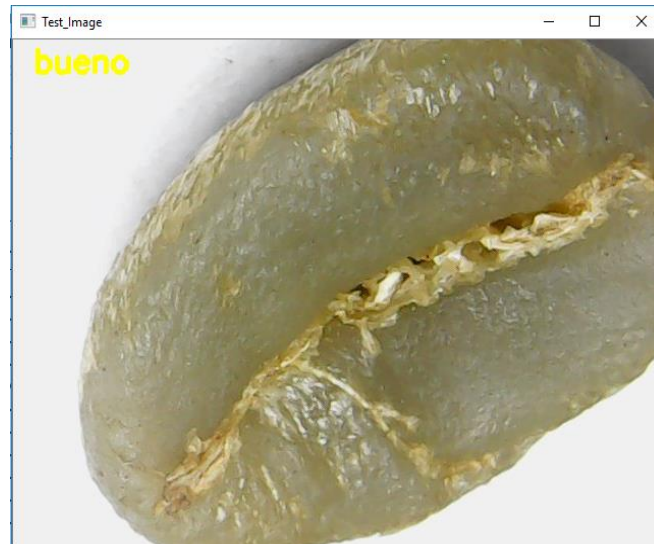


Figura 32. Resultado de clasificación de café bueno.

Fuente: Elaboración propia.

En la Figura 32 nos muestra el resultado de con un nombre que es grano de café bueno las cual es el resultado de las 50 imágenes de prueba la cual fue optimo el resultado con un 0% de error en imágenes de granos de café bueno.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

En la siguiente investigación se logró diagnosticar automáticamente con 4 % y 10% de error de la clasificación de café trillado con broca mediante procesamiento de imágenes con Deep Learning y cumpliendo con el objetivo trazado según se comprueba.

El modelo SVM clasificó las imágenes de granos café en sus dos estados, se obtuvo resultados óptimos, porque el entrenamiento se realizó con 3000 imágenes por cada estado del grano y con una muestra de 50 imágenes que es aparte de las imágenes de entrenamiento las cuales al tener los resultados fue del 96% de la data correcta de los resultados.

El modelo de CNN obtuvo resultados con un margen de error un poco más elevado que el modelo SVM, la diferencia del error fue el 6% con los dos procedimientos que se realizó de 50 y 100 pasos las cuales se realizó la prueba con las 50 imágenes que se tomó fueron las imágenes de prueba muy aparte de las imágenes de entrenamiento y se obtuvo un 90 % de datos correctos de los resultados.

5.2 Recomendaciones

- Se recomienda continuar con el proceso del algoritmo hasta una implementación con una maquina procesadora y sea muy útil para los agricultores de café.
- Se recomienda implementar a un aplicativo móvil y una página web con el algoritmo de clasificación para que sea más útil ya que todas las personas.
- Se recomienda el uso del algoritmo de haralick y LBP con el modelo SVM ya que es muy óptimo para realizar clasificaciones y el algoritmo de redes convulsionales.

- Se recomienda a futuro continuar con el algoritmo e implementar más herramientas para que pueda reconocer los granos en cantidad para de esa manera facilitar el proceso.
- Se recomienda utilizar el algoritmo por que disminuirá la pérdida de economía y generaría un buen ingreso económico para los agricultores y la cooperativa. También mejorará en las cooperativas en la parte administrativa y no habrá una demora en la selección con el personal porque todo será automatizado.

REFERENCIAS BIBLIOGRÁFICAS

Acosta, R. H. (n.d.). La Broca del Fruto del Café.

Agroecotur. (2016). Revista Agroecotur: PLAGAS DEL CAFÉ EN NUESTRA REGIÓN. Retrieved March 29, 2019, from <http://revistaagroecotur.blogspot.com/2016/12/plagas-del-cafe-en-nuestra-region.html>

Alegre, E., Pajares, G., & Junio, E. (2016). *No Title*.

ana loyo. (n.d.). Visión Computacional | SG Buzz. Retrieved April 4, 2019, from <https://sg.com.mx/revista/55/visi-n-computacional>

Arias y Sierra. (2016). Procesamiento de imágenes para la clasificación de café verde.

Barriga y Arrasco. (2018). *Pontificia Universidad Católica del Perú Escuela de Posgrado*. Pontificia Universidad Católica del peru.

Briega, R. E. L. (2015). Raul E Varela.

Bustillo, A. E. (2015). La broca del café. Plagas cuarentenarias. Perspectivas. Glosario.

Camilo. (2014). *PROYECTO Impacto económico de la broca del café (Hypothenemushampeii) en la caficultura de Jarabacoa*.

Carvajal, J. (2004). Comparison of Models of Automatic Classification of Textural Patterns of Mineral Presents in Colombian Coals. *Institución Universitaria de Envigado*, 72(146), 115–124.

Chacon Murguía, M. I., Sandoval Rodriguez R., Vega Pineda J. (2015). *PERCEPCIÓN VISUAL - Aplicada a la robótica - CHACÓN MURGUÍA , Mario I. - Google Libros*. Retrieved from <https://books.google.com.pe/books?id=pCHaDAAAQBAJ&pg=SA1-PA84&dq=segmentacion+basada+en+regiones&hl=es&sa=X&ved=0ahUKEwjUi5y4gKnhAhVOT98KHWvED-QQ6AEIPDAE#v=onepage&q=segmentacion+basada+en+regiones&f=false>

- Comstor. (2017). vision COMPUTACIONAL.
- Cuenca, J. S. (2008). *Facultat de Matemàtiques*.
- Diego Calvo. (2017). Red Neuronal Convolutacional CNN - Diego Calvo. Retrieved April 7, 2019, from <http://www.diegocalvo.es/red-neuronal-convolutacional/>
- Goodfellow, Bengio&, & Courville. (2016). Book Review: Deep Learning. *Deep Learning*, 22(4), 351–354. <https://doi.org/10.4258/hir.2016.22.4.351>
- Guerrero. (2016). librerias. Retrieved from <https://medium.com/@ricardo.guerrero/frameworks-de-deep-learning-un-repaso-antes-de-acabar-el-2016-5b9bf5b9f9af>
- Hackeling, G. (2014). *Mastering machine learning with scikit-learn: apply effective learning algorithms to real-world problems using scikit-learn*. Retrieved from <http://books.google.com/books?id=fZQeBQAAQBAJ&pgis=1>
- Hamza Bendemra. (2018). Construye tu primer clasificador de Deep Learning con TensorFlow: Ejemplo de razas de perros. Retrieved April 7, 2019, from <https://medium.com/datos-y-ciencia/construye-tu-primer-clasificador-de-deep-learning-con-tensorflow-ejemplo-de-razas-de-perros-ed218bb4df89>
- INE. (2014). 12. Agrario. <https://doi.org/10.1007/s13398-014-0173-7.2>
- INEI. (2017). url. Retrieved from <https://gestion.pe/economia/inei-produccion-cafe-aumento-13-6-junio-2017-142288>
- Jason Brownlee. (2016). ¿Qué es una matriz de confusión en el aprendizaje automático? Retrieved March 31, 2019, from <https://machinelearningmastery.com/confusion-matrix-machine-learning/>
- Lecun, B. &Geoffrey. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- NAVARRETE, P. F. H. (2018). PROCESAMIENTO DE IMÁGENES DIGITALES.

- ojala. (1996). capitulo 5. Retrieved from
<http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/230/A8.pdf?sequence=10>
- opencv. (2018). opencv. Retrieved from <https://opencv.org/about.html>
- python. (1996). aprender python.
- Raizel. (2014). segmentaciones.
- Ramirez. (2017). *RECONOCIMIENTO DE GRANOS DE CAFE VERDE ARABIGA SIN DEFECTOS FISICOS EN MUESTRAS USANDO VISI ON ARTIFICIAL Tesis*. UNIVERSIDAD NACIONAL DE SAN AGUST'IN DE AREQUIPA.
- Recio, J.A.; Fdez. Sarría, A.; Ruiz, L.A.; Hermosilla, T. (2008). Actualización de bases de datos de usos agrícolas mediante la integración de cartografía e imágenes de alta resolución.
- Rios, G. (2018). *APLICACIÓN DE TÉCNICAS DE APRENDIZAJE PROFUNDO PARA LA CLASIFICACIÓN Y RECONOCIMIENTO DE OBJETOS EN IMÁGENES*. universidad santo tomas.
- Schönberger, Nunez, Boulogne, Warner, Yager, Gouillart, Y. (2014). skit.
- Shreedhar Vellayaraj. (2017). Anaconda, Jupyter, Spyder - Cosas que necesitas saber - Shreedhar vellayaraj - Medio. Retrieved May 26, 2019, from <https://medium.com/@Shreedharvellay/anaconda-jupyter-spyder-things-you-need-to-know-3c808d824739>
- Solangel Rodríguez Vázquez, Andy Vidal Martínez Borges, J. V. L. G. (2016). Revista cubana de ciencias informáticas. *Revista Cubana de Ciencias Informáticas*, 10(1), 95–109. Retrieved from http://scielo.sld.cu/scielo.php?pid=S2227-18992016000100007&script=sci_arttext&tlng=en
- Suárez, E. J. C. (2014). Máquinas de Vectores Soporte (SVM). *Dpto. de Inteligencia Artificial, ETS de Ingeniería Inforática, Universidad Nacional de Educación a*

Distancia (UNED), 1–25. Retrieved from

[http://www.ia.uned.es/~ejcarmona/publicaciones/\[2013-Carmona\] SVM.pdf](http://www.ia.uned.es/~ejcarmona/publicaciones/[2013-Carmona] SVM.pdf)

Torres, D. P. L. (2018). *UNIVERSIDAD CENTRAL DEL ECUADOR “ SEDE - GALÁPAGOS ” CARRERA : CIENCIAS BIOLÓGICAS Y AMBIENTALES .*

Trevino, A. (2016). andrea trevino. Retrieved from <https://www.datascience.com/blog/k-means-clustering>

Varoquaux, G., Raamanac, P. R., Engemann, D. A., Hoyos-Idrobo, A., Schwartz, Y., & Bertrand Thirion. (2017). NeuroImage Assessing and tuning brain decoders : Cross-validation , caveats , and guidelines. *NeuroImage*, 145(October 2016), 166–179. <https://doi.org/10.1016/j.neuroimage.2016.10.038>

Viera. (2017). *CLASIFICACIÓN DEL CACAO Gabriela Viera-Maza.*

ANEXOS

Anexo A. Resultado obtenidos según SVN Y CNN

100 pasos por Época y 50 pasos de validación

	CNN	SVN	CNN
test_1	bueno	bueno	bueno
test_2	broca	bueno	broca
test_3	broca	broca	broca
test_4	broca	broca	broca
test_5	broca	bueno	broca
test_6	broca	broca	broca
test_7	broca	broca	broca
test_8	bueno	bueno	bueno
test_9	bueno	bueno	bueno
test_10	bueno	bueno	bueno
test_11	broca	broca	broca
test_12	bueno	bueno	bueno
test_13	bueno	bueno	bueno
test_14	broca	broca	broca
test_15	broca	broca	broca
test_16	broca	broca	broca
test_17	broca	broca	broca
test_18	broca	broca	broca
test_19	bueno	bueno	bueno
test_20	broca	broca	broca
test_21	bueno	bueno	bueno
test_22	bueno	broca	bueno
test_23	bueno	broca	bueno
test_24	broca	bueno	broca
test_25	bueno	bueno	bueno
test_26	broca	broca	broca
test_27	broca	broca	broca
test_28	bueno	bueno	bueno
test_29	bueno	bueno	bueno
test_30	bueno	bueno	bueno
test_31	broca	broca	broca
test_32	bueno	bueno	bueno
test_33	bueno	bueno	bueno
test_34	broca	broca	broca
test_35	broca	bueno	broca
test_36	bueno	bueno	bueno
test_37	broca	broca	broca
test_38	broca	broca	broca

test_39	bueno	bueno	bueno
test_40	broca	broca	broca
test_41	broca	broca	broca
test_42	broca	broca	broca
test_43	bueno	bueno	bueno
test_44	bueno	bueno	bueno
test_45	broca	bueno	broca
test_46	broca	broca	broca
test_47	bueno	bueno	bueno
test_48	broca	broca	broca
test_49	bueno	bueno	bueno
test_50	bueno	bueno	bueno

Anexo B. CNN redes convulcionales

```
import os

img_folder = 'images'

from IPython.display import Image

from __future__ import print_function, division
import numpy as np
import random
import os
import glob
import cv2
import datetime
import pandas as pd
import time
import h5py
import csv

from scipy.misc import imresize, imsave

from sklearn.cross_validation import KFold, train_test_split
from sklearn.metrics import log_loss, confusion_matrix
from sklearn.utils import shuffle

from PIL import Image, ImageChops, ImageOps

import matplotlib.pyplot as plt

from keras import backend as K
from keras.callbacks import EarlyStopping, Callback
from keras.utils import np_utils
from keras.preprocessing.image import ImageDataGenerator, load_img, img_to_array

from keras import optimizers
from keras.models import Sequential, model_from_json
from keras.layers import Convolution2D, MaxPooling2D, ZeroPadding2D, Activation, Dropout, Flatten, Dense

from keras.models import Sequential
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.layers import Dense
```

```

# Initialising the CNN
classifier = Sequential()

classifier.add(Conv2D(32, (3, 3), input_shape=(64, 64, 3), activation="relu"))

classifier.add(MaxPooling2D(pool_size = (2, 2)))

classifier.add(Conv2D(32, (3, 3), activation = 'relu'))
classifier.add(MaxPooling2D(pool_size = (2, 2)))

classifier.add(Flatten())

classifier.add(Dense(activation="relu", units=128))
classifier.add(Dense(activation="sigmoid", units=1))

classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])

from keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True)
test_datagen = ImageDataGenerator(rescale = 1./255)

training_data = train_datagen.flow_from_directory('./images/train',
                                                  target_size = (64, 64),
                                                  batch_size = 32,
                                                  class_mode = 'binary')

test_data = test_datagen.flow_from_directory('./images/test',
                                             target_size = (64, 64),
                                             batch_size = 32,
                                             class_mode = 'binary')

classifier.fit_generator(training_data,
                        steps_per_epoch = (50),
                        epochs = 1,
                        validation_data = test_data,
                        validation_steps = 30)

```

```
from IPython.display import Image
Image("./newimages/test4.jpg")

import numpy as np
from keras.preprocessing import image
test_image = image.load_img('./newimages/test4.jpg', target_size = (64, 64))
test_image = image.img_to_array(test_image)
test_image = np.expand_dims(test_image, axis = 0)
result = classifier.predict(test_image)
# training_set.class_indices
if result[0][0] == 1:
    prediction = 'bueno'
else:
    prediction = 'broca'

print(result)
print(prediction)
```