UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE POSGRADO

MAESTRÍA EN INGENIERÍA MECATRÓNICA

"SISTEMA DE CLASIFICACIÓN CON VISIÓN E INTELIGENCIA ARTIFICIAL PARA LA OPTIMIZACIÓN DEL PROCESO DE POSTCOSECHA DE FOLLAJE RUSCUS"

Proyecto del Trabajo de Titulación previo a la obtención del Título de Magíster en Mecatrónica: Mención Procesos Industriales

AUTOR: MSc. Fernando Wladimir Ortega Loza, Ing.

TUTOR: MSc. Darío Fernando Yépez Ponce, Ing.

IBARRA - ECUADOR

2023

CERTIFICACIÓN DEL DIRECTOR DE TESIS

Como director del trabajo de investigación con el tema: "SISTEMA DE CLASIFICACIÓN CON VISIÓN E INTELIGENCIA ARTIFI-CIAL PARA LA OPTIMIZACIÓN DEL PROCESO DE POSTCOSECHA DE FOLLAJE RUSCUS", trabajo que fue realizado por Ortega Loza Fernando Wladimir, previo a la obtención del título de Magister en mecatrónica mención procesos industriales, doy fe de que la obra mencionada reúne los requisitos y méritos suficientes para ser públicamente sustentada en juicio para ser oportunamente aprobada.

Ibarra, 20 de mayo de 2023

MSc, Dario Fernando Yepez Ponce

C.C.: 1004182000

Director de Tesis

UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajoa la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1002635793		
APELLIDOS Y NOMBRES:	Ortega Loza Fernando Wladimir		
DIRECCIÓN:	Calle 5 de junio C3 y Santiago Garrido		
EMAIL:	fwortega@utn.edu.ec		
TELÉFONO FIJO:	062933283	**TELÉFONO MÓVIL:**	0981846427

DATOS DE LA OBRA	
TÍTULO:	Sistema de clasificación con Visión e Inteligencia Artificial para la optimización del proceso de postcosecha de follaje Ruscus.
AUTOR (ES):	Fernando Wladimir Ortega Loza
FECHA: DD/MM/AAAA	21/04/2023
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	☐ **PREGRADO** ☒ **POSGRADO**
TITULO POR EL QUE OPTA:	Magister en Mecatrónica mención Procesos Industriales
ASESOR /DIRECTOR:	Ing. Fernando Yépez, MSc.

2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 9 días del mes de noviembre de 2023

EL AUTOR:

FERNANDO WLADIMIR ORTEGA LOZA

Digitally signed by FERNANDO WLADIMIR ORTEGA LOZA
Date: 2023.11.09 11:36:20 -05'00'

……

Fernando Wladimir Ortega Loza

REGISTRO BIBLIOGRÁFICO

Guide: POSGRADO – UTN

Date: Ibarra, 20 de mayo de 2023

Ortega Loza Fernando Wladimir: "SISTEMA DE CLASIFICACIÓN CON VISIÓN E INTELIGENCIA ARTIFI-CIAL PARA LA OPTIMIZACIÓN DEL PROCESO DE POSTCOSECHA DE FOLLAJE RUSCUS" / **GRADO MAGISTER EN MECATRÓNICA MENCIÓN PROCESOS INDUSTRIALES, Universidad Técnica del Norte, Ibarra.**

DIRECTOR: MSc. Darío Fernando Yépez Ponce

El objetivo general de esta tesis fue: Implementar un sistema de clasificación con visión e inteligencia artificial para la optimización del proceso de postcosecha de follaje Ruscus.

Entre los objetivos específicos estaban: Identificar los parámetros representativos del proceso de postcosecha, visión e inteligencia artificial para clasificación de Ruscus. Diseñar un algoritmo de visión e inteligencia artificial para clasificar follaje de Ruscus según los clústeres planteados. Probar el algoritmo de clasificación mediante una base datos fotográfica de validación. Implementar el algoritmo de clasificación como asistente en tiempo real del proceso post de cosecha de Ruscus.

MSc. Darío Fernando Yépez Ponce

Director

FERNANDO WLADIMIR ORTEGA LOZA Digitally signed by FERNANDO WLADIMIR ORTEGA LOZA Date: 2023.11.09 11:36:46 -05'00'

MSc. Fernando Wladimir Ortega Loza

Autor

DEDICATORIA

Este trabajo está dedicado a mi esposa Johana Fuentes, a mis hijos Alejandro y Sebastián Ortega ya que ellos han sido, son y serán mi motivación en todos los aspectos de mi vida, todo mi esfuerzo converge en el amor hacia ellos. Alejandro y Sebastián que este trabajo sea un pequeño ejemplo de que la perseverancia y la pasión que se ponen en los proyectos emprendidos al final tiene su recompensa y más allá de un título esta la satisfacción de la culminación.

Fernando

AGRADECIMIENTOS

Mi gratitud a todos los profesores de la maestría, en especial a mi tutor el MSc. Fernando Yépez y al coordinador de este programa, el MSc. Cosme Mejía, sepan que ustedes son gigantes en cuyos hombros he subido y he podido ver más allá.

Fernando

ÍNDICE DE CONTENIDOS

DEDICATORIA 6

AGRADECIMIENTOS 7

AUTORIZACIÓN DE USO Y PUBLICACIÓN 3

ÍNDICE DE CONTENIDOS 8

ÍNDICE DE FIGURAS 12

ÍNDICE DE TABLAS 14

RESUMEN 15

ABSTRACT 16

CAPITULO I. EL PROBLEMA 1

1.1. Planteamiento del problema 1

1.2. Objetivos de la investigación 2

1.2.1. Objetivo general 2

1.2.1. Objetivos específicos 2

1.3. Justificación 3

1.4. Alcance 5

CAPITULO II. MARCO REFERENCIAL 6

2.1. Antecedentes 6

2.2. Marco Teórico 7

2.2.1. Ruscus 7

2.2.2. Software Matlab 7

2.2.3. Visión Artificial 8

La luz. 8

Distribución espectral de energía 8

La imagen digital. 9

La cámara 9

Proceso de Visión Artificial 9

Escala de Grises. 10
Binarización por umbral (Threshold binarization). 10
Sustracción de fondo (Background substraction). 11
Mediana. 11
Erosión y Dilatación. 11
Bag of features. 12
2.2.4. Inteligencia Artificial 12
Redes Neuronales. 13
Neurona Biológica. 13
Neurona artificial. 13
Perceptrón. 16
Adaline 17
Perceptrón multicapa 18
Red neuronal Backpropragation. 18
Red Neuronal de Hopfield. 21
Red neuronal de Kohonen 21
Red de base radial. 22
Aprendizaje automático supervisado 22
Regresión Lineal. 23
Regresión Logística. 24
Naive Bayes. 25
Máquinas de soporte vectorial 26
Árboles de decisión 27
Clasificador KNN 28
2.2.4.1. Aprendizaje no supervisado 29
Algoritmos Genéticos 29
Clustering. 29

Reducción de dimensionalidad. 30
2.3. Marco Legal 31
CAPITULO III. MARCO METODOLÓGIO 32
3.1. Descripción del área de estudio 32
3.2. Enfoque y tipo de investigación 32
3.3. Procedimiento 33
3.4. Consideraciones Bioéticas 35
CAPITULO IV. RESULTADOS Y DISCUCIÓN 36
4.1. Fase1: Identificación 36
4.2. Fase 2: Diseño 37
4.3. Fase 3: Realización de pruebas 40
4.3.1. Árbol de decisión fina. La Figura …. 40
4.3.2. Árbol de decisión media. 41
4.3.3. Árbol de decisión gruesa. 41
4.3.4. Regresión Logística Binaria. 42
4.3.5. Regresión Logística Eficiente. 42
4.3.6. Naive Bayes Gaussina. 43
4.3.7. Kernel Naive Bayes. 43
4.3.8. SVM Lineal. 44
4.3.9. SVM Cuadrática. 44
4.3.10. SVM Cúbica. 45
4.3.11. SVM Gaussiana Fina. 45
4.3.12. SVM Gaussiana Media. 46
4.3.13. SVM Gaussiana Gruesa. 46
4.3.14. KNN Fina. 47
4.3.15. KNN Media. 47
4.3.16. KNN Gruesa. 48

4.3.17. KNN Coseno. 48
4.3.18. KNN Cúbico. 49
4.3.19. KNN Ponderado. 49
4.3.20. Red Neuronal Estrecha. 50
4.3.21. Red Neuronal Media. 50
4.3.22. Red Neuronal Amplia. 51
4.3.23. Red Neuronal Bicapa. 51
4.3.24. Red Neuronal de tres capas. 52
4.4. Fase 4: Implementación del algoritmo 54
CAPITULO V. CONCLUSIONES Y RECOMENDACIONES 56
5.1. Conclusiones 56
5.2. Recomendaciones 57
REFERENCIAS 58
ANEXOS 64
Anexo 1. Guión de la Entrevista 64
Anexo 2. Muestra Fotográfica 65
Anexo 3. Script de Medición 66
Anexo 4. Script de Entrenamiento y Ejecución 68
Anexo 5. Función de tiempo real 69

ÍNDICE DE FIGURAS

Figura 2.1. Espectro visible y no visible..... 8
Figura 2.2. Luz espectral de: (a) Luz verde. (b) Luz blanca..... 8
Figura 2.3. Proceso de visión artificial 9
Figura 2.4. Transformación de una imagen R.G.B. a Escala de Grises..... 10
Figura 2.5. Imagen binarizada por umbral..... 10
Figura 2.6. Sustracción de fondo 11
Figura 2.7. (a)Imagen original, (b) Imagen aplicando Mediana..... 11
Figura 2.8. Proceso de extracción de características mediante “Bag of features” 12
Figura 2.8. Esquema de una neurona artificial 14
Figura 2.9. Tipos de funciones de activación 15
Figura 2.10. Arquitectura de una red neuronal..... 15
Figura 2.11. Red neuronal a) Monocapa, b) Multicapa, c) Recurrente 16
Figura 2.12. Red neuronal Perceptrón 17
Figura 2.13. Clasificación mediante perceptrón..... 17
Figura 2.14. Red neuronal Adaline..... 18
Figura 2.15. Red neuronal Perceptrón multicapa 18
Figura 2.16. Sobre entrenamiento de una rede neuronal 21
Figura 2.17. Red neuronal Hopfield 21
Figura 2.18. Red neuronal de base radial 22
Figura 2.19. Ejemplo de clasificador SVM 27
Figura 2.20. Comparación de precisión entre una Regresesión Logistica y un Árbol de Desición..... 27
Figura 2.21. Comparación de clasificadores KNN con tamaño a) K=1 y b) K=20 28
Figura 3.1. Localización de la plantación de Ruscus GM Familiar..... 32
Figura 4.1. Algoritmo de medición de tallo..... 38
Figura 4.2. Matriz de confusión del modelo: Árbol de decisión fina 41
Figura 4.3. Matriz de confusión del modelo: Árbol de decisión medio 41
Figura 4.4. Matriz de confusión del modelo: Árbol de decisión gruesa..... 42
Figura 4.5. Matriz de confusión del modelo: Regresión logística binaria..... 42
Figura 4.6. Matriz de confusión del modelo: Regresión logística eficiente 43
Figura 4.7. Matriz de confusión del modelo: Naive Bayes Gaussina..... 43

Figura 4.8. Matriz de confusión del modelo: Naive Bayes Kernel 44
Figura 4.9. Matriz de confusión del modelo: SVM Lineal..... 44
Figura 4.10. Matriz de confusión del modelo: SVM Cuadrática..... 45
Figura 4.11. Matriz de confusión del modelo: SVM Cúbica..... 45
Figura 4.12. Matriz de confusión del modelo: SVM Gaussiana Fina 46
Figura 4.13. Matriz de confusión del modelo: SVM Gaussiana Media 46
Figura 4.14. Matriz de confusión del modelo: SVM Gaussiana Gruesa 47
Figura 4.15. Matriz de confusión del modelo: KNN Fina..... 47
Figura 4.16. Matriz de confusión del modelo: KNN media 48
Figura 4.17. Matriz de confusión del modelo: KNN gruesa..... 48
Figura 4.18. Matriz de confusión del modelo: KNN Coseno 49
Figura 4.19. Matriz de confusión del modelo: KNN Cúbico 49
Figura 4.20. Matriz de confusión del modelo: KNN Ponderado..... 50
Figura 4.21. Matriz de confusión del modelo: Red Neuronal Estrecha..... 50
Figura 4.22. Matriz de confusión del modelo: Red Neuronal Media 51
Figura 4.23. Matriz de confusión del modelo: Red Neuronal Amplia 51
Figura 4.24. Matriz de confusión del modelo: Red Neuronal Bicapa 52
Figura 4.25. Matriz de confusión del modelo: Red Neuronal de tres capas..... 52
Figura 4.26. Diagrama de dispersión entre dos características del modelo Lineal SVM 54
Figura 4.27. Flujograma de entrenamiento y clasificación en tiempo real..... 55

ÍNDICE DE TABLAS

Tabla 1. Comparativa de posible software a utilizar 37
Tabla 2. Resultados de exactitud de modelos 53

UNIVERSIDAD TÉCNICA DEL NORTEFACULTAD DE POSGRADO

PROGRAMA DE MAESTRÍA MECATRÓNICA
MENCIÓN PROCESOS INDUSTRIALES

SISTEMA DE CLASIFICACIÓN CON VISIÓN E INTELIGENCIA ARTIFICIAL PARA LA OPTIMIZACIÓN DEL PROCESO DE POSTCOSECHA DE FOLLAJE RUSCUS

Autor: Fernando Wladimir Ortega Loza

Tutor: Darío Fernando Yépez Ponce

Año: 2023

RESUMEN

En la presente investigación se implementó un sistema de clasificación con visión e inteligencia artificial para la optimización del proceso de postcosecha de follaje ornamental Ruscus en la plantación GM Familiar. Actualmente, el proceso se lo hace manualmente bajo el criterio subjetivo de la persona encargada de la clasificación, lo cual conlleva a un detrimento de la calidad y una amenaza en el nicho de mercado logrado. Con la finalidad de solventar esta problemática se realizó una revisión sistemática de literatura acerca de la visión e inteligencia artificial; así mismo, se investigó las características morfológicas intervinientes de la planta para su clasificación. Para llevar a cabo la clasificación, se creó una base de datos de imágenes sobre las cuales se detecta el tamaño del tallo mediante técnicas de visión artificial y se extraen características morfológicas de las plantas mediante el procedimiento *Bag of Features*. Se realizó el entrenamiento de 24 modelos con diferentes técnicas de inteligencia artificial que permitieron la clasificación entre plantas en buen y mal estado. Los mejores resultados se obtuvieron con el modelo de Máquina de Soporte Vectorial Lineal, el cual arrojó una precisión del 88.6% mediante la validación cruzada de cinco elementos. Finalmente, el modelo del algoritmo fue implementado en Matlab, como un aplicativo en tiempo real corriendo en Windows, y se utilizó una cámara web con condiciones mejoradas de iluminación para realizar la clasificación automática en tiempo real de los tallos de Ruscus.

Palabras clave: Clasificación Automática, Visión Artificial, Inteligencia Artificial, Aprendizaje Automático, Clasificación de Ruscus.

UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE POSGRADO

PROGRAMA DE MAESTRÍA MECATRÓNICA
MENCIÓN PROCESOS INDUSTRIALES

VISION AND ARTIFICIAL INTELLIGENCE CLASIFICATION SYSTEM FOR THE OPTIMIZATION OF POST-HARVEST PROCESS OF RUSCUS FOLIAGE

Author: Fernando Wladimir Ortega Loza

Tutor: Nombre completo del tutor

Año: 2023

ABSTRACT

The objective of this research is to implement a classification system with vision and artificial intelligence for the optimization of the postharvest process of Ruscus ornamental foliage in the GM Family plantation, which is currently done manually and in a subjective way according to the criteria of the person who is being classified, which leads to a loss of quality and a threat to the achieved market niche. To solve this problem, an investigation of the morphological characteristics of the plant involved in its classification was carried out, as well as a systematic review of the literature on vision and artificial intelligence. In this sense, an image database was built on which the size of the stem is detected by means of artificial vision techniques and morphological characteristics of the plants are extracted by means of the “Bag of Features” procedure to then carry out the analysis training of 24 models with different artificial intelligence techniques that allow the classification between good and bad plants, from this process, the Linear Vector Support Machine model was selected, which has an accuracy of 88.6% through cross validation of five elements. Finally, the model was then implemented in a real-time application with Matlab running in Windows, and that allows automatic classification of Ruscus stems, using a webcam and improved lighting conditions.

Palabras clave: Automatic classification, Artificial vision, Artificial Intelligence, Machine Learning, Ruscus.

CAPITULO I
EL PROBLEMA

1.1. Planteamiento del problema

La exportación de flores y follaje a lo largo de la historia ha aportado significativamente en la economía del Ecuador siendo las flores el producto más representativo de este nicho; sin embargo, se está cultivando cada vez más otros tipos de plantas ornamentales como es el Ruscus Hypophyllum (Ruscus). La Ruscus es una planta de follaje cuya principal característica es la resistencia a la degradación, lo cual la hace una planta apetecible en el mercado internacional, siendo los principales destinos EEUU, la Unión Europea y Rusia (Díaz, 2019). La mayor producción de Ruscus se encuentra en la Sierra norte (Campaña et al., 2014). En Imbabura, existen varios invernaderos que están cultivando esta planta como es el caso de la plantación GM Familiar que se encuentra ubicada en la parroquia de Quiroga en el cantón Cotacachi. La plantación GM Familiar está produciendo Ruscus desde el año 2019 y es la única de la parroquia dedicada a esta actividad.

Actualmente, la plantación posee 7 000 m² cultivados y cuenta con 8 0000 m² en terrenos disponibles para escalar la producción. El Sr. Fabian Guerra, productor de Ruscus en GM Familiar manifiesta que la producción promedio es de 25 000 tallos mensuales; sin embargo, se han registrado pedidos de hasta 480 000 tallos mensuales. La demanda insatisfecha no puede ser suplida debido al deficiente proceso de clasificación de tallos, y a los limitados recursos económicos de la plantación. El factor económico, no permite contratar más personal para expandir la cosecha debido a que los miembros de la asociación se encuentran pagando un crédito por las tierras. La plantación se maneja con mano de obra familiar; es decir, las familias de los socios son quienes prestan su contingente.

La principal actividad económica de los trabajadores de la plantación es la ganadería, razón por la cual la producción de Ruscus la dejan en segundo limitando la producción del follaje. La cosecha y postcosecha de esta planta se lleva a cabo manualmente, dentro de la postcosecha los dos procesos principales son el control de calidad y la clasificación de los tallos por sus características morfológicas. Estos procesos toman la mayor cantidad de tiempo dentro del flujo de producción y es realizado por distintas personas; razón por la cual, son susceptibles a la subjetividad

contribuyendo al detrimento de la calidad del producto. Si esta situación persiste, se podría perder mercado al no cubrir la demanda insatisfecha y no tener un sistema de clasificación homogéneo que permita garantizar estándares de calidad. En la actualidad, esta problemática afecta a 25 personas que laboran en este invernadero y a futuro afectaría a la consecución de la visión de la asociación de ser un referente de producción local que genere empleo para la zona.

GM Familiar, necesita optimizar el proceso de clasificación de Ruscus para que sea más rápido y deje de lado la subjetividad que tiene el ser humano. La entidad define los siguientes clústeres de discriminación:

- Talla pequeña: de 25 a 35 cm.
- Talla Mediana: de 35 a 45 cm.
- Talla grande: de 45 a 55cm.
- Tallo en buenas condiciones: tallo cargado o medianamente cargado y que esté maduro según coloración.
- Tallo descartable: tallo poco cargado o por madurar según coloración del follaje y follaje enfermo o destruido.

Se propone responder a la necesidad planteada optimizando el proceso de clasificación de Ruscus mediante un dispositivo tecnológico basado en visión e inteligencia artificial.

1.2. Objetivos de la investigación

1.2.1. Objetivo general

Implementar un sistema de clasificación con visión e inteligencia artificial para la optimización del proceso de postcosecha de follaje Ruscus.

1.2.1. Objetivos específicos

- Identificar los parámetros representativos del proceso de postcosecha, visión e inteligencia artificial para clasificación de Ruscus.
- Diseñar un algoritmo de visión e inteligencia artificial para clasificar follaje de Ruscus según los clústeres planteados.
- Probar el algoritmo de clasificación mediante una base datos fotográfica de validación.

- Implementar el algoritmo de clasificación como asistente en tiempo real del proceso post de cosecha de Ruscus.

1.3. Justificación

Ecuador ha experimentado un crecimiento continuo en la exportación de productos agrícolas creciendo de 5 000 millones de dólares en el año 2009 a 6 866 millones de dólares en el año 2019. La actividad agrícola, aporta un 13% del PIB del Ecuador, de esta actividad un 13% a su vez pertenece a flores y follaje aproximadamente. (Banco Central del Ecuador, 2020, Expoflores, 2019, Telégrafo, 2020) siendo un elemento de aporte significativo en la economía del Ecuador. La presente investigación al estar relacionada con la optimización del proceso agrícola adquiere especial relevancia económica.

Según el estudio realizado por (Breilh, 2007), en las poblaciones agrícolas del norte del país se verificó que en el 31% de las familias al menos un miembro trabaja en el sector de las flores y el follaje. De estas personas, un 56% se encuentra en estado de estrés moderado y/o severo y un 43% con malnutrición. Las extendidas jornadas de trabajo de procedimientos manuales y repetitivos generan estrés y otras enfermedades (Gonzales *et al*., 2002). Los aspectos laborales anteriormente mencionados, pueden solucionarse reduciendo la jornada laboral y mejorando las condiciones ergonómicas. De igual manera, los sistemas tecnológicos que asisten al ser humano en una actividad repetitiva reduciendo el tiempo de la jornada laboral ayudan a aliviar el estrés y prevenir enfermedades; por esta razón, el trabajo planteado también es importante desde el punto de vista social (AEPSAL, 2017).

La agricultura de precisión, el uso de aprendizaje de máquinas y la visión artificial son tendencias en el campo de la mecatrónica (Erdélyi & Jánosi, s. f.; Hassanien et al., 2012; Sanchez Calle, 2005). El aporte generado en base a estas tendencias y la aplicación en el contexto local, brindan a la presente investigación relevancia académica y científica que sirven de base para el desarrollo de futuras aplicaciones en diferentes ámbitos. La implementación del sistema permite optimizar el proceso de postcosecha de Ruscus en la plantación GM Familiar, reduciendo el tiempo y mejorando la calidad de la clasificación del producto final. Las mejores anteriormente mencionadas, aumentan la producción, contratación de personal y capacidad de cubrir más pedidos lo que conlleva a un mayor beneficio económico para los participantes de este emprendimiento.

1.4. Alcance

La presente investigación se realiza con muestras y requerimientos de la plantación de Ruscus GM Familiar. Se implementó un sistema de asistencia para la clasificación de tallos, el cual utiliza visión artificial mediante una cámara web, inteligencia artificial y una red neuronal cuyos parámetros de entrada son al menos dos características morfológicas del tallo de Ruscus. Los clústeres de salida son el conjunto de clasificación de talla y si el tallo está en buen estado y/o es descartable. El software utilizado es Matlab, debido a que la Universidad Técnica del Norte posee la licencia. La licencia del software es válida para el ámbito educativo e investigativo; en consecuencia, para ser utilizado por la plantación, tienen que adquirir una licencia comercial del software y/o migrar a un programa de licencia libre, lo cual no se contempla en la presente investigación.

Finalmente, la presente investigación no contempla hardware de clasificación que ejecute acciones automáticas de separación; razón por la cual, esta parte queda para futuros proyectos los cuales se basen en el algoritmo desarrollado en esta investigación.

CAPITULO II
MARCO REFERENCIAL

2.1. Antecedentes

Desde principios del siglo XX se ha estudiado la morfología del Ruscus Hypophyllum. En (Arber, 1924), se describen las características de las hojas, tallos y frutos. Algunas páginas webs como (*Brusco de hojas anchas, Laurelillo, Laurel de Alejandría, Planta de la mosquita, Ruscus - Ruscus hypoglossum*, s. f.; *Laurel de Alejandria-Laurelillo-Ruscus Hypoglossum – Para Mi Jardín*, s. f.) detallan las características físicas, reproductivas y de cuidado de la planta. En (Ramírez, 2010), explica la producción y tratamiento de postcosecha del Ruscus en Colombia.

Múltiples autores han aplicado la visión e inteligencia artificial para la clasificación de imágenes. En (Mohamed et al., 2015) utilizan un perceptrón multicapa para clasificar imágenes de electrocardiograma, obteniendo robustez en un descriptor tipo Haar-like (Haar) combinado con una red neuronal artificial. En (LeCun et al., 2010) emplean una red neuronal convolucional para clasificar etiquetas con letras, consiguieron una buena efectividad con un gran numero muestras. En (Batioua et al., 2018) comparan la clasificación mediante un algoritmo de invariantes de momento discreto clásico con un algoritmo de invariantes de momento discreto separable basado en polinomios Racah, concluyendo que este último expresa un avance significativo en la precisión del reconocimiento. En (Facco et al., 2017) aplican caracterización morfológica de imágenes para estimar el tamaño de la partícula en productos granulados mediante visión artificial.

Entre los trabajos relacionadas a la agricultura con visión e inteligencia artificial, se destacan las aportaciones realizadas por (Al Ohali, 2011), quien utiliza un algoritmo de propagación hacia atrás (*backpropagation*) para clasificar de manera binaria dátiles obteniendo un 80% de precisión. En (Fuentes et al., 2020) emplean un algoritmo de aprendizaje profundo (*deep learning*) para clasificar granos de café consiguiendo una eficiencia de clasificación del 97.6%. Por otra parte en (Andrea et al., 2017) hacen uso de una red neuronal convolucional para discriminar plantas de maíz de maleza alcanzando un 97.2% de precisión en el entrenamiento neuronal. En (Pantazi et al., 2020) se proporciona una descripción sobre los antecedentes científicos de la clasificación por máquinas vectoriales de soporte (SVM, por sus siglas en inglés) y los

algoritmos de aprendizaje activo y su aplicación en el campo de la agricultura de precisión.

Se puede interpretar a partir de los autores referenciados, que existen múltiples algoritmos de visión e inteligencia artificial que se han utilizado para la clasificación de imágenes y específicamente en la agricultura de precisión; sin embrago, no se evidencian trabajos de clasificación de Ruscus mediante visión e inteligencia artificial.

2.2. Marco Teórico

2.2.1. Ruscus

El nombre científico es Ruscus Hypoglossum, comúnmente conocido como brusco de hojas anchas, laurelillo, laurel de Alejandría, planta de mosquita o Ruscus. Pertenece a la familia Liliaceae y es de origen europeo, es un arbusto con ramas de consistencia herbácea, rizomatoso, ramificado de hasta un metro de altura. El Rusucs pertenece a las plantas dioicas con flores en la parte central del tallo aplanado en la axila de la bráctea transparente, si la planta se deja madurar lo suficiente posee frutos en forma de baya globosa de color rojo de un centímetro de diámetro aproximadamente (*Brusco de hojas anchas, Laurelillo, Laurel de Alejandría, Planta de la mosquita, Ruscus - Ruscus hypoglossum*, s. f.).

El uso del Ruscus es ornamental especialmente como fondo en arreglos florales o como soluciones para zonas de sombra media en jardinería, debido a que la principal característica del Ruscus es su resistencia a la degradación, puede conservarse hasta un año luego de ser cortado (Campaña et al., 2014).

2.2.2. Software Matlab

Según la empresa Mathworks “MATLAB es una plataforma de programación y cálculo numérico utilizada por millones de ingenieros y científicos para analizar datos, desarrollar algoritmos y crear modelos” (*MATLAB - El lenguaje del cálculo técnico*, 2023). Adicionalmente, este software cuenta con paquetes de herramientas de uso especifico (*Toolbox*) de Visión Artificial, Aprendizaje Automático, Inteligencia Artificial, entre muchos otros. El *Toolbox* de Visión por Computador de Matlab permite trabajar con algoritmos y aplicaciones para probar y diseñar sistemas de adquisición y procesamiento de imágenes (*Computer Vision Toolbox*, s. f.); por su parte, el *Toolbox* de Aprendizaje Automático y Estadística permite trabajar con algoritmos y

funciones para describir, analizar y modelar datos con diferentes técnicas utilizando herramientas estadísticas (*Statistics and Machine Learning Toolbox*, s. f.).

2.2.3. Visión Artificial

En (Maduell, 2012) se indica que la visión artificial es la ciencia y la tecnología que permite a las máquinas extraer información de las imágenes digitales para que mediante un procesamiento de esa información se puedan resolver tareas o entender la escena que se está adquiriendo.

La luz. En (Vélez Serrano, 2003) se considera desde el punto de vista del procesado de imágenes a la luz como una onda (ver Figura 2.1); en la cual, las características del rayo de luz vienen determinadas por la amplitud y la longitud de onda. El sistema sensorial humano interpreta diferentes amplitudes y longitudes de onda, lo cual se percibe como brillo y color respectivamente.

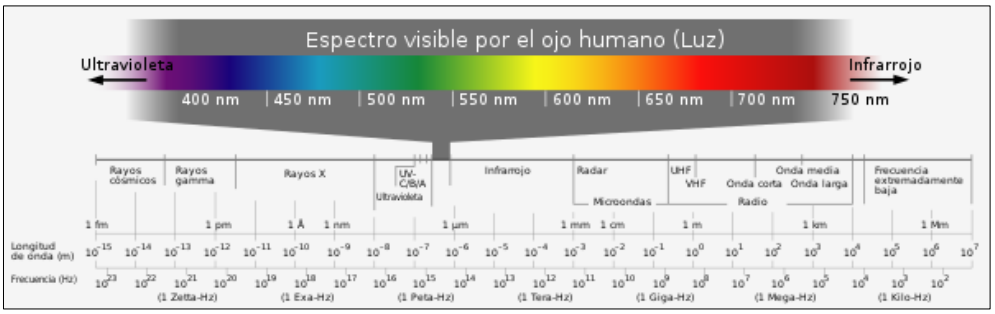


Figura 2.1. ***Espectro visible y no visible***
Fuente: Tomado de Espectro visible (2021), https://tinyurl.com/2naetrbw

Distribución espectral de energía. Es un diagrama que representa en el eje ordenado la energía en vatios y en el eje de las abscisas la longitud de onda. En general, las radiaciones vienen como una mezcla de diferentes longitudes de ondas, cada una de ellas con diferente potencia (Vélez Serrano, 2003). De esta manera, se forma una distribución espectral como la de la Figura 2.2.

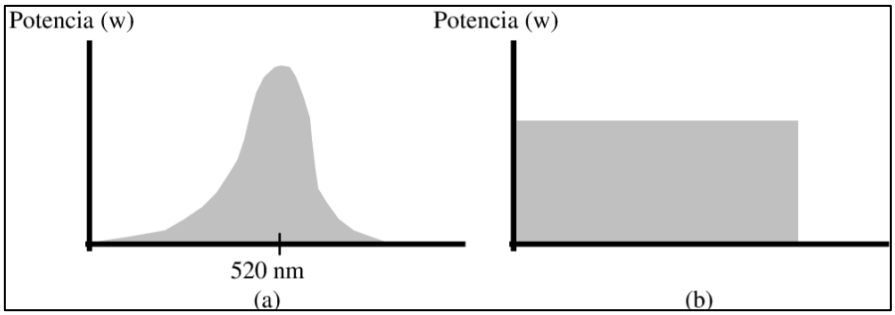


Figura 2.2. ***Luz espectral de: (a) Luz verde. (b) Luz blanca***

Fuente: Obtenido de Vélez Serrano (2003).

La imagen digital. La imagen digital se compone de una matriz bidimensional de unidades elementales de color llamados pixeles. Cada píxel puede adquirir valores de color codificados en base a tres parámetros RGB (*red, green, blue*). Los valores de R, G y B se suelen expresar en un dato de 8 bits; es decir, valores entre 0 y 255 (Maduell, 2012). Una característica importante a tener en cuenta en una imagen en movimiento, es la frecuencia de imagen (*frame rate*) que indica el número de imágenes por segundo de un video.

La cámara. Es un dispositivo sensible a la luz que permite el trabajo con visión artificial. Las cámaras web y la mayoría de cámaras de vídeo se pueden conectar directamente a los ordenadores por medio de los puertos USB, *FireWire* o *Thunderbolt* que permiten la captura de los fotogramas en tiempo real (Maduell, 2012).

Proceso de Visión Artificial. Por lo general, las técnicas de visión artificial siguen un proceso que inicia con la adquisición de la imagen, acto seguido se procesa para la extracción de datos; por último, se extrae información. Este proceso puede ser visto en la Figura 2.3.

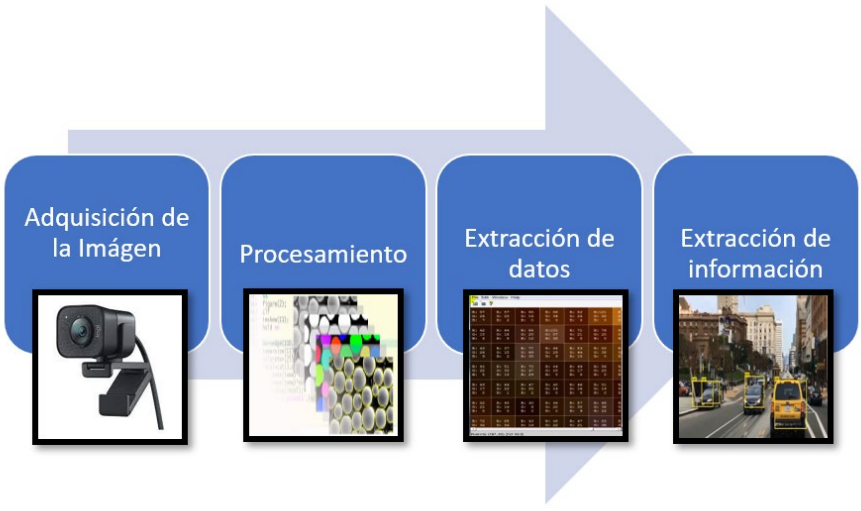


Figura 2.3. ***Proceso de visión artificial***

Fuente: Adaptado de Maduell (2012).

La adquisición de las imágenes se lleva a cabo por medio de sensores digitales provistos en las cámaras, los cuales generan una matriz bidimensional de $M \times N$ píxeles. Cada posición de la matriz tiene un arreglo de valores para el color rojo, verde y azul, la mezcla de los tres colores genera el espectro sensado en ese punto por la matriz.

El procesamiento se lo hace previo a extraer los datos de la imagen para facilitar el proceso de extracción de datos (Maduell, 2012). Los procesamientos más comunes se describen a continuación:

Escala de Grises. Las imágenes normalmente proporcionan tres números enteros en el rango de 0 a 255 por cada ubicación de la matriz. Para la mayoría de las aplicaciones, esta información es poco relevante o difícil de extraer; razón por la cual, se procede a obtener un solo número entero que proporciona información de la intensidad del píxel en escala de grises. Este procedimiento reduce la cantidad de memoria ocupada por la imagen manteniendo la información relevante. En la Figura … se puede observar la transformación de una imagen a color a escala de grises.

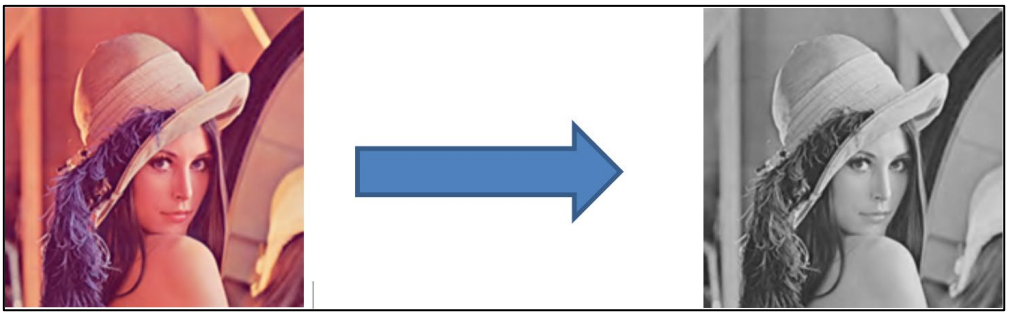


Figura 2.4. ***Transformación de una imagen R.G.B. a Escala de Grises***

Fuente: Adaptado de Maduell (2012*).*

Binarización por umbral (Threshold binarization). Esta operación parte de una imagen en escala de grises y mediante un nivel de comparación asigna a cada píxel un valor binario de pixeles negros o blancos. Esta operación reduce la memoria ocupada en disco en comparación con la imagen de partida (ver Figura 2.5).

Figura 2.5. ***Imagen binarizada por umbral***

Fuente: Tomado de Maduell (2012*).*

Sustracción de fondo (Background substraction). Esta operación permite restar una imagen de fondo manteniendo el objeto de interés; de esta manera, mediante la comparación de dos imágenes (ver Figura 2.6) se puede determinar si ha existido movimiento.

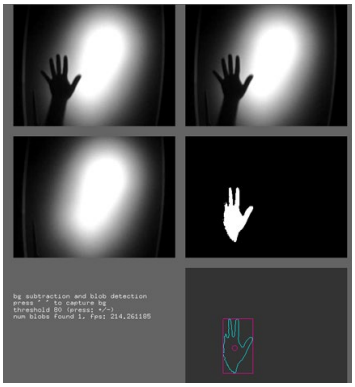


Figura 2.6. ***Sustracción de fondo***

Fuente: Recuperado de Maduell (2012*).*

Mediana. Este algoritmo divide la imagen en un conjunto de divisiones de radio en pixeles definible y calcula el valor de luminosidad medio de cada una de las subdivisiones, acto seguido se sustituye los píxeles que hay dentro de cada división por una única mancha gris con el valor de luminosidad medio de los píxeles iniciales (Maduell, 2012), como se muestra en la Figura 2.7.

Figura 2.7. ***(a)Imagen original, (b) Imagen aplicando Mediana***

Fuente: Tomado de Maduell (2012*).*

Erosión y Dilatación. Estos algoritmos generalmente se utilizan en serie. La erosión se sirve para contraer los contornos de las áreas blancas de una imagen ruidosa

configurando el número de pixeles de la contracción. A continuación, se aplica la dilatación que ayuda a recuperar el tamaño original de las áreas erosionadas (Maduell, 2012).

Una vez procesada la imagen, se debe extraer los datos relevantes de la imagen y con ellos la información mediante la aplicación de algoritmos los cuales pueden, por ejemplo: detectar un contorno, determinar movimiento, determinar áreas, entre otros. Todo dependerá de la aplicación que se quiera realizar.

Bag of features. Se deriva de la técnica Bag of Words, la cual extrae descriptores visuales de la imagen mediante un algoritmo K-means que agrupa iterativamente los descriptores en *k* grupos mutuamente excluyentes. Los racimos resultantes son compactos y separados por características similares. Cada centro de grupo representa una característica o palabra visual. El proceso inicia extrayendo puntos estratégicos diferenciadores, esta información se almacena en vectores de características que a su vez se agrupan en conjunto de datos (*clústeres*). Los *clústeres* se sintetizan en un grupo de particularidades llamado "vocabulario básico", las cuales son partes de la imagen que tienen características diferenciadoras útiles para el proceso de clasificación (*Image Classification with Bag of Visual Words - MATLAB & Simulink - MathWorks América Latina*, s. f.).

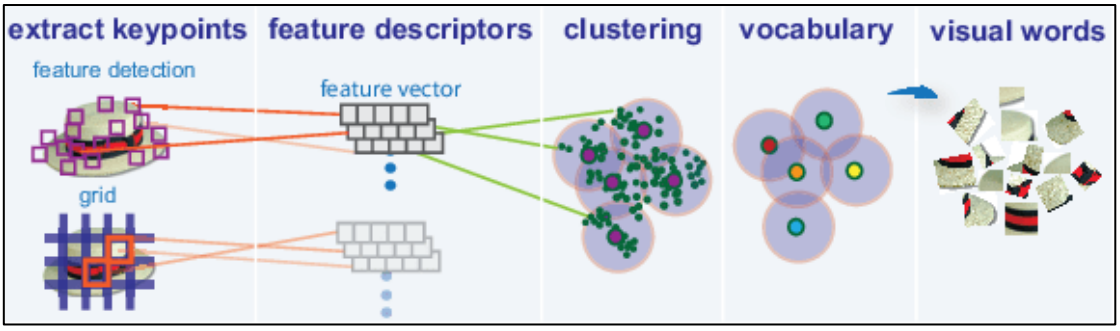


Figura 2.8. ***Proceso de extracción de características mediante "Bag of features"***

Fuente: Tomado de MATLAB & Simulink - MathWorks América Latina (*s.f.*).

2.2.4. Inteligencia Artificial

La inteligencia artificial es un amplio campo que puede desagregarse en 16 áreas, las cuales son: razonamiento, programación, vida artificial, revisión de creencias, minería de datos, inteligencia artificial distribuida, sistemas expertos, algoritmos genéticos, sistemas, representación del conocimiento, aprendizaje automático,

comprensión del lenguaje natural, redes neuronales, demostración de teoremas, satisfacción de restricciones y teoría de la computación (Oke, 2004).

La clasificación de objetos se puede lograr mediante técnicas de aprendizaje automático y redes neuronales. A continuación, se conceptualiza estos dos campos.

El aprendizaje automático puede ser supervisado o no y se ocupa del diseño de algoritmos avanzados que a partir de reglas y el análisis de datos realizan funciones, estos programas pueden adaptarse y mejorar el desempeño con la experiencia (Blum, 2007). Se diferencia de otras áreas debido a que el entrenamiento necesita supervisión humana (*Machine Learning vs Neural Networks*, 2020).

Redes Neuronales. Es un proceso masivamente distribuido e interconectado en paralelo que se basa en el funcionamiento sináptico de las neuronas biológicas. Las redes neuronales tienen una propensión natural a almacenar conocimiento experimental y convertirlo en disponible para su uso. Este tipo de algoritmo no necesita de la intervención humana durante el entrenamiento (*IBM Docs*, 2021), por lo que puede ser supervisado y no supervisado. El primero se caracteriza por que el entrenamiento es controlado por un supervisor que guía el ajuste de los parámetros de la red y conoce las salidas esperadas; mientras que, en el segundo el vector de salidas esperadas se presenta directamente a la red, por lo que no se necesita de un supervisor.

Neurona Biológica. Es el elemento base de la actividad sináptica en el cerebro. El funcionamiento de la neurona comienza con la recepción de señales de entrada, luego estas son bloqueadas, atenuadas o amplificadas por los pesos sinápticos, siendo la sinapsis la forma de transmitir información entre neuronas mediante neurotransmisores. A continuación, los elementos de proceso suman las entradas afectadas por la sinapsis y el resultado se transmite como salida bajo circunstancias de umbral definidas. La señal de salida puede convertirse en la señal de entrada de otra neurona o llegar a un elemento final como puede ser un músculo (Caicedo Bravo & Lopez Sotelo, 2009).

Neurona artificial. Según Caicedo (2009), el proceso que rige el funcionamiento de una neurona artificial se basa en el funcionamiento de una neurona biológica; es decir, esta recibe entradas de estímulo provenientes de elementos sensoriales o de otras neuronas. La información entrante se define como $X = [x_1, x_2, \dots, x_n]$. La información recibida se modifica por un vector de pesos sinápticos w , que atenúan o amplifican las señales que se desean propagar hacia la salida, la cual se activa dependiendo de un

parámetro de umbral θ_j. Los valores que se modifican por los pesos sinápticos se suman y producen una entrada neta Net_j, la cual acompañada de una función de activación determinan si la neurona se activa o no. Al activarse la neurona produce una señal de salida y_j (ver figura 2.9).

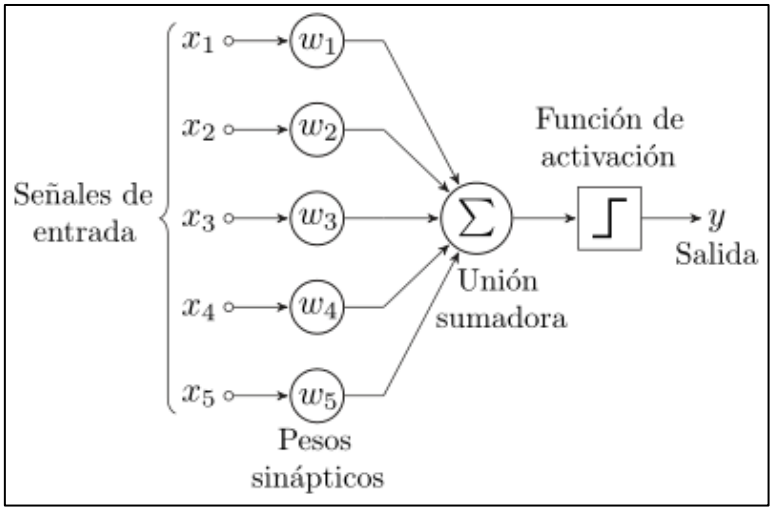


Figura 2.9. ***Esquema de una neurona artificial***

Fuente: Tomado de Perceptrón (2021).

En una neurona artificial, la entrada neta se define mediante las ecuaciones vectoriales 1 a 3:

$$Net_j = \sum_{i=1}^{N} x_i w_{ji} + \theta_j \tag{1}$$

$$Net_j = w_{j1}x_1 + w_{j2}x_2 + \ldots + w_{ji}x_i + \ldots + w_{jN}x_N + \theta_j \tag{2}$$

$$Net_j = \boldsymbol{w^T X_j} + \boldsymbol{\theta_j} \tag{3}$$

La función de activación (*Fact*) que determina la salida de la neurona, puede ser de tipo escalón, lineal o sigmoidal, entre otras (ver Figura 2.10).

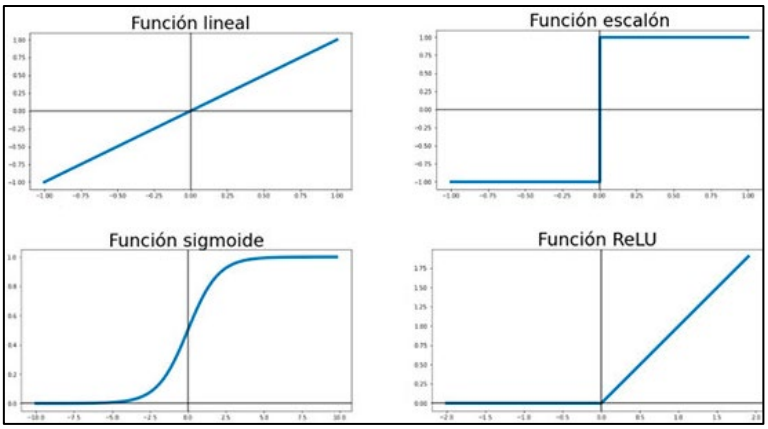


Figura 2.10. ***Tipos de funciones de activación***

Fuente: Tomado de Redes Neuronales y Deep Learning (2021).

Una neurona por si sola tiene baja utilidad y capacidad de procesamiento; por esta razón, se suelen agrupar en arreglo para que formen una red neuronal. Las redes neuronales se asemejan a un cerebro biológico en:

- Mediante un proceso de aprendizaje, la red neuronal adquiere el conocimiento.
- El conocimiento se almacena en los pesos sinápticos, los cuales se ajustan en función del aprendizaje.

Una red neuronal se compone de las siguientes tres capas: a) capa de entrada, que recibe directamente la información del exterior; b) capa oculta, que corresponde las neuronas interconectadas entre sí y que no tienen contacto con el exterior; y, c) capa de salida, que son el conjunto de neuronas que transfieren la información procesada al exterior (ver Figura 2.11).

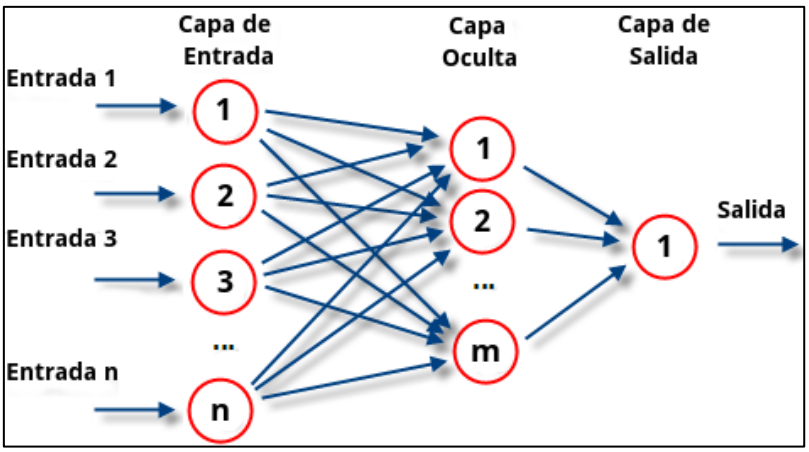


Figura 2.11. Arquitectura de una red neuronal

Fuente: Tomado de "¿Qué son las redes neuronales y sus funciones?" (2019).

Dependiendo del número de capas o niveles de la red neuronal, esta puede ser monocapa (tiene una capa oculta con flujo de información en un solo sentido), multicapa (múltiples capas ocultas con flujo de información en un solo sentido) o recurrente (la información puede realimentarse hacia capas posteriores), como se puede ver en la figura 2.12.

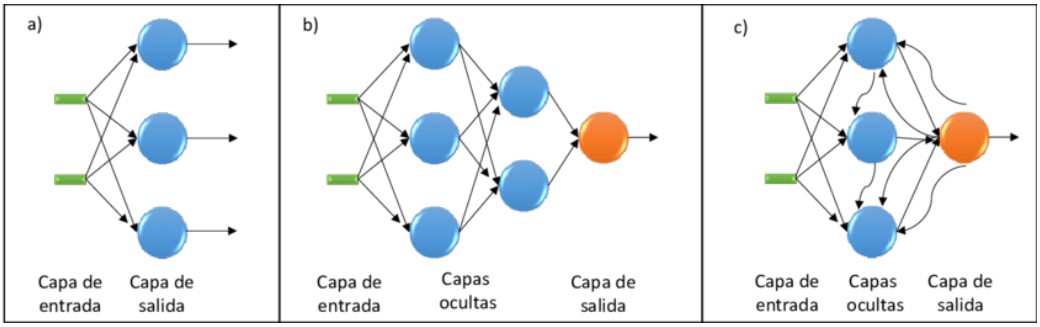


Figura 2.12. ***Red neuronal a) Monocapa, b) Multicapa, c) Recurrente***

Fuente: Tomado de Manjarrez (2014).

El aprendizaje de una red neuronal artificial es un proceso a través del cual los parámetros son ajustados mediante la estimación del ambiente en el que esta embebida la red (Caicedo Bravo & Lopez Sotelo, 2009).

El modelo de aprendizaje se expresa mediante:

$$w(t+1) = w(t) + \Delta w(t) \tag{4}$$

Donde, $w(t+1)$ es el nuevo valor del peso sináptico, $w(t)$ es el valor anterior del peso sináptico y $\Delta w(t)$ es la variación del peso sináptico.

Perceptrón. Es una red monocapa que posee una capa de procesamiento, la misma que se utiliza para la salida, comúnmente se implementan unidades de umbral que son útiles en problemas de clasificación (ver figura 2.13). La función de activación es de tipo escalón $[0, +1]$.

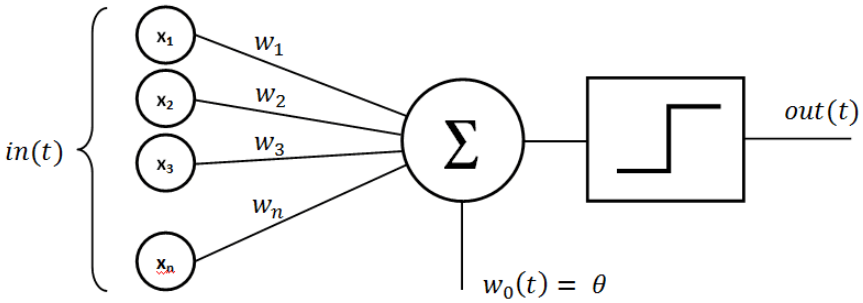


Figura 2.13. ***Red neuronal Perceptrón***

Fuente: Tomado de (DataScients, 2022)

La expresión de la entrada Neta es:

$$Neta = x_1 w_1 + x_2 w_2 + \theta \tag{5}$$

El perceptrón se utiliza generalmente en problemas de clasificación como la que se muestra en la Figura 2.14.

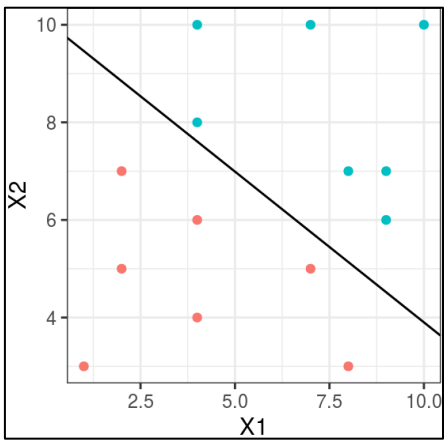


Figura 2.14. ***Clasificación mediante perceptrón***

Fuente: Adaptado de Algoritmo Perceptrón (2021).

La principal limitación del perceptrón es que los datos a clasificar deben ser linealmente separables; debido a que, la separación se hace mediante rectas.

Adaline. Es una red monocapa similar al perceptrón, a la cual se le implementa una función de activación lineal. El diagrama de esta red se puede observar en la Figura 14.

El valor de la salida se expresa como:

$$y = \sum_{i=0}^{N} x_i w_i + \theta \tag{6}$$

La red Adaline puede tener más de una neurona, por lo que puede disponer de N entradas y M salidas, como se observa en la figura 2.15.

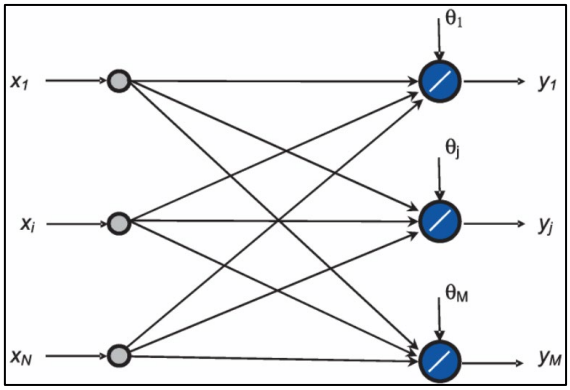


Figura 2.15. ***Red neuronal Adaline***

Fuente: Tomado de Caicedo (2009).

Perceptrón multicapa. Este puede superar la limitación del perceptrón monocapa, ya que es capaz de separar regiones complejas que no se separan con una sola recta. Las funciones de activación de las redes multicapa generalmente es lineal o sigmoidal, y como se puede apreciar en la figura 2.16. Todas las entradas están conectadas con las neuronas de la capa oculta, y estas a su vez con la capa siguiente; es por ello, que se dice que la red multicapa tiene conectividad total.

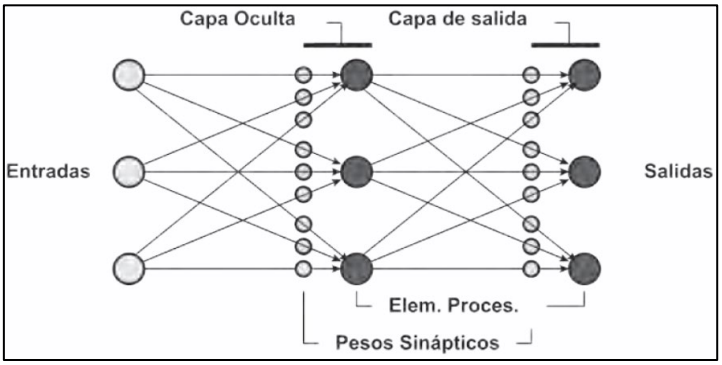


Figura 2.16. ***Red neuronal Perceptrón multicapa***

Fuente: Tomado de Caicedo (2009).

Red neuronal Backpropragation. Es un algoritmo multicapa que propone calcular el error de salida de las neuronas de la capa o capas ocultas a partir del error de la capa de salida, que es un dato conocido, esto con el fin de calibrar los pesos sinápticos en la capa oculta.

Sea el vector de entrada $x_p = [x_{p1,}x_{p2,}\ldots, x_{pi}, \ldots, x_{pN,}]^T$*, la entrada neta de la* $j-$ *ésima neuroma de la capa oculta se define por:*

$$Neta_{pj}^{h} = \sum_{i}^{N} w_{ji}^{h} x_{pi} + \theta_j^h \tag{7}$$

Siendo: h el número de neuronas de la capa oculta, w_{ji}^{h} el peso de interconexión entre la neurona $i-ésima$ de la entrada y la $j-ésima$ de la capa oculta, θ_j^h el termino de tendencia de la neurona $j-ésima$ de la capa oculta.

La salida de la neurona $j-ésima$ se calcula en función de la entrada neta y de la función de activación f_j^h.

$$i_{pj}^{h} = f_j^h(Neta_{pk}^{h}) \tag{8}$$

La entrada neta de la neurona $k-$ *ésima de la capa de salida o, viene dada por:*

$$Neta_{pk}^{o} = \sum_{j=1}^{L} w_{kj}^{o} i_{pj}^{o} + \theta_k^o \tag{9}$$

Siendo: w_{ji}^{o} el peso de interconexión entre la neurona $j-ésima$ de la capa oculta y la $k-ésima$ de la capa de salida, i_{pj}^{o}, salida de $j-ésima$ neurona de la capa de la capa oculta, θ_k^o, termino de tendencia de la $k-ésima$ neurona de la capa de salida.

La salida de la red neuronal ante un estímulo de entrada se calcula en base a la función de activación $k-esima$ definida como f_k^o:

$$y_{pk} = f_k^o(Neta_{pk}^{o}) \tag{10}$$

Los ajustes de los pesos dependen del error en las capas ocultas y de salida e_p, se basa en la regla delta o en los mínimos cuadrados promedio, y depende de la tasa de aprendizaje α:

$$w_i(t+1) = w_i(t) + 2\alpha e_p x_i \tag{11}$$

El ajuste del error mínimo se fundamenta en el cálculo de su gradiente descendente ∇E_p. Uno de los limitantes del algoritmo *Backpropagation* es el tiempo de aprendizaje, esto se debe entre otras cosas a que la tasa de aprendizaje α es fija. Para superar este limitante se puede variar la tasa de aprendizaje, esto acelera el proceso de aprendizaje o ajuste de los pesos. El algoritmo de *Backpropagation* con tasa de aprendizaje variables, para algunas aplicaciones aún es lento, para ello existen algunas variantes del algoritmo

como son: los algoritmos de aprendizaje de Gradiente Conjugado y de Levenberg (Marquardt. , 2009).

Parámetros de una red neuronal. Para medir el desempeño de una red neuronal se tiene los siguientes parámetros:

- Dimensión de la red neuronal. Es el número de capas y neuronas que se utilizan, ya que a mayor número de capas y neuronas aumenta el tiempo de entrenamiento y a menor número de capas y neuronas se puede generar un menor desempeño. Es por ello por lo que autores como (Caicedo Bravo & Lopez Sotelo, 2009), recomiendan el uso de una sola capa oculta para la mayoría de las aplicaciones.
- Velocidad de convergencia del algoritmo. Este parámetro depende de la tasa de aprendizaje α. En (Caicedo Bravo & Lopez Sotelo, 2009) sugieren que sea un valor pequeño de entre 0.05 a 0.25. Con valores más pequeños se puede asegurar el entrenamiento, pero se aumenta el tiempo de aprendizaje.
- Funciones de activación. En (Caicedo, 2009) recomienda usar una técnica de preprocesamiento, adaptando los rangos de las variables de entrada como de salida al intervalo $[-1, +1]$. Este procedimiento ayuda a la prueba de las diferentes funciones de activación.
- Regularización. En (Caicedo, 2009), indica que esta técnica se emplea para eliminar errores de sobre entrenamiento; es decir, que la red neuronal responda precisamente a los valores de entrenamiento, y si existe un ruido en los mismos se verá afectada por dicho ruido. Por ejemplo, una red neuronal de dos capas entrenada para ajustar los puntos de una función cuadrática, al tener dos capas, sus ajustes son muy precisos, tanto que son susceptibles al ruido, por lo cual se puede decir que la rede neuronal está sobre entrenada (ver figura 2.17). La solución es quitarle precisión al entrenamiento.

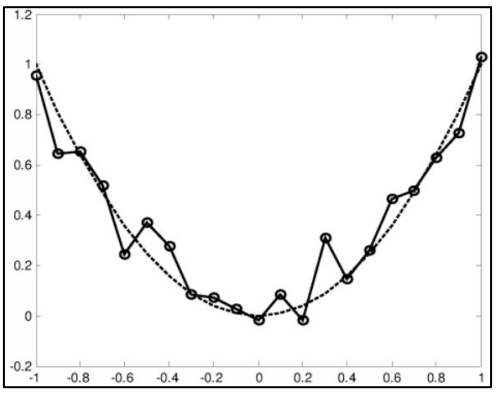


Figura 2.17. ***Sobre entrenamiento de una rede neuronal***

Fuente: Adaptado Caicedo (2009).

Red Neuronal de Hopfield. Formulada por Jhon Hopfield en 1982, se considera como una red neuronal dinámica auto asociativa; es decir, que el conocimiento se asocia dependiendo del entrenamiento. En este tipo de red se utiliza neuronas BAM, que es un tipo de arquitectura de memoria asociativa bidireccional, en donde el flujo de información va desde la entrada hacia la salida y viceversa, con la peculiaridad de que Hopfield usa una sola capa de neuronas BAM, donde la salida se lleva a la entrada (Caicedo, 2009). La arquitectura se puede ver en la figura 2.18.

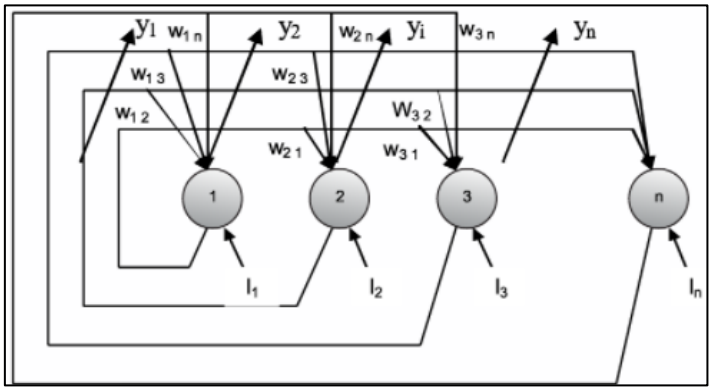


Figura 2.18. ***Red neuronal Hopfield***

Fuente: Adaptado Caicedo (2009).

Red neuronal de Kohonen. Introducido por Teuvo Kohonen en 1982, es una peculiar rede neuronal que auto-organiza la topología de las neuronas, de manera que la distancia entre los datos se conservan (Caicedo, 2009). Esto permite realizar una proyección del espacio $n - dimensional$ de entrada en un espacio $m - dimensional$ a la salida, reduciendo la dimensión y conservando las características de los datos, como se puede observar en la figura 2.19.

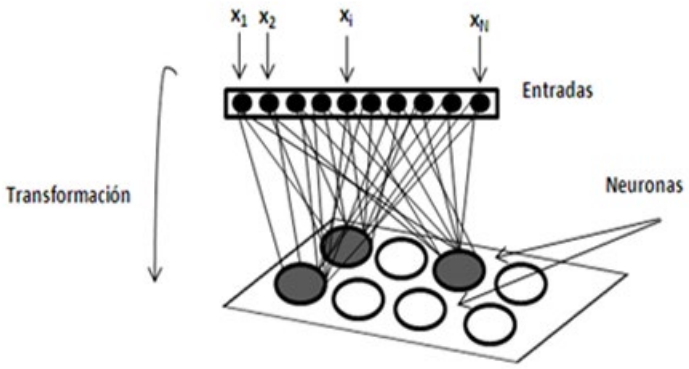


Figura 2.19. ***Red neuronal Kohonen***

Fuente: Tomado de (Gámez et al., 2016)

Red de base radial. Es una red unidireccional multicapa de aprendizaje híbrido que se basa en el principio de regularización y que sigue un entrenamiento no supervisado en la capa oculta y en la capa de salida el aprendizaje es supervisado. Esta red es simple y veloz, porque se utiliza en aplicaciones de clasificación de patrones, modelado de sistemas no lineales y aplicaciones de reconocimiento. La arquitectura de este tipo de red se puede ver en la figura 2.20.

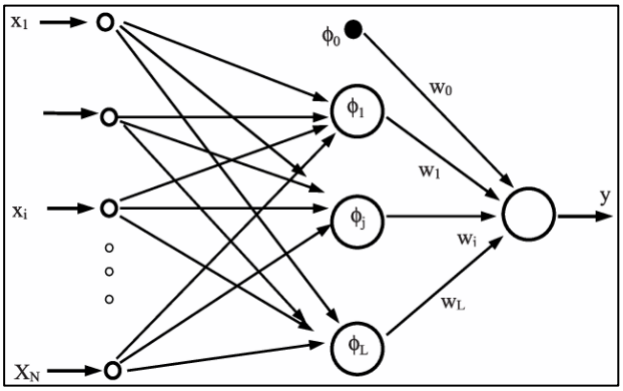


Figura 2.20. ***Red neuronal de base radial***

Fuente: Adaptado de Caicedo (2009).

Aprendizaje automático supervisado. En estos algoritmos se proporciona un conjunto de datos de entrenamiento formado por características de entrada y salida esperados (ver Figura 2.21). A partir de esta información, el agente puede corregir sus parámetros para reducir la magnitud de una función de pérdida global mediante un procesos iterativo de entrenamiento (Bonaccorso, 2017).

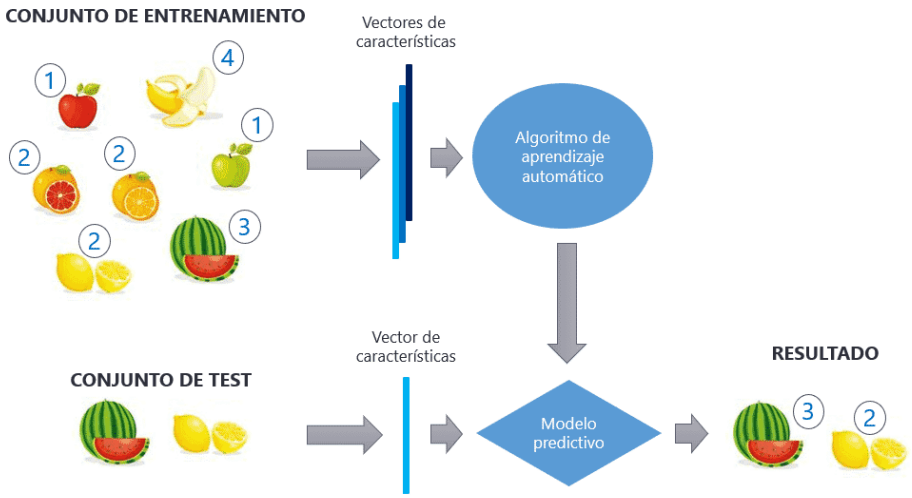


Figura 2.21. ***Aprendizaje supervisado***

Fuente: Recuperado de (Calvo, 2019). https://tinyurl.com/2okt2xx7

Este tipo de aprendizaje es usado comúnmente en:

- Análisis predictivo basado en regresión o clasificación categórica.
- Detección de correo basura.
- Procesamiento natural del lenguaje.
- Análisis de sentimientos.
- Clasificación automática de imágenes.
- Procesamiento secuencial automático de audio.

Más conocido como *Machine Learning*, se compone de una serie de técnicas que permiten que el software aprenda sin estar explícitamente programado para hacerlo, con su rendimiento optimizado a medida que están expuestos a más muestras de datos (Shinde & Shah, 2018). A continuación, se analiza los principales algoritmos de aprendizaje automático:

Regresión Lineal. Considerando un conjunto de datos referentes a una o varias variables (*dataset*) de vectores reales:

$$x = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n\}, donde\ \bar{x}_i \in \mathbb{R}^n$$

Donde cada vector $\bar{x}_i$, se relaciona con un valor $\bar{y}_i$, que se expresa como sigue:

$$y = \{\bar{y}_1, \bar{y}_2, \dots, \bar{y}_n\}, donde\ \bar{y}_i \in \mathbb{R}$$

El modelo lineal se basa en la suposición de que la variable y se puede expresar a través de un proceso de regresión de la siguiente manera:

$$y = a_0 + \sum_{i=1}^{m} a_i x_i \, , donde \; A = \{a_0, a_1, \dots, a_m\}$$

El modelo se basa en la suposición es que el *dataset* y todos los demás puntos desconocidos se encuentran en un hiperplano y el error máximo es proporcional tanto a la calidad del entrenamiento como a la adaptabilidad del conjunto de datos original. Uno de los problemas más comunes surge cuando el conjunto de datos es evidentemente no lineal y otros modelos deben ser considerados (Bonaccorso, 2017).

Regresión Logística. En este clasificador se debe encontrar un hiperplano óptimo que separe dos clases. En problemas de más de dos clases, se adopta la estrategia uno contra todos (Bonaccorso, 2017). Para un clasificador binario se parte del siguiente conjunto de datos:

$$x = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n\}, donde \; \bar{x}_i \in \mathbb{R}^n$$

Este conjunto de datos se asocia al conjunto de objetivo y, de Fuentedo como:

$$y = \{\bar{y}_1, \bar{y}_2, \dots, \bar{y}_n\}, donde \; \bar{y}_i \in [0,1]$$

Se dispone de un vector de pesos representado de la siguiente manera:

$$w = \{\bar{w}_1, \bar{w}_2, \dots, \bar{w}_n\}, donde \; \bar{w}_i \in \mathbb{R}$$

Se define el cuantificador z como:

$$z = \bar{x} \cdot \bar{w} = \sum_{i=1}^{m} \bar{x}_i \cdot \bar{w}_i \, , \forall \; \bar{x} \in \mathbb{R}^n$$

Donde, z es el valor determinado por la ecuación del hiperplano; en consecuencia, si los pesos w son los correctos se comprueba que:

$$sign(z) \begin{cases} +1, si \; x \in clase \; 1 \\ -1, si \; x \in clase \; 2 \end{cases}$$

Siendo las clases los conjuntos del problema de clasificación binaria. El problema entonces es optimizar w para minimizar el error, si la separación de clases se la puede realizar con una función lineal se dice que es linealmente separable, caso contrario es no linealmente separable.

Cuando la separación de clases se la realiza con una función sigmoidal (ver figura 2.9) la regresión toma el nombre de logística, la cual se representa como sigue:

$$\sigma(z) = \frac{1}{1+e^{-z}} \; ; \; \sigma(\bar{x};\bar{w}) = \frac{1}{1+e^{-\bar{x}\bar{w}}}$$

La probabilidad de que una muestra pertenezca a una clase es:

$$P(y|\bar{x}) = \sigma(\bar{x};\bar{w})$$

Se maximiza la probabilidad logarítmica para optimizar los parámetros y se minimiza la entropía cruzada entre una distribución objetivo y una aproximada, de este proceso se obtiene la ecuación de la Regresión Logística:

$$H(X) = -\sum_{x \in X} p(x) log_2 q(x)$$

Naive Bayes. Son una familia de clasificadores potentes y fáciles de entrenar que determinan la probabilidad de un resultado dado un conjunto de condiciones utilizando el teorema de Bayes, cuyo objetivo de aprendizaje es construir un clasificador dado un conjunto de ejemplos de entrenamiento con etiquetas de clase (Zhang, 2004). El desempeño es particularmente bueno en todas aquellas situaciones donde la probabilidad de una clase está determinada por las probabilidades de algunos factores causales (Bonaccorso, 2017).

El teorema de Bayes considera dos eventos probabilísticos A y B, con probabilidades $P(A)$ y $P(B)$, en donde la probabilidad condicional $P(A|B)$ y $P(B|A)$, siguen la siguiente regla:

$$\begin{cases} P(A \cap B) = P(A|B)P(B) \\ P(B \cap A) = P(B|A)P(A) \end{cases}$$

Se amplía la definición introduciendo un factor α de normalización y eventos concurrentes e independientes, de la siguiente manera:

$$P(A|C_1 \cap C_2 \cap C_3 \cap \ldots \cap C_n) = \alpha P(C_1|A)P(C_2|A)P(C_3|A) \ldots P(C_n|A)P(A)$$

.

Se considera un conjunto de datos de características definidos como vectores de la siguiente forma:

$$X = \{\bar{x}_1, \bar{x}_2, \bar{x}_3, \ldots, \bar{x}_n | \; \bar{x}_i \in \mathbb{R}^m\}$$

También se considera el conjunto de datos objetivo o de salida de la forma:

$$Y = \{\bar{y}_1, \bar{y}_2, \bar{y}_3, \dots, \bar{y}_P\}$$

Los valores de la probabilidad marginal a priori $P(y)$ y de las probabilidades condicionales $P(x_i|y)$ se obtienen mediante un conteo de frecuencias; es así que, dado un vector de entrada x, la clase predicha es aquella para la que la probabilidad a posteriori es máxima (Bonaccorso, 2017), lo cual se puede escribir como sigue:

$$P(y|x_1, x_2, x_3, \dots, x_P) = \alpha P(y) \prod_i P(x_1|y)$$

Máquinas de soporte vectorial. Considerando un conjunto de datos de características tipo vector, defino como:

$$X = \{\bar{x}_1, \bar{x}_2, \bar{x}_3, \dots, \bar{x}_n |\ \bar{x}_i \in \mathbb{R}^m\}$$

Se considera un conjunto de datos objetivo para una clasificación binaria, donde:

$$Y = \{\bar{y}_1, \bar{y}_2, \bar{y}_3, \dots, \bar{y}_n\}, donde\ \bar{y}_n \in \{-1,1\}$$

El objetivo es encontrar el hiper plano que satisface la siguiente ecuación:

$$\bar{w}^T \bar{x} + b = 0, donde\ \bar{w}, \bar{x} \in \mathbb{R}^n$$

Según (Bonaccorso, 2017) en un escenario realista, las dos clases normalmente están separadas por un margen con dos límites donde se encuentran algunos elementos. Esos elementos se llaman vectores de soporte. Para una expresión matemática más genérica, es preferible volver a normalizar nuestro conjunto de datos para que los vectores de soporte se encuentren en dos hiperplanos con ecuaciones:

$$\begin{cases} \bar{w}^T \bar{x} + b = -1 \\ \bar{w}^T \bar{x} + b = 1 \end{cases}$$

Un ejemplo de clasificador mediante una máquina de soporte vectorial se muestra en la figura 2.22.

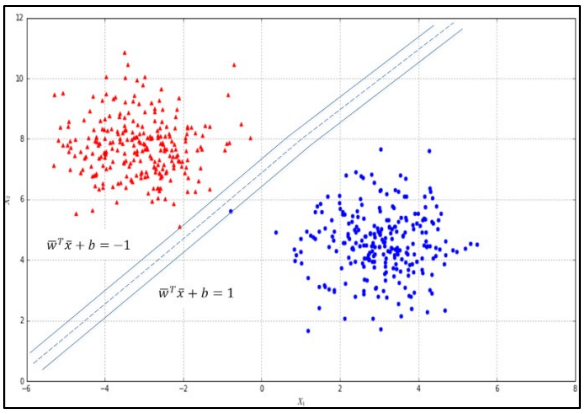


Figura 2.22. Ejemplo de c*lasificador SVM*

Fuente: Adaptado de Bonaccorso (2017).

Árboles de decisión. El clasificador más simple de esta familia es el Binario, el cuál es una estructura basada en un proceso secuencial de decisión. A partir de la raíz, se evalúa una característica y se selecciona una de las dos ramas. Este procedimiento se repite hasta llegar a una rama final, que normalmente representa el objetivo de clasificación que se busca (Bonaccorso, 2017).

En la figura 2.22, se muestra un conjunto de datos bidimensionales no normalizados y las puntuaciones de validación cruzada obtenidas mediante una regresión logística y un árbol de decisión, se observa que el resultado de precisión de la aplicación del Árbol de Decisión es mayor que de la Regresión Logística.

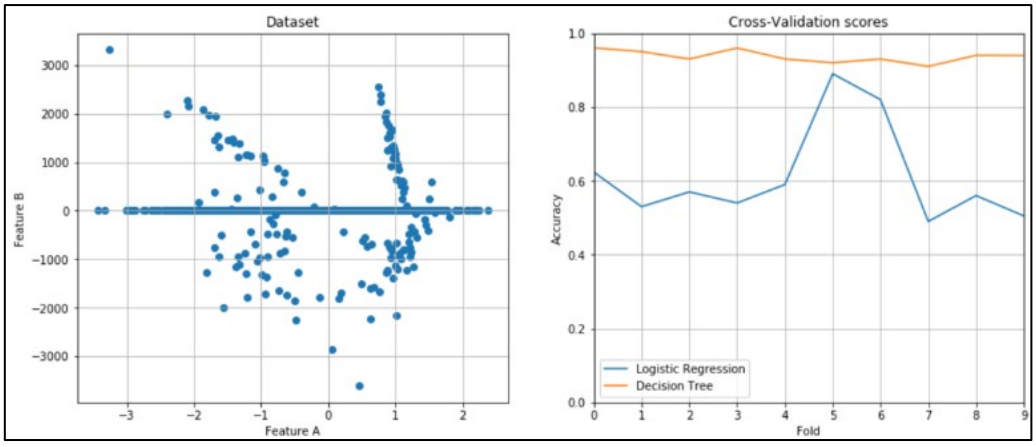


Figura 2.23. Comparación de precisión entre una Regresesión Logistica y un Árbol de Desición

Fuente: Adaptado de Bonaccorso (2017).

Clasificador KNN1. El clasificador KNN o de K vecinos cercanos está basado en la idea de que los patrones más cercanos a un patrón objetivo x', para ello KNN asigna la clase de la mayoría de los K-patrones más cercanos en el espacio de datos en función de los datos vecinos (Kramer, 2013). En consecuencia, se define una medida de similitud en el espacio $\mathbb{R}^q$ y se emplea la métrica de Minkowski (p-norm), como sigue:

$$\|x' - x_j\|^p = \left(\sum_{i=1}^{q} |(x_i)' - (x_i)_j|^p\right)^{1/p}$$

Donde la distancia euclidiana para $p = 2$, y con la función de salida $Y = \{-1,1\}$, se define como:

$$f_{KNN(x')} = \begin{cases} 1, & si \sum_{i \in \mathcal{N}_K(x')} y_i \geq 0 \\ -1, & si \sum_{i \in \mathcal{N}_K(x')} y_i < 0 \end{cases}$$

Donde se considera tamaño de vecindad K y con el conjunto de índices $\mathcal{N}_K(x')$ de los K patrones más cercanos, en la figura 2.23 se puede observar la comparación de dos clasificadores KNN con tamaño de vecindad diferente.

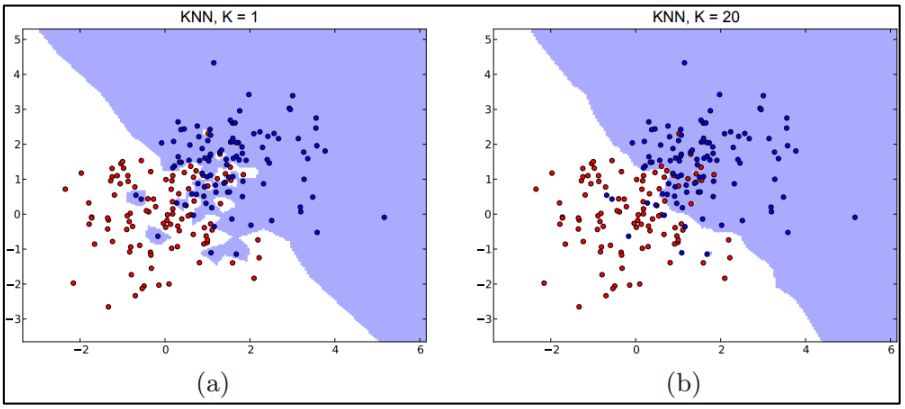


Figura 2.24. Comparación de clasificadores KNN con tamaño a) K=1 y b) K=20

Fuente: Adaptado de Bonaccorso (2017).

El fundamento del clasificador KNN de dos clases se puede extender hacia el clasificador multiclase donde la función es la siguiente:

$$f_{KNN(x')} = arg \max_{y \in Y} \sum_{i \in \mathcal{N}_K(x')} \mathcal{I}(y_i = y)$$

Donde el indicador $\mathcal{I}$ retorna uno si el argumento es verdadero y cero si es falso.

[1] KNN son las siglas de K-nearest neighbors

Aprendizaje no supervisado. Para (Bodenhofer, 2002) este enfoque se basa en la ausencia de cualquier supervisor y por lo tanto de medidas de error absoluto. Es utilizado cuando se requiere agrupar un conjunto de elementos según su similitud, este proceso se puede observar en la figura 2.24.

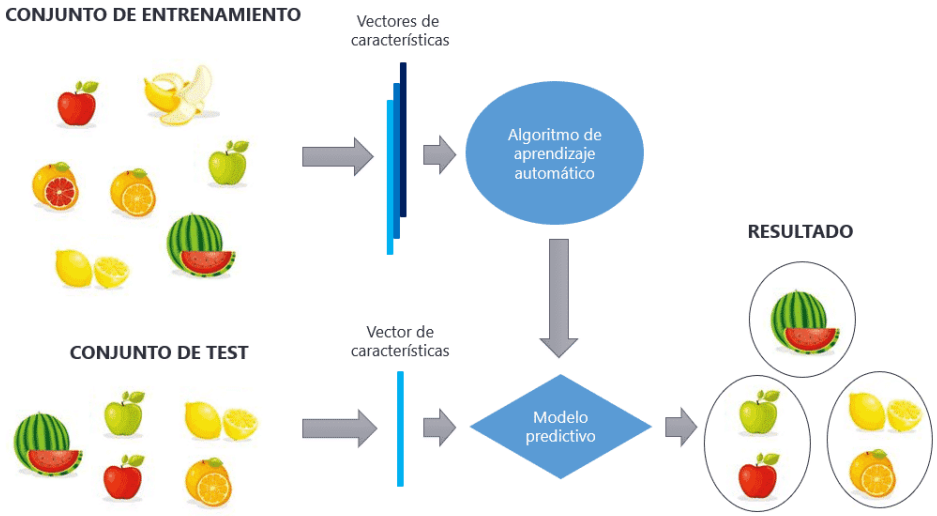


Figura 2.25. ***Aprendizaje no supervisado***

Fuente: Recuperado de (Calvo, 2019). https://tinyurl.com/2okt2xx7

Algoritmos Genéticos. Son algoritmos no supervisados avanzados de aproximación probabilística basados en la evolución biológico con el fin de generar un refinamiento y mejoramiento continuo en una población (Bodenhofer, 2002), como se aprecia en la figura 2.25.

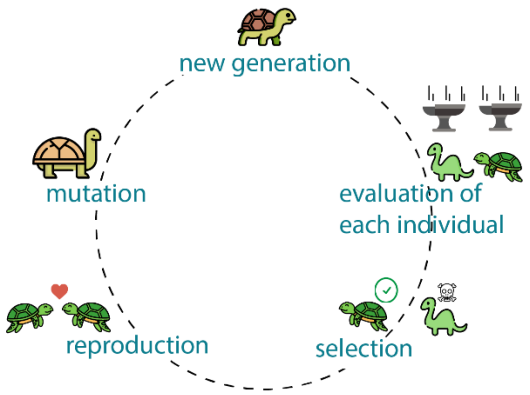


Figura 2.26. ***Pasos de un algoritmo genético***

Fuente: Tomado de (*What Is a Genetic Algorithm?*, s. f.). https://tinyurl.com/2lmxnu3f

Clustering. Según Sancho Caparrini (2020), un algoritmo de clustering tiene como objetivo agrupar los objetos de un dataset según su similaridad, de forma que los objetos

que hay dentro de un grupo (cluster) sean más similares que aquellos que caen en grupos distintos.

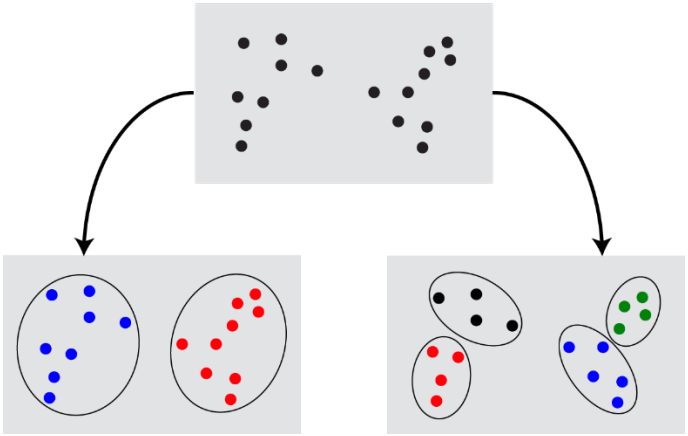


Figura 2.27. ***Reducción en 2 o 4 clusters***

Fuente: Tomado de (Sancho Caparrini, 2020). https://tinyurl.com/y2lr337r

Reducción de dimensionalidad. Son técnicas estadísticas que mapean un conjunto de los datos a subespacios derivados del espacio original, de menor dimensión (Ayala, 2020), un ejemplo de este tipo de técnicas es el Análisis de Componentes Principales.

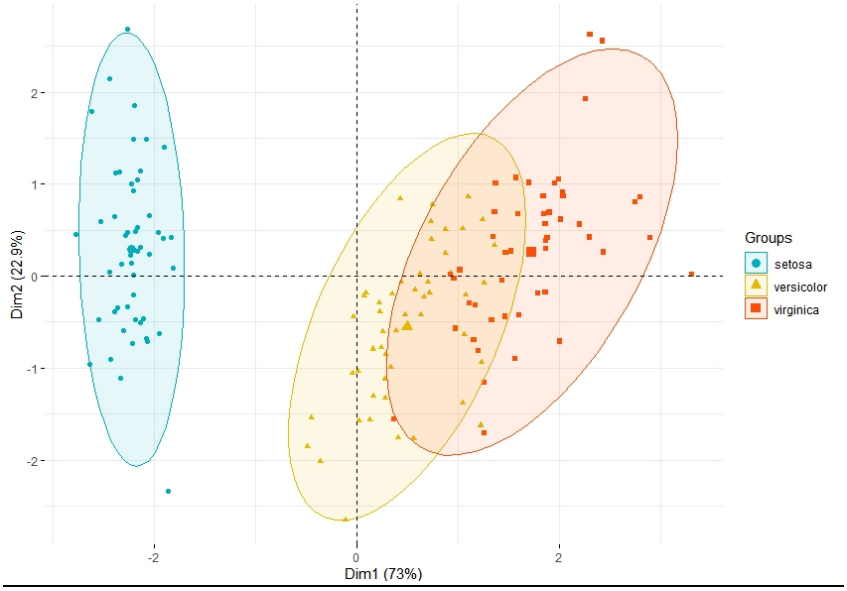


Figura 2.28. ***Ejemplo de Análisis de Componentes Principales***

Fuente: Tomado de Ayala (2020), https://tinyurl.com/2znpv8va

2.3. Marco Legal

El diseño del sistema de clasificación con Visión e Inteligencia Artificial, con el propósito de optimizar el proceso de postcosecha de follaje Ruscus, se desarrolla con base a los siguientes fundamentos legales:

La constitución del Ecuador, artículo 328, numeral 5, establece que: "Toda persona tendrá derecho a desarrollar sus labores en un ambiente adecuado y propicio, que garantice su salud, integridad, seguridad, higiene y bienestar" (Constitución de la Republica del Ecuador, 2008). Por lo expuesto y sabiendo que los trabajos repetitivos aumentan el riesgo de contraer una enfermedad laboral, el presente trabajo aporta al cumplimiento de la carta magna del Ecuador.

En el Reglamento de Seguridad y Salud de los Trabajadores, en el artículo 2, literal 2, se establece que: "Adoptar las medidas necesarias para la prevención de los riesgos que puedan afectar a la salud y el bienestar de los trabajadores en los lugares de trabajo de su responsabilidad" (Reglamento de Seguridad y Salud de los Trabajadores, 2003).

La Organización Internacional del Trabajo, en su convenio C184, sobre la seguridad y la salud en la agricultura, establece en su artículo 7, sobre las obligaciones del empleador, en el numeral c, qué: "El empleador tome medidas inmediatas para suspender cualquier operación que suponga un peligro inminente y grave para la seguridad y salud, y para evacuar a los trabajadores como convenga."(*Convenio C184 - Convenio sobre la seguridad y la salud en la agricultura, (núm. 184)*, 2001).

Por otro lado, según Sein (2019), en su artículo "Innovaciones tecnológicas, inteligencia artificial y derechos humanos en el trabajo", establece que existen preocupaciones y desafíos en cuanto a la interacción de la Inteligencia Artificial y los derechos del trabajador, en ese sentido propone un marco ético y jurídico inspirado en las directrices éticas aprobadas en el seno de la Unión Europea por el Grupo de Expertos de Alto Nivel en Inteligencia Artificial.

CAPITULO III
MARCO METODOLÓGIO

3.1. Descripción del área de estudio

La presente investigación se realizó en la plantación GM Familiar, emprendimiento que pertenece al Sr. Fabian Guerra socio de la Asociación de Desarrollo Integral el Viejo San Martín. La plantación se encuentra ubicada en la Provincia de Imbabura, cantón Cotacachi, parroquia Quiroga, en la vía Quiroga – Otavalo km 5 (coordenadas geográficas 0.2693180, -78.2827140). En la Figura … se puede ver la localización de la plantación respecto a la Universidad Técnica del Norte. La Asociación comenzó su actividad productiva en el año 2011, actualmente producen tomate riñón, pimiento, cebolla, maíz, alfalfa y Ruscus. La producción de comestibles abastece al mercado local y el Ruscus por medio de intermediarios se exporta a diferentes mercados internacionales.

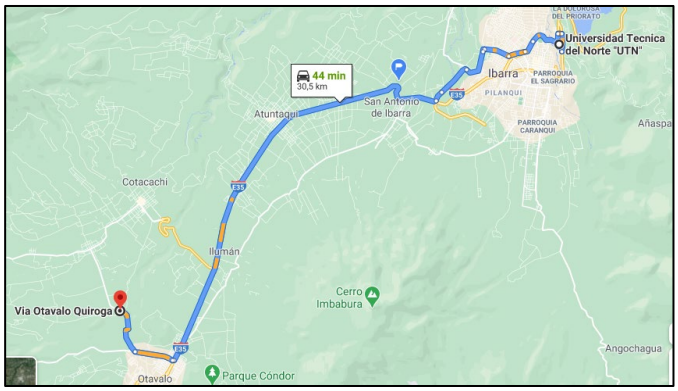


Figura 3.29. Localización de la plantación de Ruscus GM Familiar

Fuente: Google Maps (2023).

3.2. Enfoque y tipo de investigación

La presente investigación tiene un enfoque de diseño de ingeniería, ya que el objetivo está orientado a la solución de un problema particular. Para la solución del problema se parte de los requerimientos y especificaciones de los interesados y se plantea una solución concreta a través de la implementación de un sistema tecnológico.

La metodología utilizada comprende de cinco etapas: a) identificación y formulación del problema, b) análisis del problema, c) búsqueda de soluciones, d) evaluación de alternativas y selección de la solución óptima; y e) la especificación de la solución escogida (Ortiz & Rozo, 2013).

La investigación es de tipo aplicada, debido a que se creó una solución tecnológica a la problemática planteada de un invernadero en particular; sin embargo, la solución propuesta puede generalizarse a otras plantaciones de Ruscus. Antes de proponer la solución, se realizó una investigación documental sistemática con la finalidad de conocer los aspectos agrícolas y del proceso de postcosecha del Ruscus, así como de los algoritmos de visión e inteligencia artificial utilizados para la clasificación de imágenes. De igual manera, se llevó a cabo una investigación de campo, debido a que fue necesario realizar entrevistas al personal encargado de la plantación y realización de una base de datos fotográfica. Finalmente, se realizó una investigación experimental, ya que se realizó pruebas de los diferentes algoritmos de visión e inteligencia artificial y el entrenamiento de los modelos de clasificadores automáticos.

Por otra parte; la investigación contempla también aspectos descriptivos, debido a que se describió el proceso de postcosecha para entender las peculiaridades de éste y aspectos explicativos, puesto que se fue necesario dar a conocer las consecuencias de la problemática, los impactos y, los posibles trabajos futuros asociados a la solución planteada.

3.3. Procedimiento

Fase1: Identificación

Se investigó los parámetros relevantes de la morfología del Ruscus y su proceso de postcosecha; asimismo, los algoritmos de visión e inteligencia artificial más utilizados en identificación de características morfológicas.

Actividad 1: Identificación de parámetros relevantes del Ruscus. En esta actividad se identificó los parámetros morfológicos relevantes del follaje; para ello, se realizó una búsqueda sistemática de literatura identificando la teoría relevante sobre las características morfológicas del Ruscus (ver sección 2.2.1).

Actividad 2: Identificación de los parámetros importantes en postcosecha de Ruscus. Se detalló el proceso de postcosecha manual existente, con el fin de seleccionar los parámetros morfológicos más convenientes a utilizarse posteriormente. Se consiguió identificar los parámetros importantes en postcosecha de Ruscus mediante la investigación documental y una entrevista al Sr. Fabian Guerra propietario de la plantación GM Familiar.

Actividad 3: Identificación de algoritmos de visión artificial. Se indagó sobre los algoritmos y/o técnicas relevantes de identificación de características morfológicas en imágenes. Esto puede ser apreciado en la sección 2.2.3.

Actividad 4: Identificación de algoritmos de inteligencia artificial. Se investigó los algoritmos de inteligencia artificial y aprendizaje automático aplicados a la clasificación.

Actividad 5: Identificación de software. Se analizó el software disponible, sus características y licencias. Posteriormente se seleccionó el software en base a un análisis comparativo del programa a utilizarse.

Fase 2: Diseño

Se diseñó el sistema de visión e inteligencia artificial mediante el procesamiento y transformación de una base de datos de imágenes de Ruscus tomadas en campo de manera aleatoria.

Actividad 1: Recopilación de base de datos. En esta actividad se tomó 500 fotografías de Ruscus con diferentes características morfológicas.

Actividad 2: Diseño del algoritmo de visión artificial. En esta actividad se diseñó un algoritmo de visión artificial que extraiga las características morfológicas de la planta Ruscus previamente establecidas. Con esta actividad, se obtiene una base de datos refinada que posteriormente será procesada.

Actividad 3: Diseño de los algoritmos de clasificación. Se desarrolló 24 algoritmos de clasificación mediante redes neuronales y clasificadores automáticos.

Actividad 4: Entrenamiento de los algoritmos. Con los algoritmos ya diseñados se entrenó utilizando la base de datos resultante de la Actividad 2 de la Fase 2.

Fase 3: Realización de pruebas

Para el análisis del rendimiento de los algoritmos de clasificación empleados se utilizó el método de validación cruzada.

Actividad 1: Tabulación de los algoritmos. Se tabularon los resultados de la validación de los algoritmos con la base de datos obtenida en la Actividad 2 de la Fase 2.

Actividad 2: Selección del algoritmo. Se empleó el análisis comparativo de exactitud para seleccionar el mejor algoritmo en base a las tablas obtenidas en la actividad anterior.

Fase 4: Implementación del algoritmo

Se implementó el clasificador en tiempo real, mediante el uso de una cámara web y un espacio de trabajo con condiciones mejoradas de iluminación.

3.4. Consideraciones Bioéticas

El presente trabajo no causa afectaciones al ser humano ni al ambiente, debido a que la tecnología utilizada en esta aplicación es inofensiva; al contrario, mejora la salud laboral de los trabajadores al evitar procesos repetitivos por largar jornadas de trabajo en el proceso de clasificación de Ruscus.

CAPITULO IV
RESULTADOS Y DISCUCIÓN

Los resultados se organizan de acorde a las fases estipuladas en el marco metodológico.

4.1. Fase1: Identificación

De la investigación documental y la entrevista realizada (Anexo 1), se obtuvo la siguiente información:

Los tallos se ordenan en tres conjuntos por sus tamaños:

- Pequeños: [25, 35] cm.
- Medianos:]35, 45] cm.
- Grandes:]45, 55] cm.

El Ruscus se clasifica en dos conjuntos por su estado:

- Bueno: Ruscus con hojas verdes oscuras y abundantes. Las hojas no tienen huecos, líneas ni pigmentaciones de color blanco, amarillo y/o café. La base del tallo posee un color blanco indicativo de madurez, el contorno de las hojas es regular sin daños por la cosecha o por plagas.
- Malo: Ruscus descartable con hojas verdes claras. Base de tallo completamente verde, que es indicativo de falta de madurez. Las hojas presentan huecos, pigmentaciones o irregularidades. Poco follaje debido a maltrato en cosecha o por plagas.

En relación con el software empleado, se realizó una tabla comparativa entre los softwares Matlab y Python (ver Tabla 1).

Tabla 1

Comparativa de posible software a utilizar

Características	Matlab	Python
Soporta funciones matemáticas avanzadas	Si	Si
Soporta matrices y arreglos	Si	Si
Soporta visión artificial	Si	Si
Soporta Aprendizaje Automático	Si	Si
Soporta Redes Neuronales	Si	Si
Información disponible en la red	Si	Si
Soporte	Si	Limitado
Licencia	Si	Libre

Se eligió el software Matlab debido a las licencias que dispone la Universidad Técnica del Norte, las cuales incluyen soporte en vivo gratuito. "Los usuarios pueden llamar por teléfono o enviar un email para obtener ayuda del equipo experto de Matlab sobre su proyecto concreto" (*MATLAB vs. Python ¿Cuál se adapta mejor a sus necesidades?*, s. f.), y por el grado de experticia del autor en el manejo de éste, el utilizar un software nuevo implicaba tiempo de entrenamiento que no se disponía, al final se decantó por el software dominado.

4.2. Fase 2: Diseño

De las 500 fotografías tomadas a las plantas Ruscus con diferentes características morfológicas. La base de datos creada consta de 411 fotografías (Anexo 2); ya que, 89 fueron descartadas debidos a que presentaron datos atípicos por defectos en la recopilación. Posteriormente, se diseñó una función en Matlab que mediante visión artificial permite detectar el tamaño del tallo. El flujograma de la función creada se presenta en la Figura 4.1. La compilación del código arroja como resultado 233 plantas están en buen estado y 178 en mal estado.

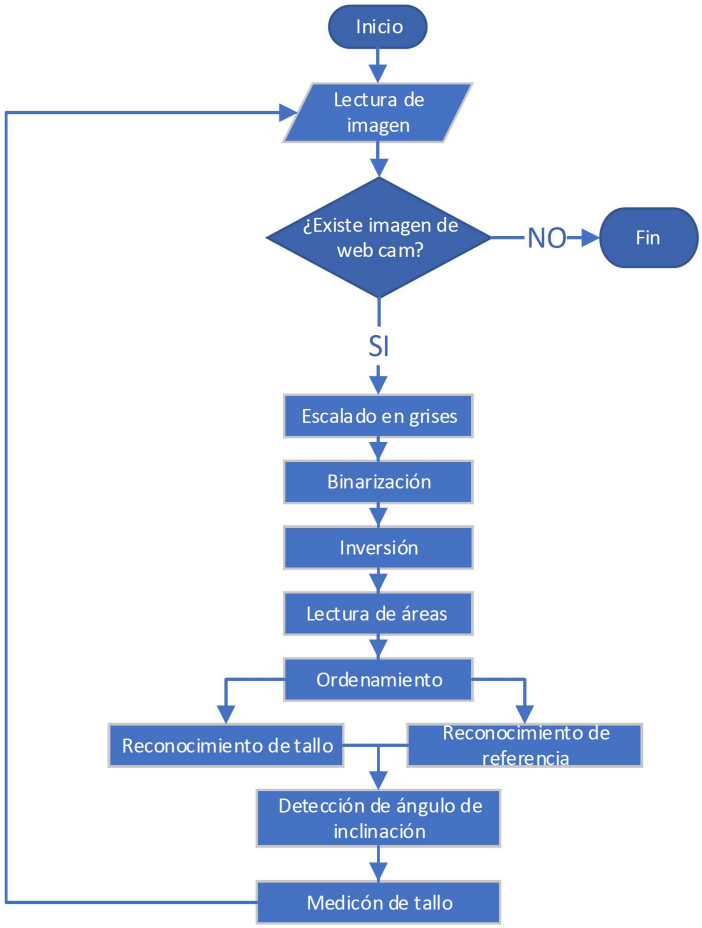


Figura 4.30. Algoritmo de medición de tallo

Adicional se indagó sobre los algoritmos y/o técnicas relevantes de identificación de características morfológicas en imágenes, se decantó por la técnica *Bag of features*. Esta técnica permite extraer las características morfológicas del Ruscus mediante una detección automática de particularidades de una imagen obtenida del conjunto de entrenamiento. El fundamento teórico de esta técnica se presentó en la sección 2.2.3.

Se realizó un programa en Matlab (*scipt*) que permite extraer las características comunes del conjunto de 411 imágenes de entrenamiento, las cuales previamente se han clasificado como “buenas” y/o “malas”. Como resultado de esta operación se obtiene una matriz de 411×201 , una fila por imagen con 201 columnas; de las cuales 200 son de los *clústeres* resultado de codificar 63 650 características, y una columna adicional que identifica el estado objetivo de la imagen (buena o descartable). La compilación se procesa en aproximadamente 2.7 segundos en un computador con procesador AMD Ryzen 9 con 16GB de memoria RAM, corriendo sobre Matlab 2023A.

Posteriormente, se utilizó el paquete *Classification Learner* de Matlab, en el cual se probaron los 24 modelos de clasificación que a continuación se detalla:

- Arboles de decisión
    - Árbol de decisión fina. Muchas hojas para hacer muchas distinciones finas entre clases, con un número máximo de 100.
    - Árbol de decisión media. Número medio de hojas para decisiones más finas entre clases, con número máximo de 20.
    - Árbol de decisión gruesa. Pocas hojas para hacer decisiones gruesas, con un número máximo de 4.
- Regresión Logística
    - Regresión logística binaria. No se puede cambiar los parámetros del modelo.
    - Regresión logística eficiente. Se puede cambiar los híperparámeteros, la tolerancia de alfa y beta.
- Naive Bayes
    - Naive Bayes Gaussiano. Bajo el supuesto de una distribución normal.
    - Kernel Naive Bayes. Utiliza una función de estimación de densidad de suavizado de kernel.
- SVM
    - SVM Lineal. Usa coeficientes de predicción lineal.
    - SVM Cuadrática. Usa coeficientes de predicción cuadráticos.
    - SVM Cúbica. Usa coeficientes de predicción cúbicos.
    - SVM Gaussiano fino. Permite variaciones rápidas en la función de respuesta.
    - SVM Gaussiano medio. Brinda una función de respuesta menos flexible que el SVM Gaussiano fino.
    - SVM Gaussiano grueso. Brinda una función de respuesta rígida.
- KNN
    - KNN Fino. Se establece el número de vecinos en 1.
    - KNN Medio. Se establece el número de vecinos en 10.
    - KNN Grueso. Se establece el número de vecinos en 100.
    - KNN Coseno. Usa distinciones medias entre clases, usando una métrica de distancia Coseno.

  - KNN Cúbico. Distinciones medias entre clases, usando una métrica de distancia Cúbica.
  - KNN Ponderado. Usa distinciones medias entre clases, utilizando un peso de distancia.
- Redes Neuronales
  - Red neuronal estrecha. Aumenta con la configuración del tamaño de la primera capa.
  - Red neuronal media. Aumenta con la configuración del tamaño de la primera capa.
  - Red neuronal amplia. Aumenta con la configuración del tamaño de la primera capa.
  - Red neuronal bicapa. Aumenta con la configuración del tamaño de la primera y segunda capa.
  - Red neuronal con tres capas. Aumenta con la configuración del tamaño de la primera, segunda y tercera capa.

4.3. Fase 3: Realización de pruebas

Para la validación no se dividió el *dataset* en dos conjuntos (entrenamiento y validación), sino se lo hizo mediante validación cruzada de cinco elementos debido a la limitada cantidad del *dataset*, los resultados fueron:

4.3.1. ***Árbol de decisión fina.*** La figura 4.2 representa la matriz de confusión en la que se puede observar que existen una cantidad considerable de Ruscus mal clasificadas, de la intersección de las filas que representan las clases reales y las columnas que representan las clases de predicción se puede observar que 45 tallos buenos se predicen como malos y 40 tallos buenos se predicen como buenos.

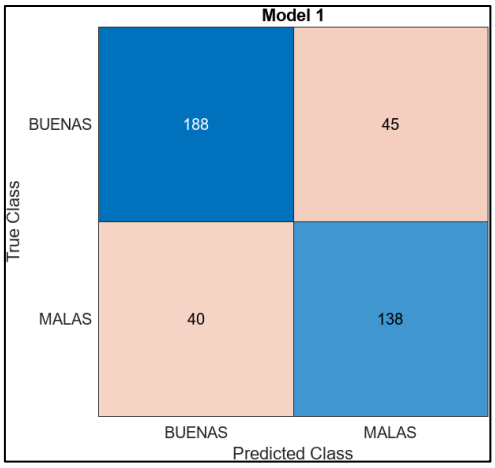


Figura 4.31. Matriz de confusión del modelo: Árbol de decisión fina

Fuente: Imagen realizada por el autor en Matlab 2023A.

4.3.2. ***Árbol de decisión media.*** En la matriz de confusión se puede observar que existen una cantidad considerable de Ruscus mal clasificados, al igual que en modelo anterior (ver figura 4.3).

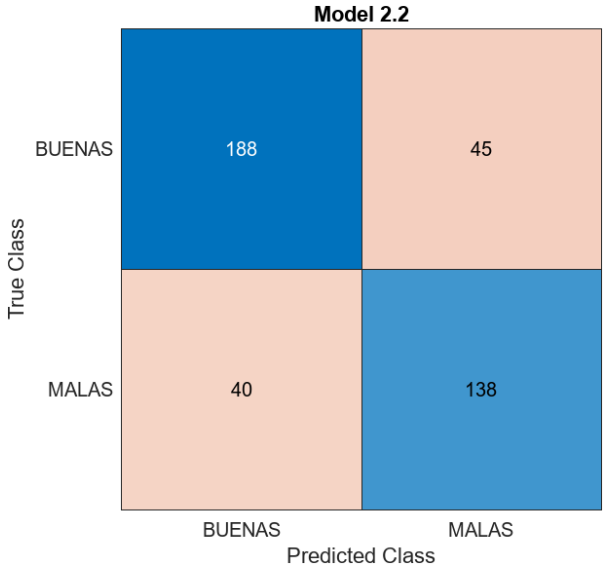


Figura 4.32. Matriz de confusión del modelo: Árbol de decisión medio

Fuente: Imagen realizada por el autor en Matlab 2023A.

4.3.3. ***Árbol de decisión gruesa.*** En la matriz de confusión se puede observar una mejora ante los dos modelos anteriores, como se puede observar en la figura 4.4.

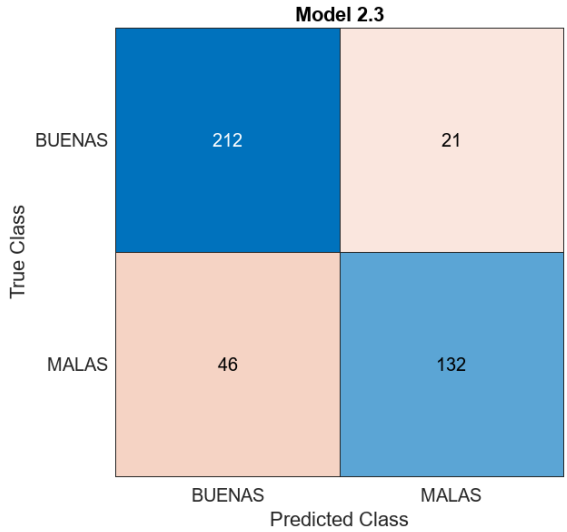


Figura 4.33. Matriz de confusión del modelo: Árbol de decisión gruesa

Fuente: Imagen realizada por el autor en Matlab 2023A.

4.3.4. ***Regresión Logística Binaria.*** En la matriz de confusión se puede observar un detrimento en la exactitud en comparación a los modelos de Árbol de decisión (ver figura 4.5).

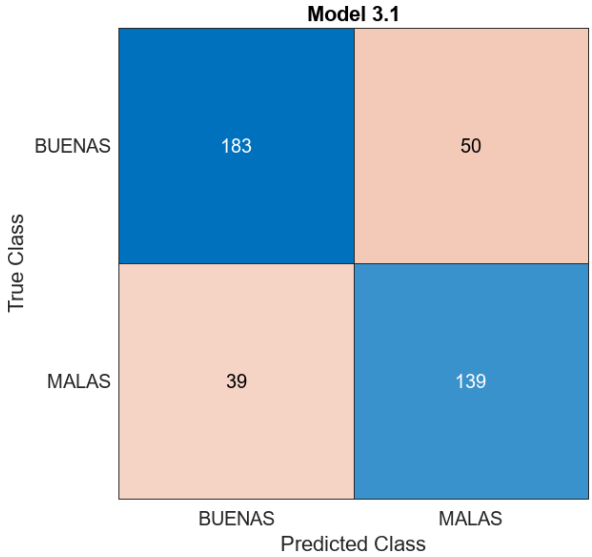


Figura 4.34. Matriz de confusión del modelo: Regresión logística binaria

Fuente: Imagen realizada por el autor en Matlab 2023A.

4.3.5. ***Regresión Logística Eficiente.*** En la figura 4.6 se puede visualizar la matriz de confusión, donde se puede observar una mejora en la exactitud en comparación al modelo anterior.

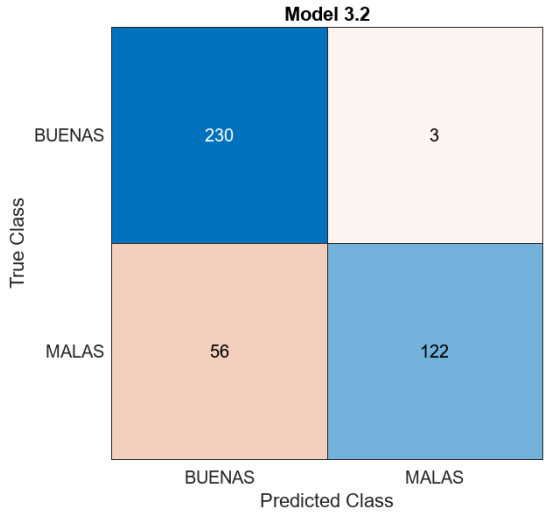


Figura 4.35. Matriz de confusión del modelo: Regresión logística eficiente

Fuente: Imagen realizada por el autor en Matlab 2023A.

4.3.6. ***Naive Bayes Gaussina.*** En la matriz de confusión se puede observar una mejora en la exactitud en comparación al modelo anterior (ver figura 4.7).

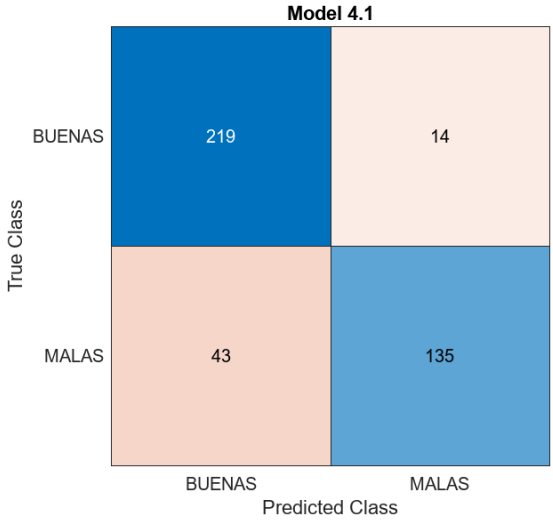


Figura 4.36. Matriz de confusión del modelo: Naive Bayes Gaussina

Fuente: Imagen realizada por el autor en Matlab 2023A.

4.3.7. ***Kernel Naive Bayes.*** En la matriz de confusión se puede observar un detrimento en la exactitud en comparación al modelo anterior, como se parecía en la figura 4.8.

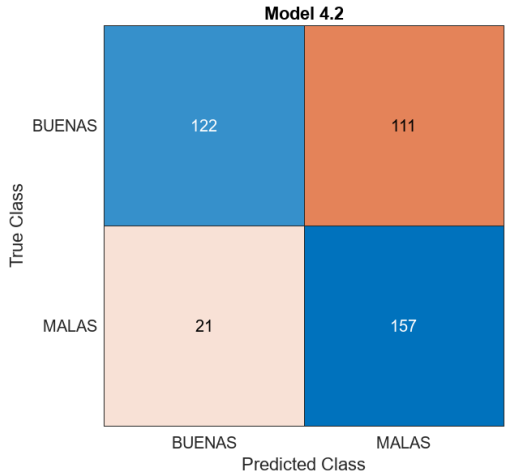


Figura 4.37. Matriz de confusión del modelo: Naive Bayes Kernel

Fuente: Imagen realizada por el autor en Matlab 2023A.

4.3.8. ***SVM Lineal.*** En la matriz de confusión se puede observar un aumento significativo en la exactitud en comparación a todos los modelos (ver figura 4.9).

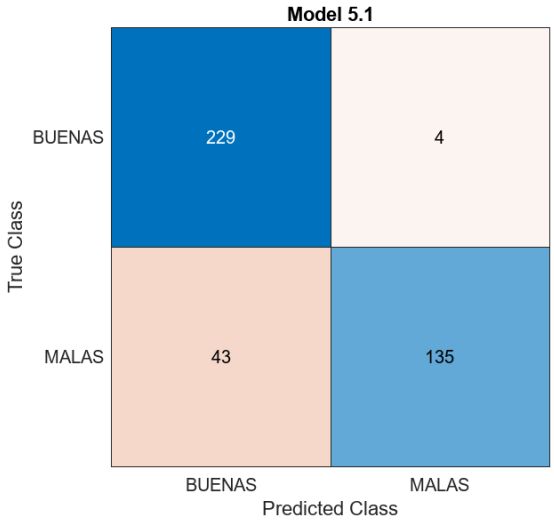


Figura 4.38. Matriz de confusión del modelo: SVM Lineal

Fuente: Imagen realizada por el autor en Matlab 2023A.

4.3.9. ***SVM Cuadrática.*** En la figura 4.10 se observa la matriz de confusión, en la cual se puede observar un ligero detrimento o en la exactitud en comparación al anterior modelo.

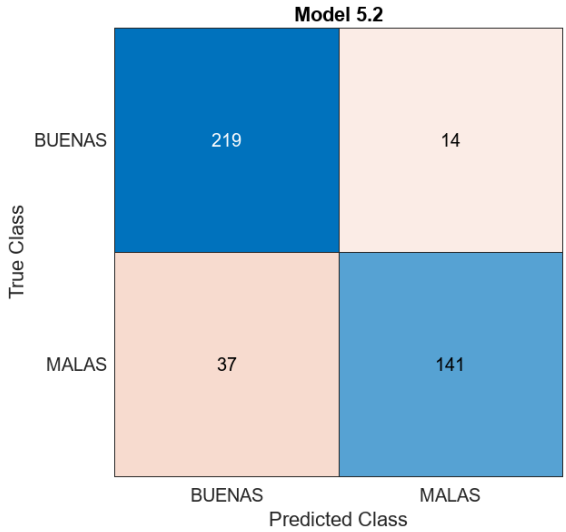


Figura 4.39. Matriz de confusión del modelo: SVM Cuadrática

Fuente: Imagen realizada por el autor en Matlab 2023A.

4.3.10. ***SVM Cúbica.*** En la matriz de confusión se puede observar un ligero detrimento o en la exactitud en comparación al anterior modelo (ver figura 4.11).

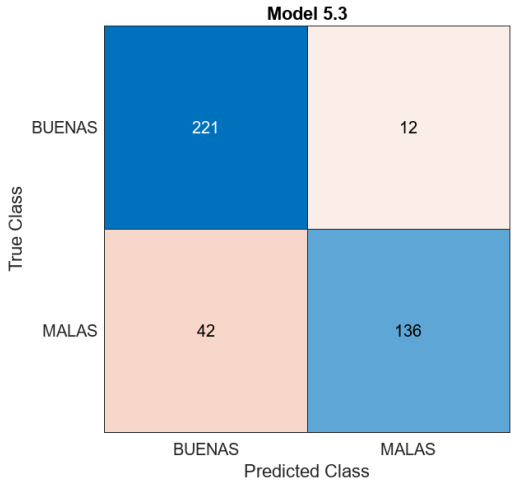


Figura 4.40. Matriz de confusión del modelo: SVM Cúbica

Fuente: Imagen realizada por el autor en Matlab 2023A.

4.3.11. ***SVM Gaussiana Fina.*** En la matriz de confusión se puede observar un total detrimento en la exactitud en comparación a todos los anteriores modelos, como se muestra en la figura 4.12.

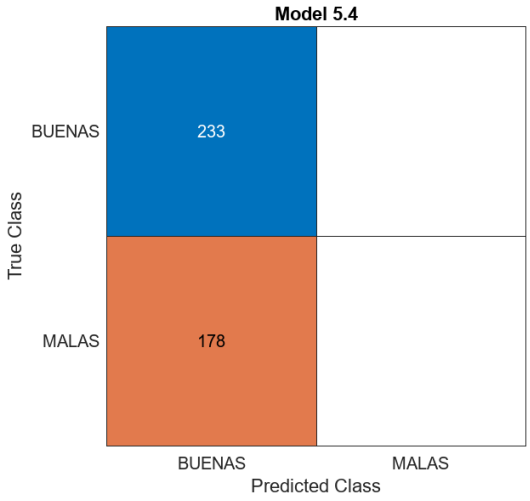


Figura 4.41. Matriz de confusión del modelo: SVM Gaussiana Fina

Fuente: Imagen realizada por el autor en Matlab 2023A.

4.3.12. ***SVM Gaussiana Media.*** En la figura 4.13. se muestra la matriz de confusión, donde se puede observar un incremento en la exactitud en comparación al anterior modelo.

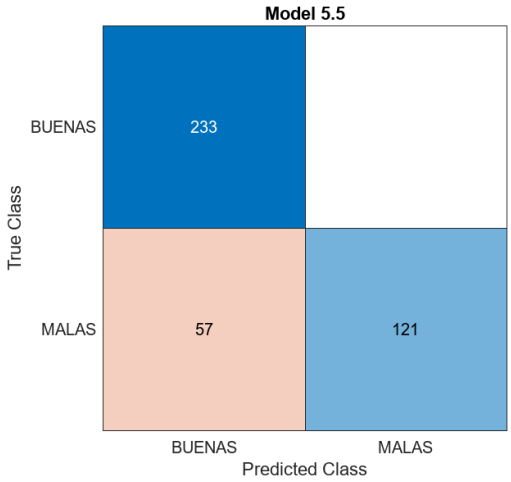


Figura 4.42. Matriz de confusión del modelo: SVM Gaussiana Media

Fuente: Imagen realizada por el autor en Matlab 2023A.

4.3.13. ***SVM Gaussiana Gruesa.*** En la matriz de confusión se puede observar un incremento en la exactitud en comparación al anterior modelo (ver figura 4.14).

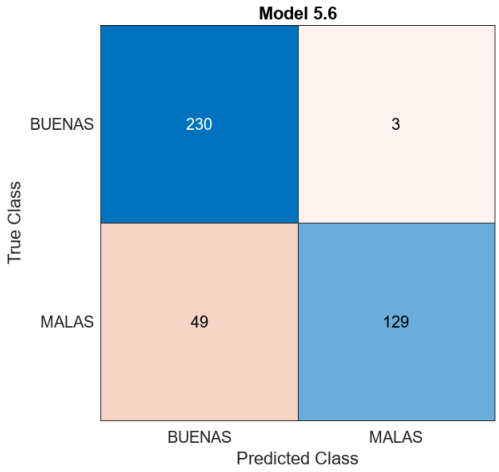


Figura 4.43. Matriz de confusión del modelo: SVM Gaussiana Gruesa

Fuente: Imagen realizada por el autor en Matlab 2023A.

4.3.14. ***KNN Fina.*** En la figura 4.15 se observa la matriz de confusión, donde se aprecia una baja exactitud en la clasificación de plantas buenas.

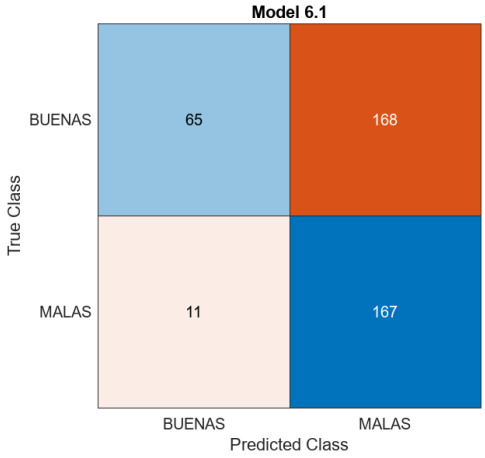


Figura 4.44. Matriz de confusión del modelo: KNN Fina

Fuente: Imagen realizada por el autor en Matlab 2023A.

4.3.15. ***KNN Media.*** En la matriz de confusión se pudo observar una leve mejora en comparación al modelo anterior, no obstante, se evidenció una baja exactitud en la clasificación de plantas buenas (ver figura 4.16).

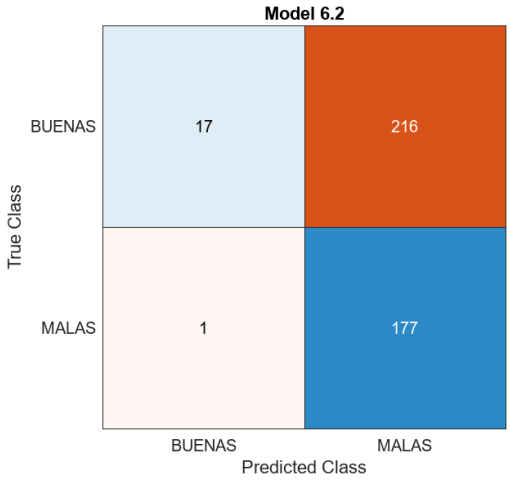


Figura 4.45. Matriz de confusión del modelo: KNN media

Fuente: Imagen realizada por el autor en Matlab 2023A.

4.3.16. *KNN Gruesa.* En la matriz de confusión se pudo observar un detrimento en la exactitud en comparación al modelo anterior, como se muestra en la figura 4.17.

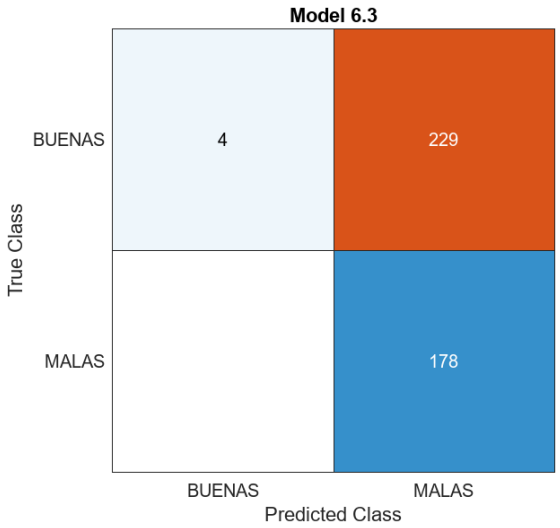


Figura 4.46. Matriz de confusión del modelo: KNN gruesa

Fuente: Imagen realizada por el autor en Matlab 2023A.

4.3.17. *KNN Coseno.* En la matriz de confusión se pudo observar un aumento significativo en la exactitud en comparación al modelo anterior (ver figura 4.18).

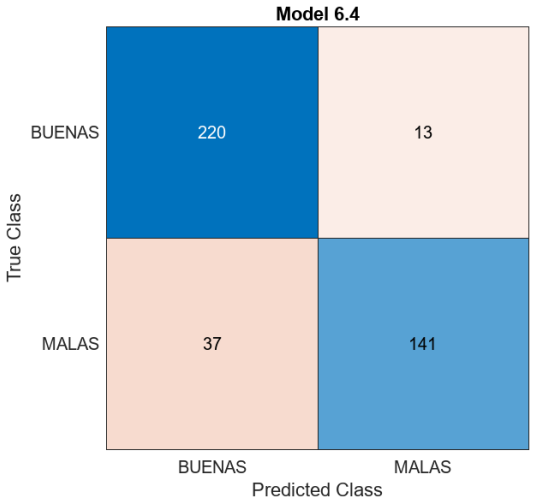


Figura 4.47. Matriz de confusión del modelo: KNN Coseno

Fuente: Imagen realizada por el autor en Matlab 2023A.

4.3.18. ***KNN Cúbico.*** En la figura 4.19 se muestra la matriz de confusión, donde se pudo observar un detrimento leve en la exactitud en comparación al modelo anterior.

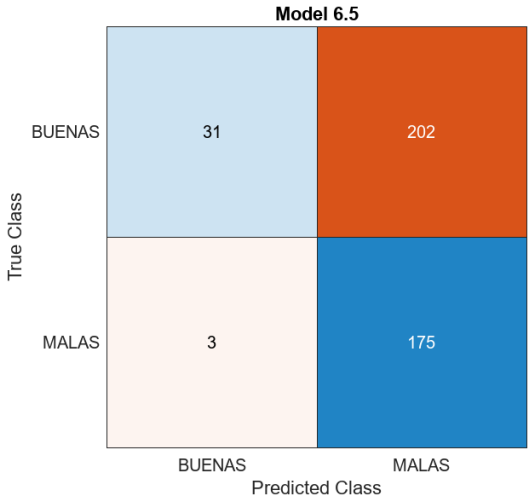


Figura 4.48. Matriz de confusión del modelo: KNN Cúbico

Fuente: Imagen realizada por el autor en Matlab 2023A.

4.3.19. ***KNN Ponderado.*** En la matriz de confusión se pudo observar un leve detrimento significativo en la exactitud en comparación al modelo anterior (ver figura 4.20).

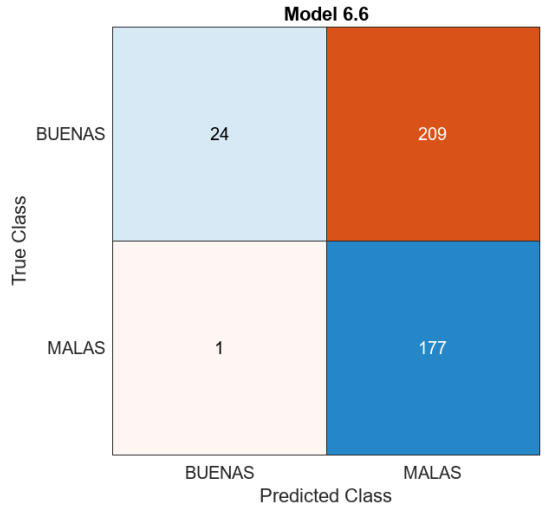


Figura 4.49. Matriz de confusión del modelo: KNN Ponderado

Fuente: Imagen realizada por el autor en Matlab 2023A.

4.3.20. ***Red Neuronal Estrecha.*** En la matriz de confusión se pudo observar un aumento significativo en la exactitud en comparación al modelo anterior, como se observa en la figura 4.21.

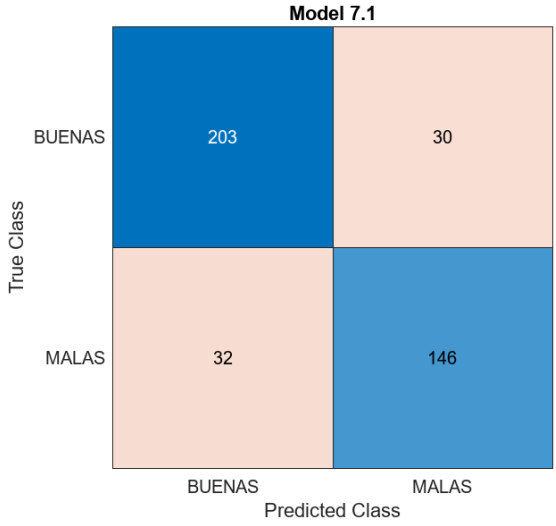


Figura 4.50. Matriz de confusión del modelo: Red Neuronal Estrecha

Fuente: Imagen realizada por el autor en Matlab 2023A.

4.3.21. ***Red Neuronal Media.*** En la matriz de confusión se pudo observar un leve aumento en la exactitud en comparación al modelo anterior (ver figura 4.22).

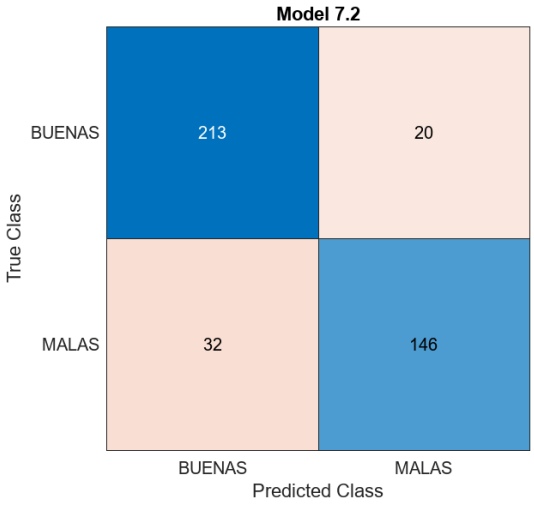


Figura 4.51. Matriz de confusión del modelo: Red Neuronal Media

Fuente: Imagen realizada por el autor en Matlab 2023A.

4.3.22. ***Red Neuronal Amplia.*** En la figura 4.23 se muestra la matriz de confusión con un leve detrimento en la exactitud en comparación al modelo anterior.

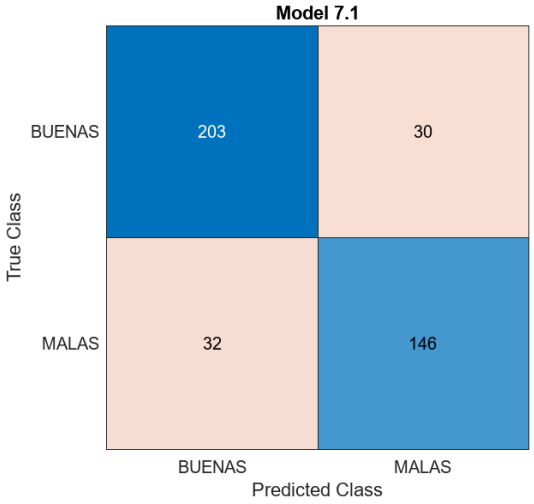


Figura 4.52. Matriz de confusión del modelo: Red Neuronal Amplia

Fuente: Imagen realizada por el autor en Matlab 2023A.

4.3.23. ***Red Neuronal Bicapa.*** En la matriz de confusión se pudo observar un leve detrimento en la exactitud en comparación al modelo anterior (ver figura 4.24).

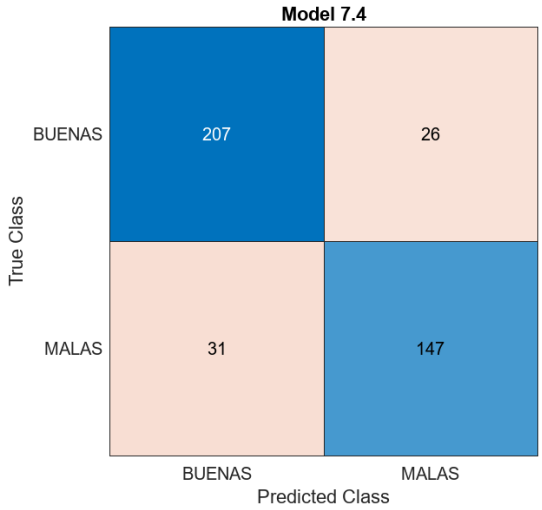


Figura 4.53. Matriz de confusión del modelo: Red Neuronal Bicapa

Fuente: Imagen realizada por el autor en Matlab 2023A.

4.3.24. ***Red Neuronal de tres capas.*** En la matriz de confusión se pudo observar un leve detrimento en la exactitud en comparación al modelo anterior, como se muestra en la figura 4.25.

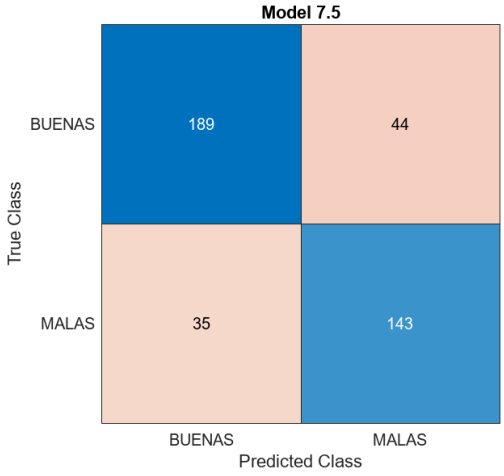


Figura 4.54. Matriz de confusión del modelo: Red Neuronal de tres capas

Fuente: Imagen realizada por el autor en Matlab 2023A.

Actividad 10: Tabulación de los algoritmos; se tabuló los resultados de la validación de los algoritmos descritos en la actividad 7 (ver tabla 2).

Tabla 2
Resultados de exactitud de modelos

Modelo	Exactitud
Árbol de decisión fina	79,3 %
Árbol de decisión media	79,3 %
Árbol de decisión gruesa	83,7 %
Regresión logística binaria	78,3 %
Regresión logística eficiente	85,6 %
Naive Bayes Gaussiano	86,1 %
Kernel Naive Bayes	67,9 %
SVM Lineal	**88,6 %**
SVM Cuadrático	**87,6 %**
SVM Cúbico	86,9 %
SVM Gaussiano Fino	56,7 %
SVM Gaussiano Medio	86,1 %
SVM Gaussiano Grueso	87,3 %
KNN Fino	56,4 %
KNN Medio	47,2 %
KNN Grueso	44,3 %
KNN Coseno	**87,8 %**
KNN Cúbico	50,1 %
KNN Ponderado	48,9 %
Red Neuronal Estrecha	84,9 %
Red Neuronal Media	87,3 %
Red Neuronal Amplia	86,6 %
Red Neuronal Bicapa	86,1 %
Red Neuronal con tres capas	80,8 %

El algoritmo se seleccionó en base al análisis comparativo presentado en la tabla 2, los clasificadores con mayor precisión en validación cruzada se encuentran marcados en negrita. El algoritmo que presenta el mejor rendimiento es el SVM Lineal, el cual presenta una exactitud del 88.6%. La figura 4.26 es el diagrama de dispersión de este algoritmo, en el cual las “x” indican los elementos mal clasificados, en función de dos características diferentes.

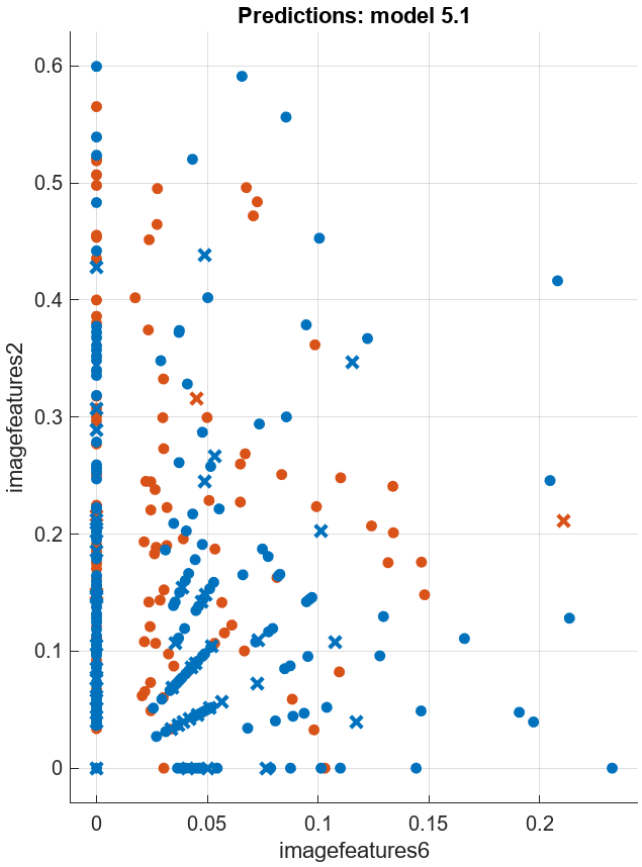


Figura 4.55. Diagrama de dispersión entre dos características del modelo Lineal SVM

Fuente: Imagen realizada por el autor en Matlab 2023A.

4.4. Fase 4: Implementación del algoritmo

Se implementó el clasificador SVM Lineal en tiempo real utilizando una cámara web emulada mediante la cámara de un celular Xiami Redmi Note 9 Pro conectado a la aplicación Iriun cam, con resolución de fotografía 3472x4640 pixeles. El flujograma comparativo entre el entrenamiento y la predicción en tiempo real puede ser vistos en la Figura 4.27. El *script* empleado se presenta en el Anexo 3.

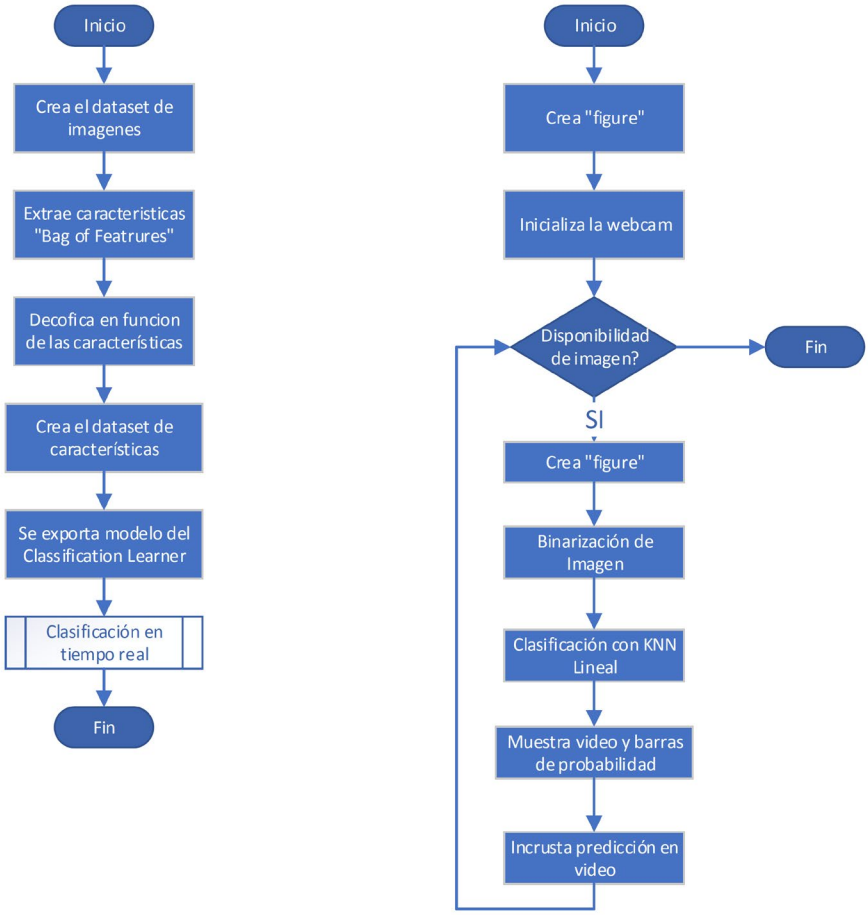
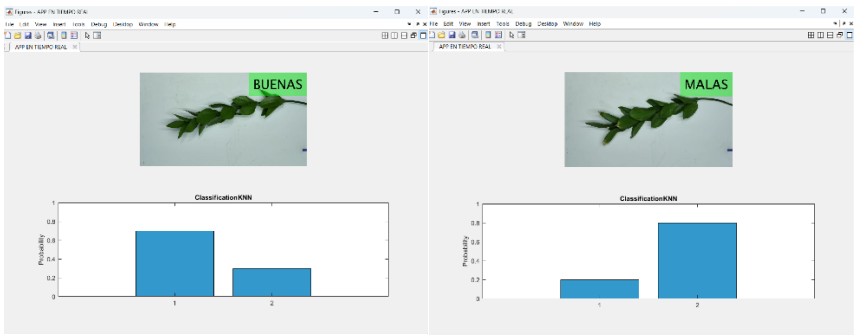


Figura 4.56. Flujograma de entrenamiento y clasificación en tiempo real

En la figura 4.28, se puede observar el aplicativo en tiempo real, que se compone de una ventana donde se muestra el video y un grafico que índica la probabilidad de ser clasificado como tallo bueno o malo.



(a) (b)

Figura 4.57. Aplicativo en tiempo real: (a) Tallo bueno (b)Tallo malo

Fuente: Imagen realizada por el autor en Matlab 2023A.

CAPITULO V
CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- Los parámetros adecuados para la clasificación del Ruscus dentro del contexto de la plantación GM Familiar son el tamaño y el estado de los tallos, que pueden ser buenos o malos, entre las plantas malas o descartables se encuentran las plantas tiernas, con imperfecciones, poco follaje y atacadas por plagas o enfermedades, en ese sentido las redes neuronales o los clasificadores automáticos son métodos apropiados para la clasificación de las plantas.

- Es pertinente realizar la medición del tamaño mediante técnicas de visión artificial y mediante comparación de elementos de imagen, para la clasificación es válido utilizar técnicas de inteligencia artificial.

- Debido a la heterogeneidad de las características de las imágenes es pertinente utilizar la técnica Bag of features de extracción de características morfológicas de las imágenes, de esta forma se extraen 63 650 particularidades diferenciadoras agrupados en 200 clústeres.

- La comparación con un amplio abanico de técnicas de redes neuronales y clasificadores automáticos soportados por el software Matlab enriquece el estudio y sustenta la selección del modelo de Máquina de Soporte Vectorial Lineal, el cual tiene una precisión del 88,6% aprobada mediante validación cruzada de cinco elementos.

- La implementación del sistema de clasificación en tiempo real requiere de una cámara web de calidad y condiciones mejoradas de iluminación que permiten estandarizar las condiciones de captura de imagen.

5.2. Recomendaciones

- Se recomienda para futuras investigaciones contactar con exportadores nacionales e importadores en el extranjero que permitan conocer los requerimientos en cuanto a características morfológicas de los diferentes actores de la cadena de comercialización.
- Contrastar la extracción de las características morfológicas de la técnica utilizada *Bag of features* con otras técnicas de visión artificial y en otros softwares.
- Se recomienda ejecutar los algoritmos planteados con un *dataset* de imágenes ampliado para mejorar la precisión de los modelos generados.
- Se recomienda utilizar los avances planteados en la presente investigación para generar un proyecto integral de automatización industrial que permita generar maquinaria especializada en clasificación de Ruscus y otros productos.

REFERENCIAS

AEPSAL. (2017, mayo 25). Eliminar el trabajo repetitivo puede aumentar la productividad. *Asociación de especialistas en prevención y salud laboral*. https://www.aepsal.com/eliminar-trabajo-repetitivo-aumenta-productividad/

Al Ohali, Y. (2011). Computer vision based date fruit grading system: Design and implementation. *Journal of King Saud University - Computer and Information Sciences*, *23*(1), 29-36. https://doi.org/10.1016/j.jksuci.2010.03.003

Algoritmo Perceptrón. (s. f.). Recuperado 18 de noviembre de 2021, de https://www.cienciadedatos.net/documentos/50_algoritmo_perceptron

Andrea, C.-C., Mauricio Daniel, B. B., & Jose Misael, J. B. (2017). Precise weed and maize classification through convolutional neuronal networks. *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, 1-6. https://doi.org/10.1109/ETCM.2017.8247469

Arber, A. (1924). Danae, Ruscus, and Semele: A Morphological Study. *Annals of Botany*, *38*(150), 229-260.

Ayala, J. (2020). *RPubs—Reducción de dimensionalidad (PCA)*. https://rpubs.com/JairoAyala/574796

Banco Central del Ecuador. (2020, noviembre 30). La economía ecuatoriana se recuperará 3,1% en 2021. https://www.bce.fin.ec/index.php/boletines-de-prensa-archivo/item/1394-la-economia-ecuatoriana-se-recuperara-3-1-en-2021

Batioua, I., Benouini, R., Zenkouar, K., & Zahi, A. (2018). Image classification using separable invariants moments based on Racah polynomials. *Procedia Computer Science*, *127*, 320-327. https://doi.org/10.1016/j.procs.2018.01.128

Blum, A. (2007). Machine Learning Theory. *Department of Computer Science Carnegie Mellon University*, *26*, 4.

Bodenhofer, U. (2002). Genetic Algorithms: Theory and Applications. *Johannes Kepler Universitat*, 108.

Bonaccorso, G. (2017). *Machine learning algorithms: A reference guide to popular algorithms for data science and machine learning*. Packt.

Breilh, J. (2007). Nuevo modelo de acumulación y agroindustria: Las implicaciones ecológicas y epidemiológicas de la floricultura en Ecuador. *Ciência & Saúde Coletiva*, *12*(1), 91-104. https://doi.org/10.1590/S1413-81232007000100013

Brusco de hojas anchas, Laurelillo, Laurel de Alejandría, Planta de la mosquita, Ruscus—Ruscus hypoglossum. (s. f.). Recuperado 22 de febrero de 2021, de https://fichas.infojardin.com/arbustos/ruscus-hypoglossum-brusco-laurelillo.htm

Caicedo Bravo, E. F., & Lopez Sotelo, J. A. (2009). *Una aproximacion practica a las redes neuronales artificiales*. Programa Editorial Universidad del Valle. https://elibro.net/es/lc/utnorte/titulos/129183

Campaña, R., Tutillo, M., & Verjan, L. (2014). *Proyecto asociativo para los pequeños productores del follaje ornamental (Ruscus Hypophyllum) de las parroquias de Guallabamba, Yaruquí y Checa hacía el mejoramiento de la comercialización y distribución en las florícolas del sector Cayambe* [Universidad Politécnica Salesiana sede Quito]. https://dspace.ups.edu.ec/bitstream/123456789/7416/1/UPS-QT06282.pdf

Computer Vision Toolbox. (s. f.). Recuperado 17 de abril de 2023, de https://la.mathworks.com/products/computer-vision.html

Constitución de la Republica del Ecuador, Registro oficial No.449 79 (2008).

Convenio C184—Convenio sobre la seguridad y la salud en la agricultura, 2001 (núm. 184). (s. f.). Recuperado 19 de noviembre de 2021, de

https://www.ilo.org/dyn/normlex/es/f?p=NORMLEXPUB:12100:0::NO::P12100_ILO_CODE:C184

DataScients. (2022, marzo 7). Perceptron: ¿qué es y para qué sirve? *Formation Data Science | DataScientest.com*. https://datascientest.com/es/perceptron-que-es-y-para-que-sirve

Díaz, D. F. G. (2019). *Plan de exportación de tallos de follaje Ruscus de la empresa Drag Filding del cantón Quinche al mercado de Miami-EEUU.* Pontificia Universidad Católica del Ecuador sede Ibarra.

Erdélyi, V., & Jánosi, L. (s. f.). *Mechatronics in Agriculture*. 1.

Expoflores. (2019). *Informe Anual de Exportaciones 2019*. https://expoflores.com/wp-content/uploads/2020/04/reporte-anual_Ecuador_2019.pdf

Facco, P., Santomaso, A. C., & Barolo, M. (2017). Artificial vision system for particle size characterization from bulk materials. *Chemical Engineering Science*, *164*, 246-257. https://doi.org/10.1016/j.ces.2017.01.053

Fuentes, M. S., Zelaya, N. A. L., & Avila, J. L. O. (2020). Coffee Fruit Recognition Using Artificial Vision and neural networks. *2020 5th International Conference on Control and Robotics Engineering (ICCRE)*, 224-228. https://doi.org/10.1109/ICCRE49379.2020.9096441

Gámez, H., Orejuela, J. P., Salas, Ó., & Bravo, J. J. (2016). Aplicación de mapas de Kohonen para la priorización de zonas de mercado: Una Aproximación práctica. *Revista de la escuela de Ingeniería de Antioquia*. http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1794-12372016000100012

Gonzales, R. R., Doval, Y. R., & Pérez, O. M. (2002). Estrés Laboral, consideraciones sobre sus características y formas de afrontamiento. *Revista Internacional de Psicología*, *3*(01), 1-19. https://doi.org/10.33670/18181023.v3i01.13

Google Maps. (s. f.). Google Maps. Recuperado 26 de febrero de 2021, de https://www.google.com.ec/maps

Hassanien, A. E., Salem, A.-B. M., Ramadan, R., & Kim, T. (Eds.). (2012). *Advanced Machine Learning Technologies and Applications* (Vol. 322). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-35326-0

IBM Docs. (2021, noviembre 9). https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us-south.containers.appdomain.cloud/docs/es/spss-statistics/SaaS?topic=networks-what-is-neural-network

Kramer, O. (2013). K-Nearest Neighbors. En O. Kramer, *Dimensionality Reduction with Unsupervised Nearest Neighbors* (Vol. 51, pp. 13-23). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-38652-7_2

Laurel de Alejandria-Laurelillo-Ruscus Hypoglossum – Para Mi Jardín. (s. f.). Recuperado 22 de febrero de 2021, de https://paramijardin.com/plantas/arbustos/laurel-de-alejandria-laurelillo-ruscus-hypoglossum/

LeCun, Y., Kavukcuoglu, K., & Farabet, C. (2010). Convolutional networks and applications in vision. *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, 253-256. https://doi.org/10.1109/ISCAS.2010.5537907

Machine Learning vs Neural Networks: What is the Difference? (2020, febrero 13). UpGrad Blog. https://www.upgrad.com/blog/machine-learning-vs-neural-networks/

Maduell, E. (2012). *Vision Artificial* (Universitat Oberta de Catalunya).

Manjarrez, L. (2014). *Relaciones Neuronales Para Determinar la Atenuación del Valor de la Aceleración Máxima en Superficie de Sitios en Roca Para Zonas de Subducción*.

MATLAB - El lenguaje del cálculo técnico. (2023). https://la.mathworks.com/products/matlab.html

MATLAB vs. Python ¿Cuál se adapta mejor a sus necesidades? (s. f.). Recuperado 8 de mayo de 2023, de https://la.mathworks.com/products/matlab/matlab-vs-python.html

Mohamed, B., Issam, A., Mohamed, A., & Abdellatif, B. (2015). ECG Image Classification in Real time based on the Haar-like Features and Artificial Neural Networks. *Procedia Computer Science*, *73*, 32-39. https://doi.org/10.1016/j.procs.2015.12.045

Oke, S. A. (2004). *A Literature Review on Artificial Intelligence*. 37.

Ortiz, O. G., & Rozo, M. E. V. (2013). *Introducción a la ingeniería* (primera). Ecoe Ediciones.

Pantazi, X. E., Moshou, D., & Bochtis, D. (2020). Chapter 2—Artificial intelligence in agriculture. En X. E. Pantazi, D. Moshou, & D. Bochtis (Eds.), *Intelligent Data Mining and Fusion Systems in Agriculture* (pp. 17-101). Academic Press. https://doi.org/10.1016/B978-0-12-814391-9.00002-9

Qué son las redes neuronales y sus funciones. (2019, octubre 22). *ATRIA Innovation*. https://www.atriainnovation.com/que-son-las-redes-neuronales-y-sus-funciones/

Ramírez, L. M. P. (2010). *Ruscus (Ruscus hypoglossum L.): Produccinón y manejo poscosecha*.

Redes Neuronales y Deep Learning. Capítulo 2: La neurona. (2021, abril 8). *Future Space S.A.* https://www.futurespace.es/redes-neuronales-y-deep-learning-capitulo-2-la-neurona/

Reglamento de Seguridad y Salud de los Trabajadores, Decretro Ejecutivo 2393, 71 (2003).

Sanchez Calle, A. (2005). *Aplicaciones de la vision artificial y la biometria informatica*. Dykinson. https://elibro.net/es/lc/utnorte/titulos/60927

Sancho Caparrini, F. (2020). *Algoritmos de Clustering—Fernando Sancho Caparrini*. http://www.cs.us.es/~fsancho/?e=230

Sein, J. L. G. (2019). *Innovaciones tecnológicas, inteligencia artificial y derechos humanos en el trabajo*. *II*, 57-62.

Statistics and Machine Learning Toolbox. (s. f.). Recuperado 17 de abril de 2023, de https://la.mathworks.com/products/statistics.html

Telégrafo, E. (2020, febrero 5). *Exportación de flores sube en volumen, pero su precio bajó*. El Telégrafo. https://www.eltelegrafo.com.ec/noticias/economia/4/flores-exportacion-ecuador

Vélez Serrano, J. F. (2003). *Vision por computador*. Dykinson. https://elibro.net/es/lc/utnorte/titulos/104894

What is a Genetic Algorithm? (s. f.). Recuperado 5 de mayo de 2023, de https://www.generativedesign.org/02-deeper-dive/02-04_genetic-algorithms/02-04-01_what-is-a-genetic-algorithm

Zhang, H. (2004). *The Optimality of Naive Bayes*.

ANEXOS

Anexo 1. Guion de la Entrevista

Tema: Características morfológicas deseables y proceso postcosecha.

Entrevistado: Sr. Fabian Guerra (Dueño de la Plantación de Ruscus GM Familiar)

Objetivo: Recolectar información sobre las características morfológicas deseables, plagas y enfermedades para la postcosecha y la comercialización del Ruscus, en base a la experiencia del productor.

Estimador 1. Parámetros de cosecha de Ruscus.

Respuesta. "Los cosechadores observan los tamaños y follaje de los tallos, así mismo que no tengan enfermedades ni ninguna imperfección, pero siempre se pasan algunos tallos malos".

Estimador 2. Parámetros de postcosecha.

Repuesta. "En la postcosecha tenemos una tabla con medidas donde se coloca el tallo para verificar su tamaño, también se hace la limpieza de las hojas al mismo tiempo que se verifica si las hojas no tienen imperfecciones de plagas, de lancha o de cualquier enfermedad".

Estimador 3. Conjuntos de clasificación.

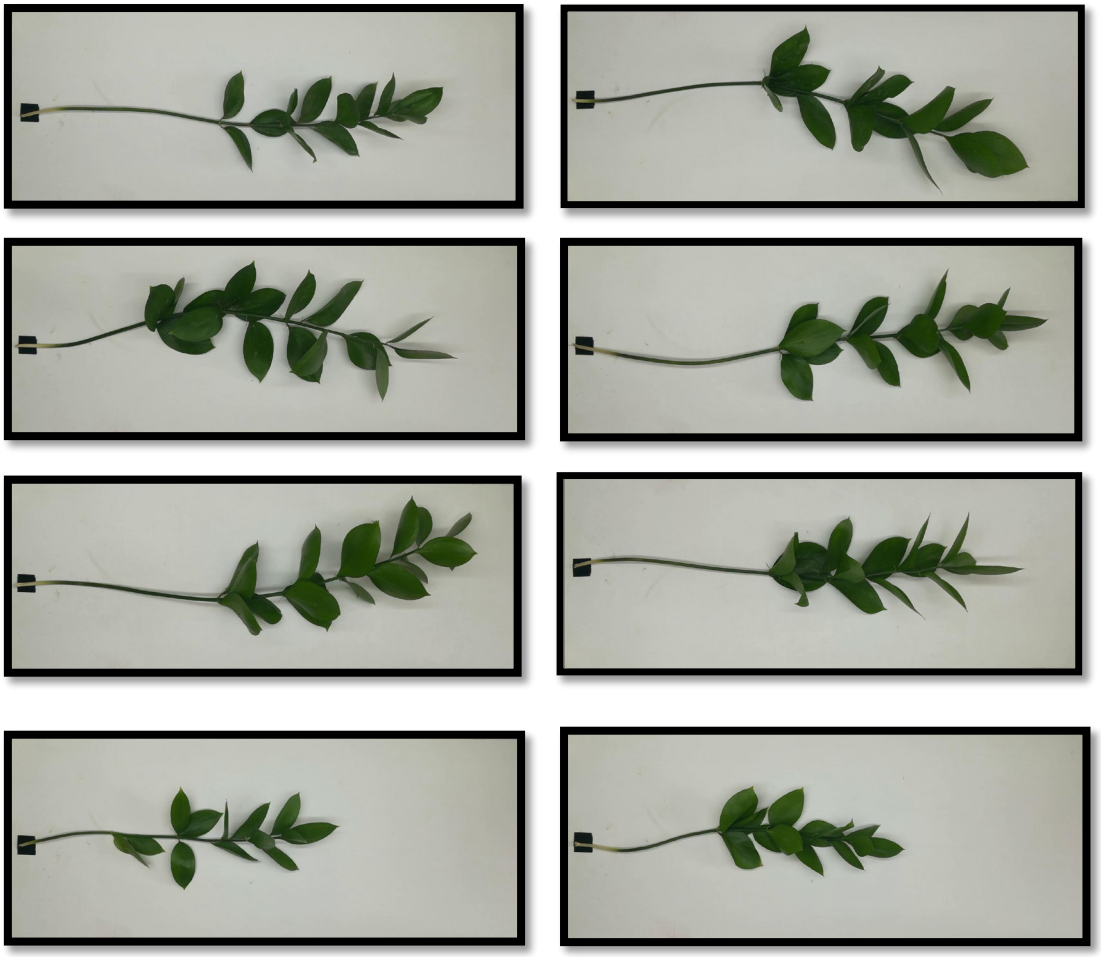
Repuesta. "Bueno primero en la limpieza se desecha todo lo malo y quedan solo tallos buenos, y luego pasa a medición y se clasifican en tallo pequeños de entre 25cm a 35 cm, tallos medianos de 35cm a 45 cm y tallos grandes de 45 cm o más".

Estimador 4. Tiempo y ergonomía.

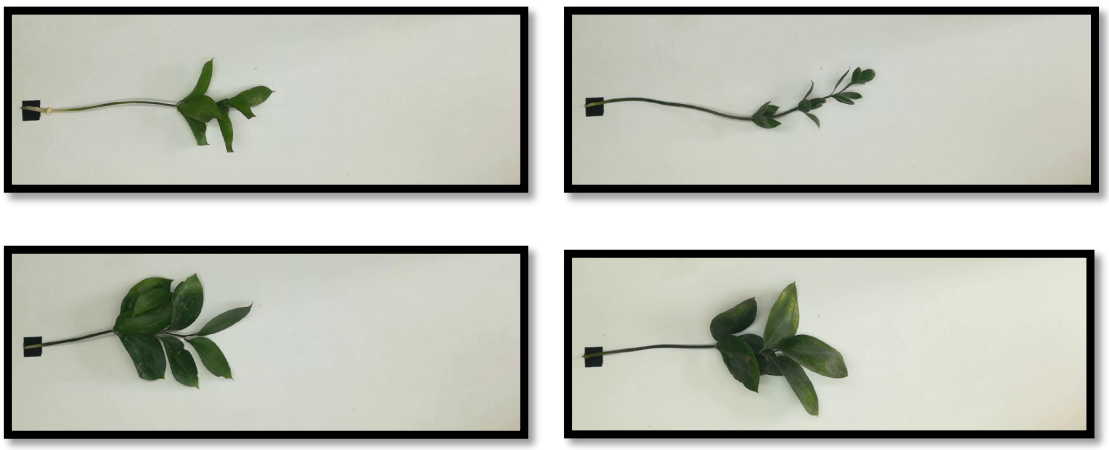
Respuesta. "Pues si es demoroso el tener que estar midiendo los tallos, o a veces llegan nuevas personas a la postcosecha y clasifican mal y toca volver a clasificar o toca estarles guiando a que lo hagan bien y eso nos quita tiempo, a veces hacemos jornadas de postcosecha de ocho horas seguidas o más".

Anexo 2. Muestra Fotográfica

Buenas:



Malas:



El dataset completo está disponible en el siguiente enlace: https://tinyurl.com/2qsfpekz

Anexo 3. Script de Medición

```
clear;clc;
%% Medición del tallo
I=imread('3.jpg');
imshow(I)
whos I;
Igray= rgb2gray(I); imshow(Igray)
Ibin= imcomplement(imbinarize(Igray));
imshow(Ibin)
cc = bwconncomp(Ibin)
%% Deteccion de Elementos
ruscus = false(size(Ibin));
referencia = false(size(Ibin));
inicio = false(size(Ibin));
numPixels = cellfun(@numel,cc.PixelIdxList);↙
[biggest,idx] = max(numPixels); BW(cc.↙
PixelIdxList{idx}) = 0;
C = size(cc.PixelIdxList);
C1=[0];
for k = 1:C(2)
    C1(k) = bwarea( cc.PixelIdxList{k});
end
C2=sort(C1);
R1=find(C1==C2(C(2)))
R2=find(C1==C2(C(2)-3))
R3=find(C1==C2(C(2)-4))
ruscus(cc.PixelIdxList{R1}) = true;
inicio(cc.PixelIdxList{R2}) = true;
referencia(cc.PixelIdxList{R3})=true;
imshow(ruscus+referencia)
hold on
```

```
%% Creación de recuadros de detección
ini=regionprops(inicio,'BoundingBox')
ref= regionprops(referencia,'BoundingBox')
med = regionprops(ruscus,'BoundingBox')
ang=regionprops(ruscus, 'Orientation')
%% Medición del tamaño
tam=((med.BoundingBox(3)+(med.BoundingBox(1)-↙
ini.BoundingBox(1)))*10.1/137)/cosd(ang.↙
Orientation)
rectangle('Position',[ini.BoundingBox(1),med.↙
BoundingBox(2),med.BoundingBox(3)+(med.↙
BoundingBox(1)-ini.BoundingBox(1)),med.↙
BoundingBox(4)],'EdgeColor','r','LineWidth',2 )
```

Anexo 4. Script de Entrenamiento y Ejecución

```
clear all;
clc;
%% 1. Se carga la data como un conjunto de↙
imagenes
imset = imageSet('TRAIN','recursive');
%% 2. Se caracteriza las imagenes mediante bag↙
of features
bag = bagOfFeatures(imset,'VocabularySize',↙
200,...
    'PointSelection','Detector');
%% Decodifica las imagenes en funcion de las↙
caracteristoicas extraidas anteriormente
imagefeatures = encode(bag,imset);
%% Crea una tabla usando las caracteristicas↙
decodificadas
RuscusData          = array2table(imagefeatures);
RuscusData.Estado   = getImageLabels(imset);
%% Se utiliza las caracteristicas para generar↙
un modelo apropiado
classificationLearner
%% Prueba del modelo entrado
Ruscus2FinderLive(trainedClassifier,bag)
```

Anexo 5. Función de tiempo real

```
function Ruscus2FinderLive(trainedClassifier,↙
bag)
[fig, ax1, ax2] = figureSetup↙
(trainedClassifier);
wcam = webcam;

while ishandle(fig)

    img = snapshot(wcam);
    grayimg = rgb2gray(img);
    imagefeatures = double(encode(bag,grayimg));
    [imagepred, probabilities] = predict↙
(trainedClassifier.ClassificationKNN,↙
imagefeatures);
    try
        imshow(insertText(img,[640,1],upper↙
(cellstr(imagepred)),...
            'AnchorPoint','RightTop','FontSize',↙
50,'BoxColor','Green',...
            'BoxOpacity',0.4),'Parent',ax1);
            ax2.Children.YData = probabilities;
            ax2.YLim = [0 1];
    catch err
    end
    drawnow
end

function cname = getClassifierName↙
(trainedClassifier)
cname = class (trainedClassifier.↙
ClassificationKNN);
if isa(trainedClassifier,'ClassificationECOC')
    cname = 'SVM';
end
pos = strfind(cname,'.');
if ~isempty(pos)
  cname = cname(pos(end)+1:end);
end
```

```
function [fig, ax1, ax2] = figureSetup↙
(trainedClassifier)
warning('off','images:imshow:↙
magnificationMustBeFitForDockedFigure')
set(0,'defaultfigurewindowstyle','docked')
fig = figure('Name','Car Finder↙
Go!','NumberTitle','off');
ax1 = subplot(2,1,1);
ax2 = subplot(2,1,2);
bar (ax2, zeros (1, numel (trainedClassifier.↙
ClassificationKNN.ClassNames)), 'FaceColor',↙
[0.2 0.6 0.8])
title(getClassifierName(trainedClassifier)),↙
ylabel('Probability')
set(0,'defaultfigurewindowstyle','normal')
```