

DISEÑO DE UN SISTEMA DE VISIÓN ARTIFICIAL PARA EL ANÁLISIS DE
CALIDAD Y PRODUCCIÓN DE ROSAS

Trabajo de Grado

Presentado por:

Omar Alfonso Muñoz Amaya

Director:

M. Sc. Luis Jorge Herrera Fernández

UNIVERSIDAD PEDAGÓGICA NACIONAL
FACULTAD CIENCIA Y TECNOLOGÍA
LICENCIATURA EN ELECTRÓNICA
BOGOTÁ D.C

2019

DISEÑO DE UN SISTEMA DE VISIÓN ARTIFICIAL PARA EL ANÁLISIS DE
CALIDAD Y PRODUCCIÓN DE ROSAS

Trabajo de Grado


Presentado por:

Omar Alfonso Muñoz Amaya

Director:

M. Sc. Luis Jorge Herrera Fernández

UNIVERSIDAD PEDAGÓGICA NACIONAL
FACULTAD CIENCIA Y TECNOLOGÍA
LICENCIATURA EN ELECTRÓNICA
BOGOTÁ D.C 2019

 UNIVERSIDAD PEDAGÓGICA NACIONAL <small>República de Colombia</small>	FORMATO	
	RESUMEN ANALÍTICO EN EDUCACIÓN - RAE	
Código: FOR020GIB	Versión: 01	
Fecha de Aprobación: 10-10-2012	Página 2 de 118	

1. Información General	
Tipo de documento	Trabajo de Grado
Acceso al documento	Universidad Pedagógica Nacional. Biblioteca Central
Título del documento	Diseño de un sistema de visión artificial para el análisis de calidad y producción de rosas.
Autor(es)	Muñoz Amaya, Omar Alfonso
Director	Herrera Fernández, Luis Jorge
Publicación	Bogotá, Universidad Pedagógica Nacional, 2019. 118p.
Unidad Patrocinante	Universidad Pedagógica Nacional.
Palabras Claves	APRENDIZAJE DE MÁQUINA; FLORICULTURA; PROCESAMIENTO DE IMÁGENES; SISTEMA; VISIÓN ARTIFICIAL.

2. Descripción
<p>El presente trabajo de grado se desarrolló con el fin de explorar técnicas de visión artificial aplicadas a los procesos de producción agrícola del país como lo es la industria floricultora, entendiendo que, en las temporadas de gran producción los operarios contratados no tienen suficiente experiencia para realizar una correcta evaluación de la calidad de la rosa, entrenarlos y capacitar dicho personal toma demasiado tiempo. Se desarrolla entonces el diseño de un sistema entrenado con técnicas de visión artificial que sea capaz de realizar esta labor y que cumpla con los criterios de calidad acotados. Por</p>

lo que al final del desarrollo se obtiene una aplicación a modo de interfaz de usuario que permite ver análisis de la calidad de las fotografías de rosas ingresadas al mismo.

3. Fuentes

- Bayer CropScience. (s.f.). *Mildeo veloso. El Mildeo Velloso acecha su cultivo Protéjalo con el mejor portafolio*. [Folleto]. Autor.
- Cárdenas, M., & Rodríguez, M. (2011). *Colombia y la creación de una empresa productora de flores*. Universidad de La Sabana, 166.
- Csurka, G., C. R. Dance, L. Fan, J. Willamowski, and C. Bray. *Visual Categorization with Bags of Keypoints*. Workshop on Statistical Learning in Computer Vision. 2004, ECCV 1 (1–22), 1–2.
- Dharwadkar, S., Bhat, G., Reddy, N. V. S., & Aithal, P. K. (2017). Floriculture classification using simple neural network and deep learning. *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, 619–622. <https://doi.org/10.1109/RTEICT.2017.8256671>
- Do, T. B., Nguyen, H. H., Nguyen, T. T. N., Vu, H., Tran, T. T. H., & Le, T. L. (2017). Plant identification using score-based fusion of multi-organ images. *Proceedings - 2017 9th International Conference on Knowledge and Systems Engineering, KSE 2017, 2017-Janua*, 191–196. <https://doi.org/10.1109/KSE.2017.8119457>
- García Santillán, I. D. (2008). *Visión artificial y procesamiento digital de imágenes usando Matlab*. Quito, Ecuador: Cámara Ecuatoriana del Libro - Núcleo de Pichincha
- Gavai, N. R., Jakhade, Y. A., Tribhuvan, S. A., & Bhattad, R. (2018). MobileNets for flower classification using TensorFlow. *2017 International Conference on Big Data, IoT and Data Science, BID 2017, 2018-Janua*, 154–158. <https://doi.org/10.1109/BID.2017.8336590>
- Ghosh, S., Kumar, H., Kumar, P.V., & Nayak, J.S. (2015). *Study on Classification of Plants Images using Combined Classifier*.
- Gogul, I., & Kumar, V. S. (2017). Flower species recognition system using convolution neural networks and transfer learning. *2017 4th International Conference on Signal Processing, Communication and Networking, ICSCN 2017*, 1–6. <https://doi.org/10.1109/ICSCN.2017.8085675>
- Hiary, H., Saadeh, H., Saadeh, M., & Yaqub, M. (2015). Flower Classification using Deep Convolutional Neural Networks, *I2*, 1–8. <https://doi.org/10.1049/iet-cvi.2017.0155>

- Hussein, S. A., Naby, H. El, & Youssif, A. A. A. (2016). Image Multi-Classification using PHOW Features. *IOSR Journal of Computer Engineering*, 18(5), 2278–2661.
<https://doi.org/10.9790/0661-1805063136>
- Marín, M., & Rangel, J. (2000). *Comercialización Internacional de Flores: Antecedentes y Evolución: 1990-1999*. Universidad Nacional de Colombia, 278.
- Norvig, P., & Russell, S. (2014). *Inteligencia artificial. Elsevier Brasil*. (Vol. 1).
<https://doi.org/M-26913-2004>
- Núcleo Ambiental S.A.S. (2015). *Flores & follajes*, 1–42. Recuperado de:
https://www.google.com.co/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&cad=rja&uact=8&ved=0ahUKEwiMq_TrkpXQAhVFKyYKHcIgAMwQFghNMAY&url=https://www.ccb.org.co/content/download/13733/175129/file/Flore_Follajes.pdf&usg=AFQjCNFGTwQCPrE2PPnoFqW55y-LHixNvA&sig
- Liu, W., Rao, Y., Fan, B., Song, J., & Wang, Q. (2017). Flower classification using fusion descriptor and SVM. *2017 International Smart Cities Conference, ISC2 2017*.
- Palma, J., y Marín, R. (2008). *Inteligencia Artificial. Inteligencia Artificial* (Vol. 2). McGraw-Hill. España. <https://doi.org/10.4114/ia.v2i6.614>
- Perez, R. (2002). *Plagas Y Enfermedades Importantes Del Rosal*, 63.
- Ponce, P. (2010). *Inteligencia Artificial con Aplicaciones a la Ingeniería*. (Alfaomega, Ed.) (Primera Ed). México.
- Purohit, S., Gandhi, S., Viroja, R., & Chaudhary, N. (2015). Automatic Plant Species Recognition Technique using Machine learning approaches. *2015 International Conference on Computing and Network Communications (CoCoNet)*, 710–719.
<https://doi.org/10.1109/CoCoNet.2015.7411268>
- Ravindran, P., Costa, A.M., Soares, R.E., & Wiedenhoeft, A.C. (2018). *Classification of CITES-listed and other neotropical Meliaceae wood images using convolutional neural networks. Plant Methods*.
- Shukla, T., Mishra, N., & Sharma, S. (2013). *Automatic image annotation using SURF Features. 2013 Annual IEEE India Conference, INDICON 2013*, 68(4), 920–924.
<https://doi.org/10.1109/INDCON.2012.6420748>
- Tárrega, P., (2015, marzo 5). *Procesado de imagen de forma sencilla* [Archivo de video]. Recuperado de https://la.mathworks.com/videos/image-processing-made-easy-100161.html?fbclid=IwAR3fYY1kCVzKTJgKpjXAe_QIDEVoidW_E-3dWbOtjDW4f5pRXjUKMQcadTQ

Tárrega, P., (2015, mayo 7). *Aprendizaje automático (Machine Learning) de forma sencilla* [Archivo de video]. Recuperado de <https://la.mathworks.com/videos/machine-learning-made-easy-101277.html>

Tiay, T., Benyaphaichit, P., & Riyamongkol, P. (2014). *Flower Recognition System Based on Image Processing*, 99–102.

Uddin, M.A., Joolee, J.B., Alam, A., & Lee, Y. (2017). *Human Action Recognition Using Adaptive Local Motion Descriptor in Spark. IEEE Access*, 5, 21157-21167.

Xia, X., Xu, C., & Nan, B. (2017). Inception-v3 for flower classification. *2017 2nd International Conference on Image, Vision and Computing, ICIVC 2017*, 783–787. <https://doi.org/10.1109/ICIVC.2017.7984661>

Zamora, P. E. (2016). *Algunas Fisiopatías De Frutos, Tallos Y Hojas En Cultivos Protegidos*.

Zawbaa, H. M., Abbass, M., & Basha, S. H. (2014). *An Automatic Flower Classification Approach Using Machine Learning Algorithms. Icacci*, 895–901. <https://doi.org/10.1109/ICACCI.2014.6968612>

4. Contenidos

Objetivos

El objetivo general del trabajo consiste en implementar un sistema para el análisis de la calidad de las rosas por medio de procesamiento de imágenes y técnicas de visión artificial, con la finalidad de cumplir con los criterios de exportación acotados, para lograrlo se establecieron 4 objetivos específicos de la siguiente manera: 1 manipular las imágenes por medio de procesamiento digital de imágenes para lograr la atenuación de posibles ruidos y distorsiones presentes al momento de la captura, 2 clasificar la imágenes manipuladas mediante la extracción de características y la aplicación de algoritmos de aprendizaje de máquina, 3 evaluar el desempeño de las técnicas implementadas en el sistema en términos de rapidez y precisión de clasificación, 4 implementar una interfaz gráfica para la visualización de los resultados generados por el sistema.

Justificación

La justificación del proyecto se apoya básicamente en la exploración de nuevas tecnologías para apoyar el sector agrícola del país entendiendo que la capacitación de personal nuevo para la labor de clasificación de rosas y verificación de calidad, por lo

general toma tiempo y muchas veces debido a su inexperiencia el trabajador puede pasar por alto alguna imperfección, por lo que esta es una labor de suma importancia ya que en el peor de los casos se puede producir enormes pérdidas en la producción de la rosa.

Antecedentes

Los antecedentes que se encontraron sobre clasificación de flores con visión artificial se enfocan en su mayoría en la clasificación de flores por especies o variedades de flores por lo que ninguno de los documentos encontrados trabaja el tema de la verificación de calidad. No obstante, brindan un excelente punto de partida en el sentido de las diferentes estrategias que se deben emplear para reconocer patrones, texturas y elementos que brindan información sobre la imagen a analizar. Muchas de estas estrategias incluyen el análisis de los órganos de las plantas, técnicas de extracción de características, clasificadores entrenados y flujos para lograr el entrenamiento de estructuras de clasificación.

Marco teórico o de referencia

En este apartado inicia explicando la formación de la imagen analógica desde una mirada física para luego entender cómo se lleva a cabo el proceso de digitalización y su conversión a los diferentes sistemas de digitalización como lo es RGB o CMYK, siendo RGB uno de los más empleados. Se describe como se realiza el proceso de muestreo en términos de píxeles con diferentes intensidades de color que combinados producen una imagen, se explica también las diferentes técnicas para lograr el preprocesamiento de imágenes, conversión a escala de grises, segmentación, binarización y las diferentes operaciones morfológicas que permiten eliminar elementos que no son de interés en la imagen, un ejemplo de esto es el ruido. Después se trata el tema de la inteligencia artificial explicando el cómo se tratan los datos contenidos en la imagen y como encontrar patrones que describen características específicas de cada categoría; para ello se realiza una descripción de los diferentes tipos de extractores de características, como también se realiza una descripción de los diferentes clasificadores que se han desarrollado con el fin de predecir o clasificar. Finalmente se trata el tema de la floricultura en el país, los diferentes términos que se emplean y las principales afectaciones que dan cabida a la generación de patologías que afectan el estado de la rosa y no hacen de la misma un producto en óptimas condiciones de exportación.

Desarrollo y resultados

En este capítulo se detalla el procedimiento llevado a cabo para lograr la implementación de cada una de las etapas que conforman el algoritmo esto es, la descripción de cada

uno de los momentos propuestos en la metodología y paralelamente se obtienen los diferentes resultados que dan cabida a la continuación del proyecto, ya que en todo momento se están llevando a cabo pruebas de funcionamiento con el fin de detectar errores y corregir estos mismos a la vez.

Interpretación y discusión de los resultados

En este capítulo se realiza un análisis que permita el entendimiento de los resultados obtenidos del programa en su totalidad, teniendo como material de análisis las matrices de confusión de las estructuras entrenadas y los tiempos que le toma al sistema realizar las diversas clasificaciones, repasando detenidamente si el método empleado como extractor de características es el más adecuado y evaluando al mismo tiempo la exploración de técnicas más robustas que puedan ofrecer mejores resultados pero que por limitaciones de tiempo no es posible efectuar.

5. Metodología

La metodología seguida para el desarrollo de la investigación es ideada por el autor y comprende cinco momentos principales y cada uno de los mismos conllevan una serie de pasos y actividades requeridas antes de pasar al siguiente momento, además algunos de estos pasos son iterativos y requieren de reevaluar los resultados y volver a diseñar actividades:

Momento 1: Documentación e indagación sobre las técnicas de visión artificial aplicadas al sector agrícola, además, es un momento adecuado para aprender sobre cómo aplicar estas técnicas usando Matlab y la utilización de los comandos que sirven como equivalentes de procesamiento de imagen en la teoría.

Momento 2: Se evalúa la viabilidad de la implementación en Matlab de las diferentes técnicas encontradas y se realizan pruebas de validación con conjuntos de datos disponibles de forma gratuita en la red. Se determina la elaboración de un dataset propio que incluyan parámetros de calidad de la rosa, como también se decide la construcción de un ambiente con condiciones controladas para la toma de fotografías de las rosas.

Momento 3: En este momento intermedio del diseño del sistema es importante porque se lleva a cabo la consecución de las muestras a trabajar, se realizan desplazamientos a empresas de producción, toma de fotografías y etiquetado de las mismas.

Momentos 4 y 5: En estos momentos fue de gran importancia el trabajo en paralelo ya que consistió en el desarrollo de los diferentes algoritmos que hacen parte del programa y que se ejecutan con la activación de los botones de la interfaz de usuario, por lo que la realización de pruebas y corrección de errores fueron acciones constantes en estos dos momentos.

6. Conclusiones

- Se da cumplimiento a los objetivos específicos del proyecto en cuanto se logra manipular las imágenes por medio de preprocesamiento digital de imágenes logrando la atenuación de ruidos y distorsiones, esto empleando filtros morfológicos, segmentación basada en color, conversión a escala de grises y conversión binaria.
- Se logra la clasificación de imágenes basada en extracción de características, aunque la técnica seleccionada para tal fin no muestra una alta precisión mostrando errores de clasificación de flor buena en la categoría flor con veloso de hasta el 80%.
- Se logra una segunda clasificación de las imágenes basada en tamaño de botón y longitud del tallo usando técnicas de preprocesamiento sin error alguno.
- Se evalúa el desempeño de las técnicas empleadas determinando que el sistema le toma demasiado tiempo debido a varios factores, entre ellos la técnica empleada *bag of features* con 190 características por imagen y la cantidad de imágenes por cada flor.
- Se implementa una interfaz gráfica de usuario que permite ver los resultados generados por el sistema y a su vez permite la interacción con un usuario.
- Se propone la exploración de técnicas de visión artificial más robustas que permitan realizar la tarea de forma precisa y que optimicen el tiempo de clasificación.
- Para trabajos futuros, también se propone el desarrollo de estas técnicas en plataformas gratuitas como Python, los cuales cuentan un buen soporte en el campo de la inteligencia artificial.

Elaborado por:	Muñoz Amaya, Omar Alfonso
Revisado por:	Herrera Fernández, Luis Jorge

Fecha de elaboración del			
Resumen:	22	2	2020

Contenido

Índice de Ecuaciones	13
CAPÍTULO 1	14
1 Introducción.	14
1.1 Planteamiento del Problema.....	16
1.2 Objetivos	18
1.2.1 Objetivo General.	18
1.2.2 Objetivos Específicos.....	18
1.3 Justificación.....	19
1.4 Antecedentes	20
1.4.1 <i>Flower Recognition System Based on Image Processing</i>	20
1.4.2 <i>An Automatic Flower Classification Approach Using Machine Learning Algorithms</i>	21
1.4.3 <i>Automatic Plant Species Recognition Technique using Machine Learning Approaches</i>	21
1.4.4 <i>Flower Classification Using Deep Convolutional Neural Networks (CNN)</i>	23
1.4.5 <i>Plant Identification using score-based fusion of multi-organ images</i>	23
1.4.6 <i>Flower Classification using Fusion Descriptor and SVM</i>	24
1.4.7 <i>Flower Species Recognition System Using Convolution Neural Networks and Transfer Learning</i>	25
1.4.8 <i>Floriculture Classification Using Simple Neural Network and Deep Learning</i>	26

1.4.9	<i>Inception-V3 For Flower Classification</i>	27
1.4.10	<i>MobileNets for Flower Classification Using TensorFlow</i>	28
CAPÍTULO 2		30
2	Marco Teórico o de Referencia.....	30
2.1	Creación de la imagen.....	30
2.2	La Luz.	30
2.3	El color.	31
2.4	La imagen digital.....	32
2.4.1	Muestreo	33
2.4.2	Cuantización	33
2.5	Procesado de imagen.....	33
2.5.1	Ajuste de Contraste:.....	33
2.5.2	Segmentación.....	34
2.5.3	Umbralizado.....	34
2.5.4	Binarizado.	34
2.5.5	Operaciones Morfológicas	34
2.6	Inteligencia Artificial (IA)	35
2.6.1	<i>Machine Learning</i>	36
2.6.2	Red de Neuronas Artificiales	36
2.6.3	Aprendizaje o entrenamiento de las RNA	37

2.6.4	Clasificación de imágenes por categorías por medio de la extracción de características: bag of features.	38
2.6.5	Extracción de características SIFT.....	40
2.6.6	<i>Local Binary Patterns (LBP)</i>	40
2.6.7	<i>Histogram of Oriented Gradients (HOG)</i>	41
2.6.8	<i>Speed Up Robust Features (SURF)</i>	41
2.6.9	<i>Bag of Words (BOW)</i> (Bolsa de palabras)	42
2.6.10	Random Forest RF	42
2.6.11	Vecinos próximos o <i>Near neighbor (NN)</i>	43
2.6.12	K vecinos más próximos o <i>K - nearest neighbors (kNN)</i>	43
2.6.13	Análisis de Componentes Principales o <i>Principal Component Analysis (PCA)</i>	44
2.6.14	Clasificador de Naïve Bayes o <i>Naive Bayes classifier NBC</i>	45
2.6.15	Clasificación Bayesiana o <i>Bayesian classification</i>	45
2.6.16	Máquinas de soporte vectorial o <i>Support vector machines</i>	46
2.6.17	Redes Neuronales Convolucionales o <i>Convolutional neural networks CNN</i>	47
2.7	Floricultura.	48
2.7.1	Especies de flores más vendidas	49
2.7.2	Terminología en floricultura	49
2.7.3	Enfermedades en la floricultura	50
2.7.4	Fisiopatías en las rosas	52

CAPÍTULO 3	55
3 Procedimiento Metodológico.....	55
3.1 Descripción general del sistema propuesto	55
3.2 Metodología.	57
CAPÍTULO 4.....	59
4 Desarrollo y Resultados	59
4.1.1 Momento 1	59
4.1.2 Momento 2.....	60
4.1.3 Momento 3	64
4.1.4 Momento 4.....	65
4.1.5 Momento 5.....	78
CAPÍTULO 5	88
5 Interpretación y Discusión De Resultados.....	88
CAPÍTULO 6.....	98
6 Conclusiones, Recomendaciones y Trabajo Futuro.....	98
REFERENCIAS	102
ANEXOS.....	105
Anexo 1	105
Anexo 2	106
Anexo 3	108

Anexo 4	108
Anexo 5	110
Anexo 6	111

Índice de Ecuaciones

Ecuación 1. Matriz de Hesse.....	41
Ecuación 2. Probabilidad de clase en RF.....	43
Ecuación 3. Función de decisión.....	43
Ecuación 4. Teorema de Bayes.....	46

CAPÍTULO 1

1 Introducción.

Desde el enfoque de la inteligencia artificial que estudia el comportamiento humano, este proyecto trata el tema de la visión artificial aplicado a la solución de problemas de clasificación para control de calidad de rosas rojas. Además, también se utilizan técnicas de procesamiento de imagen para realizar clasificaciones basadas en dimensiones y color. Los criterios de clasificación se dan en términos de las características de mala calidad que puedan adquirir las rosas en el cultivo, tales como: enfermedad vellosa, la malformación tallo torcido y deformaciones por botones extremadamente pequeños o grandes. El diseño de un sistema que sea acorde a los requerimientos plantea dos etapas fundamentales: Etapa de preprocesamiento y etapa de análisis de la imagen con algoritmo de inteligencia artificial, dicho algoritmo consiste en una técnica de Machine Learning denominado aprendizaje supervisado, que precisa de entrenamiento de clasificadores los cuales se disponen en la herramienta *Classification Learner* de Matlab. Para entrenar el clasificador, fue necesario extraer características aplicando la técnica *Bag Of Words*. Por otra parte, en la etapa de procesamiento se aplicaron técnicas que permitieron el ajuste de contraste, segmentación basada en color, filtros morfológicos y medición de órganos de las rosas con análisis por regiones binarias.

La razón del desarrollo de este proyecto es proponer un diseño exploratorio en el campo de la floricultura, con el fin de analizar las diferentes técnicas aplicables que permiten optimizar los procesos de producción agrícola del país, entendiendo que el entrenamiento de personal nuevo por temporadas toma demasiado tiempo, agregando también condiciones de estrés, cansancio y error humano, podrían ocasionar problemas mayores en términos de calidad. El problema de no tener control sobre una enfermedad tan peligrosa como el vellosa, perjudica todo el proceso de

cultivo y cuidados que requieren las rosas antes de ser exportadas. Por lo tanto, el objetivo principal de toda empresa productora de flor es cumplir con los estándares mínimos de calidad que eviten devoluciones que resulten en pérdidas económicas.

La metodología usada para la elaboración del proyecto es ideada por el autor y comprende cinco momentos principales. En cada momento, se llevan a cabo una serie de actividades para conseguir finalmente una interfaz gráfica usada para verificar el desempeño del diseño propuesto. Para efectos de pruebas y validación del sistema se contó con más de seiscientas fotografías de rosas capturadas en un ambiente encerrado con fondo blanco e iluminación constante blanca, con el fin de reducir la interferencia ambiental evitando la eliminación de fondo complejo y los diferentes niveles de iluminación que dependen de la hora del día.

El documento está compuesto por seis capítulos dispuestos de la siguiente manera:

Capítulo 1, se realiza el planteamiento del problema, justificación del proyecto, planteamiento del objetivo general y específicos, como también se realiza un estudio de antecedentes relacionados con visión artificial aplicados a la agricultura.

Capítulo 2, desarrolla el marco teórico o de referencia sobre los temas que aborda el proyecto, conceptos sobre luz, imagen digital, procesamiento de imagen y análisis de imágenes con técnicas de inteligencia artificial. Además, se expone el tema de la floricultura en el país y algunos conceptos básicos necesarios relacionados con la floricultura, son los temas que principalmente se tratan en este capítulo.

Capítulo 3, describe de forma general el sistema que se propone desarrollar y la metodología que se va a seguir para lograr el desarrollo de dicho sistema. La metodología se plantea en forma de esquema y está compuesto por cinco momentos principales.

Capítulo 4, en este capítulo se detalla el desarrollo de cada uno de los momentos planteados en el capítulo anterior, como también, los resultados que se obtienen en la medida que se desarrollan. Al final, se obtiene una interfaz gráfica que integra todas las etapas y a la vez permite validar el funcionamiento del diseño propuesto.

Capítulo 5, se elabora un análisis de los resultados obtenidos a lo largo del desarrollo de los momentos planteados en la metodología, comparando el rendimiento de los dos mejores clasificadores que se obtuvieron en la aplicación de Matlab, usándolos son fotografías nuevas que no entraron en el proceso de entrenamiento. De la misma manera, se analizan las matrices de confusión de cada uno de los clasificadores y el porcentaje de precisión en cada categoría.

Capítulo 6, este capítulo presenta las conclusiones con base a los resultados obtenidos, planteando como posibilidad evaluar mejores alternativas en inteligencia artificial que solucionen eficientemente el problema de clasificación de rosas, y que además sean capaces de tratar una mayor variedad de enfermedades, y más criterios de calidad.

1.1 Planteamiento del Problema.

“Actualmente Colombia es uno de los principales países exportadores de flor, exportó a alrededor de 90 países y generó ingresos de hasta 785 millones de dólares en sólo el primer semestre del 2017, equivalente a 129.000 toneladas.”

(Asocolflores, citado por, Revista Dinero, 2017)

En la sabana centro de Bogotá, hay gran concentración de fincas pequeñas, medianas y grandes que se dedican al cultivo y cosecha de rosas. En el proceso de obtención final del producto intervienen múltiples etapas que van desde la siembra, cuidado durante el crecimiento hasta la cosecha.

Cuando finalmente la rosa está lista, es llevada a la etapa de postcosecha, donde la rosa es revisada y clasificada de forma manual por personal con bastante experiencia y siguiendo ciertos parámetros, como lo son la longitud del tallo, el tamaño del botón, las enfermedades, el grado de apertura, el color, el maltrato y las deformidades principalmente.

Sin embargo, los anteriores parámetros varían según la finca que las produzca. Las enfermedades son uno de los aspectos que más preocupa a las empresas, pues es indispensable que no llegue el producto en malas condiciones debido a la presencia de estas, que generalmente provocan irregularidades en la uniformidad del color. Algunas enfermedades son más comunes que otras, de igual manera hay especies de rosas que son más vulnerables a ciertos padecimientos. Las enfermedades más reconocidas son: Moho gris o botritis, Velloso, Oídio, Roya y Agallas o tumores.

Además de las enfermedades, las rosas también sufren problemas provocados generalmente por tropismos, los cuales son respuestas a estímulos y condiciones ambientales, estos comportamientos resultan en la pérdida de la calidad de la rosa. Algunas de estas son el tallo torcido (Cuello de ganso), las cabezas grandes y pétalos desordenados y sin forma.

La producción de rosas, especialmente la de color rojo, aumenta en fechas especiales como es el caso en el mes de febrero del día de San Valentín en los Estados Unidos, que provoca el aumento de la demanda de manera considerable. Es en estas épocas cuando las fincas contratan personal por temporada, que generalmente no tienen suficiente experiencia, y por lo tanto el proceso de enseñanza de las técnicas de clasificación al nuevo personal puede tomar bastante tiempo, el cual es limitado dada la demanda de producción.

Por otro lado, al proceso de postcosecha ingresa una cantidad determinada de producto para clasificar, pero no es posible tener la certeza de cuánto producto ha sido procesado. Es decir,

ingresa a la etapa de postcosecha una cierta cantidad de rosa resultando para exportación una cantidad menor. Lo anterior debido que en el proceso de control de calidad se eliminan rosas que no cumplen los requisitos para exportación, de acuerdo con el criterio y a la experiencia del operario. Es sobre este proceso en el que generalmente no se tiene control, convirtiéndose en un problema que es importante, en términos de la cantidad desechada de rosas que pueden cumplir las condiciones de calidad de exportación.

Con base en lo anterior se puede plantear la siguiente pregunta de investigación: ¿Cuál es el sistema de clasificación más eficiente, empleando técnicas de procesamiento de imágenes y visión artificial, que se puede aplicar al proceso de calidad en la selección de rosas a exportar?

1.2 Objetivos

1.2.1 Objetivo General.

Implementar un sistema para el análisis de la calidad de las rosas por medio de procesamiento de imágenes y técnicas de visión artificial con la finalidad de cumplir los criterios de exportación

1.2.2 Objetivos Específicos.

- Manipular las imágenes por medio de procesamiento digital de imágenes para lograr la atenuación de posibles ruidos y distorsiones presentes al momento de la captura.
- Clasificar las imágenes manipuladas mediante la extracción de características y la aplicación de algoritmos de aprendizaje de máquina.
- Evaluar el desempeño de las técnicas implementadas en el sistema en términos de rapidez y precisión de clasificación.
- Implementar una interfaz gráfica para la visualización de los resultados generados por el sistema.

1.3 Justificación

Este proyecto puede tener un impacto positivo en empresas de producción de rosa, las cuales pueden llegar a tener un mayor control sobre la calidad de la rosa de tipo exportación, al implementar en su línea de producción equipos que sirvan de sensores para capturar la información, para su posterior procesamiento empleando un algoritmo el cual proporciona un informe acerca de la calidad y cantidad obtenida durante el proceso de producción en una computadora.

Debido a múltiples factores como el cansancio, el estrés, la falta de experiencia, la dificultad y el tiempo que toma el aprendizaje, entre otros, los operarios pueden pasar por alto algún imperfecto en la rosa, lo cual resulta en devoluciones por parte de los clientes, provocando pérdidas y haciendo infructífero todo el cuidado y procesos anteriores a la cosecha de la flor. Además, el sistema ayudaría en la toma de decisiones, por parte de los encargados de producción, las cuales aporten en la mejora de la calidad del producto al ser más evidentes ciertas características frecuentes en la producción.

Por otra parte, se plantea la exploración de técnicas de Inteligencia Artificial mediante obtención, procesamiento y análisis de fotografías por medio de Machine Learning como una solución a la mejora de los procesos de inspección de calidad aplicado a la floricultura del país, haciendo del desarrollo de este proyecto un aporte interesante a este tipo de procesos. La razón para implementar Machine Learning es porque el proceso de entrenamiento del sistema toma menos tiempo y podría ser más eficiente en relación con el proceso de entrenamiento de personas para realizar la misma tarea.

1.4 Antecedentes

En este apartado se referenciarán los trabajos, artículos e investigaciones relacionados con algoritmos de clasificación e inspección aplicados a los procesos de producción del sector agrícola principalmente, con el fin de establecer parámetros y metodologías que puedan servir de base y den continuidad al proyecto propuesto.

1.4.1 Flower Recognition System Based on Image Processing

En este documento, se describe el desarrollo de un sistema de reconocimiento de flores basado en el procesado de imágenes empleando algoritmos para reconocer las características de los bordes de las flores en las imágenes, los colores se obtienen de los histogramas de las imágenes y finalmente se usa el algoritmo de vecinos más próximos para la clasificación de las flores. Los autores elaboran un sistema liviano y portable que sea capaz de ser ejecutado en dispositivos móviles.

Los resultados experimentales señalan que el sistema se usa con 10 especies de flores y que para la prueba del sistema se emplean 50 imágenes para cada especie. El sistema fue entrenado con 100 imágenes obtenidas de datasets gratuitos en línea. Con este conjunto de datos el sistema logra una precisión mayor al 80% en la mayoría de los experimentos realizados así:

Para el caso de la rosa roja y la flor llamada guisante de mariposa se obtiene una precisión del 100% debido a que el color de estas especies es distintivo, mientras que en el caso de Marigold presenta la menor precisión debido a que su conjunto de colores es similar al conjunto de colores del girasol y el conjunto de características de los bordes es similar a las características de borde de la flor conocida con el nombre amaranto de globo.

De este documento es posible evidenciar la posibilidad de usar algoritmos para obtener los bordes de las plantas y así usar esta información para conocer la posición de las partes de interés y trabajar sobre estas.

1.4.2 An Automatic Flower Classification Approach Using Machine Learning Algorithms

Este artículo, consistió en el desarrollo de un clasificador de 8 categorías de flores utilizando algoritmos de Machine Learning. Para esto se emplean los algoritmos *Scale Invariant Feature Transform (SIFT)* y *Segmentation Based Fractal Texture Analysis (SFTA)* para extraer características de las flores. La estrategia que se propone consta de 3 fases: Segmentación, extracción de características y clasificación. En la fase de segmentación se elimina el fondo complejo que no hace parte del conjunto de datos de interés, luego se extraen las características de la imagen de la flor y, por último, en la fase de clasificación se utilizaron los algoritmos *Support Vector Machine (SVM)* y *Random Forest (RF)* con el fin de clasificar diferentes tipos de flores.

El sistema se implementó en Matlab 2013a con el sistema operativo Windows 8.1. La precisión de la clasificación se logra en un 100% para las categorías Anthurium y Barbeton Daisy, la más baja fue 66.67% con Californian Poppy cuando se usa *SIFT* como extractor de características y *RF* como clasificador. Se observa además que *SFTA* alcanza una precisión de 77.78% a 100%, y con *SIFT* se logra una precisión de 66.67% a 100%.

1.4.3 Automatic Plant Species Recognition Technique using Machine Learning Approaches

En este artículo se propone la identificación de plantas basada en imágenes de especies de plantas. Las imágenes pueden ser de específicamente algún órgano de la planta como el tallo, las hojas, flores o frutos y para esto se utilizan diferentes métodos para el reconocimiento de las

especies y según la parte de la planta que se observa en la imagen el sistema elige de forma dinámica que enfoque emplear. Además, para el reconocimiento de la flor, se agregan datos adicionales como fecha, latitud, longitud asociadas a la imagen de la planta y así hacer más sencillo y preciso el proceso de identificación.

El software que se implementó en Matlab recibe de entrada la imagen asociada más los metadatos de entrada; después reconoce las especies de la imagen ingresada y produce una salida que muestra el nombre de Familia, Género y Especie.

En el desarrollo del sistema, se probaron diferentes enfoques para decidir cuáles eran los más precisos, estos enfoques se probaron en 10 especies de plantas con 5 subcategorías que son:

- Subcategoría Flor: En la cual se observa que la clasificación usando *Combined Vocabulary* en color, forma y características de textura alcanza una precisión máxima de 73.18%, siendo la mayor en esta subcategoría comparada con otros dos enfoques realizados; el enfoque *SIFT* con *Smart Process Management (SPM)* con una precisión de 69.74% y el enfoque basado en pixeles con una precisión de 33.62%
- Subcategoría Fruto: En esta subcategoría se encuentra que el enfoque basado en características *SIFT* alcanza la máxima precisión, la cual es del 67.3% comparado con otros dos enfoques que alcanzan 57% el enfoque *Combined Vocabulary* y 18.9% para el enfoque basado en intensidad es decir basado en pixeles.
- Subcategoría Hoja: Se encuentra que el enfoque de características *SIFT* alcanza la precisión máxima con 69.17% en cambio el enfoque basado en pixeles alcanza 22.9% y el enfoque *Combined Vocabulary* alcanza 45%.
- Subcategoría *Leafscan*: Esta subcategoría contiene imágenes de hojas sin fondo. Se alcanza la máxima precisión es 98% con características *SIFT*, mientras que con el

enfoque basado en píxeles la precisión es de 72.77% y el enfoque *Combined Vocabulary* alcanza una precisión de 82.67%.

- Subcategoría Tallo: El enfoque que mayor precisión alcanza es *SIFT* el cual alcanza 76.57% comprado con los enfoques *Combined Vocabulary* que logra una precisión de 58.76% y el enfoque basado en píxeles que alcanza una precisión de 20.75%.

1.4.4 Flower Classification Using Deep Convolutional Neural Networks (CNN)

En este artículo se muestra la utilización de los métodos de aprendizaje profundo, siendo un ejemplo de esto las *Convolutional Neural Networks (CNN)*. Se emplea además grupos de flores de tamaños moderados y teniendo como estrategia el recorte de la imagen seleccionando la parte de interés. El método que se plantea en el artículo detecta la región que corresponde a la flor y luego estas imágenes recortadas se usan para entrenar el clasificador *CNN*, el cual distingue diferentes clases de flores. Según los resultados, el método es exitoso debido a que la detección de la flor por medio de la segmentación permite que la clasificación *CNN* se centre únicamente en la región de interés, descartando las regiones que no lo son. La detección de la flor se logra mediante clasificación binaria. El método se evalúa con diferentes grupos de flores y consigue tener al menos un 97% de precisión al momento de clasificar. Se concluye además que el aumento de datos aumenta la precisión en la segmentación y en la detección de la flor.

1.4.5 Plant Identification using score-based fusion of multi-organ images.

En este artículo los autores describen una técnica de fusión para la identificación de diferentes tipos de plantas a partir de imágenes de sus órganos: hoja, tallo, rama.

El conjunto de datos que recopilaron lo dividieron en tres partes. La primera parte se usó como datos de entrenamiento de la *CNN* para la identificación de un solo órgano. La segunda parte del conjunto de datos que regresaron de la *CNN* se utilizaron como conjunto de datos de

entrenamiento del modelo *SVM*. Se usó la tercera parte del conjunto de datos para evaluar los resultados de los métodos de fusión tardía y *CNN*. Para equilibrar el número de muestras positivas y negativas, se recolectaron al azar los puntos negativos en lugar de tomarlos todos. El esquema de fusión híbrida que propusieron, utiliza los esquemas de prueba de la regla del producto y los enfoques basados en la clasificación.

Finalmente, los experimentos mostraron que las técnicas de fusión aumentan el rendimiento de la tarea de identificación de las plantas. Por otro lado, el modelo de fusión híbrida presentó los mejores resultados en todas las evaluaciones. Según los resultados obtenidos, al combinar dos tareas, ayuda al usuario final a obtener un alto rendimiento.

1.4.6 Flower Classification using Fusion Descriptor and SVM

En este documento se desarrolla un enfoque de clasificación de flores en el software Matlab2015a, utilizando la tecnología de extracción de características. Para esto, se utiliza un descriptor basado en *Pyramid Histogram of Visual Words (PHOW)* por sus siglas en inglés, con el fin de extraer el color, la textura y la información del contorno de la imagen de la flor, este descriptor es una extensión del modelo *Bag Of Words (BOW)* en el que las características obtenidas por *SIFT* se tratan como palabras. Considera las características locales de la imagen y divide la imagen en subregiones más finas (pirámides) y concatena el histograma de cada una de estas regiones para el histograma de la imagen original. Por otra parte, se utiliza en la función *PHOW, Dictionary Learning y Locality- constrained Linear Coding (LLC)*, el cual se emplea para codificar las características de las imágenes de las flores al usar la restricción de localidad para proyectar cada descriptor en su diccionario de coordenadas locales. Las imágenes se presentan mediante *Spatial Pyramid Pooling (SPP)*, el cuál es un modelo de codificación, utilizado en la extracción de características de imagen. La imagen de entrada se divide en una

secuencia de cuadrículas crecientes superpuestas en cada nivel de pirámide. Los descriptores se extraen de todas las cuadrículas y se concatenan para formar el descriptor de imagen final. A un clasificador *Support Vector Machines (SVM)* se envía una función integrada, para el entrenamiento y pruebas.

Finalmente se concluye que el enfoque propuesto promete hacer una clasificación automatizada de la flor y reducir el tiempo y el trabajo. Los resultados de la experimentación en el conjunto de datos Oxford 17 demostraron que las características utilizadas fueron discriminativas y robustas, ya que dan una alta precisión en la clasificación de las flores, siendo de 86.17%.

1.4.7 Flower Species Recognition System Using Convolution Neural Networks and Transfer Learning

En este trabajo se utilizó el enfoque de aprendizaje profundo o *Deep Learning* utilizando *CNN*, con el fin de reconocer especies de flores. Las imágenes de las diferentes especies de flores se obtienen usando el módulo de la cámara que viene incorporado en un celular. La extracción de las características de las imágenes de flores se realiza mediante un enfoque de aprendizaje por transferencia. En el desarrollo del algoritmo, emplean también un clasificador de aprendizaje de máquina, con el fin de obtener índices de precisión más altos al momento de clasificar las especies de flores.

Finalmente, se observa que las redes neuronales de convolución (*CNN*) combinadas con el método de aprendizaje por transferencia o *Transfer Learning* como extractor de características superó a todos los métodos de extracción de características que se elaboran a mano, como *Local Binary Patterns (LBP)*, estadísticas de canales de color, histogramas de color, textura Haralick, momentos Hu y Momentos de Zernike. Según los resultados, se concluye que *CNN* combinado

con el enfoque de *Transfer Learning* produce muy buenas precisiones en rangos de 73.05%, 93.41%, y 90.60% para los cuales se usó las arquitecturas *OverFeat*, *Inception-v3* y *Xception*, respectivamente, como extractores de características en el *dataset* usado, el cual fue FLOWERS102; esta base de datos gratuita contiene el registro de varias especies de flores disponibles para ser descargados y utilizados en proyectos investigativos, procesamiento y tratamiento de imágenes con flores y rosas.

1.4.8 Floriculture Classification Using Simple Neural Network and Deep Learning

En el caso de la identificación de flores, los autores presentan un enfoque de aprendizaje no supervisado basado en *Deep Learning* para la identificación y clasificación de estas, de tal manera que se pueda usar en análisis de patentes de la industria floricultora en todas sus formas: viveros, comercio de flores, extracción de aceites naturales, producción de semillas y bulbos.

Se señala que el uso de una red neuronal desde cero requiere mucho tiempo y es ineficiente, además, la *Simple Neural Network* debido a la falta de conocimiento que obtiene al entrenar con pocos datos predice de igual manera con probabilidades más bajas, lo cual sugiere que para el sistema que se propone desarrollar podría no ser buena idea optar por un clasificador partiendo completamente desde cero, pues como se viene observando, pareciera ser conveniente usar algoritmos previamente elaborados y evaluar su desempeño.

En las conclusiones finales se señala que los valores de pérdida debido a la falta de conocimiento se podrían reducir si se utiliza *Transfer Learning* en una *Deep Neural Network* previamente entrenada.

Se observa un aumento en la tasa de precisión usando *Transfer Learning*. En las gráficas presentadas por los autores se evidencia que la tasa de precisión usando *Simple Neural Network* es de 83% y la tasa de precisión usando *Transfer Learning* es de 99%.

1.4.9 Inception-V3 For Flower Classification

Este documento se basa en el desarrollo del algoritmo clasificador de flores en el modelo Inception-v3 de la plataforma TensorFlow, el cual es una biblioteca de código abierto para *Machine Learning* desarrollado por Google con el fin de construir y entrenar redes neuronales para detectar patrones y correlaciones. Adicionalmente se usa el enfoque *Transfer Learning* para volver a entrenar los conjuntos de datos de las categorías de las flores.

Para hacer el entrenamiento y pruebas del algoritmo se emplea el conjunto de datos disponible Oxford -17 y Oxford -102 los cuales contienen 17 y 102 especies de flores respectivamente. Por cada especie contienen de 40 a 208 imágenes de flores con diferente luminosidad, ángulo de toma, postura y escala. Para el desarrollo específico de este algoritmo se elabora para la clasificación de 28 especies de flores.

Para la construcción del modelo de clasificación de flores, se realizan 4 pasos que son: preprocesamiento de imagen, proceso de entrenamiento, proceso de verificación y proceso de prueba. Como el algoritmo está basado en el modelo de red Inception-v3, se explica que esta es una red neuronal profunda, la cual requiere de varios días para capacitarla si no se emplea la ayuda de la técnica *Transfer Learning*, este método, según los autores mantiene los parámetros de la capa anterior y elimina la última capa del modelo Inception-v3. El número de nodos de salida en la última capa corresponde al número de categorías de clasificación.

Se concluye que la precisión de clasificación del modelo es del 95% en el conjunto de datos de flores de Oxford-17 y del 94% en el conjunto de datos de flores de Oxford-102, demostrando ser mayor que en otros métodos que utilizan el mismo conjunto de datos a clasificar.

1.4.10 MobileNets for Flower Classification Using TensorFlow

En este Proyecto, se presenta la experimentación de clasificación de flores, con un modelo basado en la arquitectura de Google, llamado MobileNets, siendo esta una (CNN). Se logra construir una aplicación con MobileNets más pequeña y rápida. Para el desarrollo de la aplicación se utilizó Python con TensorFlow, la cuál es una biblioteca de software libre.

El modelo que los autores plantean se divide principalmente en 4 etapas que son preprocesamiento de imágenes, capacitación, verificación y fase de prueba.

Se concluye que MobileNets demuestra ser una aplicación eficiente probándose con una gran cantidad de conjuntos de datos de imágenes. Se logra obtener una correcta clasificación de flores con un margen de precisión de entre el 85% y el 99%.

De acuerdo con la revisión general de antecedentes, el proyecto que se propone se desarrollará acorde a las sugerencias que se realizan en cada uno de las producciones académicas, donde las distintas maneras de extraer las características, lograr la clasificación y de entregar los informes por pantalla, dependiendo de los criterios de cada sistema, se especifican mediante diferentes algoritmos, los cuales serán la base del diseño y punto de partida para el desarrollo del sistema en general y que se adaptarán a los criterios previamente establecidos. Se opta entonces por emplear una técnica de *Machine Learning*, basada en extracción de características por medio de *Bag Of Words*, y con la información de las características extraídas por este método se realizará entrenamiento de distintos clasificadores con el fin de determinar cuál presenta una mayor precisión con los datos para integrarlo al producto final. Por otra parte, y antes de aplicar el algoritmo clasificador en el programa, se aplicarán las determinadas etapas de procesado de imagen para lograr mejoras a las imágenes que ingresan al sistema. Durante la etapa de tratamiento de imagen se aprovecharán las operaciones que se pueden realizar allí para obtener

características de las rosas como los son tamaño del botón (anchura y altura) y longitud del tallo, por medio de conversión en escala de grises, segmentación, umbralizado, operaciones morfológicas, conversión binaria y filtros morfológicos para eliminación de ruido y así reducir el coste computacional que de igual manera logre dar respuesta a los objetivos planteados inicialmente. Todo lo anterior implementado en una interfaz gráfica de usuario desarrollada en el software Matlab haciendo uso de los toolbox que incluye y que facilitan este proceso.

CAPÍTULO 2

2 Marco Teórico o de Referencia

2.1 Creación de la imagen.

Como se sabe, una imagen es una representación bidimensional $f(x, y)$ de un objeto o escena tridimensional, la manera de generar una imagen se describe a continuación:

La visión recoge la luz reflejada por los objetos en la escena y crea una imagen bidimensional sobre un plano de imagen. El plano de imagen está recubierto con material fotosensible: moléculas de rodopsina en la retina, haluro de plata en una película fotográfica o una matriz CCD (*Charged-Coupled Device*) en una cámara digital. Cada punto de un CCD integra los electrones liberados por la absorción de fotones durante un período de tiempo establecido. En una cámara digital el plano de imagen se subdivide en una rejilla rectangular de unos pocos millones de píxeles. El ojo tiene una matriz de píxeles similar que consiste en unos 100 millones de bastoncillos y cinco millones de conos, dispuestos en una malla hexagonal.

La escena real es muy grande comparada con el pequeño plano de imagen, por lo que es necesario disponer de alguna forma de enfocar la luz sobre este plano. Esta labor se puede realizar con lentes o sin ellas. En cualquier caso, la clave está en definir la geometría de manera que podamos decir dónde se encontrará cada punto de la escena sobre el plano de imagen. (Russel y Norvig, 2004, p.981).

2.2 La Luz.

Russel y Norvig (2004, p.984) afirman que “la luz es un requisito para la visión: sin luz, todas las imágenes estarían totalmente oscuras sin importar el interés de la escena.”. El brillo en un píxel es proporcional a la cantidad de luz que llega a la cámara desde una porción de superficie de la escena y que se proyecta en determinado píxel. Pero a su vez existen factores que afectan la cantidad de luz que inciden en el píxel, pues en ese píxel están presentes componentes de luz de toda la escena en general y debido a las propiedades reflectantes de cada uno de los objetos que conforman la escena, estos podrían funcionar como fuentes de luz directas sobre el píxel cuando reflejan luz.

2.3 El color.

La luz juega un papel importante para la formación de la imagen, sin embargo, para tener una forma correspondiente a un objeto, los bordes que a este lo componen y entre las características especiales que describen al objeto, es el color una de estas características, de acuerdo con Russel y Norvig (2004) quienes señalan que la luz visible que llega a nosotros se encuentra en un rango de longitudes de onda que varían desde 400 nm que corresponden al extremo violeta del espectro de luz hasta 700 nm en el extremo rojo, y que algunas de estas luces son puras debido a que se forman de una única longitud de onda y corresponden a uno de los colores del arcoíris, las demás luces son combinaciones de longitudes de ondas de estas luces puras. Es decir que, si necesitamos representar de forma exacta la física de la luz, tendríamos una mezcla de valores para un nivel de intensidad, en lugar de un único valor. Pero para imitar la percepción de la vista humana es posible delimitar este asunto. “Algunos experimentos (si nos remontamos a los trabajos de Thomas Young en 1801) han demostrado que cualquier mezcla de longitudes de onda, sin importar su complejidad, puede reproducirse con la mezcla de únicamente tres colores primarios” (Russel y Norvig, 2004, p.985). Si es posible tener una fuente de luz, la cual puede combinar de forma lineal tres longitudes de onda, que normalmente se seleccionan como rojo (700nm), el verde (546nm) y el azul (436nm), y si además es posible ajustar los valores de intensidad de cada color, es posible obtener cualquier combinación de longitudes de onda, es decir cualquier color. Por lo tanto, se puede suponer, según Russel y Norvig (2004, p.986) “las imágenes pueden representarse como un vector con únicamente tres valores de intensidad por píxel: uno para cada una de las tres longitudes de onda primarias.”. Es así como comúnmente se hace en la práctica, donde se asigna un byte (8 bits) para cada uno de los valores, pues con esto es suficiente para representar una imagen con alta fidelidad. La elección de longitudes de ondas

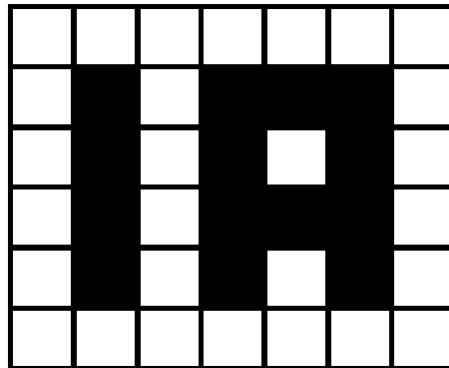
de los tres colores primarios rojo, verde y azul está relacionada con el hecho de que la retina del ojo humano presenta también tres tipos de conos máximos de receptividad en 650 nm, 530 nm y 430 nm respectivamente.

2.4 La imagen digital.

Una imagen digital, es una función de dos dimensiones $f(X, Y)$ donde X e Y representan coordenadas en el plano, la amplitud de esta función representa un nivel de intensidad o el nivel de gris en un punto determinado del plano. Además, si (X, Y) son finitos y discretos es posible afirmar que la imagen es digital (García, 2008).

Al ampliar una imagen, esta se compone de una cuadrícula que se forma debido a un número fijo de M filas por N columnas, estas cuadrículas individuales se les llama píxeles (figura 2.1).

Figura 2.1. Representación de imagen digital de M filas por N columnas.



Fuente. Elaboración propia.

En el píxel se almacena la información de la intensidad promedio de cada una de las tres capas: red, green y blue (R, G, B), esta intensidad por cada capa se dispone en un rango que depende de la cantidad de bits de resolución que se van a emplear, los cuales pueden ir de 0 hasta $(2^n - 1)$, donde 0 representa negro y $(2^n - 1)$, el valor extremo máximo blanco.

En el proceso de construcción de la imagen digital, generalmente hay dos pasos a seguir: el muestreo y la cuantización.

2.4.1 Muestreo

Es el procedimiento realizado por los dispositivos de captura que consiste en tomar determinado número de muestras o determinada cantidad de datos del mundo continuo (analógico), los cuales se acomodan en una matriz con m número de filas y n número de columnas. En el muestreo se establece el tamaño del píxel y el valor del brillo (García, 2008).

2.4.2 Cuantización

La cuantización es el proceso que consiste en asignar los valores específicos a los elementos que conforman la matriz, donde cada valor representa la amplitud en intensidad de la variable física en ese punto. Se asume que el valor más pequeño de nivel de gris corresponde al negro y el valor más grande de nivel de gris corresponde al blanco. En este rango mientras más niveles de grises se puedan representar, mayor cantidad de matices se pueden representar (resolución en bits, profundidad del píxel) (García, 2008).

2.5 Procesado de imagen.

Esta etapa consiste en realizar ajustes a las imágenes digitales para que los resultados sean más adecuados para la visualización o procesamiento posterior.

Las principales acciones que se suelen llevar a cabo en esta etapa son:

2.5.1 Ajuste de Contraste:

Es una técnica que sirve para mejorar una imagen, esta mejora se puede dar en, por ejemplo, hacer que algunas características de interés sean más fáciles de apreciar modificando parámetros de las imágenes, tales como intensidades o los colores y así poder apreciar más detalles en la imagen. Cuando se ajusta la intensidad de una imagen, se mapean los valores de las intensidades de la imagen en un nuevo rango de intensidades y así lograr una mejor distribución de los datos.

2.5.2 Segmentación.

García (2008, p. 71), define la segmentación como “un proceso que consiste en dividir una imagen digital en regiones homogéneas o similares con respecto a una o más características (como por ejemplo el brillo, el color, tamaño, longitud, forma) con el fin de facilitar su posterior análisis y reconocimiento automático”, de tal manera que se aprovechan las características específicas de las imágenes para realizar las respectivas operaciones que permitan alcanzar el objetivo que se pretende.

2.5.3 Umbralizado.

García (2008, p. 71) sobre el proceso de Umbralización señala que “es un proceso que permite convertir una imagen de niveles de gris o en color en una imagen binaria, de tal forma que los objetos de interés se etiqueten con un valor distinto de los píxeles del fondo”. Agregando, además que es una técnica de segmentación rápida con un bajo coste computacional lo que acelera el rendimiento en procesamiento de las imágenes en sólo la etapa de preprocesamiento.

2.5.4 Binarizado.

Este es un proceso que consiste en convertir las imágenes en blanco y negro (binarias), en el que generalmente se elige un umbral específico para los cuales los valores inferiores a dicho umbral serán cero y los superiores serán uno.

2.5.5 Operaciones Morfológicas

Primero, tres definiciones:

Morfología Matemática: Esta es una técnica que procesa la imagen de forma lineal que se enfoca principalmente en la geometría de los objetos. De aquí, es posible obtener información sobre la forma y la estructura de la imagen.

Análisis Morfológico: Con este análisis se obtienen componentes de la imagen que son necesarios en la representación y descripción de la forma de las regiones que pueden ser: Esqueletos, Fronteras y bordes.

Procesado Morfológico: Es un proceso que permite transformar la forma o la estructura de los objetos en una imagen. En el procesado morfológico existen tres elementos esenciales:

- a) Conjuntos (imágenes)
- b) Elementos estructurantes (en Matlab: discos, rectángulos, líneas y otros más en las nuevas versiones)
- c) Operadores Morfológicos: Dilatación, Erosión, apertura/cierre.

2.6 Inteligencia Artificial (IA)

Ponce (2010), hace alusión sobre los métodos de inteligencia artificial como “una respuesta al deseo de aproximar el comportamiento y el pensamiento humano a diversos sistemas para la solución de determinadas problemáticas”, siendo estos métodos no solo novedosos sino también muy eficientes y mejor planeados. El estudio de la IA es interesante porque plantea, por un lado, la comprensión sobre como aprenden los seres humanos y la construcción de agentes inteligentes que imiten este comportamiento, indica que en su momento la IA se abordó de diversas formas, por lo tanto, el campo de estudio derivó en diversos enfoques, los cuales están sustentados por los autores que defienden su postura sobre la IA.

En el texto de Russell y Norvig (2004), se presentan los siguientes cuatro enfoques:

A. Comportamiento humano: El enfoque de la prueba de Turing

En este enfoque es pertinente hacer una ampliación debido los aportes que permitieron establecer seis disciplinas, de las cuales dos hacen parte del desarrollo de este proyecto.

- Procesamiento de lenguaje natural
- Representación del conocimiento
- Razonamiento automático
- Aprendizaje automático
- Visión Computacional
- Robótica

B. Pensar como un humano: El enfoque del modelo cognitivo

C. Pensamiento racional: El enfoque de las «leyes del pensamiento»

D. Actuar de forma racional: El enfoque del agente racional.

2.6.1 *Machine Learning*

Uno de los problemas más antiguos de la ciencia experimental es encontrar funciones que ajusten, o expliquen, datos que se observan de fenómenos naturales. La principal ventaja de la existencia de tales funciones es la posibilidad de predecir el comportamiento del sistema natural en el futuro y controlar sus salidas mediante la aplicación de las entradas adecuadas. Algunos ejemplos interesantes podrían ser la predicción de valores en bolsa, la predicción de meteorológica o la clasificación de formas tumorales. La dificultad estriba en que los datos observados tienden a ir acompañados de ruido, y los mecanismos exactos que los generan normalmente son desconocidos. En ocasiones será posible encontrar un modelo matemático exacto que explique el proceso del que provienen los datos que observamos. Muchas veces, sin embargo, no podemos dar detalles de ese proceso. El objetivo, en este caso, será *estimar* el modelo subyacente que genera los datos observados.

En estos casos, las técnicas de aprendizaje automático nos permiten establecer estos modelos utilizando datos de ejemplo o experiencias pasadas. De este modo se podrán identificar ciertos *patrones* o regularidades en los datos y así podremos construir buenas *aproximaciones* al problema. Dentro de estas técnicas, que podríamos llamar de aproximación semiparamétrica, se encuentran las *Redes de Neuronas Artificiales* (Palma y Marín, 2008, p. 649).

2.6.2 **Red de Neuronas Artificiales**

Una Red de Neuronas Artificiales (en adelante, RNA) es un paradigma de procesamiento de información inicialmente inspirado en el modo en el que lo hace el cerebro. El elemento clave de este paradigma es su estructura. Las RNA están compuestas por un cierto número de elementos de procesamiento o *neuronas* que trabajan al unísono para resolver un problema específico. Las redes neuronales actuales se basan en el modelo

matemático de neurona propuesto por McCulloch y Pitts en 1943 [McCulloch y Pitts, 1943]. (...) Cada neurona recibe un conjunto de entradas $\{x_1, x_2, \dots, x_D\}$ y devuelve una única salida y . Además, dentro de una RNA existen numerosas conexiones entre las distintas neuronas que la forman. Estas conexiones simulan las conexiones interneuronales del cerebro y, al igual que éstas, pueden establecerse con mayor o menor intensidad. En el caso de las RNA esta intensidad la determinan los *pesos sinápticos* (o simplemente pesos). De este modo, cada entrada x_i de una neurona se encuentra afectada por un peso w_i (Palma y Marín, 2008, p. 649).

Existen Básicamente dos tipos de problemas que se pueden solucionar con una Red Neuronal Artificial: Problemas de clasificación y Problemas de Regresión.

En cuanto a los problemas de clasificación una definición muy precisa explica:

(...) el objetivo consiste en crear un procedimiento mediante el cual un nuevo caso representado por unos atributos observados o *características* que constituyen los datos de entrada a la RNA, se asigne a una de entre un conjunto de clases predefinidas. A la construcción de estos clasificadores también se les conoce como *reconocimiento de patrones* (Palma y Marín, 2008, p. 651).

Para los problemas de regresión, por su parte, se explica que: “(...) son problemas de *ajuste de funciones*. Es decir, se trata de obtener un número en función de los atributos de entrada a la red o, lo que es lo mismo, se trata de encontrar una función continua de ciertas variables” (Palma y Marín, 2008, p. 652).

2.6.3 Aprendizaje o entrenamiento de las RNA

(...) Las RNA reciben unos datos de entrada que se transforman para producir una salida, con el objetivo de clasificarlos o ajustar una función. La justificación teórica para este tipo de aplicaciones es que, suponiendo que la red tenga un número suficiente de neuronas, podrá ajustar cualquier función continua con una cierta precisión con tan sólo escoger los valores adecuados para los parámetros ajustables de dicha red. Estos parámetros son, en general, los pesos sinápticos y son, por tanto, el medio que la red emplea para almacenar su conocimiento sobre el problema a resolver.

Este conocimiento se almacena en la RNA a través de un proceso de aprendizaje o *entrenamiento*, que no es más que la modificación de los parámetros de RNA mediante un procedimiento preestablecido, al objeto de conseguir una mejora en su rendimiento. Como el proceso de aprendizaje humano, el de las RNA está basado el uso de ejemplos que representan al problema. A este conjunto se le conoce como *conjunto de entrenamiento*. Es importante recalcar que el objetivo del aprendizaje no es memorizar

las relaciones entrada/salida que hay en el conjunto de entrenamiento, sino modelar el proceso que ha generado estos datos. Para ello es conveniente que el número y tipo de ejemplos disponibles para el entrenamiento de la red sea suficientemente representativo de la relación que se desea aprender. De este modo, una vez entrenada, la red será capaz de manejar no sólo los datos de entrenamiento, sino nuevos datos distintos a los primeros, sin que por ello se degrade su rendimiento. (Palma y Marín, 2008, p. 649).

Se dispone de dos tipos de entrenamiento (aprendizaje supervisado y no supervisado), y la elección de este depende de cómo se constituya el conjunto de entrenamiento.

Palma y Marín (2008, p.652) señalan que, para el caso del aprendizaje supervisado “se parte de una serie de observaciones o entradas y unas salidas deseadas que la red debería obtener, y el objetivo es aprender la correspondencia entre ambas.” Para ello se tendrá una expresión que representa el conjunto de entrenamiento a pares de la forma: $X = \{(X^n, t^n)\}_{n=1}^N$, donde X es un vector de entradas, t corresponde al vector de salidas deseadas y N será el tamaño del conjunto de entrenamiento.

Por otro lado, para el caso del aprendizaje no supervisado:

(...) En este caso, en el conjunto de entrenamiento los datos de entrada X no vienen acompañados de una *salida deseada*, y el objetivo no es encontrar un mapeo entrada-salida, sino encontrar patrones que definan regularidades subyacentes en X. Además, estos patrones ocurren con distintas frecuencias y la red deberá encontrar lo que ocurre con más generalidad. En algunos casos se tratará de encontrar una estimación de la *distribución de probabilidad* de x, y en otros el objetivo será inferir las clases existentes en las que agrupar los ejemplos de entrada que presenten similitudes. (Palma y Marín, 2008, p. 653).

2.6.4 Clasificación de imágenes por categorías por medio de la extracción de características: bag of features.

Según el trabajo efectuado por Csurka, et al. (2004) la categorización de imágenes visuales es un proceso de asignación de etiquetas para diferentes categorías de imágenes, y estas categorías pueden contener imágenes que representan cualquier objeto. Una bolsa de puntos clave,

corresponde a un histograma del número de apariciones de patrones de imagen particulares en una imagen dada. Las principales ventajas de este método “*bag of features*”, son principalmente su simplicidad, eficiencia computacional y su invarianza a las transformaciones afines, así como la oclusión, la iluminación y las variaciones intraclase.

Csurka, et al. (2004) plantean 4 pasos mínimos necesarios para lograr la clasificación de imágenes por medio de esta técnica:

- Detección y descripción de parches en la imagen.
- Asignación de descriptores de parches a un conjunto de grupos determinados entendidos como vocabulario, con un algoritmo de cuantificación vectorial.
- Construcción de una bolsa (*Bag*) de puntos clave, que lleva cuenta del número de parches asignados a cada grupo.
- Tratar la bolsa de puntos clave (*bag of features*), como un vector de características, y con esto determinar por medio de un clasificador, la categoría o categorías que se asignan a la imagen.

Con estos pasos es posible garantizar la máxima precisión de la clasificación y al mismo tiempo minimizar el esfuerzo computacional. Es de suponer que los descriptores extraídos en el primer paso deben ser invariables a las variaciones que no son importantes en la tarea de clasificación. (Transformaciones de imagen, variaciones de iluminación y oclusiones), pero que deben aportar la suficiente información para poder ser discriminatorio en cada nivel de categoría. El vocabulario que se emplea en el segundo paso debe ser lo suficientemente amplio como para distinguir los cambios relevantes en las partes de la imagen, pero a la vez no ser tan grande como para distinguir variaciones irrelevantes como el ruido (Csurka, et al., 2004).

2.6.5 Extracción de características SIFT

Scale Invariant Feature Transform (SIFT) es un descriptor de imagen para la coincidencia y el reconocimiento basados en imágenes; Este descriptor se utiliza con fines de visión por computadora, como la coincidencia de puntos entre diferentes vistas y el reconocimiento de objetos en la misma vista. Las características de *SIFT* se consideran una forma eficiente de clasificación de imágenes debido a su demostración de utilidad en condiciones reales (Hussein, Naby y Youssif, 2016).

2.6.6 Local Binary Patterns (LBP)

Al igual que las texturas de Haralick, los *LBP* también se utilizan para cuantificar la imagen en función de la textura. Usando *LBP*, se da importancia a los detalles finos en la imagen. La principal diferencia con las texturas de Haralick es que las *LBP* procesan píxeles localmente utilizando el concepto de vecindad. La imagen de entrada en escala de grises se divide en celdas. Para cada píxel en una celda, en función de su vecindad, se calcula un valor *LBP* (decimal) mediante un umbral simple a lo largo de la vecindad. Después de calcular los valores de *LBP* para todos los píxeles en una celda, se calcula un histograma. Este histograma se normaliza y concatena aún más para todas las demás celdas de la imagen. Si el número de puntos a lo largo de la vecindad se elige como 9 y el radio se elige como 3, entonces el tamaño del vector de la característica *LBP* resultante es de 11-d. Se propuso el concepto extendido de patrones uniformes *LBP* para reducir el tamaño del vector de características resultantes para fines computacionales. Se dice que un *LBP* es uniforme, si el patrón binario calculado para un píxel en una celda tiene como máximo dos transiciones 0-1 y 1-0 (Gogul y Kumar, 2017).

2.6.7 Histogram of Oriented Gradients (HOG)

Este es un descriptor global muy usado en la detección de objetos. Emplea el concepto de contar las ocurrencias de orientaciones de gradiente en regiones específicas localizadas en una imagen. El descriptor *HOG* calcula estas ocurrencias de orientaciones de gradiente en una cuadrícula que tiene celdas uniformemente espaciadas con normalización de contraste local superpuesta (Gogul y Kumar, 2017).

2.6.8 Speed Up Robust Features (SURF)

Una de las razones por la cual se prefieren los descriptores *SIFT* y *SURF* es que son invariables para la rotación de la imagen, la escala y los cambios de iluminación. Pero normalmente se prefiere los descriptores *SURF* sobre *SIFT* debido a la longitud del vector. Mientras que la implementación habitual de *SIFT* utiliza un descriptor que consta de 128 valores de punto flotante, *SURF* comprime la longitud de este descriptor a 64 valores de punto flotante.

El detector *SURF* se basa en la matriz de Hesse. Dado un punto $X = (x, y)$ en una imagen I , la matriz de Hesse $H(X, \sigma)$, X en la escala de σ se define de la siguiente manera:

Ecuación 1. Matriz de Hesse.

$$H(X, \sigma) = \begin{pmatrix} L_{xx}(X, \sigma) & L_{xy}(X, \sigma) \\ L_{xy}(X, \sigma) & L_{yy}(X, \sigma) \end{pmatrix}$$

Donde $L_{xx}(X, \sigma)$ es la convolución de la derivada gaussiana de segundo orden $\frac{\partial^2}{\partial x^2} g(\sigma)$ con la imagen I en el punto X , y de manera similar para $L_{xy}(X, \sigma)$ y $L_{yy}(X, \sigma)$.

Los detectores basados en Hess son más estables. Además, una aproximación como la diferencia de Gauss puede disminuir la velocidad a bajo costo. El rango del punto de interés se mantiene en un vecindario de $3 \times 3 \times 3$ para que el punto de interés se puede localizar en la escala y en el espacio de la imagen utilizando la supresión no máxima (Shukla, Mishra y Sharma, 2013).

2.6.9 *Bag of Words (BOW)* (Bolsa de palabras)

El modelo de bolsa de palabras (*BOW*) es una forma de representación de imagen conocida para fines de clasificación, que utiliza un histograma de palabras visuales para una representación de imagen eficiente. Una parte esencial de la representación de imágenes *BOW* es el libro de códigos visual. Es el proceso que utiliza el agrupamiento de k -medidas con una métrica de distancia euclidiana para la cuantificación de vectores. Estos vectores representan las características de una imagen en palabras visuales. El requisito computacional de esta etapa es muy alto y, por lo tanto, se considera la parte más costosa del proceso de modelado de *BOW*, y los rastros de reducción del tiempo de cálculo a menudo conducen a palabras visuales ruidosas (Hussein, Naby y Youssif, 2016).

2.6.10 Random Forest RF

Este es un clasificador de conjunto que emplea varios árboles de decisión que se aplican para la clasificación o regresión. Hay muchas ventajas de *random forest* que incluye la producción de un clasificador altamente preciso y que también puede funcionar en grandes conjuntos de datos de manera eficiente. El algoritmo *random forest* fue desarrollado por Breiman. Pero fue Ho quien lo presentó para la selección de características aleatorias.

Este clasificador de conjunto involucra un grupo de clasificadores estructurados en árbol $\{r(n, \theta_k), k=1, \dots, L\}$ donde θ_k son árboles aleatorios distribuidos independientemente. Aquí cada árbol contribuye con un voto para la clasificación final de la entrada n . Similar al árbol de clasificación y regresión *CART* (por sus siglas en inglés), se emplea el índice de GINI para calcular el resultado final en cada árbol.

Cada árbol aporta un voto a una clase para la característica de entrada n . Las probabilidades de clase se determinan por votación mayoritaria, que se puede escribir de la siguiente forma:

Ecuación 2. Probabilidad de clase en RF.

$$p(n) = \arg \max \left(\frac{1}{L} \sum_{K=1}^L F_{rk(n)=c} \right)$$

La función de decisión $rk(n)$ devuelve la clase de resultado c ,

Ecuación 3. Función de decisión.

$$F_{rk(n)=c} = \begin{cases} 1, & rk(n) = c \\ 0, & \text{para otros} \end{cases}$$

(Uddin et al, 2017)

2.6.11 Vecinos próximos o *Near neighbor (NN)*

(...) Este clasificador presenta diversas ventajas a pesar de ser bastante costoso computacionalmente. Las principales de ellas son su simplicidad y su generalidad. Además, no es un clasificador paramétrico en el sentido de que cada clase no debe modelarse en base al conjunto de muestras, sino que el conjunto de muestra es el modelo mismo de la clase. Es ciertamente costoso porque necesita determinar la distancia de una forma incógnita a todas las muestras de la clase, el *síndrome de la distancia a todos* que generalmente se trata de evitar en el diseño de clasificadores (...) (Palma y Marín, 2008, p. 992).

(...) Existen además técnicas basadas en el diagrama de Voronoi que permiten obtener clasificadores NN sin tener que computar la distancia a las muestras.

El clasificador NN presenta ventajas a priori, principalmente porque permite definir reglas de clasificación sin esquemas de aprendizaje. Sin embargo, esta estructura de clasificador es de naturaleza exhaustiva y, por lo tanto, si el número de muestras de aprendizaje es elevado, el coste computacional en la toma de decisión también lo es (Palma y Marín, 2008, p. 992).

2.6.12 K vecinos más próximos o *K - nearest neighbors (kNN)*

La clasificación según la regla del vecino más próximo puede modificarse en el sentido de que la regla no suministre el prototipo más cercano a la muestra incógnita, sino el conjunto de prototipos más cercanos. Este tipo de regla es la denominada regla de los k -vecinos más próximos (KNN), que suministra los k prototipos más próximos y a continuación, según un criterio de mayoría entre los k resultados es posible obtener la clasificación de la muestra incógnita. En un sentido probabilístico, este procedimiento puede entenderse como un clasificador Bayesiano, o pseudo-Bayesiano, por cuanto la

distribución de clases entre los k vecinos más próximos constituye una estima de las distribuciones de probabilidad.

Esta regla es útil en ciertas situaciones en que las muestras de clase diferentes se encuentran muy próximas. La regla NN suministra resultados más fiables que la kNN sólo si las distancias entre muestras de la misma clase son más pequeñas que las distancias entre muestras de diferentes clases. La principal ventaja que aporta el clasificador kNN es que la decisión se realiza en función de la mayoría de puntos que rodean al punto incógnita, lo que permite un comportamiento más suave, pero al mismo tiempo se puede obtener una mayor tasa de error (Palma y Marín, 2008, p. 992).

2.6.13 Análisis de Componentes Principales o *Principal Component Analysis (PCA)*

El objetivo de *PCA* es presentar la información de los datos originales como la combinación lineal de ciertas variables lineales irrelevantes. Matemáticamente, *PCA* transforma los datos en un nuevo sistema de coordenadas, de modo que la varianza más grande por cualquier proyección de los datos se encuentre en la primera coordenada, la segunda mayor varianza en la segunda coordenada, y así sucesivamente. A cada coordenada se le llama un componente principal.

El análisis de componentes principales es también un procedimiento de reducción de variable. Es útil cuando se ha obtenido datos sobre una serie de variables (posiblemente una gran cantidad de variables) y se cree que se existe cierta redundancia en esas variables. *PCA* permite extraer información de datos dimensionales superiores proyectándolos en una dimensión inferior. *PCA* también se utiliza porque reduce la dimensión del vector de entrada de la red neuronal. Este método genera un nuevo conjunto de variables, llamadas componentes principales. Cada componente principal es una combinación lineal de las variables observadas y óptimamente ponderadas. Con este método se logra que todos los componentes principales sean ortogonales entre sí, por lo que no hay información redundante. Los componentes principales en su conjunto forman una base ortogonal para el espacio de los datos. *PCA* realiza una reducción de dimensionalidad mientras se preserva la mayor cantidad de aleatoriedad posible en los espacios de alta dimensión.

La limitación con *PCA* es que depende de la escala de las variables y no siempre es fácil interpretar el componente principal (Ghosh et al, 2015).

2.6.14 Clasificador de Naïve Bayes o *Naive Bayes classifier NBC*

Un *NBC* es un clasificador probabilístico simple que se basa en la aplicación del teorema de Bayes (estadísticas bayesianas) con suposiciones de independencia fuertes (naive). Un término más descriptivo para el modelo de probabilidad subyacente sería “modelo de característica independiente”. En términos simples, un *NBC* supone que la presencia de (o ausencia) de una característica particular de la clase no está relacionada con la presencia (o ausencia) de cualquier otra característica. Un ejemplo de esto es que una fruta puede considerarse manzana si es roja, y tiene un diámetro de aproximadamente cuatro pulgadas. Incluso si estas características dependen unas de otras o de la existencia de otras características, un *NBC* considera que todas estas propiedades contribuyen independientemente a la probabilidad de que esta fruta sea una manzana.

Dependiendo de la naturaleza precisa del modelo de probabilidad, el *NBC* puede entrenarse de manera muy eficiente en un entorno de aprendizaje supervisado. En muchas aplicaciones prácticas, la estimación de parámetros para modelos Naive Bayes (NB) usan el método de máxima verosimilitud; en otras palabras, es posible trabajar con el modelo NB sin creer en la probabilidad bayesiana o sin usar algún método bayesiano (Ghosh et al, 2015).

2.6.15 Clasificación Bayesiana o *Bayesian classification*.

Este tipo de clasificador es estadístico. Son capaces de predecir las probabilidades de pertenencia a una clase, como la probabilidad de que un elemento de datos dado pertenezca a una etiqueta de clase particular. La clasificación bayesiana se basa en el teorema de Bayes tal y como

se mostrará: sea X una muestra de datos cuya etiqueta de clase no se conoce y sea H una hipótesis, de modo que los datos X pertenezcan a una clase específica:

Ecuación 4. Teorema de Bayes.

$$P(H|X) = \frac{P(X|H) \cdot P(H)}{P(X)}$$

Una ventaja de Naive Bayes es que solo requiere una pequeña cantidad de datos de entrenamiento para estimar los parámetros (medias y variaciones de las variables) necesarios para la clasificación. Debido a que se suponen variables independientes, sólo deben determinarse las variaciones de las variables para cada clase y no la matriz de covarianza completa (Ghosh et al, 2015).

2.6.16 Máquinas de soporte vectorial o *Support vector machines*

Desde el punto de vista del trabajo efectuado por Ghosh, et al. (2015) *SVM* es un clasificador no lineal. La idea detrás de este método es mapear de forma no lineal los datos de entrada a un espacio de alta dimensión, donde los datos se pueden separar linealmente, lo que proporciona un gran rendimiento de clasificación. *SVM* es una herramienta de aprendizaje de máquina y se ha convertido en una técnica poderosa para aprender de los datos y, en particular, para resolver problemas de clasificación binaria. Los conceptos principales de *SVM* son transformar primero los datos de entrada en un espacio dimensional superior por medio de una función de núcleo y luego construir un hiperplano separado óptimo *OSH* (en inglés), entre las dos clases en el espacio transformado.

SVM encuentra el *OSH* maximizando el margen entre las clases. Los vectores de datos más cercanos a la línea construida en el espacio transformado se denominan vectores de soporte. El *SVM* estima una función para clasificar datos en dos clases. Usando una transformación no lineal que depende de un parámetro de regularización, los vectores de entrada se colocan en un espacio

de características de alta dimensión, donde se emplea una separación lineal. Para construir un clasificador de vector de soporte no lineal, el producto interno (x, y) se reemplaza por una función de núcleo $K(x, y)$.

Se asume que los sujetos normales fueron etiquetados como -1 y otros como +1. El *SVM* tiene dos capas durante el proceso de aprendizaje, la primera capa selecciona la base $K(x_i, x)$, $i=1, 2 \dots n$ de los núcleos establecidos, mientras que la segunda capa construye una función lineal en el espacio. Esto es el equivalente a encontrar el hiperplano óptimo en el espacio de características correspondiente. El algoritmo *SVM* puede construir una variedad de máquinas de aprendizaje utilizando diferentes funciones de núcleo (Ghosh et al, 2015).

2.6.17 Redes Neuronales Convolucionales o *Convolutional neural networks CNN*.

Las redes neuronales convolucionales *CNN* son clasificadores vanguardia que han impulsado muchos avances recientes en la clasificación de imágenes. Los *CNN* tienen una arquitectura multicapa de operaciones convolucionales intercaladas con funciones de activación no lineales y operaciones de agrupación que les permiten aprender representaciones no lineales para la clasificación de imágenes. Los parámetros de las *CNN* pueden aprenderse automáticamente de extremo a extremo con datos suficientes.

Si bien el aprendizaje automatizado de la representación de datos es una característica atractiva, la capacitación de *CNN* desde cero generalmente requiere de grandes conjuntos de datos que pueden no estar disponibles. Una forma práctica de construir clasificadores de imágenes basados en *CNN* utilizando conjuntos de datos de tamaño moderado es mediante la técnica de transfer learning donde las características aprendidas utilizando grandes conjuntos de datos en un dominio relacionado se aprovechan para la tarea en cuestión (Ravindran et al., 2018).

En consecuencia, y en relación con los resultados que se obtienen de las primeras pruebas realizadas con el software Matlab es evidente que *SVM* representan una buena elección para trabajos de clasificación. Por lo tanto, es necesario realizar pruebas e implementar clasificadores con *SVM* de los diferentes tipos de este clasificador, pues Matlab ofrece la posibilidad de elegir entre seis tipos distintos del método. Estos son: *Linear SVM*, *Quadratic SVM*, *Cubic SVM*, *Fine Gaussian SVM*, *Medium Gaussian SVM* y *Coarse Gaussian SVM*.

Se expone a continuación lo relacionado con la industria florícola del país y los principales aspectos, terminologías y temas de importancia en términos de calidad y producción de este producto.

2.7 Floricultura.

Esta es una actividad que consiste en cultivar plantas para obtener flores y posteriormente ser comercializadas. Los usos que se dan a los productos finales van desde la ornamentación, industria o la medicina (Cárdenas y Rodríguez, 2011).

En el caso especial de Colombia este es un proceso que tiene gran importancia en el sector industrial y agrícola, generando significativos ingresos al país.

Las actividades florícolas en Colombia se inician a mediados de la década de los 60s, cuando los costos y condiciones de producción del sector permitieron encontrar elementos altamente competitivos respecto a otros actores del comercio mundial.

Las principales áreas de cultivo como la sabana de Bogotá y la zona de Rionegro en Antioquia, representaban tierras fértiles con temperatura adecuada (de 13°C a 21°C), uniformidad entre las horas de luz y sombra y ausencia de estaciones, que permitían obtener hasta 3.2 cosechas por año. Adicionalmente la relación de costos de la tierra por hectárea alrededor de Bogotá, respecto a zonas floricultoras de Estados Unidos, era de 1 a 9 y el valor de envío de flores a Miami era bajo (el costo de envío de un paquete de claveles de Bogotá a Miami en 1969 era de US \$0.08 en tanto que el precio final de venta era de US \$1,05 por paquete) (...).

(...) Así, la floricultura del país fue convirtiéndose en una actividad destacada dentro del sector agropecuario colombiano caracterizándose por el uso intensivo de recursos, el aprovechamiento de la tecnología y la mayor optimización posible del espacio, conservando una gran incidencia social puesto que es la actividad agrícola con más mano de obra trabajando por hectárea (En la actividad cafetera, el otro gran generador de exportaciones agrícolas del país, por cada hectárea

sembrada son requeridos 0.6 hombres, mientras que en la floricultura son requeridos 16 hombres para la misma hectárea).

Hoy Colombia es el mayor exportador de flores de corte en América y el segundo después de Holanda, llegando a lograr que dos de cada tres flores vendidas en Estados Unidos, el 60% de los claveles, el 20% de los pompones, el 8% de los crisantemos y el 4% de las rosas comercializadas a nivel mundial sean colombianas (Núcleo Ambiental S.A.S., 2015, p.10).

2.7.1 Especies de flores más vendidas

Alemania es el mayor consumidor de flores a nivel mundial, lo que lo convierte en un mercado objetivo importante para los países productores. En la actualidad compra el mayor porcentaje de flores a Holanda. En el 2005, el nivel de importaciones de éste país fue de más de US\$ 976 millones, siendo las flores preferidas las Rosas (37%), Tulipán (10%), Crisantemo (5%) y Gérbera (5%).

Las rosas y los claveles son los principales productos florícolas comercializados a nivel mundial. En 1995, último año para el se dispone información completa, la importación mundial de rosas alcanzó US\$ 688 millones, que representan el 18% del valor mundial importado de flores para el mismo año. Por su parte los claveles participaron en un 12%, seguido por crisantemos 7,5% y orquídeas 1,3%. Sin embargo, las preferencias de las especies varían según sea su mercado de destino (SUPERINTENDENCIA DE SOCIEDADES, 2013, citado por Núcleo Ambiental S.A.S., 2015, p.12).

2.7.2 Terminología en floricultura

- **Clasificación**

Es un proceso que se realiza en la zona de postcosecha y dependiendo la empresa se realiza por colores, tamaños, longitud y grosor del tallo (Marín y Rangel, 2000).

- **Corte**

Es un proceso que lleva a cabo en el invernadero, el cual se realiza de forma manual y bajo ciertos parámetros de desinfección de tijeras y en cierta zona de corte en el tallo (Marín y Rangel, 2000).

- **Cosecha:**

La cosecha es una actividad que por lo general se realiza a mano empleando herramientas como cuchillos o tijeras debidamente afiladas y desinfectadas.

- **Empaque de las flores**

Esta una labor que puede variar dependiendo la empresa o finca productora, es un trabajo generalmente realizado por hombres y de forma manual. En este momento se consideran varios aspectos relacionados con el patrón estándar que hace referencia a los grados de las flores tales como (tamaño, la longitud y la forma de la flor). A estos grados generalmente se les llama de la siguiente forma: Select, son los tallos de las flores que son mayores a 60 cm de longitud, estos suelen ser fuertes y resistentes. Fancy, son los tallos de las flores que miden entre 50-59 cm de longitud, el tallo suele ser un poco más delgado. Standard, son los tallos de las flores que miden entre 40 – 49 cm de longitud, el tallo es por lo general demasiado débil y delgado (Marín y Rangel, 2000).

- **Follajes**

Los follajes son especies vegetales que se cultivan con el fin de que este producto sirva de acompañante de las flores de arreglo.

- **Postcosecha**

Corresponde a las actividades de selección, clasificación, armado de ramos, tratamiento sanitario, empaque y el almacenamiento a los cuartos fríos para la conservación de las flores.

2.7.3 Enfermedades en la floricultura

- **Mildeo veloso (peronospora sparsa):**

Es considerada una de las enfermedades más limitante en cultivos bajo invernadero, ya que está presente en todo el ciclo del cultivo y bajo condiciones ideales puede dispersarse tan rápidamente que en una sola semana puede afectar gravemente la productividad del cultivo ocasionando importantes pérdidas económicas. (Bayer CropScience, s.f.)

Esta es una enfermedad que ha tomado mayor fuerza en los cultivos de rosa y limonum principalmente, es además un hongo que necesita de un huésped vivo para poder prosperar y

completar su ciclo de vida y se sabe que es mayor su incidencia en las fechas entre abril y junio y entre octubre y diciembre.

La sintomatología que presentan las flores se evidencia en los tallos, cáliz y pedúnculos en forma de manchas de color púrpura a negro con secreciones de color blanco, se pueden presentar también manchas irregulares rojizas a café oscuro junto con amarillamiento de las hojas jóvenes (figura 2.2), que pueden generar la momificación del botón floral. También es posible encontrar la enfermedad en el envés de las hojas. Por otro lado, los síntomas pueden ser similares a quemazones de toxicidad debido a productos químicos. (Bayer CropScience, s.f.)

Figura 2.2. Hojas de rosa con enfermedad vellosa.



Fuente. Elaboración Propia.

- **Oídio (*sphaerotheca pannosa*):**

Algunos síntomas como manchas blancas y pulverulentas se manifiestan sobre los tejidos tiernos tales como: brotes, hojas, botones de la flor y base de las espinas. Por otro lado, las hojas también se pueden deformar apareciendo retorcidas o curvadas (Núcleo Ambiental S.A.S., 2015).

- **Roya (*phragmidium disciflorum*):**

Esta es una enfermedad que se caracteriza por la producir la aparición de manchas de color naranja en el envés de las hojas. Por lo general suele aparecer en zonas donde hay concentración de humedad. En el manual “Flores y Follajes” de Núcleo Ambiental S.A.S. (2015, p.26), se

señala que “Una fertilización nitrogenada excesiva favorece la aparición de la roya. Por el contrario, la sequía estival y la fertilización potásica frenan su desarrollo”.

- **Moho gris o botrytis (Botrytis cinérea):**

El desarrollo de esta enfermedad se ve favorecido por las bajas temperaturas y elevada humedad relativa, que origina la aparición de un crecimiento fúngico gris sobre cualquier zona de crecimiento, tales como: flores. Asimismo, es recomendable cuidar de no ocasionar heridas durante las operaciones de poda, ya que son lugares fácilmente conquistados por el patógeno (Núcleo Ambiental S.A.S., 2015).

- **Agallas o tumores (agrobacterium tumofaciens):**

Las agallas o tumores producidos por esta enfermedad aparecen generalmente en el tallo hasta una altura de 50 cm sobre el nivel del suelo o en otras ocasiones en las raíces, penetrando por las heridas de la planta cuando ésta se desarrolla sobre un suelo infectado (Núcleo Ambiental S.A.S., 2015).

2.7.4 Fisiopatías en las rosas

Las fisiopatías o desordenes abióticos en las plantas son enfermedades fisiológicas que no son infecciosas y que no se transmiten por contagio. Estas son deformaciones que se manifiestan en frutos, tallos y hojas causados principalmente por el estrés causado en la planta durante su crecimiento y desarrollo, puede ser debido a un ambiente mecánicamente desfavorable (Zamora, 2016).

- **Distorsión de la hoja:**

Durante el periodo de crecimiento de la hoja que suele llevar entre 10 a 12 días, es durante este periodo que el tamaño embrionario de la yema, como también las aspersiones suelen dañar

la hoja y detienen el normal desarrollo de esta (Perez, 2002). También, la exposición prolongada a rayos ultravioleta suele quemar la hoja.

- **Tallo torcido:**

Es una malformación ocasionada por factores externos que alteran el normal desarrollo del tallo, ya sea por elementos que obligan a la desviación de tallo propios del cultivo como estacas, alambrado y plásticos. Un ejemplo, es el cuello de ganso el cual se produce en la parte de la rosa.

- **Cuello de Ganso:**

Esta es una deformación muy común en los cultivos de rosa y es debida principalmente a elementos que alteran el normal crecimiento de la rosa y afecta el tallo de la rosa cerca al botón floral provocando torceduras notorias en la planta.

Figura 2.3. Tallo de rosa con deformación cuello de ganso.



Fuente. Elaboración Propia.

- **Cabeza de toro:**

Es una enfermedad que provoca una distorsión de las flores y se caracteriza por presentar pétalos más cortos de lo normal y es frecuente que flores con esta deformidad aparezcan sobre

tallos muy vigorosos. También es común encontrar flores distorsionadas debido a los ácaros conocidos como thrips pues muchas veces se culpa a estos por estas deformidades (Perez, 2002).

CAPÍTULO 3

3 Procedimiento Metodológico

3.1 Descripción general del sistema propuesto

Para tener un mayor control en el proceso de calidad y cantidad de producción el sistema debe contar con dos secciones principales:

1. Sección uno: Las rosas entran en una banda que transporta las rosas una a una, esta banda finaliza con un brazo que toma la flor y la introduce en un ambiente controlado con condiciones de luz y fondo constantes. El brazo se encarga de mantener la rosa a una distancia que permita obtener fotografías que abarque los 25cm superiores de la rosa incluyendo la cabeza. Al mismo tiempo, el brazo gira la rosa para obtener fotos de varias vistas. Se determina que sean dieciocho fotos, es decir, se tomarán fotos cada 20°. En cada ocasión el brazo espera unos segundos para la toma de la foto y así sucesivamente hasta completar la rotación completa. Adicionalmente, se tomarán cuatro fotografías en tamaño completo para identificar el tamaño total. Dependiendo de los resultados que genere el programa, el brazo ubica la rosa en el lugar correspondiente de acuerdo con los parámetros de clasificación, que son: Rosa exportación, Rosa nacional y otras.
2. Sección dos: Consiste en un programa que tiene como entrada 22 imágenes para ser analizadas por un algoritmo que emplea técnicas de visión artificial, y más específicamente aprendizaje de máquina. El programa se desarrolla en el software Matlab, el cual lleva cuenta de la cantidad y el porcentaje de cada criterio de acuerdo con el total procesado.

Para este trabajo solo se plantea la elaboración de la segunda sección. Con el fin de delimitar el proyecto, los criterios de clasificación para tener en cuenta son:

- Color (únicamente se tendrá en cuenta la flor de color rojo)
- longitud del tallo en 3 grupos (Marín y Rangel, 2000)
 - Tallo Select (superior a 60 cm)
 - Tallo Fancy (entre 50 – 59 cm)
 - Tallo Estándar (entre 40 – 49 cm)
- tamaño del botón: ancho y alto
 - Botón Altura Pequeño (entre 4 – 4.9 cm)
 - Botón Altura Mediano (entre 5 – 5.9 cm)
 - Botón Altura Grande (entre 6 – 6.9 cm)
 - Botón Anchura Pequeño (entre 4 – 4.9 cm)
 - Botón Anchura Mediano (entre 5 – 5.9 cm)
 - Botón Anchura Grande (entre 6 – 6.9 cm)

Además, se tendrán en cuenta las características asociadas a enfermedades o malformaciones:

- mildiu vellosa o tizón (*Peronospora sparsa*)
- rectitud del tallo (cuello de ganso)

Finalmente, la interfaz del programa deberá mostrar una gráfica por cada categoría así:

- Grupo Rosas Nacionales: donde se lleva un conteo de la cantidad de rosas y el porcentaje correspondiente a aquellas que son descartadas por presentar la enfermedad mildiu vellosa, el tallo deforme o tamaños de cabezas muy pequeñas o demasiado grandes.

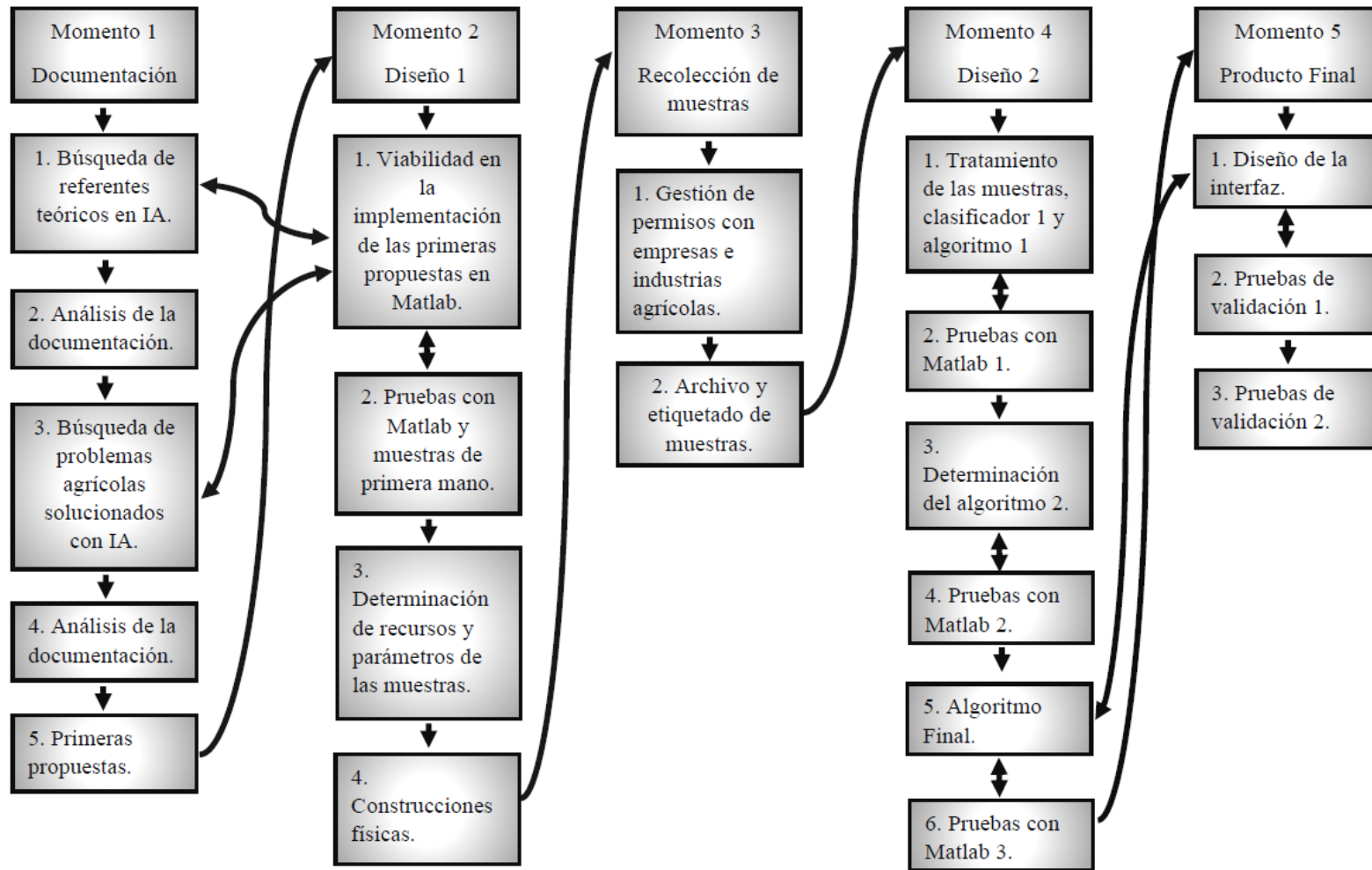
- Grupo tipo exportación: Especificará cuántas y que porcentaje del producto procesado presentan tamaño del botón pequeño, mediano y grande. Qué cantidad y que porcentaje presentan medidas de tallo corto, mediano y largo.
- Grupo Otras: Presentará cuántas no son de color rojo, más no se garantizará su calidad.

Se pretende determinar qué tan eficiente es el programa en términos de precisión de detección de los parámetros establecidos y el tiempo que le toma hacer dicha clasificación frente a las distintas características que pueden presentar las rosas al momento de ser inspeccionadas.

3.2 Metodología.

La metodología planteada para el desarrollo del proyecto consistió en un modelo diseñado por el autor, el cual consta de 5 momentos principales y en cada uno de estos se trabajan determinadas actividades como se ilustra en la figura 3.1.

Figura 3.1. Modelo metodológico.



Fuente. Elaboración Propia.

CAPÍTULO 4

4 Desarrollo y Resultados

A continuación, se detalla el desarrollo de cada uno de los momentos de acuerdo con las actividades realizadas y los diferentes resultados que se obtienen a medida que se llevan a cabo cada uno de los momentos.

4.1.1 Momento 1

En este momento inicial del proyecto se realiza una búsqueda de los principales referentes teóricos y conceptuales que han trabajado el tema de la IA.

Se realizó además una búsqueda de trabajos que abordarán el tema de la IA para solucionar problemas de la agricultura y más específicamente en flores, reconocimiento de especies y clasificación de estas.

En esta búsqueda se encontró principalmente algunas bases de datos con imágenes de flores que están disponibles online y son de acceso libre. Se revisó este aporte para determinar si era posible trabajar con estas imágenes, sin embargo, se encontró que esta base de datos contiene imágenes que corresponden a varias especies de flores, ya que la finalidad de estas bases de datos es el ser útiles para clasificar las flores por especies. Por lo tanto, no fueron útiles a lo que refiere a este proyecto ya que la intención es evaluar la calidad en cuanto a presencia de enfermedades en una única especie de flor (rosa) la cuál es muy empleada en la industria.

Al no encontrar bases de datos gratuitas con ejemplos de rosas enfermas y sanas, se determina tomar muestras propias, etiquetarlas y garantizar un ambiente que permita centrar la atención únicamente en las características de la rosa.

4.1.2 Momento 2

Para este momento se evaluaron las primeras propuestas en términos de técnicas de procesado de imágenes y clasificación de imágenes usando el software Matlab 2017b.

El software Matlab ofrece un gran soporte para la parte de tratamiento y clasificación de imágenes, ya que contiene una amplia variedad de comandos, algoritmos predefinidos y toolboxes que permiten realizar pruebas de forma sencilla, estos comandos evitan la tarea de reescribir líneas de código extensas debido a que estas se encuentran resumidas en funciones que piden en muchos casos sólo un par de valores de entrada.

Por otro lado, es un lenguaje sencillo con una estructura similar a C++ que permite la creación de funciones, para que estas puedan ser reutilizadas en cualquier momento las veces que se desee.

Se realizaron pruebas usando las bases de datos gratuitas que se encuentran online para uso en tareas de clasificación y entrenamiento de sistemas de *Machine Learning*. Esto con el fin de determinar inicialmente, con las herramientas que ofrece Matlab cual es la estrategia más eficiente. Se utiliza la herramienta *classification learner* de Matlab y así evaluar su desempeño y confiabilidad. Para el entrenamiento y pruebas se usa la base de datos Oxford-17, la cual se usó en uno de los documentos encontrados. Esta base de datos contiene imágenes de flores de 17 especies, pero estas imágenes no son de flores enfermas o de tipo producción. Se decide realizar las pruebas con flores del mismo color, pero de diferentes especies, y así determinar qué tan eficiente es un clasificador entrenado por Matlab en el caso de flores del mismo color, pero con características específicas diferentes, pues esto se asemeja a la clasificación de rosas rojas, pero con características en tallo y hojas específicas.

Las flores escogidas son: Botón de Oro, Girasol, Narciso y Uña de caballo, todas amarillas. Para estas categorías se emplea el método de extracción de características *Bag of features* la cual es una función de Matlab basado en *Bag of Keypoints* propuesto por Csurka, et al. (2004). El clasificador que logra la mayor precisión de clasificación es 75% usando *SVM Medium Gaussian*, Seguido por 74.4% obtenido por *SVM Linear* y *SVM quadratic* quienes obtienen la misma precisión de clasificación. Otros clasificadores como *SVM cubic* y *SVM coarse Gaussian* obtienen 73.1% y 67.5% respectivamente (figura 4.1). Estos resultados se obtienen con la pre-validación realizada por Matlab, ya que al principio de entrenamiento permite seleccionar el porcentaje de datos que serán usados en el entrenamiento y el porcentaje de datos que no entra en el entrenamiento pero que se usará en validación. Para este caso se seleccionó lo máximo permitido por el programa 50% para entrenamiento y 50% para validación.

Figura 4.1. Resultados obtenidos en la aplicación classification learner con diferentes clasificadores a las categorías con flores de especies amarillas.

1	☆ Tree	Accuracy: 55.0%
	Last change: Disabled...	200/200 features
2	☆ Tree	Accuracy: 55.0%
	Last change: Medium ...	200/200 features
3	☆ Tree	Accuracy: 53.8%
	Last change: Coarse T...	200/200 features
4	☆ Linear Discrimi...	Accuracy: 48.1%
	Last change: Linear Di...	200/200 features
5	☆ Quadratic Discriminant	Failed
	Last change: Quadrati...	200/200 features
6	☆ SVM	Accuracy: 74.4%
	Last change: Linear S...	200/200 features
7	☆ SVM	Accuracy: 74.4%
	Last change: Quadrati...	200/200 features
8	☆ SVM	Accuracy: 73.1%
	Last change: Cubic SVM	200/200 features
9	☆ SVM	Accuracy: 30.6%
	Last change: Fine Gau...	200/200 features
10	☆ SVM	Accuracy: 75.0%
	Last change: Medium ...	200/200 features
11	☆ SVM	Accuracy: 67.5%
	Last change: Coarse ...	200/200 features

Fuente. Elaboración Propia.

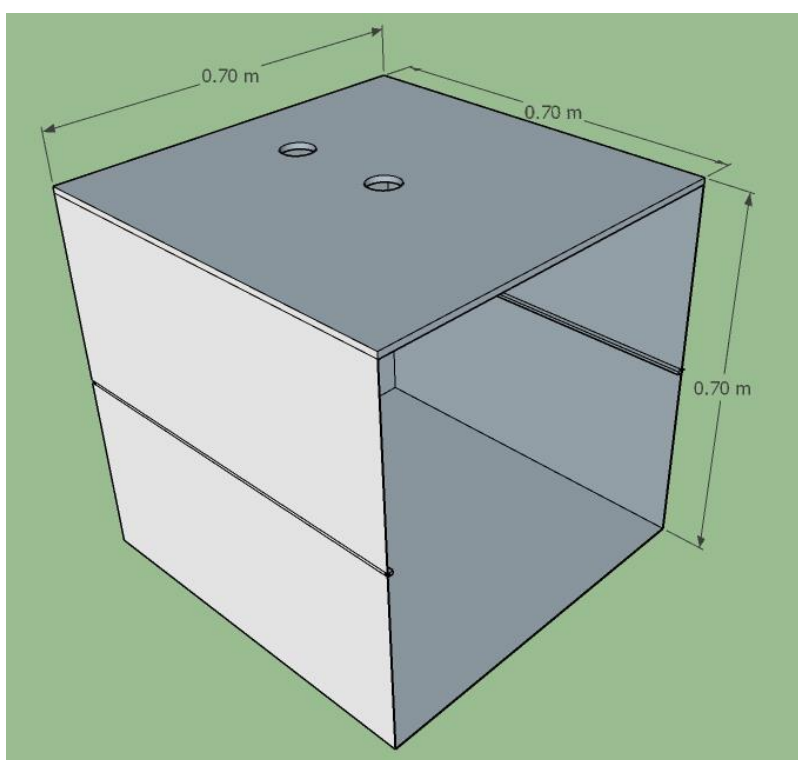
Es claro que la herramienta *Classification Learner* de Matlab ofrece aproximaciones suficientes para implementar un clasificador. Cabe mencionar que el porcentaje obtenido pudo ser mayor, pero teniendo en cuenta que la base de datos usada suministra imágenes de flores con fondo complejo además de variaciones en luz, escala y tamaño, los resultados no dejan de ser satisfactorios.

Es conveniente entonces, que la escena en donde se encuentra la rosa para este caso sea limitada. Para minimizar al máximo el coste computacional y mejorar la aproximación del clasificador ya que, si bien existen técnicas para remover el fondo complejo en una imagen por medio de procesamiento de imágenes, tales como segmentación o incluso técnicas de visión artificial, no es un tema que se trabaje en este proyecto. Para este proyecto se tienen dos aspectos importantes. El primero como ya se mencionó relacionado con el fondo complejo, ya que las muestras pueden tomarse en el ambiente del cultivo bajo el invernadero o sobre una mesa de trabajo, pero como se ha venido indicando no es propiamente de interés en el desarrollo de este proyecto, el cual se centra específicamente en las características particulares de cada rosa en cuanto a calidad como: la forma del tallo o presencia irregular de machas debidas a la enfermedad vellosa. Para el fondo se trabajará con un fondo de color sólido que contraste con los colores de la rosa. El segundo está relacionado con la intensidad lumínica, ya que cuando el sistema esté realizando las tareas de clasificación, la iluminación ambiente cambiará dependiendo la hora. Por lo tanto, se propuso crear un ambiente con condiciones de luz relativamente constantes. Este ambiente estará encerrado, con el fin de impedir que las condiciones lumínicas naturales interfieran con la adquisición de fotografías de entrada al sistema. Aun así, es posible que estas condiciones lumínicas sean variables, y esto puede ser debido a la potencia que quizá alguna otra máquina o elemento eléctrico consuma o deje de

consumir de la red eléctrica y esto puede que afecte en algo las condiciones al interior del ambiente.

Con ayuda del software SketchUp se diseñó un ambiente con forma de cubo y aristas de 70 cm de longitud (figura 4.2), pues esta es la medida máxima de longitud en una rosa como lo indican Marín y Rangel (2000). Este ambiente se construyó en madera y el interior se pintó en color blanco mate. Además, se instaló iluminación led blanca en el interior.

Figura 4.2. Diseño del modelo de ambiente propuesto.

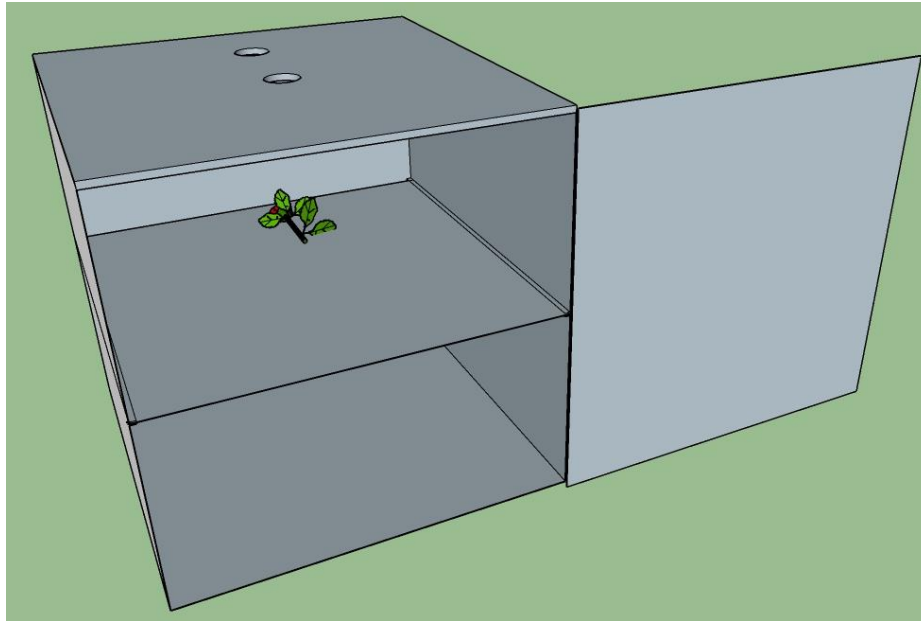


Fuente. Elaboración Propia.

El ambiente necesario para la toma de muestras y realización de pruebas requería de dos agujeros en la parte superior, con diámetro suficiente para el lente de la cámara. Se decidió dividirlo en dos secciones. En la parte superior se desliza una división en madera (figura 4.3), la cual permitiría tomar fotos de los 25 cm superiores de la rosa, pues en esta zona es la que se suele dejar algo de follaje y no debe tener ninguna enfermedad. También, verificar que esta parte del tallo de la rosa no presente la deformidad cuello de ganso. Además, con la información de la

parte superior de la rosa es posible medir el tamaño del botón y así realizar la clasificación por tamaños usando técnicas de pre procesado. La parte inferior, al retirar la división, permitiría que el lente captara la longitud total de la rosa y con esto realizar la clasificación por tamaño de tallos.

Figura 4.3. Diseño del modelo de ambiente propuesto con división y tapa.



Fuente. Elaboración Propia.

Para tener información completa de la rosa, era necesario garantizar una rotación longitudinal de 360° . Se necesitan de 18 fotos los cuales son los parámetros de entrada del sistema. Es decir, que la flor debe girar 20 grados y en cada ocasión se toma una foto de la rosa. Para el efecto de realización de pruebas, esto se logró con un motor paso a paso que garantiza una rotación precisa cada 20 grados hasta lograr el giro completo. Se recomienda que, en la implementación a futuro, sea también un motor paso a paso el que garantice la rotación completa de la rosa.

4.1.3 Momento 3

Se gestionó permiso con tres empresas de flores del sector de las cuales dos accedieron a colaborar con el proyecto.

Se obtuvieron 139 muestras de diferentes flores para efectos de entrenamiento y pruebas de validación primarias que realiza Matlab, de las cuales 59 imágenes se usaron como datos de entrenamiento de rosa en buen estado o categoría (F_Buena). Para datos de entrenamiento de rosa torcida o categoría (F_Torcida) se usaron 30 imágenes y para rosa con enfermedad veloso o categoría (F_Velloso) el algoritmo se entrenó con 50 imágenes.

Adicionalmente, se tomaron 583 fotografías a 24 rosas para efecto de validación final del algoritmo las cuales no entran al entrenamiento de ninguna manera. Un último grupo de 93 imágenes de rosas se usan como material de pruebas en procesamiento de imágenes, tales como medición de longitud del tallo, medición de tamaño de botón y revisión de color.

Para efecto de toma de fotografías de todas las rosas, se llevó a cabo con una cámara Nikon de 20.1 Megapíxeles, o 5.152 x 3.864 píxeles, y usando el ambiente controlado con interior blanco mate e iluminación de 2 paneles de luz led blanca en la parte superior con consumo de 6W de potencia y un flujo luminoso de 270lm.

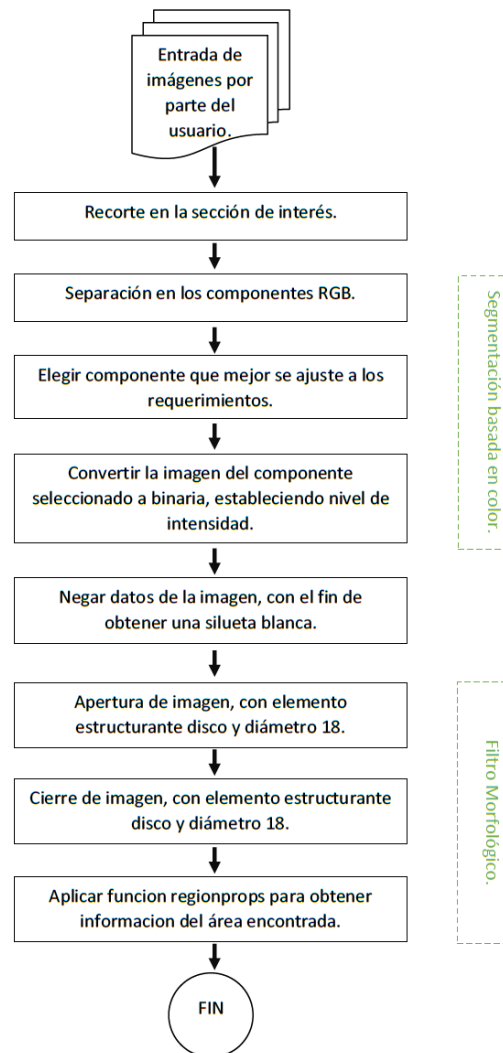
4.1.4 Momento 4

En esta etapa se construyen los diferentes algoritmos que trabajan la parte de pre procesamiento de imagen, clasificación y sus respectivas pruebas. En primer lugar, se trabaja el algoritmo de verificación de color de la rosa, y después el algoritmo de pre procesamiento de imagen. De la etapa de preprocesamiento, se realiza la medición del botón y la posterior clasificación de la flor por tamaños. Además, en este momento se realizan dos tipos de medición, la medición de la cabeza de la flor y la medición de la longitud del tallo.

- Algoritmo 1: Preprocesamiento y clasificación por medidas.

Para lograr el preprocesamiento digital de la imagen se siguió la estrategia que se muestra en el diagrama de flujo de la figura 4.4, esta estrategia es la misma en todos los casos de preprocesamiento digital de imagen para clasificación por medidas mostrados en este documento.

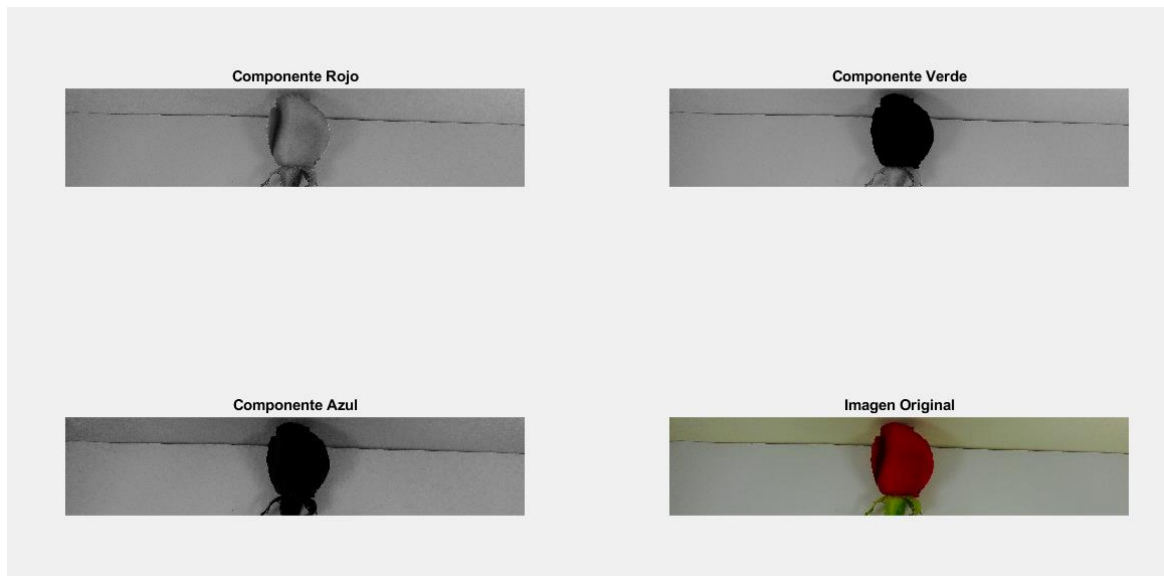
Figura 4.4. Estrategia seguida para preprocesamiento digital de imágenes.



Fuente. Elaboración Propia

Lo primero que realiza el programa es revisar que la rosa que se va a procesar es de color rojo. Para ello se convierte la imagen en escala de grises usando el componente azul de la imagen, pues en este componente se encuentra mayor intensidad en color de los tres componentes, como se observa en la figura 4.5. También se recorta la imagen en el espacio donde es seguro que se encontrará el botón floral (Véase código en el Anexo 1).

Figura 4.5. Separación en los componentes RGB de color.



Fuente. Elaboración Propia.

Luego, se procede a convertir el componente azul a imagen a binaria, con el fin de obtener una silueta en negro de la región de interés. Se convierte en binaria usando un nivel de intensidad específico, con el fin de eliminar componentes que agregan ruido a la imagen, por lo tanto, ciertas intensidades por debajo del umbral establecido tomarán valor igual a 0 y las intensidades con valor por encima del umbral establecido serán 1. Se espera entonces que las secciones en blanco o grises muy claros tomen valores iguales a 1 y las regiones oscuras tomen valor 0 (figura 4.6).

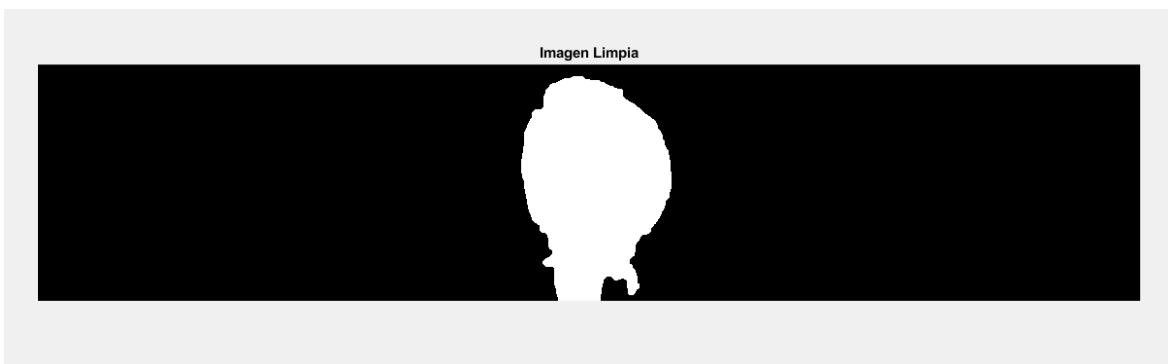
Figura 4.6. Imagen de la componente azul convertida a binaria.



Fuente. Elaboración Propia.

Como la selección de intensidades no resulta ser suficiente para eliminar el ruido en la imagen, se realiza una negación de los datos en la imagen, de manera que ahora la silueta negra será blanca y todo el fondo aparecerá en negro y después se realiza un filtrado morfológico usando los comandos `imopen` y `imclose` en ese orden, con elemento estructurante disco de radio 18 para ambos casos. Se obtiene una imagen limpia y con una silueta claramente definida como se observa en la figura 4.7.

Figura 4.7. Resultado final después del filtro morfológico.



Fuente. Elaboración Propia.

El comando `regionprops` y `bwlabel`, entregan información de las regiones que se identifiquen en la imagen, y dado que la única región que se identifica es la que corresponde al botón floral y parte del tallo, se usa dicha información para obtener las coordenadas de un rectángulo que encierran perfectamente la única silueta encontrada. Este rectángulo se lleva a la imagen original

para que recorte esta área y determinar si el color que más predomina es rojo (figura 4.8). De ser así la función devolverá “Es roja”, para cualquier otro caso, la función devolverá no es roja.

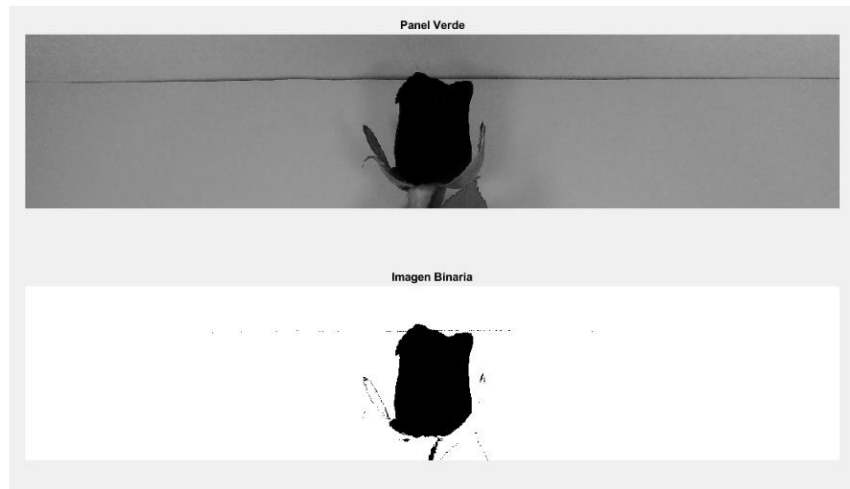
Figura 4.8. Recorte aplicado a la imagen original.



Fuente. Elaboración Propia.

Una vez se ha revisado que la imagen corresponde a una rosa roja, lo siguiente es valorar la calidad con un algoritmo entrenado, en el cual se revisa si la rosa presenta la enfermedad vellosa o si está torcida. Si la rosa no presenta ninguna de estas dos categorías, entonces es medida. El proceso de valoración de calidad en cuanto enfermedad y malformación del tallo pertenece al algoritmo 2 que se tratará más adelante. Para lograr la medición del botón de la rosa se realiza un procedimiento similar al que se realizó para determinar el color, solo que en este caso el componente de color que se utiliza es el verde, ya que permite obtener una mejor silueta del botón como se observa en la figura 4.9. Se determina un umbral de intensidad específico para eliminar la mayor cantidad de componente verde y convertir a binaria.

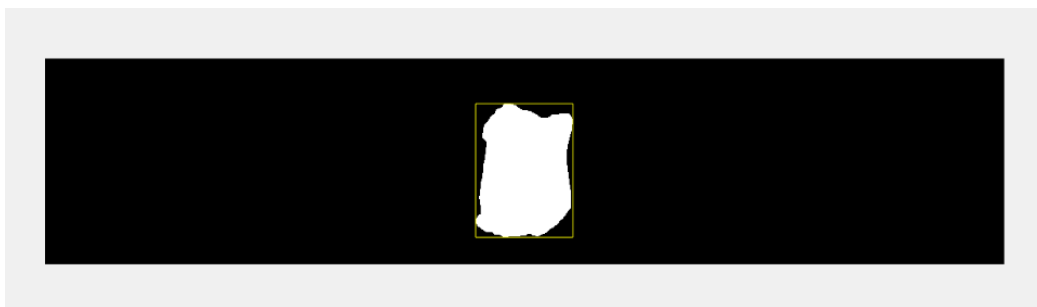
Figura 4.9. Imagen de componente verde convertida en binaria.



Fuente. Elaboración Propia.

De la misma forma que en el procedimiento anterior, lo siguiente es obtener la imagen negativa para que la silueta aparezca blanca y pueda ser encontrada por los comandos `bwlabel` y `regionprops`, los cuales brindarán información sobre las siluetas encontradas (en este caso una), es de interés la información acerca de las dimensiones del rectángulo que encierra perfectamente el botón (figura 4.10), en cuanto altura y anchura.

Figura 4.10. Imagen de componente verde limpia convertida en binaria.



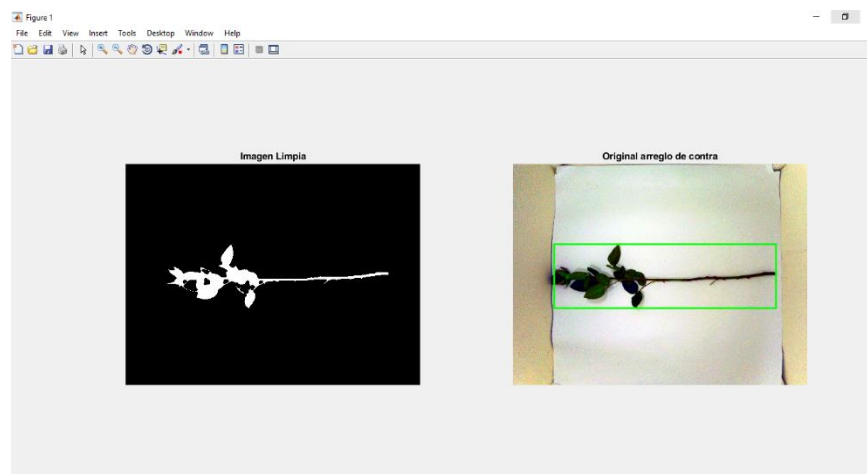
Fuente. Elaboración Propia.

La información sobre las dimensiones de dicho rectángulo está dada en píxeles, por lo que es necesario realizar una conversión de píxeles a milímetros para poder obtener una función que entregue las dimensiones en unidades de medida familiares al usuario. Con las medidas que entrega la función se realiza una clasificación en tres grupos de medida definidos al principio por

el usuario en intervalos de medida en milímetros. Esta operación es realizada la cantidad de veces igual al número de imágenes de la parte superior ingresadas, pues al final se realiza un promedio de las medidas obtenidas para tener un valor más confiable de la medida del botón.

Lo siguiente consiste en obtener la medida longitudinal del tallo, para el cual se realiza un procedimiento similar a los que se han realizado, además, se realiza una resta de imágenes para eliminar la silueta del botón. Dadas las condiciones del ambiente, las zonas en las esquinas presentan sombras de gran tamaño que resulta difícil eliminarlas sin afectar partes del tallo. Por lo tanto, se decide usar las propiedades del comando `regionprops`. Se realiza un filtrado por áreas menores a 300000 píxeles cuadrados, para el cual el tallo más corto y con menos follaje supera por mucho. A aquellas áreas que presentes un área menor a la estipulada, el algoritmo las pinta en negro o, toman un valor igual a 0 (negro). Al final se obtiene un área uniforme blanca que corresponde a la silueta del tallo completa a la cual se le obtiene las dimensiones de un rectángulo que lo encierra perfectamente (figura 4.11), se realiza la conversión de píxeles a centímetros y la función la clasifica en tres posibles categorías: Tallo estándar (40 cm – 49 cm), Tallo Fancy (50 cm – 59 cm), Tallo Select (superior a 60 cm), (Véase código en el Anexo 2).

Figura 4.11. Imagen de componente verde convertida en binaria.

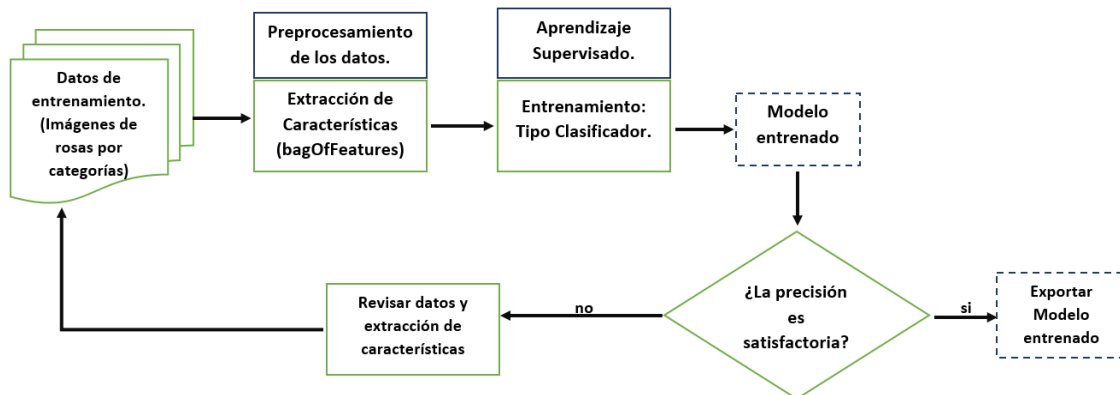


Fuente. Elaboración Propia.

- Algoritmo 2: Verificación de calidad con algoritmo entrenado.

Para el desarrollo de este algoritmo, se necesitó del entrenador de modelos de Matlab “*Classification Learner*”, el cual para entrenar los diferentes modelos incluidos por defecto requiere de información sobre el conjunto de entrenamiento. Para aportar esta información, se acudió a la extracción de características del comando *bagOfFeatures*, basado en selección de puntos clave o *Bag of Keypoints* propuesto por Csurka, et al. (2004). Es necesario repetir el proceso de entrenamiento hasta que se tenga un resultado que cumpla con lo requerido, según la figura 4.12, donde se muestra el proceso de desarrollo del algoritmo (Véase código del entrenamiento en el Anexo 3).

Figura 4.12. Proceso de entrenamiento y selección de clasificador entrenado.

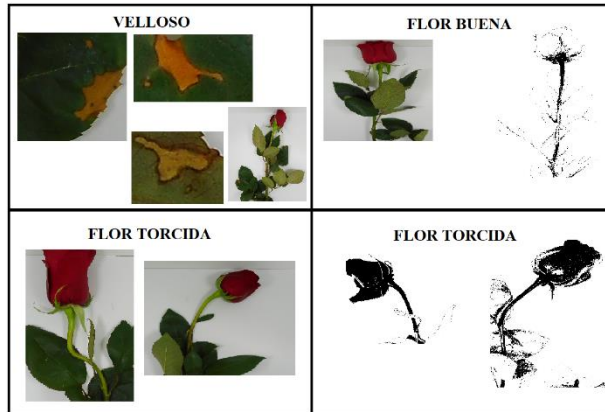


Fuente. Elaboración Propia.

Las imágenes empleadas para el entrenamiento hacen énfasis en las características que hacen que la imagen pertenezca a cada categoría, es decir, que la carpeta de entrenamiento contiene 3 carpetas con imágenes en la que se observan órganos de las flores afectadas con vellosos, en el caso de esa categoría y tallos torcidos para el caso de la categoría flor torcida. Algunas imágenes incluso solamente dan información de la imagen binaria, con el fin aportar información de “como luce” una rosa torcida dejando de lado las intensidades de gris y componentes de color. De manera similar se hace con las rosas buenas donde se observan siluetas de tallos rectos y sin

deformidades. Ejemplo de esto se observa en la figura 4.13 en donde se muestran ejemplos de algunas imágenes pertenecientes a cada categoría.

Figura 4.13. Órganos y flores pertenecientes a cada categoría.



Fuente. Elaboración Propia.

La facilidad para entrenar modelos con la aplicación permite seleccionar el clasificador con los mejores resultados y exportar una estructura que puede ser usada para analizar datos nuevos. Se entrena el modelo con 190 características de cada imagen en tres categorías distintas. La aplicación realiza una validación con el porcentaje de datos que se indique, para este caso se seleccionó el máximo permitido por la aplicación del 50% para validación y 50% para entrenamiento. De los 20 clasificadores entrenados en la aplicación se observa que los mejores clasificadores son: *Linear SVM* y *Quadratic SVM*, con un porcentaje de precisión del 67.1% y 65.8% respectivamente (figura 4.14).

Con estos resultados se decide exportar ambos clasificadores, el programa Matlab entrega una estructura entrenada que recibe como entrada 190 características de cada imagen que se desee analizar. Esta operación se lleva a una función de Matlab que recibe una carpeta con imágenes de la parte superior de las rosas. En un primer momento se extraen 190 características de las imágenes nuevas y luego la estructura entrenada revisa cada una de las características correspondientes a las imágenes y determina cuales de estas imágenes pertenecen a las categorías

“Rosa Buena”, “Rosa con Velloso” y “Rosa Torcida” (Ver código de predictor en el Anexo 4). Si la suma de la cantidad de imágenes pertenecientes a las categorías “Rosa con Velloso” y “Rosa Torcida” es superior o igual al 60%, la función determina que la rosa es para consumo nacional, de lo contrario la función determina que la rosa no presenta enfermedades ni está torcida.

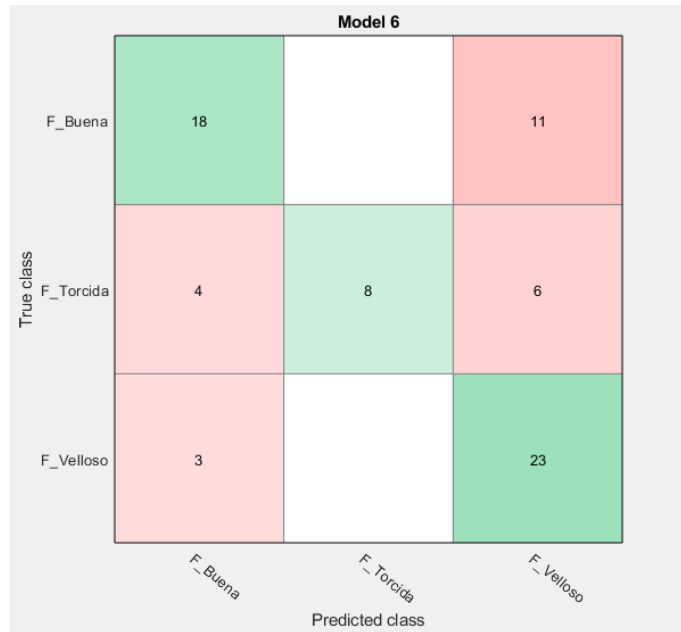
Figura 4.14. Resultados del entrenamiento con validación de Matlab.

Model	Last change	Accuracy
Linear Discriminant	Linear Discriminant	190/190 features
5 Quadratic Discriminant	Quadratic Discriminant	Failed
6 SVM	Linear SVM	Accuracy: 67.1%
7 SVM	Quadratic SVM	Accuracy: 65.8%
8 SVM	Cubic SVM	Accuracy: 58.9%
9 SVM	Fine Gaussian SVM	Accuracy: 39.7%
10 SVM	Medium Gaussian SVM	Accuracy: 64.4%
11 SVM	Coarse Gaussian SVM	Accuracy: 39.7%

Fuente. Elaboración Propia.

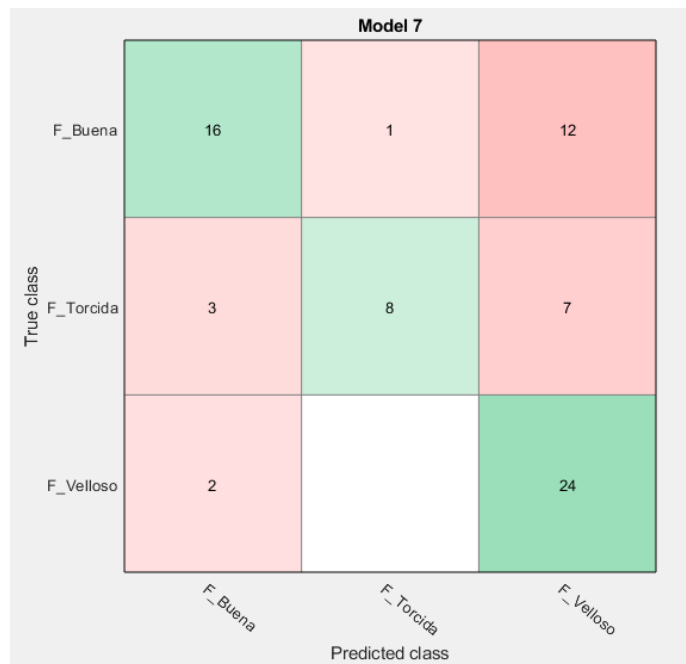
La aplicación *classification Learner* de Matlab, permite visualizar la matriz de confusión de los modelos entrenados, en términos de cantidad de observaciones o porcentajes. En estas gráficas es posible determinar en cuales casos el modelo entrenado ha tenido un desempeño bajo. Las filas muestran la clase verdadera y las columnas muestran la clase que ha sido predicha. Para los modelos escogidos, se muestra en la figura 4.15 y en la figura 4.16, la matriz de confusión de los modelos entrenados *Linear SVM* y *Quadratic SVM* respectivamente, en términos de cantidad de observaciones. En el siguiente capítulo se discutirán y analizarán en detalle estas matrices, por el momento se comenta que es de interés la diagonal principal de la matriz (en verde), ya que muestra las intersecciones óptimas entre clase verdadera y predicción.

Figura 4.15. Matriz de confusión del modelo *Linear SVM*.



Fuente. Elaboración Propia.

Figura 4.16. Matriz de confusión del modelo *Quadratic SVM*.

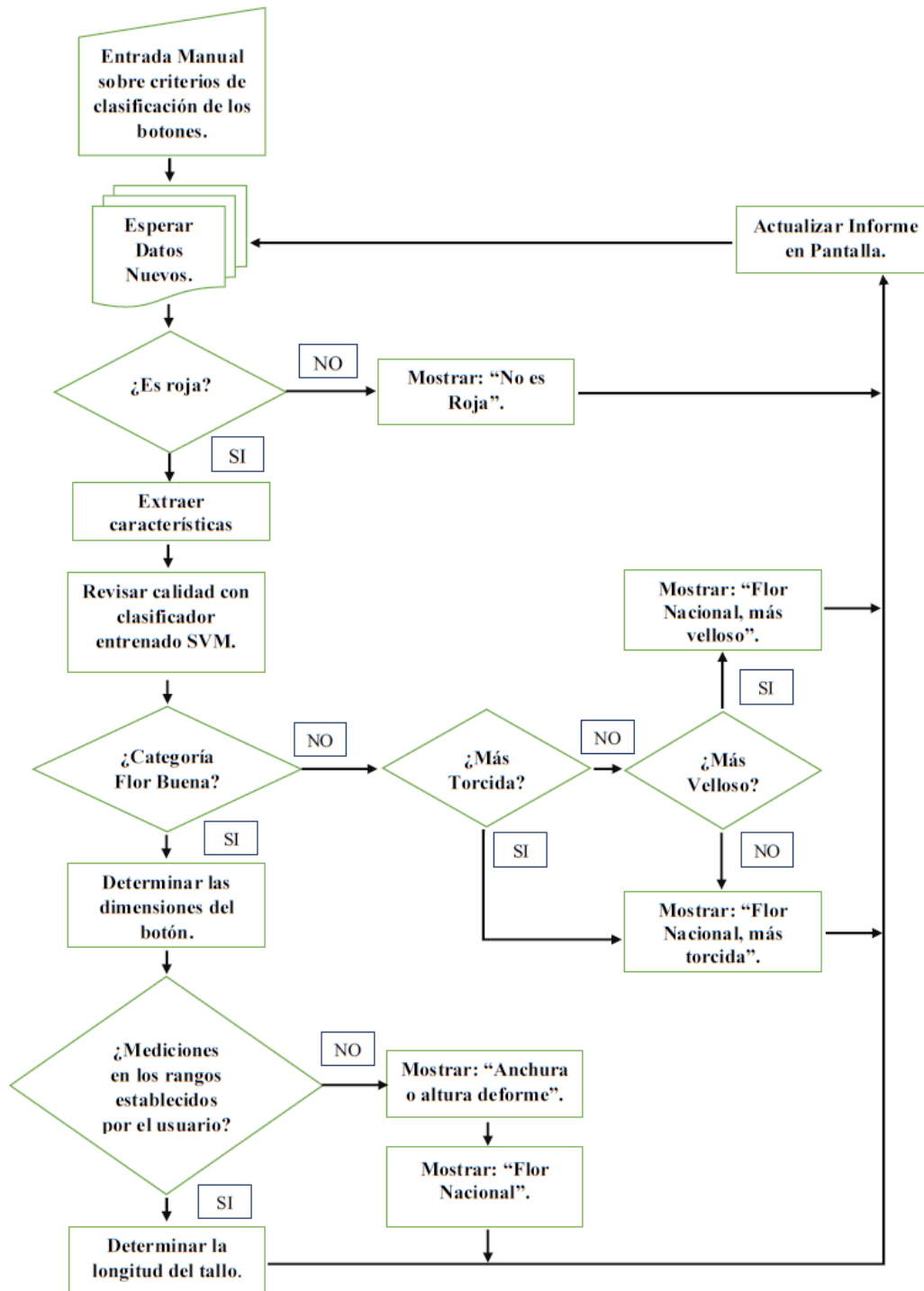


Fuente. Elaboración Propia.

- Algoritmo Final: Diagrama de flujo y unión de las dos etapas.

Este consiste básicamente en la unión de las dos etapas anteriormente desarrolladas, y en la lógica de funcionamiento del algoritmo principal encargado de llamar las respectivas funciones y generar los datos de salida para el usuario. Además, se organiza de forma paralela la interfaz con sus respectivos botones y visualizadores. El programa inicialmente determina si la rosa es de color rojo, de ser así, el algoritmo permite que las imágenes alojadas en carpeta sean examinadas por el algoritmo entrenado, con el fin de determinar si la rosa se encuentra en buena calidad, o si, por el contrario, presenta la enfermedad vellosa o si está torcida. Si el algoritmo entrenado determina que las imágenes correspondientes a determinada flor presentan alguna de los parámetros de baja calidad, entonces la rosa es clasificada en el grupo “rosa para consumo nacional” y consecuentemente no se miden el botón ni la longitud del tallo. Por el contrario, si la rosa cumple con pertenecer a la categoría “Rosa Buena” entonces el algoritmo permite que la rosa sea medida tanto el botón como la longitud del tallo. La última condición está relacionada con las condiciones impuestas por el usuario, ya que, si la rosa no pertenece a ninguno de los intervalos ingresados, entonces la rosa es clasificada como rosa deforme grande o rosa deforme pequeña. Todo lo anterior, se muestra en el diagrama de flujo de la figura 4.17.

Figura 4.17. Diagrama de flujo funcionamiento general del algoritmo.



Fuente. Elaboración Propia.

Esto sugiere que, durante el proceso de revisión, la flor puede superar varias etapas, sin embargo, es necesario que supere la totalidad de las etapas. Puede existir el caso que la rosa sea roja y que la rosa no presente deformidad del tallo ni enfermedad vellosa, pero si el botón es extremadamente grande o demasiado pequeño es una rosa deforme y por ende será clasificada como “FLOR NACIONAL”.

Con el fin de reducir el coste computacional y el tiempo de procesamiento, las imágenes nuevas se convertirán a escala de grises, es decir, que el algoritmo tendrá información únicamente de las intensidades y de las formas presentes en la imagen.

4.1.5 Momento 5

- Diseño de la Interfaz:

El diseño de la interfaz consiste básicamente en dos partes; una primera ventana (figura 4.18), inicia solicitando los nombres de las carpetas que alojan las imágenes de la rosa, el cuadro que solicita “PARTE SUPERIOR” se refiere a imágenes de los 25 cm superiores de la rosa y, el cuadro que solicita “ROSA COMPLETA” se refiere a la carpeta con imágenes de la rosa en su totalidad. Después, el usuario debe elegir qué información le interesa conocer del botón: la altura o la anchura. Si no desea medir el botón puede seleccionar la opción no medir. También se solicita los intervalos de clasificación para los botones florales en tres categorías: pequeño, mediano y grande. Si el usuario selecciona no medir, no es necesario ingresar los datos de medición. Dependiendo del interés del usuario, el programa mostrará “altura grande” o “anchura grande” en el caso de una rosa grande (en ambos casos), recién medida. Al final un botón “Empezar” oculta la ventana de configuración y abre la venta para visualizar el estado actual del análisis (véase Anexo 5, código del botón).

Figura 4.18. Ventana de configuración.

Config_Clas

Configuración

Ingresar Nombres de Carpetas Contenedoras

Parte Superior

Rosa Completa

Seleccionar Criterio de Medición

Medir Altura de Botón

Medir Anchura de Botón

¿No medir?

Establecer Intervalos de Medición

Botón Pequeño entre y mm

Botón Mediano entre y mm

Botón Grande entre y mm

Tallo Estándar (40 cm - 49 cm)

Tallo Fancy (50 cm - 59 cm)

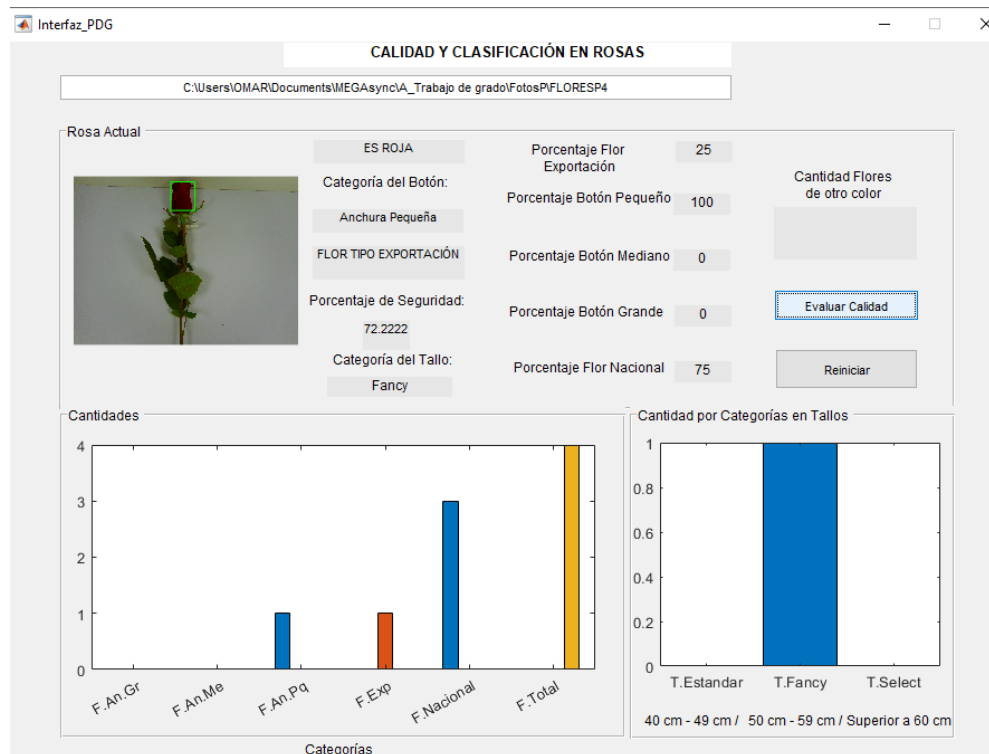
Tallo Select (Superior a 60 cm)

Empezar

Fuente. Elaboración Propia.

La segunda ventana (figura 4.19), muestra información sobre el estado actual del análisis. La ventana se actualiza cada vez que finaliza la revisión de una rosa. La ventana también muestra una imagen de la rosa que se está revisando, esto con el fin de confirmar visualmente que la rosa cambió.

Figura 4.19. Ventana de estado actual del análisis.



Fuente. Elaboración Propia.

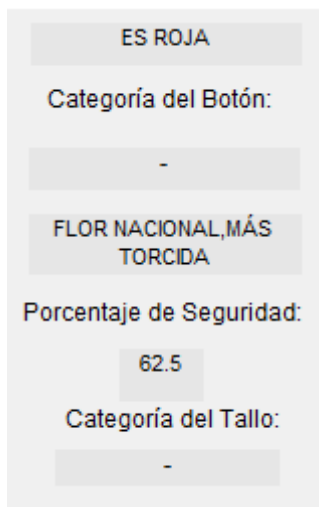
A continuación, se detalla cada uno de los elementos de la ventana:

En un recuadro blanco ubicado en la parte superior de la interfaz, se debe ingresar la ubicación de la carpeta que contiene las imágenes. No olvidar que dentro de dicha carpeta debe haber dos carpetas, una con fotos de la parte superior de la rosa y otra carpeta con imágenes de la rosa en tamaño completo.

Para el caso de la primera columna (figura 4.20), el primer cuadro muestra el mensaje “FLOR NUEVA”, cuando se está revisando una rosa, cuando finaliza de revisar el cuadro muestra “Es Roja” o “No es Roja”, dependiendo el caso. El segundo cuadro, mientras está revisando una nueva rosa muestra un guion y cuando finaliza la revisión muestra la categoría a la cual pertenece el botón floral, “Anchura Pequeña” o “Altura pequeña” dependiendo de la selección del usuario. Cuando la rosa no es medida, es porque la rosa no es tipo exportación, por lo tanto,

el botón no es clasificado en ninguna categoría, y el segundo cuadro seguirá mostrando un guion. El tercer cuadro muestra si la rosa es tipo exportación o para consumo nacional y si la rosa presenta mayor síntoma de vellosa o de tallo torcido. Esta clasificación es realizada por el algoritmo entrenado y está relacionado con la cantidad de imágenes clasificadas en cada categoría. El cuarto recuadro muestra en términos de porcentaje que tan seguro está el programa de que la rosa pertenece a la categoría dada en el cuadro anterior. El último recuadro muestra la categoría a la cual pertenece el tallo, si la rosa es clasificada como “FLOR NACIONAL, MÁS VELLOSO” o “FLOR NACIONAL, MÁS TORCIDO”, entonces el tallo no es medido y por lo tanto no es clasificado en ninguna categoría y seguirá mostrando un guion.

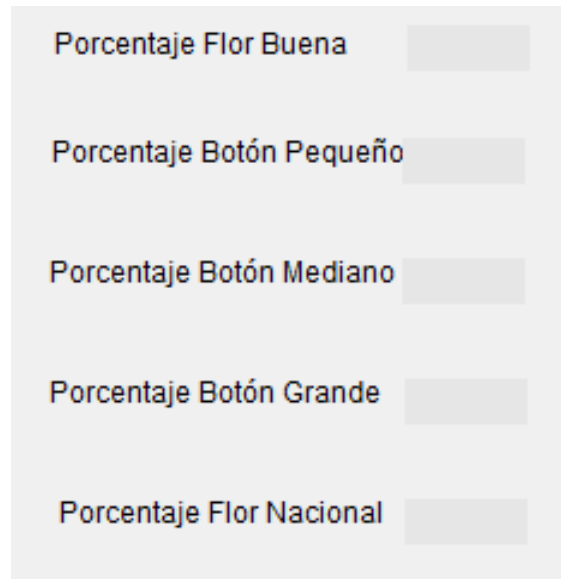
Figura 4.20. Primera columna de resultados de clasificación.



Fuente. Elaboración Propia.

La segunda columna (figura 4.21) brinda información sobre el porcentaje de rosas que han sido clasificadas en cada categoría. El “Porcentaje Flor Buena” y el “Porcentaje Flor Nacional” están relacionados con la cantidad total de flor procesada, por lo que de la suma de estos dos se obtendrá el cien por ciento. Los porcentajes correspondientes a Botón Pequeño, Botón Mediano y Botón Grande están relacionados con el “Porcentaje Flor Buena”, pues solo las flores categorizadas como Buena o exportación son aquellas que son medidas.

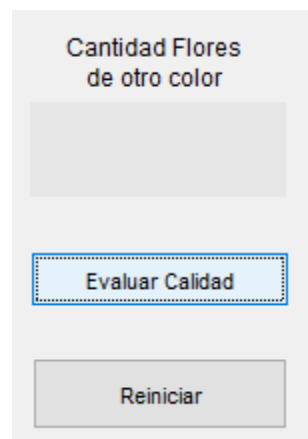
Figura 4.21. Segunda columna, porcentajes de clasificación.



Fuente. Elaboración Propia.

La tercera columna (figura 4.22) muestra la cantidad de flores que no son del color rojo y por lo tanto no se realiza ninguna acción adicional. Más abajo se encuentra el botón para iniciar el proceso de análisis a una rosa nueva (Véase Anexo 6, código del botón), y un último botón llamado “Reiniciar” que tiene como propósito reiniciar el proceso en general.

Figura 4.22. Tercera columna.

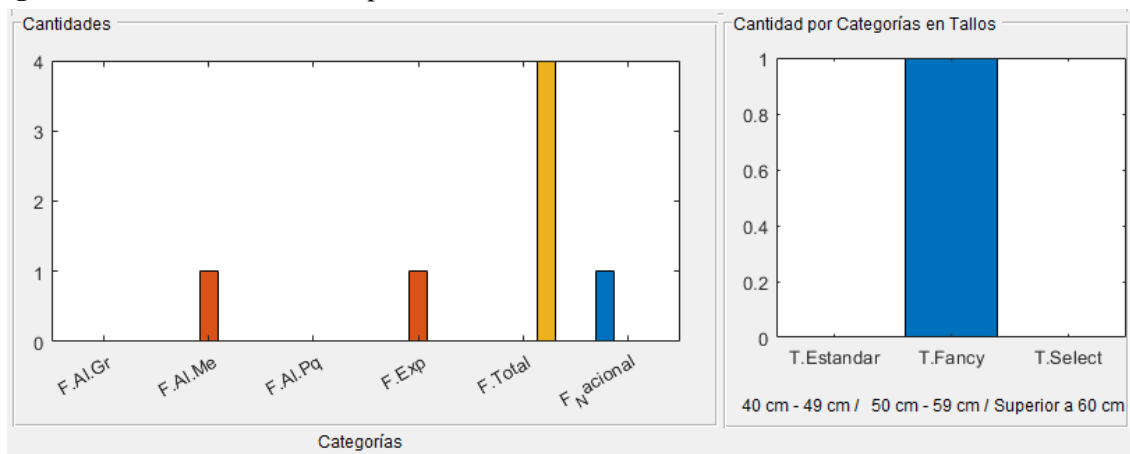


Fuente. Elaboración Propia.

Las gráficas de barras (figura 4.23), en la parte inferior izquierda de la ventana muestra el estado del análisis en términos de cantidades. Cada una de las categorías se abrevian de la siguiente manera: flor altura grande (F.Al.Gr), Flor altura mediana (F.Al.Me), flor altura pequeña (F.Al.Pq), flor para exportación (F.Exp), cantidad total de flor procesada (F.Total), flor para consumo nacional (F.Nacional), flor anchura grande (F.An.Gr), flor anchura mediana (F.An.Me) y flor anchura pequeña (F.An.Pq). Las categorías para botón en altura o anchura dependen de la selección del usuario en la ventana de configuración.

Las gráficas de barras que se muestran en la parte inferior derecha de la ventana indican el estado actual de clasificación de tallos por cantidades en cada categoría.

Figura 4.23. Gráficas de barras para visualización de cantidades.



Fuente. Elaboración Propia.

- Pruebas de validación 1:

Las pruebas de validación 1 se hicieron con datos nuevos los cuales nunca fueron parte de los datos de entrenamiento. Se realizaron algunos ajustes al programa en la medida que aparecía algún inconveniente o problema. Los ajustes fueron realizados para mejorar la etapa de preprocesamiento con el fin de dar soluciones a problemas sencillos con el fin de mejorar el diseño propuesto como también se buscaron maneras alternativas de agregar datos de entrenamiento y obtener los clasificadores finales que se implementaron. Por lo anterior fue

necesario reestructurar líneas de código, cambiar elementos y maneras de distribución de los elementos de la interfaz gráfica, lo que permitió obtener finalmente el programa y la interfaz que se han venido mostrando en las figuras.

- Pruebas de validación 2

Estas pruebas permiten los resultados finales después de realizar los ajustes necesarios y que se determinaron en las pruebas de validación 1. Para esto se probó con las imágenes que nunca fueron usadas como datos de entrenamiento. Para este fin se tiene una colección de 26 imágenes, dispuestas de la siguiente manera:

- Rosas Buenas sin enfermedad ni tallo deforme: 9
- Rosas Torcidas: 9
- Rosas Torcidas y con Velloso: 2
- Rosas con Velloso: 4
- Rosas de otro color: 1

La tabla de la figura 4.24 presenta la clasificación realizada por una persona experimentada y las predicciones realizadas por los modelos entrenados, siendo el Clasificador A, el correspondiente a *Linear SVM* y el Clasificador B, se asocia con *Quadratic SVM*. Para el caso de las clasificaciones realizadas por los modelos se adiciona el porcentaje de qué tan seguro está el programa sobre la predicción. Para abreviar los datos en la tabla se asigna la letra B para flores Buenas, T para flores con tallo torcido y V para flores con enfermedad velloso. Estas pruebas se realizan únicamente alterando la línea de código que escoge uno u otro clasificador, por lo que las demás etapas no son alteradas.

Para el caso del clasificador A, encuentra 4 rosas como rosas buenas a las cuales se les midió el tallo y se clasificó todas en la categoría Fancy. En el caso del clasificador B, encuentra 2 rosas

como rosas buenas, coincidiendo en las mismas encontradas con el clasificador A y de igual manera con clasificación de tallo en categoría Fancy, lo cual es de esperar pues las etapas de preprocesamiento son las mismas.

Figura 4.24. Resultados de clasificación obtenidos por los dos clasificadores entrenados.

#	Persona	Clasificadores entrenados			
		Clasificador A		Clasificador B	
		Clasificador	%	Clasificador	%
1	T	T	76.47	T	76.4
2	V	V	87.5	V	100
3	T y V	V	85.7	V	85.7
4	B	B	72.2	B	72.2
5	T	V	93.7	V	100
6	B	B	47	V	70.5
7	B	V	87.5	V	87.5
8	B	V	83.3	V	94.4
9	B	V	87.5	V	93.7
10	B	V	87.5	V	93.7
11	T y V	T	100	V	94.1
12	B	V	64.7	V	88.2
13	T	T	62.5	V	68.7
14	T	B	50	B	43.7
15	B	V	68.7	V	85.2
16	T	B	56.2	V	75
17	V	V	66.6	V	88.8
18	T	V	82.3	V	88.2
19	T	No es roja	NA	No es roja	NA
20	V	V	64.7	V	100
21	T	T	66.6	T	88.8
22	V	V	64.7	V	76.4
23	T	V	76.4	V	94.1
24	B	V	64.7	V	88.2
25	Otro color	No es roja	NA	No es roja	NA

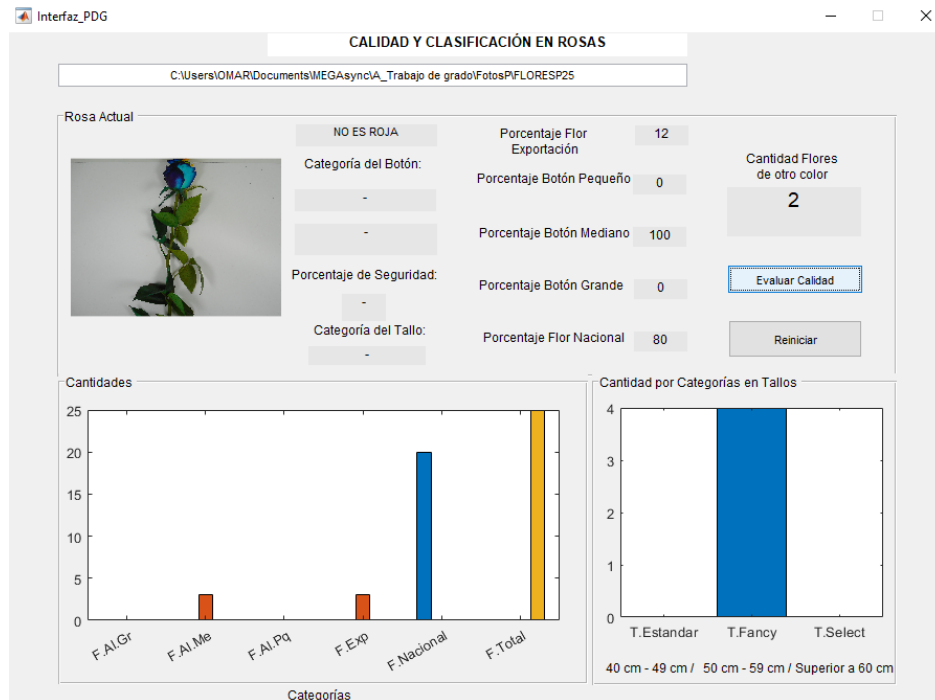
Fuente. Elaboración Propia.

En la interfaz del programa, al finalizar las pruebas con cada uno de los clasificadores entrenados presentan los informes en pantalla como se muestra en las figuras 4.25 y 4.26.

Algunas de las flores por motivos de imperfecciones en el tallo no pudieron rotar

completamente, por lo que algunas carpetas contienen una o más imágenes adicionales. Esto no se considera un problema y además se tuvo en cuenta en el desarrollo cuando se explica que la cantidad de imágenes que se ingresan al programa no tiene límite, sin embargo, no se debe olvidar que la cantidad de imágenes afecta el tiempo de clasificación.

Figura 4.25. Informes finales de clasificación en la interfaz con clasificador *Linear SVM*.



Fuente. Elaboración Propia.

Cabe mencionar que las rosas clasificadas como “otro color”, no se encuentran en los grupos de tipo exportación ni para consumo nacional, por lo tanto, al sumar los porcentajes de flor para exportación y flor nacional, no se obtendrá el cien por ciento.

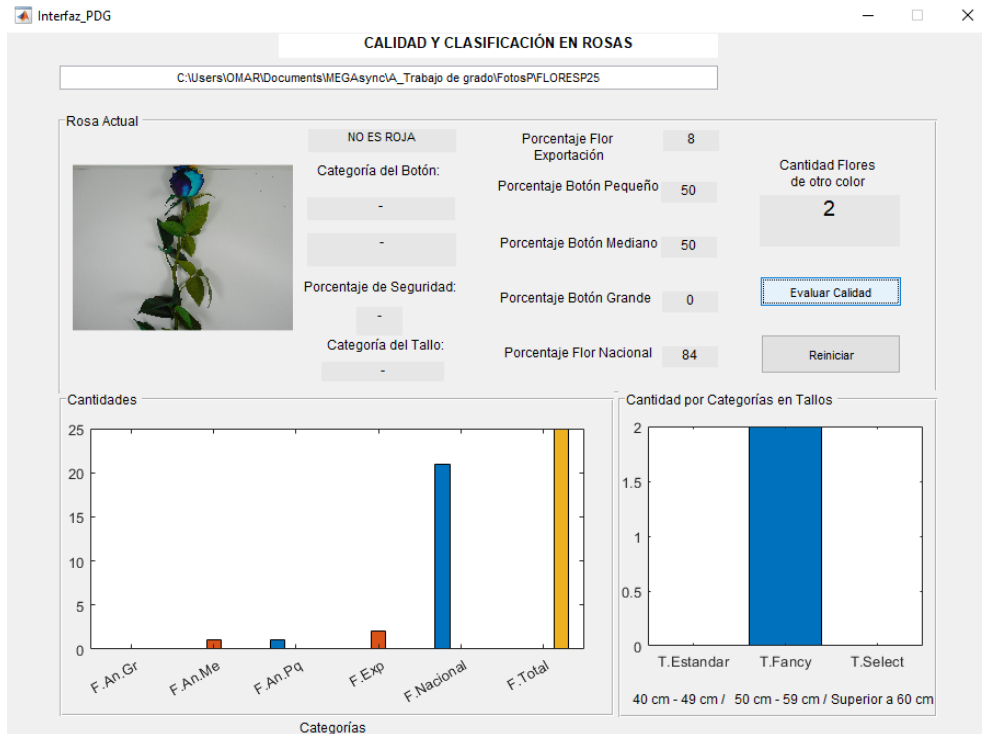
Las medidas de clasificación para los intervalos que se piden en la primera ventana fueron los siguientes:

- Altura y Anchura Pequeña: 40mm - 50 mm
- Altura y Anchura Mediana: 50mm - 60 mm
- Altura y Anchura Grande: 60mm - 70 mm

La interfaz del clasificador A clasificó en términos de Altura y la interfaz del clasificador B lo hizo en términos de anchura.

Para finalizar, la rosa de otro color que se emplea de manera intencional es una rosa tinturada artificialmente con productos especiales y presenta no solo un color, si no varios a la vez, arrojando como resultado “NO ES ROJA” tal y como se muestra en el informe final de clasificación de la figura 4.26 en la última rosa procesada. De tal manera que se demuestra que el algoritmo de preprocesamiento que cumple la función de determinar color lo hace de forma correcta.

Figura 4.26. Informes finales de clasificación en la interfaz con clasificador *Quadratic SVM*



Fuente. Elaboración Propia.

CAPÍTULO 5

5 Interpretación y Discusión De Resultados

En este capítulo se analizará el desempeño de cada una de las etapas que hacen parte del programa. Teniendo en cuenta que el desempeño del computador que ejecuta el programa es importante, ya que los tiempos de procesamiento de datos afectan el rendimiento del programa en términos de tiempo. Por esta razón se adicionan las prestaciones del computador que ejecuta el programa.

- Procesador: Intel Core i5 (tercera generación)
- Memoria Ram: 8 GB
- Velocidad de procesamiento: 2.6 GHz

Con estas características se puede apreciar que el computador presenta unas características regulares ya que normalmente en este tipo de aplicaciones las *Graphics Processing Unit (GPU)* desempeñan un papel importante en el procesamiento de gráficos, lo que representa una limitante del equipo ya que no la tiene integrada.

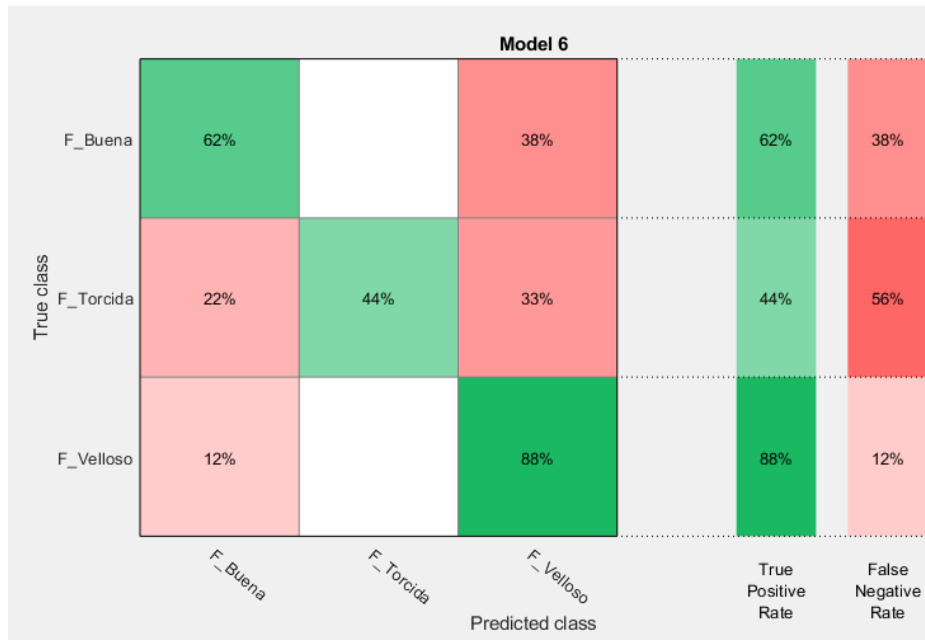
En la etapa de preprocesamiento se observó un desempeño óptimo en cuanto a obtención de las medidas tanto del botón y del tallo y se clasificaron en la categoría correcta de acuerdo con los intervalos establecidos por el usuario.

En cuanto a la etapa de clasificación realizada con los algoritmos entrenados y, con el fin de establecer una discusión entrono al desempeño del algoritmo, se analizarán los resultados obtenidos en el entrenamiento y plasmados en las matrices de confusión que proporciona Matlab. Paralelamente, se analizará el desempeño de los clasificadores con datos nuevos y se mostrarán las matrices de confusión que se obtienen de cada clasificador con estos mismos datos.

- Clasificador A: *Linear SVM*

En la figura 5.1 se observa la matriz de confusión del modelo entrenado Linear SVM con un porcentaje de precisión de 67.1%. Las filas indican las clases verdaderas, mientras que las columnas muestran las clases predichas. Por lo tanto, en la fila superior el 62% de las flores buenas se clasifican de forma correcta. Por lo que el 62% es la razón de positivos verdaderos para la categoría F_Buena. Además, es de reconocer que no existen flores pertenecientes a la categoría F_Buena y que hayan sido clasificadas como F_Torcida. Pero, por otro lado, el 38% flores fueron clasificadas en la categoría F_Velloso y realmente pertenecen a la categoría F_Buena. La razón de falsos negativos para datos clasificados en esta categoría es del 38%. En la fila del medio que corresponde a la categoría de flores torcidas (F_Torcida), la primera columna señala que el 22% de los datos que realmente son torcidos son clasificados en la categoría F_Buena, el 44% de las flores torcidas son clasificadas correctamente y un 33% de las rosas torcidas son clasificadas en la categoría F_Velloso. Por lo que la categoría de F_Torcida tiene una razón de falsos negativos de 55%. La categoría F_Velloso parece ser la que mejor puede ser clasificada, según la gráfica el 12% de los datos fueron clasificados como F_Buena, ninguno de los datos de F_Velloso es clasificado como F_Torcida y el 88% fue clasificado de forma correcta, con una razón de verdaderos positivos para esta categoría de 88% mientras la razón de falsos negativos es del 12%.

Figura 5.1. Matriz de confusión clasificador *Linear SVM*.



Fuente. Elaboración Propia.

Replicando el procedimiento que utiliza Matlab para mostrar que tan eficiente es el modelo entrenado mediante una matriz de confusión, se realiza una matriz de confusión como se muestra en la figura 5.2, a la izquierda por número de observaciones y a la derecha en términos de porcentajes, esta matriz de confusión se realiza con los resultados obtenidos en los datos nuevos y que se presentaron anteriormente en la tabla de la figura 4.24, comparando los dos modelos con la clasificación realizada por una persona. Es necesario recordar que algunas de las rosas presentan dos categorías ya que pueden padecer la enfermedad veloso y al mismo tiempo estar torcidas, en este caso, en cualquiera de las dos categorías puede ser correcta la clasificación.

El clasificador implementado presenta unos resultados regulares para la clasificación con flores buenas, ya que la mayoría de las veces clasificó las flores buenas como flores con veloso. Sin embargo, coincide con la anterior matriz de confusión en no clasificar flores buenas en la

categoría torcidas. La razón de falsos negativos para esta categoría es de 77.8% y la razón para verdaderos positivos es de 22.2%.

Por otro lado, para la categoría flor torcida el 22.2% de los datos fueron clasificados como flor buena coincidiendo con la matriz presentada por Matlab, el 44.5% de los datos fueron bien clasificados y el 33.3% de los datos fueron clasificados de forma incorrecta en la categoría flor con velloso, coincidiendo en la totalidad de esta fila con la presentada por Matlab.

Finalmente, la categoría flor con velloso presenta un excelente resultado al clasificar de manera correcta todos los datos en la categoría correcta, sin presentar datos propios en las categorías que no pertenecen; superando incluso los resultados de la matriz presentada por Matlab en esta categoría.

Figura 5.2. Matriz de confusión clasificador *Linear SVM* con datos nuevos.

	Programa Final con clasificador Linear SVM		
F_Buena	2	0	7
F_Torcida	2	4	3
F_Velloso	0	0	5
	F_Buena	F_Torcida	F_Velloso

	Programa Final con clasificador Linear SVM		
F_Buena	22.2	0	77.8
F_Torcida	22.2	44.4	33.3
F_Velloso	0	0	100
	F_Buena	F_Torcida	F_Velloso

Fuente. Elaboración Propia.

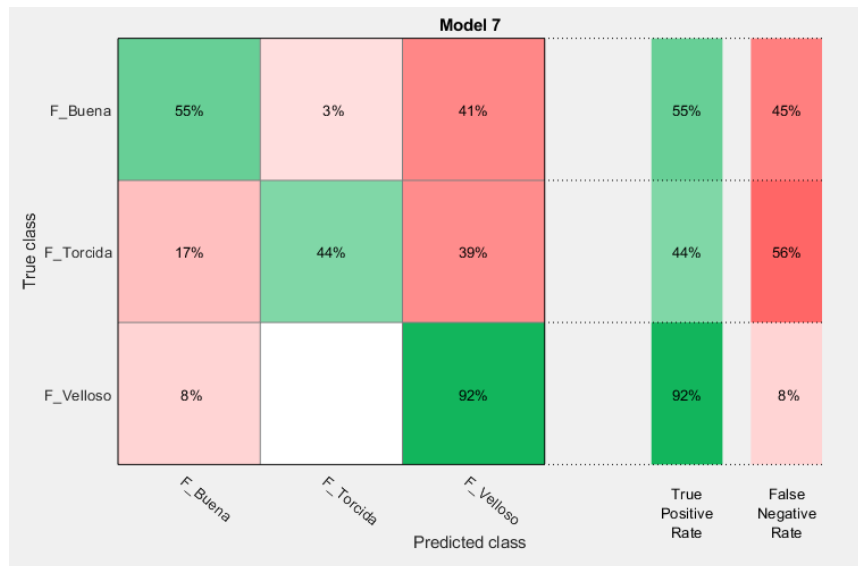
- Clasificador B: *Quadratic SVM*

Para el caso del clasificador B, con un porcentaje de precisión de 65.8%, Matlab presenta una matriz de confusión como se ilustra en la figura 5.3, en donde se observa que el porcentaje de flores clasificadas en la categoría F_Buena o flor buena, es del 55%, existe un 3% de flores buenas clasificadas en la categoría flor torcida y 41% de flores buenas clasificadas en la categoría flor con velloso. Para la categoría flor buena, Matlab presenta una razón de verdaderos positivos del 55% y una razón de falsos negativos de 45%.

Para la categoría flor torcida, el clasificador presenta un 17% de los datos clasificados en la categoría flor buena (un poco menos que el clasificador anterior), 44% de las flores torcidas clasificadas de forma correcta (el mismo porcentaje que el anterior) y un 39% de los datos clasificados de forma incorrecta en la categoría flor con vellosa. Para esta categoría se presenta una razón de verdaderos positivos del 44% y una razón de falsos negativos del 56%, manteniendo los mismos resultados que el clasificador anterior.

Finalmente, para la categoría flor con vellosa, el clasificador presenta 8% de los datos clasificados de forma incorrecta en la categoría flor buena, menos que el algoritmo anterior y prometiendo ser mejor ya que clasifica el 92% de los datos de forma correcta en la categoría flor con vellosa y ninguno en la categoría flor torcida.

Figura 5.3. Matriz de confusión clasificador *Quadratic SVM* con datos nuevos.



Fuente. Elaboración Propia.

De la realización de pruebas con datos nuevos se obtuvieron los resultados plasmados en la matriz de confusión de la figura 5.4. De los cuales se concluye que para el caso de la categoría flor buena, de la primera fila el 11.11% de los datos fueron clasificados de forma correcta, de

esta categoría ninguna de las flores fue clasificada en la categoría flor torcida y el 88.88% de las flores fue clasificada de forma incorrecta en la categoría flor con velloso. Para la categoría flor torcida, el 12 % de las flores es clasificada de forma incorrecta en la categoría flor torcida, el 25% de las flores es clasificada de forma correcta y 62.5% de los datos es clasificado de forma incorrecta en la categoría flor con velloso.

Finalmente, para la categoría flor con velloso ninguna de las flores es clasificada de forma incorrecta y presenta un porcentaje de 100% para la categoría.

Figura 5.4. Matriz de confusión clasificador *Quadratic SVM* con datos nuevos.

	Programa Final con clasificador Quadratic SVM		
F_Buena	1	0	8
F_Torcida	1	2	5
F_Velloso	0	0	6
	F_Buena	F_Torcida	F_Velloso

	Programa Final con clasificador Quadratic SVM		
F_Buena	11.1	0	88.9
F_Torcida	12.5	25	62.5
F_Velloso	0	0	100
	F_Buena	F_Torcida	F_Velloso

Fuente. Elaboración Propia.

La realización de pruebas con los dos clasificadores busca determinar si efectivamente el 1.3% que los diferencia afecta significativamente en los resultados. Aparentemente el clasificador B parecía ser un poco más flexible a la hora de categorizar las flores, pero los resultados mostraron que al contrario, fueron más las flores clasificadas en la categoría velloso.

Se analizarán las 4 flores clasificadas como buenas por el Clasificador A que son mayoría:

- Flor #4: Clasificada como flor buena con un porcentaje de seguridad del 72.2%, lo que significa que el 72% de las imágenes correspondientes a esta flor fueron clasificadas en esta categoría. La medida real de la flor es de 55 mm de alto y 40 mm de ancho, por lo tanto, debe pertenecer a la categoría altura mediana. Lo cual es correcto. En cuanto a anchura fue clasificada en la categoría anchura pequeña, lo cual es correcto. Longitud del tallo 55 cm, clasifica en la categoría Tallo Fancy

- Flor #6: Clasificada como flor buena con un porcentaje de seguridad del 47%. Esta flor clasificó en esta categoría porque la cantidad de imágenes defectuosas evidentemente no superaron el 60%, sin embargo, fue categorizada como flor nacional debido a su reducida altura, clasificando como altura deforme pequeña, lo cual es correcto pues su altura real es de 38 mm y anchura 35 mm. Longitud del tallo 56 cm, clasifica para la categoría Tallo Fancy.
- Flor #14: Clasifica de forma incorrecta en la categoría Flor Buena con un porcentaje de seguridad del 50%, realmente pertenece a la categoría Flor Torcida, por lo tanto, la medida del botón floral no es confiable ya que lo más seguro es que el botón aparezca inclinado, como tampoco es confiable la medición del tallo puesto que no está recto.
- Flor #16: Clasifica de forma incorrecta en la categoría flor buena con un porcentaje de seguridad de 56.2%, realmente es una rosa con tallo torcido, por lo cual, las medidas obtenidas del botón no son confiables, como tampoco lo son las medidas obtenidas para el tallo porque está torcido.

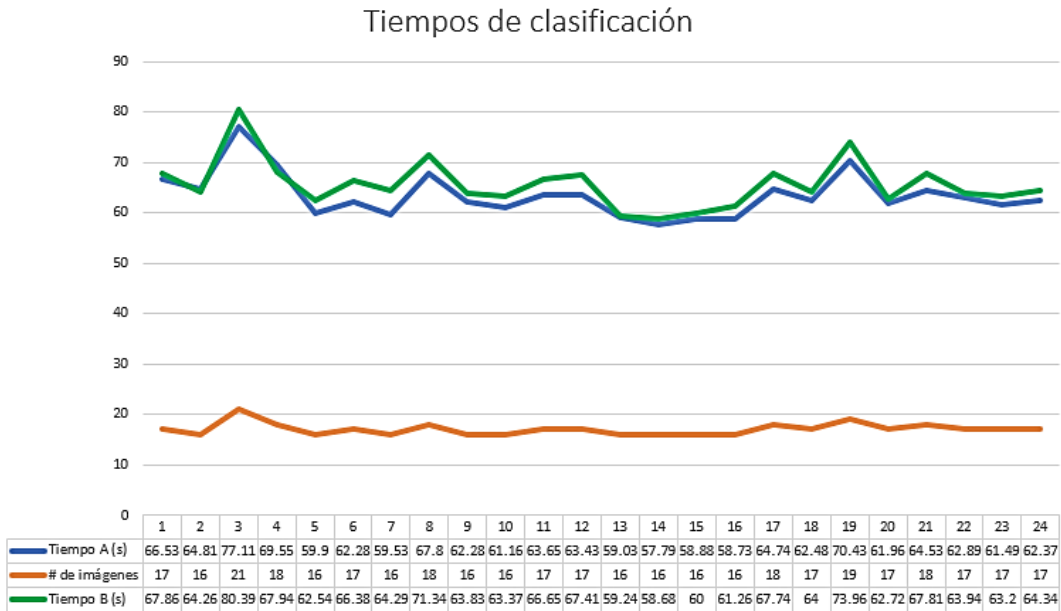
Para la parte de análisis y clasificación de flores por medidas de las dimensiones de interés del usuario es realizado de forma correcta, pues la eliminación de ruido y obtención de las dimensiones que encierran la silueta de interés permiten obtener resultados satisfactorios.

Un último análisis al desempeño del clasificador está relacionado con los tiempos de clasificación los cuales se midieron analizando el estado de la flor, ya que, si se retoma el paso a paso del funcionamiento del programa, es de notar que ciertas etapas eliminan a la flor de avanzar a la siguiente etapa, lo que provoca que el proceso termine rápido, o lo contrario, se extienda dependiendo de las características de calidad de la flor en cuestión. En pocas palabras, una flor con excelentes parámetros de calidad, al programa le tomará más tiempo en dictar una

decisión en comparación con una flor que, por ejemplo, se encuentra demasiado maltratada por la enfermedad vellosa, o por estar torcida y menos tiempo aún si no es roja. Para el caso de una flor que no es roja, el programa termina el proceso en un promedio de 25 segundos a 35 segundos. Por otro lado, si la rosa es roja, pero presenta enfermedad vellosa o está torcida, el programa termina el proceso en un máximo 1 minuto con 30 segundos con 21 imágenes de la flor en cuestión ya que en este caso, estaría ejecutando el extractor de características y después el clasificador entrenado. El último caso es que la rosa esté en excelentes condiciones y deba ser medidos tanto el botón como el tallo, para el cuál el programa le toma 2 minutos con 20 segundos como máximo. Por lo anterior expuesto se determina que el programa aún debe mejorar, ya que una persona promedio le toma alrededor de 30 segundos como máximo en clasificar una rosa, por lo que es importante pensar además en la cantidad de fotografías que logren dar información completa de la rosa, o lo que es lo mismo establecer un muestreo mínimo suficiente, recordando que para efecto de pruebas se realizaron con al menos 18 fotografías de cada rosa, un muestreo cada 20 grados, llegando a ser mucho.

Para concluir el análisis de tiempos de clasificación, se realizó una validación evaluando únicamente a la función que emplea el clasificador entrenado. De esta forma, medir cuanto le toma únicamente a la función realizar una predicción. Cabe recordar que la función contiene el extractor de características y el clasificador entrenado. El procedimiento se realiza con ambos clasificadores, con el fin de determinar cuál de los dos es además más rápido. La gráfica de la figura 5.5, realizada en Excel muestra de forma general, la rapidez de clasificación en términos de segundos de los clasificadores, teniendo en cuenta además la cantidad de imágenes ingresadas por el usuario y que las características en ambos casos son las mismas; 190 puntos clave.

Figura 5.5. Tiempos de clasificación de los modelos entrenados.



Fuente. Elaboración Propia.

Los números de 1 al 24 representa la flor en cuestión, la línea naranja muestra la cantidad de imágenes ingresadas por cada flor, mientras que las líneas azul y verde, representan el tiempo que le toma a cada clasificador *Linear SVM* y *Quadratic SVM* respectivamente, realizar una clasificación en segundos. La información de cantidad y tiempos se muestran en una misma escala en el eje Y, sin embargo, el análisis es simple y es de notar que cuando se incrementa el número de imágenes, a ambos clasificadores les toma más tiempo realizar una clasificación, algo que ya se había advertido. Por otro lado, al clasificador *Quadratic SVM*, en línea color verde, la mayoría de las veces le toma más tiempo realizar clasificaciones. Se ha observado que es el clasificador menos preciso. Es de suponer entonces, que los cálculos que realiza este clasificador requieren mayor rendimiento de la máquina que los ejecuta en comparación con los cálculos que se realizan con el clasificador *Linear SVM*.

Como resultado general de lo anterior, se logra dar cumplimiento a los objetivos específicos planteados, como lo es, manipular las imágenes por medio de procesamiento digital de imagen, logrando eliminación de ruidos, se clasificaron las imágenes usando el extractor de características *bag of features* empleando después varios clasificadores entrenados y eligiendo los dos mejores, y se evalúa también la precisión de los clasificadores analizando las matrices de confusión de los clasificadores, se midió el tiempo que tarda el programa en realizar las clasificaciones. Finalmente se diseñó una interfaz gráfica usando la guía de Matlab, que permite la visualización de los resultados y la interacción con el usuario. No se logra la totalidad del objetivo general ya que la clasificación no es 100% precisa y como ya se vio presentó errores de clasificación de rosas buenas clasificadas como flor con velloso. A manera de resumen final, se observa que el método seleccionado no es preciso, pues dado que los datos suministrados para el entrenamiento de rosas buenas y rosas con velloso fueron considerablemente amplios, fue en estas categorías en donde se presentó mayor imprecisión, en comparación con la categoría rosa torcida, donde a pesar de tener poca información es más preciso el algoritmo.

CAPÍTULO 6

6 Conclusiones, Recomendaciones y Trabajo Futuro

En este capítulo se comentarán las conclusiones, recomendaciones y posibles trabajos a futuro en el campo de la visión artificial aplicada al sector agrícola y más específicamente a la clasificación de flores de producción.

- Se da cumplimiento a los objetivos específicos del proyecto en cuanto se logra manipular las imágenes por medio de preprocesamiento digital de imágenes logrando la atenuación de ruidos y distorsiones, esto empleando filtros morfológicos, segmentación basada en color, conversión a escala de grises y conversión binaria.
- Se logra la clasificación de imágenes basada en extracción de características, aunque la técnica seleccionada para tal fin no muestra una alta precisión mostrando errores de clasificación de flor buena en la categoría flor con veloso de hasta el 80%.
- Se logra una segunda clasificación de las imágenes basada en tamaño de botón y longitud del tallo usando técnicas de preprocesamiento sin error alguno.
- Se evalúa el desempeño de las técnicas empleadas determinando que el sistema le toma demasiado tiempo debido a varios factores, entre ellos la técnica empleada *bag of features* con 190 características por imagen y la cantidad de imágenes por cada flor.
- Se observa además que el clasificador Quadratic SVM, le toma más tiempo en clasificar una rosa, en comparación con el clasificador Linear SVM, esto se evidencia en las pruebas de validación 2, agregando también que es el menos preciso.
- Se implementa una interfaz gráfica de usuario que permite ver los resultados generados por el sistema y a su vez permite la interacción con un usuario.

En cuanto al funcionamiento del sistema se observa que:

- La extracción de características se desarrolló iterativamente en busca del mejor resultado y desempeño de los clasificadores que propone la aplicación *Classification Learner*. De todos los intentos se determinó una extracción de 190 características, con menos características, el porcentaje de precisión del algoritmo está por debajo del 40% y con un número mayor toma mucho tiempo y resulta en solapamientos entre categorías. Estos solapamientos se observan en las gráficas que muestran los puntos de interés extraídos, lo que desde el principio indica menos precisión de los clasificadores.
- En la categoría flor buena, la cual se entrenó con 59 fotografías de imágenes de rosas en excelentes condiciones, se observa que el porcentaje de aciertos es de 22% que corresponde a dos aciertos de nueve en el mejor de los casos (*linear SVM*), y el 88% corresponde al porcentaje de flores mal clasificadas como flores con velloso.
- Para el caso de la categoría flor torcida, se debe considerar que los datos de entrenamiento para esta categoría fueron pobres en comparación con los datos en las otras categorías, ya que sólo se entrenó con 30 fotografías de rosas que presentaban el tallo torcido. El desempeño en esta categoría alcanzó un porcentaje de aciertos de 44.4%, mostrando un porcentaje de 22.2% de flores mal clasificadas como buenas y un 33.3% restante como flores con velloso.
- Para el caso de la categoría flor con velloso, el algoritmo presenta una alta sensibilidad a cualquier rastro de la enfermedad, esta categoría se entrena con 50 fotografías de rosas con enfermedad velloso. En este caso logra un porcentaje de aciertos del 100% para las flores con enfermedad velloso indicando que las características extraídas en este caso fueron representativas en esta categoría.

Es evidente entonces, que el método de extracción de características “*Bag Of Features*” no es un método eficiente para implementar en este tipo de problemas. Es necesario probar métodos más robustos que sean capaces de abarcar más categorías (enfermedades y criterios de calidad), y que además realicen las clasificaciones en poco tiempo. Sin embargo, una forma de mejorar los resultados de esta estrategia de clasificación, se debe contar con datos de entrenamiento más amplios, lo cual fue limitante importante en este proyecto, se deben implementar también, estrategias de preprocesamiento que resalten características importantes en las categorías; estas estrategias pueden ser enfocadas en filtros de color, y o forma. Una última sugerencia de mejora en este tipo de estrategias de clasificación, es quizá probar con métodos diferentes de extracción de características, extractores de características como SIFT y SURF u otras estrategias novedosas que permitan obtener información del conjunto de datos de entrenamiento para los clasificadores.

A modo de recomendación para trabajos futuros, se deberían tener en cuenta otros criterios para selección de calidad en las flores, más enfermedades, malformaciones y maltrato (daños mecánicos), los cuales son criterios de calidad. Por lo tanto, casos más complejos que revisen a mayor profundidad la calidad de las rosas requerirán herramientas más sofisticadas de la inteligencia artificial y con ello, también un mejor desempeño en las computadoras que ejecuten este tipo de programas. Como se observó en el desarrollo de este proyecto, lograr la clasificación completa de una rosa de tipo exportación toma como mínimo 120 segundos, en comparación con una persona experta que logra concluirlo en 30 segundos como máximo. Cabe agregar que es importante evaluar la calidad de especies de flores diferentes a las rosas, como el clavel, crisantemo, mini rosa y demás, las cuales también entran en la economía floricultora del país.

Otro aspecto importante para recomendar en este tipo de sistemas es el uso de programas de desarrollo alternativos que permitan libertad de implementación. Programas de lenguaje de

desarrollo como Python, el cual ya cuenta con un buen soporte en el ámbito de inteligencia artificial.

Para terminar, es necesario recordar que el sistema que se presentó en este proyecto únicamente presenta la parte de programación, pero que nunca se trabajó la parte mecánica por cuestiones de limitación de tiempo y recursos. Por lo tanto, para este proyecto específico, la parte mecánica debería tener una interfaz de comunicación que sincronice las capturas de imágenes con el programa de procesamiento de imágenes, y a su vez, sea capaz de controlar los actuadores y los brazos que clasifican cada flor en el lugar correspondiente.

REFERENCIAS

- Bayer CropScience. (s.f.). *Mildeo veloso. El Mildeo Velloso acecha su cultivo Protéjalo con el mejor portafolio*. [Folleto]. Autor.
- Cárdenas, M., & Rodríguez, M. (2011). *Colombia y la creación de una empresa productora de flores*. Universidad de La Sabana, 166.
- Csurka, G., C. R. Dance, L. Fan, J. Willamowski, and C. Bray. *Visual Categorization with Bags of Keypoints*. Workshop on Statistical Learning in Computer Vision. 2004, ECCV 1 (1–22), 1–2.
- Dharwadkar, S., Bhat, G., Reddy, N. V. S., & Aithal, P. K. (2017). Floriculture classification using simple neural network and deep learning. *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, 619–622. <https://doi.org/10.1109/RTEICT.2017.8256671>
- Do, T. B., Nguyen, H. H., Nguyen, T. T. N., Vu, H., Tran, T. T. H., & Le, T. L. (2017). Plant identification using score-based fusion of multi-organ images. *Proceedings - 2017 9th International Conference on Knowledge and Systems Engineering, KSE 2017, 2017–Janua*, 191–196. <https://doi.org/10.1109/KSE.2017.8119457>
- García Santillán, I. D. (2008). *Visión artificial y procesamiento digital de imágenes usando Matlab*. Quito, Ecuador: Cámara Ecuatoriana del Libro - Núcleo de Pichincha
- Gavai, N. R., Jakhade, Y. A., Tribhuvan, S. A., & Bhattad, R. (2018). MobileNets for flower classification using TensorFlow. *2017 International Conference on Big Data, IoT and Data Science, BID 2017, 2018–Janua*, 154–158. <https://doi.org/10.1109/BID.2017.8336590>
- Ghosh, S., Kumar, H., Kumar, P.V., & Nayak, J.S. (2015). *Study on Classification of Plants Images using Combined Classifier*.
- Gogul, I., & Kumar, V. S. (2017). Flower species recognition system using convolution neural networks and transfer learning. *2017 4th International Conference on Signal Processing, Communication and Networking, ICSCN 2017*, 1–6. <https://doi.org/10.1109/ICSCN.2017.8085675>
- Hiary, H., Saadeh, H., Saadeh, M., & Yaqub, M. (2015). Flower Classification using Deep Convolutional Neural Networks, *I2*, 1–8. <https://doi.org/10.1049/iet-cvi.2017.0155>
- Hussein, S. A., Naby, H. El, & Youssif, A. A. A. (2016). Image Multi-Classification using PHOW Features. *IOSR Journal of Computer Engineering*, *18*(5), 2278–2661. <https://doi.org/10.9790/0661-1805063136>

- Marín, M., & Rangel, J. (2000). *Comercialización Internacional de Flores: Antecedentes y Evolución: 1990-1999*. Universidad Nacional de Colombia, 278.
- Norvig, P., & Russell, S. (2014). *Inteligencia artificial. Elsevier Brasil*. (Vol. 1). <https://doi.org/M-26913-2004>
- Núcleo Ambiental S.A.S. (2015). *Flores & follajes*, 1–42. Recuperado de: https://www.google.com.co/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&cad=rja&uact=8&ved=0ahUKEwiMq_TrkpXQAhVFKyYKHcIgAMwQFghNMAY&url=https://www.cb.org.co/content/download/13733/175129/file/Flore_Follajes.pdf&usg=AFQjCNFGTwQCPrE2PPnoFqW55y-LHixNvA&sig
- Liu, W., Rao, Y., Fan, B., Song, J., & Wang, Q. (2017). Flower classification using fusion descriptor and SVM. *2017 International Smart Cities Conference, ISC2 2017*.
- Palma, J., y Marín, R. (2008). *Inteligencia Artificial. Inteligencia Artificial* (Vol. 2). McGraw-Hill. España. <https://doi.org/10.4114/ia.v2i6.614>
- Perez, R. (2002). *Plagas Y Enfermedades Importantes Del Rosal*, 63.
- Ponce, P. (2010). *Inteligencia Artificial con Aplicaciones a la Ingeniería*. (Alfaomega, Ed.) (Primera Ed). México.
- Purohit, S., Gandhi, S., Viroja, R., & Chaudhary, N. (2015). Automatic Plant Species Recognition Technique using Machine learning approaches. *2015 International Conference on Computing and Network Communications (CoCoNet)*, 710–719. <https://doi.org/10.1109/CoCoNet.2015.7411268>
- Ravindran, P., Costa, A.M., Soares, R.E., & Wiedenhoeft, A.C. (2018). *Classification of CITES-listed and other neotropical Meliaceae wood images using convolutional neural networks. Plant Methods*.
- Shukla, T., Mishra, N., & Sharma, S. (2013). *Automatic image annotation using SURF Features. 2013 Annual IEEE India Conference, INDICON 2013*, 68(4), 920–924. <https://doi.org/10.1109/INDCON.2012.6420748>
- Tárrega, P., (2015, marzo 5). *Procesado de imagen de forma sencilla* [Archivo de video]. Recuperado de https://la.mathworks.com/videos/image-processing-made-easy-100161.html?fbclid=IwAR3fYY1kCVzKTJgKpjXAe_QIDEVoidW_E-3dWbOtjDW4f5pRXjUKMQcadTQ
- Tárrega, P., (2015, mayo 7). *Aprendizaje automático (Machine Learning) de forma sencilla* [Archivo de video]. Recuperado de <https://la.mathworks.com/videos/machine-learning-made-easy-101277.html>

- Tiay, T., Benyaphaichit, P., & Riyamongkol, P. (2014). *Flower Recognition System Based on Image Processing*, 99–102.
- Uddin, M.A., Joolee, J.B., Alam, A., & Lee, Y. (2017). *Human Action Recognition Using Adaptive Local Motion Descriptor in Spark*. *IEEE Access*, 5, 21157-21167.
- Xia, X., Xu, C., & Nan, B. (2017). Inception-v3 for flower classification. *2017 2nd International Conference on Image, Vision and Computing, ICIVC 2017*, 783–787.
<https://doi.org/10.1109/ICIVC.2017.7984661>
- Zamora, P. E. (2016). *Algunas Fisiopatias De Frutos, Tallos Y Hojas En Cultivos Protegidos*.
- Zawbaa, H. M., Abbass, M., & Basha, S. H. (2014). *An Automatic Flower Classification Approach Using Machine Learning Algorithms*. *Icacci*, 895–901.
<https://doi.org/10.1109/ICACCI.2014.6968612>

ANEXOS

Esta sección presenta las diferentes líneas de código implementadas en el desarrollo del programa. Para todos los casos, las letras en color verde significan comentarios, los cuales solo se usan para que el programador recuerde y haga anotaciones, estas no realizan ninguna acción en la ejecución del código. El resto de las líneas se entenderán como código de programación que realizan acciones específicas.

Anexo 1

Función que se encarga recortar la sección superior de la foto encontrar la silueta y por último determina si el color rojo predomina en el recorte de la imagen.

```
function colorf = deter_color(Carpeta_imag)

imset2 =imageSet(Carpeta_imag); % Se crea la colección de imágenes
Laactual = 1;
Rcolor = 2;
Lultima = imset2.Count;

while Rcolor == 2 % ciclo que revisa mientras la rosa no sea roja

    ImInteresk = read(imset2,Laactual);% imagen de interés actual
    B = ImInteresk;

    imagen = imcrop(B, [1 1 5152 1100]); % Se corta la parte Superior de la
imagen

    matiz_B = imagen(:,:,3); %UMBRALIZACION POR PANEL DE COLOR

    nivel_B = 0.06; % nivel de intensidad específico

    I3 = im2bw(matiz_B,nivel_B); %bancaización de la imagen por cada panel

    I6 = not(I3); %Negación de los datos en las imágenes
    se = strel('disk',18); %elemento estructurante disco de radio 10
    bw2 = imopen(I6,se); % se abre la imagen binaria mediante el elemento
estructurante
    se2 = strel('disk',18); %elemento estructurante disco de radio 8
    bw3 = imclose(bw2,se2); % se cierra la imagen con un elemnto

    L = bwlabel(bw3); % Crea regiones
    stats = regionprops(L,'all'); % TODAS las Estadísticas de las regiones
    E=stats(1).BoundingBox; % Toma la frontera de la 1ra región

    % 2da parte, determinación del color-----
```

```

    recor_boton = imcrop(B, [E(1) E(2) E(3) E(4)]); % Se corta únicamente la
cabeza de la rosa

    Ajus_Cont12 = imadjust(recor_boton, stretchlim(recor_boton)); % se realiza
un ajuste de contraste

    matiz_R12 = Ajus_Cont12(:,:,1);
    matiz_G12 = Ajus_Cont12(:,:,2);
    matiz_B12 = Ajus_Cont12(:,:,3);

    ColorR = mean(matiz_R12);
    ColorG = mean(matiz_G12);
    ColorB = mean(matiz_B12);

    prom_R = mean(ColorR);
    prom_G = mean(ColorG);
    prom_B = mean(ColorB);

    % fin de la segunda parte-----

    if prom_R > prom_G && prom_R > prom_B
        rectangle('Position', [E(1) E(2) E(3) E(4)], 'EdgeColor', 'r'); %DIBUJAR
RECTANGULO EN LA REGION ENCONTRADA
        hold on
        Rcolor = 1; % el número 1 significa que la rosa es roja
        break
    else
        Rcolor = 2; % el número 2 significa que la rosa no es roja
        Laactual = Laactual + 1;

        if Laultima == Laactual+1 % si ya es la última imagen y no encontró
nada... break
            break
        end
    end
end

end
colorf = Rcolor;

end

```

Anexo 2

Función que se encarga de encontrar la silueta que corresponde al tallo de acuerdo con las dimensiones mínimas establecidas. Después obtiene la medida de la longitud del tallo en pixeles y se realiza la conversión a centímetros.

```

function medicionTallo = medir_tallos(Carp_TComp)
% Funcion para medir tallos.

```

```

img_set1 =imageSet(Carp_TComp); % Se crea la colección de imágenes

NumdIma = img_set1.Count; %obtener el número de imágenes en la carpeta
seleccionada
long_tallo = 1:1:NumdIma; % crear vector = a el número de imágenes
i = 0; % inicializar el contador.

for i = 1:NumdIma % bucle para leer las imágenes y procesarlas

    k = i;
    ImInteresk = read(img_set1,k);% imagen k de interés actual
    Contr_aju = imadjust(ImInteresk,stretchlim(ImInteresk)); %la función
    imadjust, ajusta el contraste de la imagen

    imaB = Contr_aju;
    matiz_G = imaB(:,:,2);
    matiz_B = imaB(:,:,3);

    nivel_G = 0.0002;
    nivel_B = 0.12;

    I2 = im2bw(matiz_G,nivel_G); %bancaización de la imagen por cada panel
    I3 = im2bw(matiz_B,nivel_B); %bancaización de la imagen por cada panel

    restadi = imsubtract(I2,I3);% resta de imágenes para encontrar únicamente
    el tallo

    se2 = strel('disk',35); %elemento estructurante disco de radio 12
    bw3 = imclose(restadi,se2); % se cierra la imagen con un elemento
    estructurante disco de radio 35

    % etiqueta de los elementos en la imagen-----
    [Etqt Noe] = bwlabel(bw3);
    % cálculo de las propiedades de los objetos-----
    propiedades = regionprops(Etqt);
    % buscar áreas pequeñas:-----
    buscarA = find([propiedades.Area]<90000);
    % eliminar áreas menores a la indicada:-----
    for conta3 = 1:size(buscarA,2)
        f = round(propiedades(buscarA(conta3)).BoundingBox);% elemento de
        relleno de áreas pequeñas
        bw3(f(2):f(2)+f(4),f(1):f(1)+f(3))=0;
    end

    [etiquetas cant] = bwlabel(bw3);% Identifica elementos y obtiene la
    cantidad de los mismos
    descriptor = regionprops(etiquetas);% Obtiene información sobre los
    elementos en la imagen

    long_tallo(k) = descriptor.BoundingBox(3)% almacena longitud en la matriz
    "long_tallo" en la posición k en cada iteración

end % FIN DEL BUCLE DEPENDIENDO DEL NUMERO DE IMÁGENES

```

```

    prom_long = mean(long_tallo);% promedio de las medidas obtenidas-----
-----

    longt_cm = (prom_long*55)/3434; % conversion de pixels a cm -----
-----

    medicionTallo = longt_cm;

end

```

Anexo 3

Algoritmo para extracción de características y entrenamiento de los clasificadores.

```

% Cargar datos de imágenes
imset = imageSet('Data_set', 'recursive');

%% entrenamiento
% Crear el bag of features de las imágenes de flores para extracción de
% características
bag = bagOfFeatures(imset, 'VocabularySize', 190, 'PointSelection', 'Detector');

%codificación de las imágenes con nuevas características

imagefeatures = encode(bag, imset);

%% creacion de tabla usando las características de codificación

FlorData = array2table(imagefeatures);
FlorData.EstadoFlor = getImageLabels(imset); % para extraer los nombres de
las etiquetas y asignarlas a la tabla

%% entrenador de modelos...

classificationLearner

```

Anexo 4

Función con estructura entrenada para realizar predicciones.

```

function calidad = clasificador_entrenado(CarpetaI)

bag1 = evalin('base', 'bag'); % Cargar las variables del workspace
%trainedModelB = evalin('base', 'trainedModelB'); % MODELO ENTRENADO B
trainedModelB = evalin('base', 'trainedModel6'); % MODELO ENTRENADO C

% Preprocesamiento para determinar calidad de flores con imágenes usando el
modelo entrenado

imageness = CarpetaI; % Nombre de la carpeta con imagenes para ser revisadas

```

```

imaset3 =imageSet(imageness); % Se crea la colección de imágenes
cantidadfotos = imaset3.Count %obtener el número de imágenes en la carpeta
seleccionada
Estados_Rosas = categorical(1:1:cantidadfotos);

for c = 1:cantidadfotos % bucle para leer las imágenes y procesarlas

    j = c;
    Imactual = read(imaset3,j);% imagen j de interés actual
    im_entrada = Imactual;
    imagengris = rgb2gray(im_entrada);

    imagefeatures = double(encode(bag1,imagengris));
    carac_Rosa = array2table(imagefeatures);

    Estado_Rosa = trainedModelB.predictFcn(carac_Rosa); % MODELO ENTRENADO 1

    Estados_Rosas(j) = categorical(Estado_Rosa);

end % FIN DEL BUCLE DEPENDIENDO DEL NUMERO DE IMÁGENES

% porcentaje de seguridad en cada flor
Estados_Rosas
FB = 0;
FT = 0;
FV = 0;

for c = 1:cantidadfotos % ciclo para conocer la cantidad y el % de imágenes
en las categorías establecidas

    j = c;
    categoria_actual = Estados_Rosas(j);

    if (categoria_actual == 'F_Buena') % # de imágenes que se consideran
flores buenas

        FB = FB + 1; % cantidades de imágenes

    elseif(categoria_actual == 'F_Torcida') % # de imágenes que se consideran
flores torcidas

        FT = FT + 1; % cantidades de imágenes

    elseif(categoria_actual == 'F_Velloso') % # de imágenes que se consideran
flores con velloso

        FV = FV + 1; % cantidades de imágenes

    end

end % FIN DEL BUCLE DEPENDIENDO DEL NUMERO DE IMÁGENES

PFBuena = FB/cantidadfotos % porcentaje flor buena

```

```

PFTorcida = FT/cantidadfotos % porcentaje flor torcida
PFVelloso = FV/cantidadfotos % porcentaje flor con velloso

PFLOR_BCAL = PFTorcida + PFVelloso % porcentaje flor baja calidad
PFLOR_TEXPOR = PFBuena % porcentaje flor tipo exportación

if PFLOR_BCAL >= 0.6

    Porcientos = PFLOR_BCAL*100;
    ResClas = 1;

    if PFTorcida >= PFVelloso
        mayormente = 0; % 0 significa mayormente Torcida

    elseif PFVelloso > PFTorcida
        mayormente = 1; % 1 significa mayormente velloso

    end

elseif PFLOR_BCAL < 0.6
    Porcientos = PFLOR_TEXPOR*100;
    ResClas = 2;
    mayormente = 3;
end

calidad = [Porcientos ResClas mayormente];

end

```

Anexo 5

Código del botón “Empezar” de la interfaz configuración.

```

function pushbutton2_Callback(hObject, eventdata, handles)

load('modeloB.mat')

global nom_img1 BAPI_E BAPS_E BAMI_E BAMS_E BAGI_E BAGS_E BHPI_E BHPS_E
BHMI_E BHMS_E BHGI_E BHGS_E noroja
nom_img1 = get(handles.edit9, 'String'); % en esta caja se anota el nombre de
la carpeta que contiene las imágenes

global nom_img2 selector1 selector2 selector3 contaflores FMP FANP FANM FANG
FBTP FALP FALM FALG Tallo_E Tallo_F Tallo_S
nom_img2 = get(handles.edit10, 'String'); % en esta caja se anota el nombre de
la carpeta que contiene las imágenes

    BAPI_E = str2double(get(handles.edit11, 'String')); % Botón Anchura Pequeña
lim Inferior
    BAPS_E = str2double(get(handles.edit12, 'String'));
    BAMI_E = str2double(get(handles.edit13, 'String'));
    BAMS_E = str2double(get(handles.edit14, 'String'));
    BAGI_E = str2double(get(handles.edit15, 'String'));

```

```

BAGS_E = str2double(get(handles.edit16,'String'));

BHPI_E = str2double(get(handles.edit11,'String'));%Botón Altura pequeña
lim Inferior
BHPS_E = str2double(get(handles.edit12,'String'));
BHMI_E = str2double(get(handles.edit13,'String'));
BHMS_E = str2double(get(handles.edit14,'String'));
BHGI_E = str2double(get(handles.edit15,'String'));
BHGS_E = str2double(get(handles.edit16,'String'));

selector1 = get(handles.radiobutton4,'Value'); % Revisa si el selector
está activado
selector2 = get(handles.radiobutton5,'Value'); % Revisa si el selector
está activado
selector3 = get(handles.radiobutton6,'Value'); % Revisa si el selector
está activado

contaflores = 0;
FMP = 0;
FANP = 0;
FANM = 0;
FANG = 0;
FBTP = 0;
FALP = 0;
FALM = 0;
FALG = 0;
Tallo_E = 0;
Tallo_F = 0;
Tallo_S = 0;
noroja = 0;

```

Interfaz_PDG

Anexo 6

Código del botón “Evaluar Calidad” de la interfaz principal.

```

function pushbutton6_Callback(hObject, eventdata, handles)

global nom_img1 nom_img2 BAPI_E BAPS_E BAM_I_E BAMS_E BAGI_E BAGS_E BHPI_E
BHPS_E BHMI_E BHMS_E BHGI_E BHGS_E noroja
global selector1 selector2 selector3 contaflores FMP FANP FANM FANG FBTP FALP
FALM FALG Tallo_E Tallo_F Tallo_S

contaflores = contaflores + 1;

%-----Limpiar ventanas .....
set(handles.text47,'string','FLOR NUEVA') %Escribe el texto en text
set(handles.text48,'string','-') %Escribe el texto en text
set(handles.text49,'string','-') %Escribe el texto en text
set(handles.text50,'string','-') %Escribe el texto en text
set(handles.text74,'string','-') %Escribe el texto en text

```



```

%-----Limpiar ventanas .....

ima_nueva = nom_img1; % Nombre de la carpeta con imagenes para ser revisadas
ima_rcompleta = nom_img2; % Nombre de la carpeta con imagenes para ser
revisadas

set_imag =imageSet(ima_nueva);
mostrarnueva = read(set_imag,1);
imshow(mostrarnueva)
axes(handles.axes6);

% determinar color -----

color_flor = deter_color(nom_img1);

if color_flor == 1
    set(handles.text47,'string','ES ROJA') %Escribe el texto en text
    Rev_Calidad = clasificador_entrenado(nom_img1); % FUNCION QUE LLAMA AL
CLASIFICADOR                                     % ENTRENADO
else
    set(handles.text47,'string','NO ES ROJA') %Escribe el texto en text
    noroja = noroja + 1;
    Rev_Calidad(2) = 0;
    set(handles.text91,'string',noroja) %Escribe el texto en text
end

% determinar color -----

if Rev_Calidad(2) == 0
    set(handles.text47,'string','NO ES ROJA') %Escribe el texto en text

    if selector1 == 1
        nro_rosas = contaflores;
        c =
categorical({'F.Al.Pq','F.Al.Me','F.Al.Gr';'F.Nacional','F.Exp','F.Total'});
        TOTALES_FINALES = [FALP FALM FALG; FMP FBTP nro_rosas];
        bar(handles.axes7,c,TOTALES_FINALES)
    end
    if selector2 == 1
        nro_rosas = contaflores;
        c =
categorical({'F.An.Pq','F.An.Me','F.An.Gr';'F.Nacional','F.Exp','F.Total'});
        TOTALES_FINALES = [FANP FANM FANG; FMP FBTP nro_rosas];
        bar(handles.axes7,c,TOTALES_FINALES)
    end
    if selector3 == 1
        nro_rosas = contaflores;
        c = categorical({'F.Nacional','F.Exp','F.Total'});
        TOTALES_FINALES = [FMP FBTP nro_rosas];
        bar(handles.axes7,c,TOTALES_FINALES)
    end
end
end

```

```

if (Rev_Calidad(2) == 2)

    if selector1 == 1
        mediciones =
medir_flores(nom_img1,BAPI_E,BAPS_E,BAMI_E,BAMS_E,BAGI_E,BAGS_E,BHPI_E,BHPS_E
,BHMI_E,BHMS_E,BHGI_E,BHGS_E)
        seleccion = mediciones;
        set(handles.text49,'string','FLOR TIPO EXPORTACIÓN') %Escribe el
texto en text
        set(handles.text50,'string',Rev_Calidad(1)) %Escribe el texto en text
    end

    if selector2 == 1
        mediciones =
medir_flores(nom_img1,BAPI_E,BAPS_E,BAMI_E,BAMS_E,BAGI_E,BAGS_E,BHPI_E,BHPS_E
,BHMI_E,BHMS_E,BHGI_E,BHGS_E)
        seleccion = mediciones;
        set(handles.text49,'string','FLOR TIPO EXPORTACIÓN') %Escribe el
texto en text
        set(handles.text50,'string',Rev_Calidad(1)) %Escribe el texto en text
    end

    if selector3 == 1
        set(handles.text48,'string','No medir') %Escribe el texto en text
        set(handles.text49,'string','FLOR TIPO EXPORTACIÓN') %Escribe el
texto en text
        set(handles.text50,'string',Rev_Calidad(1)) %Escribe el texto en text
        FBTP = FBTP + 1; % FLOR BUENA TOTAL PROCESADA
        nro_rosas = contaflores;
        c = categorical({'F.Nacional','F.Exp','F.Total'});
        TOTALES_FINALES = [FMP FBTP nro_rosas];
        bar(handles.axes7,c,TOTALES_FINALES)
    end

    if selector1 == 1 %---Inicio de selector 1 -----

        seleccion_user = 'alto';

        if seleccion(2) == 1 % CONDICIONES PARA DETERMINAR LA ALTURA DE LA
FLOR
            set(handles.text48,'string','Deforme Altura Pequeña') %Escribe el
texto en text
            set(handles.text49,'string','FLOR NACIONAL,DEFORME PEQUEÑA')
            FMP = FMP + 1;
            %medidaH = 1;

        end
        if seleccion(2) == 2
            set(handles.text48,'string','Altura Pequeña') %Escribe el texto
en text
            FALP = FALP + 1;
            FBTP = FBTP + 1; % FLOR BUENA TOTAL PROCESADA

```

```

end
if seleccion(2) == 3
    set(handles.text48,'string','Altura Mediana') %Escribe el texto
en text
    FALM = FALM + 1;
    FBTP = FBTP + 1; % FLOR BUENA TOTAL PROCESADA

end
if seleccion(2) == 4

    set(handles.text48,'string','Altura Grande') %Escribe el texto en
text
    FALG = FALG + 1;
    FBTP = FBTP + 1; % FLOR BUENA TOTAL PROCESADA
end
if seleccion(2) == 5
    set(handles.text48,'string','Deforme Altura Grande') %Escribe el
texto en text
    set(handles.text49,'string','FLOR NACIONAL,DEFORME GRANDE')
    FMP = FMP + 1;
end
nro_rosas = contaflores;
c =
categorical({'F.Al.Pq','F.Al.Me','F.Al.Gr';'F.Nacional','F.Exp','F.Total'});
TOTALES_FINALES = [FALP FALM FALG; FMP FBTP nro_rosas];
bar(handles.axes7,c,TOTALES_FINALES)

end % ----- Fin de selector 1 -----

if selector2 == 1 %---Inicio de selector 2 -----

    seleccion_user = 'ancho';

    if seleccion(1) == 1
        set(handles.text48,'string','Deforme Anchura Pequeña') %Escribe
el texto en text
        set(handles.text49,'string','FLOR NACIONAL,DEFORME PEQUEÑA')
        FMP = FMP +1;

    end
    if seleccion(1) == 2
        set(handles.text48,'string','Anchura Pequeña') %Escribe el texto
en text
        FANP = FANP + 1;
        FBTP = FBTP + 1; % FLOR BUENA TOTAL PROCESADA

    end
    if seleccion(1) == 3
        set(handles.text48,'string','Anchura Mediana') %Escribe el texto
en text
        FANM = FANM + 1;
        FBTP = FBTP + 1; % FLOR BUENA TOTAL PROCESADA

    end
    if seleccion(1) == 4

```

```

        set(handles.text48,'string','Anchura Grande') %Escribe el texto
en text
        FANG = FANG + 1;
        FBTP = FBTP + 1; % FLOR BUENA TOTAL PROCESADA

    end
    if seleccion(1) == 5
        set(handles.text48,'string','Deforme Anchura Grande') %Escribe
el texto en text
        set(handles.text49,'string','FLOR NACIONAL,DEFORME GRANDE')
        FMP = FMP + 1;

    end
    nro_rosas = contaflores;
    c =
categorical({'F.An.Pq','F.An.Me','F.An.Gr';'F.Nacional','F.Exp','F.Total'});
    TOTALES_FINALES = [FANP FANM FANG; FMP FBTP nro_rosas];
    bar(handles.axes7,c,TOTALES_FINALES)

end% ----- Fin de selector 2 -----

% inicio Clasificacion por tallos -----
set(handles.text74,'string','Midiendo Tallo...') %Escribe el texto en
text
LongTallo = medir_tallos(ima_rcompleta);

if LongTallo >= 40 && LongTallo < 50
    set(handles.text74,'string','Estándar') %Escribe el texto en text
    Tallo_E = Tallo_E + 1;
end

if LongTallo >= 50 && LongTallo < 60
    set(handles.text74,'string','Fancy') %Escribe el texto en text
    Tallo_F = Tallo_F + 1;
end

if LongTallo >= 60
    set(handles.text74,'string','Select') %Escribe el texto en text
    Tallo_S = Tallo_S + 1;
end
Grupo = categorical({'T.Estandar','T.Fancy','T.Select'});
TOTALES_Tallos = [Tallo_E Tallo_F Tallo_S];
bar(handles.axes8,Grupo,TOTALES_Tallos)

% fin Clasificacion por tallos -----

end

if (Rev_Calidad(2) == 1)
    set(handles.text47,'string','ES ROJA') %Escribe el texto en text
    set(handles.text48,'string','-')
    set(handles.text49,'string','FLOR NACIONAL') %Escribe el texto en text
    set(handles.text50,'string',Rev_Calidad(1)) %Escribe el texto en text

```

```

FMP = FMP +1;

if selector1 == 1
    nro_rosas = contaflores;
    c =
categorical({'F.Al.Pq', 'F.Al.Me', 'F.Al.Gr'; 'F_Nacional', 'F.Exp', 'F.Total'});
    TOTALES_FINALES = [FALP FALM FALG; FMP FBTP nro_rosas];
    bar(handles.axes7,c,TOTALES_FINALES)
end

if selector2 == 1
    nro_rosas = contaflores;
    c =
categorical({'F.An.Pq', 'F.An.Me', 'F.An.Gr'; 'F.Nacional', 'F.Exp', 'F.Total'});
    TOTALES_FINALES = [FANP FANM FANG; FMP FBTP nro_rosas];
    bar(handles.axes7,c,TOTALES_FINALES)
end

if selector3 == 1
    nro_rosas = contaflores;
    c = categorical({'F.Nacional', 'F.Exp', 'F.Total'});
    TOTALES_FINALES = [FMP FBTP nro_rosas];
    bar(handles.axes7,c,TOTALES_FINALES)
end

if Rev_Calidad(3) == 0
    set(handles.text49, 'string', 'FLOR NACIONAL,MÁS TORCIDA') %Escribe el
texto en text
end

if Rev_Calidad(3) == 1
    set(handles.text49, 'string', 'FLOR NACIONAL,MAS VELLOSO') %Escribe el
texto en text
end
end

R_Nal = (FMP * 100)/contaflores
set(handles.text75, 'string',R_Nal) % % de rosa nacional

R_Buena = (FBTP * 100)/contaflores
set(handles.text81, 'string',R_Buena) % % de rosa nacional

%----- Porcientos -----

if selector1 == 1
    PALP = FALP*100/FBTP;
    PALM = FALM*100/FBTP;
    PALG = FALG*100/FBTP;
    set(handles.text87, 'string',PALP) % % de rosa nacional
    set(handles.text88, 'string',PALM) % % de rosa nacional
    set(handles.text89, 'string',PALG) % % de rosa nacional

end

```

```

if selector2 == 1
    PANP = FANP*100/FBTP;
    PANM = FANM*100/FBTP;
    PANG = FANG*100/FBTP;

    set(handles.text87,'string',PANP) %% de rosa nacional
    set(handles.text88,'string',PANM) %% de rosa nacional
    set(handles.text89,'string',PANG) %% de rosa nacional

end

if selector3 == 1

    R_Nal = (FMP * 100)/contaflores
    set(handles.text75,'string',R_Nal) %% de rosa nacional

    R_Buena = (FBTP * 100)/contaflores
    set(handles.text81,'string',R_Buena) %% de rosa nacional

end
%----- Porcientos -----
axes(handles.axes6);

```