



UNIVERSIDAD AGRARIA DEL ECUADOR
FACULTAD DE CIENCIAS AGRARIAS
DR. JACOBO BUCARÁM ORTIZ
CARRERA COMPUTACIÓN

**ESTUDIO COMPARATIVO DE ALGORITMOS DE
APRENDIZAJE AUTOMÁTICO UTILIZADOS PARA LA
IDENTIFICACIÓN DE UNA PLANTA ORNAMENTAL
PROPUESTA TECNOLÓGICA**

Trabajo de titulación presentado como requisito para la
obtención del título de
INGENIERO EN CIENCIAS DE LA COMPUTACIÓN

AUTORES
ALMEIDA FLORES ANTONIO LENIN
DURANGO DEVIS WILMER JAVIER

TUTOR
ING. BELTRAN ROBAYO NUVIA, M.Sc

MILAGRO – ECUADOR

2023



UNIVERSIDAD AGRARIA DEL ECUADOR
FACULTAD DE CIENCIAS AGRARIAS
DR. JACOBO BUCARÁM ORTIZ
CARRERA COMPUTACIÓN

APROBACIÓN DEL TUTOR

Yo, **BELTRÁN ROBAYO NUVIA ARACELLY**, docente de la Universidad Agraria del Ecuador, en mi calidad de Tutor, certifico que el presente trabajo de titulación: **ESTUDIO COMPARATIVO DE ALGORITMOS DE APRENDIZAJE AUTOMÁTICO UTILIZADOS PARA LA IDENTIFICACIÓN DE UNA PLANTA ORNAMENTAL** realizado por los estudiantes **ALMEIDA FLORES ANTONIO LENIN**; con cédula de identidad **N°0940934615** Y **DURANGO DEVIS WILMER JAVIER**; con cédula de identidad **N°0957227739** de la carrera **COMPUTACIÓN**, Unidad Académica Milagro, ha sido orientado y revisado durante su ejecución; y cumple con los requisitos técnicos exigidos por la Universidad Agraria del Ecuador; por lo tanto se aprueba la presentación del mismo.

Atentamente,

ING. BELTRAN ROBAYO NUVIA

Milagro, 10 de noviembre del 2023



UNIVERSIDAD AGRARIA DEL ECUADOR
FACULTAD DE CIENCIAS AGRARIAS
DR. JACOBO BUCARÁM ORTIZ
CARRERA COMPUTACIÓN

APROBACIÓN DEL TRIBUNAL DE SUSTENTACIÓN

Los abajo firmantes, docentes designados por el H. Consejo Directivo como miembros del Tribunal de Sustentación, aprobamos la defensa del trabajo de titulación: **“ESTUDIO COMPARATIVO DE ALGORITMOS DE APRENDIZAJE AUTOMÁTICO UTILIZADOS PARA LA IDENTIFICACIÓN DE UNA PLANTA ORNAMENTAL”**, realizado por los estudiantes **ALMEIDA FLORES ANTONIO LENIN Y DURANGO DEVIS WILMER JAVIER**, el mismo que cumple con los requisitos exigidos por la Universidad Agraria del Ecuador.

Atentamente,

ING. MARIO CÁRDENAS, M.Sc.
PRESIDENTE

ING. TERESA SAMANIEGO, M.Sc.
EXAMINADOR PRINCIPAL

ING. OSCAR BERMEO, M.Sc.
EXAMINADOR PRINCIPAL

Milagro, 10 de noviembre de 2023

Dedicatoria

Le dedico mi éxito este esfuerzo de vida a mis padres sr. Enrique Almeida y Sra. Magaly Flores por permitirme formar parte de sus vidas y continuar con mis estudios, de la misma forma a mis hermanos que de una u otra forma me dieron su ayuda y por estar siempre cuando se los necesitan a pesar de todo.

Almeida Flores Lenin

A mis padres por haberme inculcado los valores que tengo en la actualidad; muchos de mis logros y metas se los debo a ellos, así mismo a mi hermano, amigos y pareja que me ha apoyado en todo este proceso universitario, a toda mi familia que es lo mejor y más valioso que Dios me ha dado.

Durango Devis Wilmer

Agradecimiento

En primer lugar, agradezco a Dios por darme la vida y a mis padres por permitir disfrutarla en estos años que no han sido fáciles, agradezco a mis compañeros y amigos de la U por formar parte de este recorrido largo pero corto a la vez, siento que en la universidad no solo se puede llegar aprender cosas necesarias para la vida profesional sino también amistades que se llevan en el corazón por mucho tiempo más, para concluir agradezco a mi familia Almeida Flores; un agradecimiento a mis maestros que formaron parte de mis estudios. Gracias a todos.

Almeida Flores Lenin

Doy gracias a Dios por la vida que me dio y las cosas bonitas que ha puesto en mi camino universitario, agradezco a mis padres y hermanos por el apoyo incondicional lo largo de este proceso, a los docentes y tutor de tesis por los conocimientos que me han brindado para poder realizar este proyecto; también doy gracias a mis amigos que han sido incondicionales en mi vida.

Durango Devis Wilmer

Autorización de Autoría Intelectual

Nosotros **ALMEIDA FLORES ANTONIO LENIN** y **DURANGO DEVIS WILMER JAVIER**, en calidad de autores del proyecto realizado, sobre “**ESTUDIO COMPARATIVO DE ALGORITMOS DE APRENDIZAJE AUTOMÁTICO UTILIZADOS PARA LA IDENTIFICACIÓN DE UNA PLANTA ORNAMENTAL**” para optar el título de **INGENIERO EN CIENCIAS DE LA COMPUTACIÓN** por la presente autorizo a la **UNIVERSIDAD AGRARIA DEL ECUADOR**, hacer uso de todos los contenidos que me pertenecen o parte de los que contienen esta obra, con fines estrictamente académicos o de investigación.

Los derechos que como autores nos corresponden, con excepción de la presente autorización, seguirán vigentes a mi favor, de conformidad con lo establecido en los artículos 5, 6, 8; 19 y demás pertinentes de la Ley de Propiedad Intelectual y su Reglamento.

Milagro, 10 de noviembre de 2023

ALMEIDA FLORES ANTONIO LENIN

C.I. 0940934615

DURANGO DEVIS WILMER JAVIER

C.I. 0957227739

Índice general

APROBACIÓN DEL TUTOR	2
APROBACIÓN DEL TRIBUNAL DE SUSTENTACIÓN	3
Dedicatoria.....	3
Agradecimiento.....	5
Autorización de Autoría Intelectual	6
Índice general	7
Índice de tablas	13
Índice de figuras.....	14
Resumen.....	18
Abstract.....	19
1. Introducción.....	20
1.1 Antecedentes del problema.....	21
1.2 Planteamiento y formulación del problema	22
1.2.1 Planteamiento del problema.....	22
1.2.2 Formulación del problema.....	23
1.3 Justificación de la investigación	23
1.4 Delimitación de la investigación	24
1.5 Objetivo general	25
1.6 Objetivos específicos.....	25
1.7 Hipótesis.....	25
2. Marco teórico.....	26
2.1 Estado del arte.....	26
2.2 Bases teóricas.....	31
2.2.1 Vivero	31

2.2.2 Plantas.....	32
2.2.2.1 Ixoras.....	33
2.2.2.2 Girasol.....	33
2.2.2.3 Palmeras.....	34
2.2.3 Visión por computadora	34
2.2.4 Google Colab	35
2.2.5 Algoritmos de aprendizaje automático.....	36
2.2.5.1 Redes neuronales convolucionales.....	36
2.2.5.2 Algoritmo K-medias	39
2.2.5.3 Algoritmo KNN (K-nearest neighbor).....	41
2.2.6 Lenguaje de programación Python.....	43
2.2.7 Librerías	43
2.2.7.1 TensorFlow	43
2.2.7.2 OpenCV	44
2.2.7.3 Keras	44
2.2.7.4 El Sklearn	45
2.3 Marco legal.....	45
2.3.1 Constitución de la República del Ecuador.....	46
2.3.2 Ley de Propiedad Intelectual.....	46
2.3.3 Modelo de software libre	47
3. Materiales y métodos	48
3.1 Enfoque de la investigación	48
3.1.1 Enfoque Mixto.....	48
3.1.2 Tipo de investigación.....	49
3.1.2.1 Investigación descriptiva.....	49

3.1.2.2	Investigación aplicada	49
3.1.2.3	Investigación documental	49
3.1.3	Diseño de investigación	50
3.1.3.1	Investigación Experimental	50
3.2	Metodología.....	51
3.2.1	Variables de la investigación	51
3.2.1.1	Variable independiente	51
3.2.1.2	Variable Dependiente	51
3.2.1.2.1	Precisión	51
3.2.1.2.2	Exactitud	52
3.2.1.2.3	Sensibilidad	52
3.2.1.2.4	F1	52
3.2.1.2.5	Tiempo utilizado para la detección	53
3.2.2	Fases de la investigación	53
3.2.2.1	Metodología Propuesta.....	53
3.2.2.1.1	Etapas de recopilación de información de datos de entrada	53
3.2.2.1.2	Etapas de procesamiento y transformación de los datos al modelo espacio vectorial	54
3.2.2.1.3 Etapas de selección e implementación de los algoritmos de aprendizaje automático	54
3.2.2.1.4 Etapas de evaluación y comparación de los resultados obtenidos	55
3.2.3	Recolección de datos.....	55
3.2.4.1	Recursos	55
3.2.3.1.1	Recursos Humanos.....	55

3.2.3.1.2	Recursos Materiales.....	55
3.2.3.1.3	Recursos tecnológicos	56
3.2.3.1.4	Recursos bibliográficos.....	56
3.2.3.1.5	Presupuesto del proyecto	56
3.2.4.2.	Métodos y técnicas	56
3.2.3.1.6	Método analítico	56
3.2.4	Análisis estadístico	57
4.	Resultados.....	58
4.1.	Análisis del funcionamiento de los algoritmos de aprendizaje automático que permite la identificación de imágenes mediante revisión bibliográfica para el establecimiento de parámetros que evaluarán para su comparación.....	58
4.1.1.	Observación.....	58
4.1.2.	Entrevista	59
4.2.	Programación de los algoritmos de aprendizaje automático mediante la utilización de Python y librerías para la obtención de criterios relacionados a estos.....	59
4.3.	Establecimiento del conjunto de datos a utilizar mediante la captura de imágenes para el procesamiento y entrenamiento de los algoritmos de aprendizaje automático.....	60
4.3.1.	Procesamiento de imágenes	61
4.4.	Evaluación de los criterios de exactitud, sensibilidad, precisión y F1 a través de la ejecución de los algoritmos para la comparación y establecimiento del algoritmo más óptimo que identifica plantas ornamentales.....	62

5. Discusión.....	63
6. Conclusión.....	68
7. Recomendación.....	70
8. Bibliografía.....	71
9. Anexos	86
9.1 Anexo 1. Presupuesto del proyecto.....	86
9.2 Anexo 2. Ficha de observación.....	87
9.3 Anexo 3. Entrevista	89
9.4 Anexo 4. Metodología Propuesta.....	92
9.5 Anexo 5. Etapas del Procesamiento de imágenes	92
9.6 Anexo 6. Funcionamiento de los algoritmos K medias, CNN y KNN	93
9.7 Anexo 7. Autores que han usado algoritmos de aprendizaje automático.	94
9.8 Anexo 8. Codificación del algoritmo CNN.....	98
9.9 Anexo 9. Matriz de confusión del algoritmo CNN.....	103
9.10 Anexo 10. Métricas de estudio del algoritmo CNN	103
9.11 Anexo 11. Codificación del algoritmo KNN.....	104
9.12 Anexo 12. Matriz de confusión del algoritmo de KNN.....	109
9.13 Anexo 13. Métricas de estudio del algoritmo KNN	109
9.14 Anexo 14. Codificación del algoritmo K-Means.....	110
9.15 Anexo 15. Matriz de confusión del algoritmo K-Means.....	115
9.16 Anexo 16. Métricas de estudio del algoritmo K-Medias.....	116
9.17 Anexo 17. Grafica comparativa de las métricas calculadas	117
9.18 Anexo 18. Cuadro comparativo.....	117
9.19 Anexo 19. Recursos de las herramientas utilizadas	128

9.20 Anexo 20. Establecimiento de conjunto de imágenes	128
9.21 Anexo 21. Manual Técnico.....	131

Índice de tablas

Tabla 1. Costo de los materiales	86
Tabla 2. Ficha de observación de los procesos de identificación de las plantas..	87
Tabla 3. Diferentes Autores que implementaron algoritmos de aprendizaje automático.....	95
Tabla 4. Comparativa de algoritmos de aprendizaje automático.....	118
Tabla 5. Herramientas y los recursos usados para la ejecución y realización de los algoritmos.....	128

Índice de figuras

Figura 1. Metodología propuesta para desarrollar los algoritmos.....	92
Figura 2. Etapas fundamentales del procesamiento de imágenes	92
Figura 3. Funcionamiento del algoritmo K means.	93
Figura 4. Arquitectura de las fases de una red neuronal convolucional.	93
Figura 5. K ejemplos del conjunto de vecinos más cercanos en el algoritmo KNN.....	94
Figura 6. Conexión con Google Drive para que los archivos alojados en la nube puedan ser usados y permitan la ejecución del algoritmo.....	98
Figura 7. Extracción del archivo data.zip donde se encuentran alojadas las imágenes de los tres tipos de plantas para su identificación y debido entrenamiento usando YOLOV5.	99
Figura 8. Carga de modelo Pre-entrenado de YOLOV5, importación y ejecución de las librerías para calcular y presentar la matriz de confusión.....	99
Figura 9. Mostrar la matriz de confusión la cual se guardará la imagen para poder ser descargada.....	100
Figura 10. Importar las librerías numpy, sklearn y matplotlib para calcular las métricas de medición del algoritmo y poderlas presentar mediante un gráfico. .	100
Figura 11. Impresión de los resultados de los parámetros calculados.....	101
Figura 12. Crear grafica de barras de los parámetros calculados y que permita guardar la imagen para su descarga.....	101
Figura 13. Código para la detección de la imagen.	102
Figura 14. Código para la captura de la imagen y la clasificación sobre el tipo de planta mostrada.....	102
Figura 15. Matriz de confusión del algoritmo CNN	103

Figura 16. Métricas de medición algoritmo CNN	103
Figura 17. Instalación de librerías para el funcionamiento del algoritmo KNN. ...	104
Figura 18. Carga y extracción del dataset de imágenes de las plantas para el entrenamiento del algoritmo.....	104
Figura 19. . Ejecución y entrenamiento del algoritmo KNN por medio de las imágenes extraídas del dataset y presentar las gráficas de la matriz de confusión y métricas calculadas por medio de la librería llamada sklearn.	105
Figura 20. Validación y clasificación de las imágenes obtenidas del dataset para el entrenamiento del algoritmo obtenidas a través de etiquetas a partir del nombre de la planta.....	106
Figura 21. Presentación de grafica de la matriz de confusión y las métricas evaluadas del algoritmo las cuales son la exactitud, sensibilidad, precisión y F1.....	107
Figura 22. Captura de la imagen para convertirla en formato OpenCV y pueda ser identificada por el algoritmo.....	108
Figura 23. Presentar el tipo de planta que está identificando por medio de la imagen tomada.	108
Figura 24. Matriz de confusión del algoritmo de KNN	109
Figura 25. Métricas de medición algoritmo KNN.	109
Figura 26. Instalación de librerías para el funcionamiento del algoritmo K- Means.	110
Figura 27. Carga y extracción del dataset de imágenes de las plantas para el entrenamiento del algoritmo.....	110

Figura 28. Ejecución y entrenamiento del algoritmo K-Means por medio de las imágenes extraídas del dataset y presentar las gráficas de la matriz de confusión y métricas calculadas por medio de la librería llamada sklearn.	111
Figura 29. Clasificación de las imágenes, validación, y presentación de grafica de la matriz de confusión y las métricas evaluadas del algoritmo las cuales son la exactitud, sensibilidad, precisión y F1.	112
Figura 30. Presentación de grafica de la matriz de confusión y las métricas evaluadas del algoritmo las cuales son la exactitud, sensibilidad, precisión y F1.	113
Figura 31. Importar las librerías necesarias que permitan abrir la cámara y la captura de la imagen a través de la cámara del navegador.	113
Figura 32. Cambiar la imagen a escalas de grises y aplicar la detección por bordes para la identificación de la planta.	114
Figura 33. Presentar el tipo de planta que está identificando por medio de la imagen capturada, tomando en cuenta las características de la imagen como su color y su forma.	115
Figura 34. Matriz de confusión del algoritmo de K-means	115
Figura 35. Métricas de medición algoritmo K-Means	116
Figura 36. Datos por cada Cluster algoritmo Kmeans.	116
Figura 37. Gráfico de barras comparativo de las diferentes métricas calculadas de los algoritmos KNN, K-Medias y CNN.	117
Figura 38. almacenamiento de conjunto de imágenes para el entrenamiento de los algoritmos.	128
Figura 39. Carpetas de plantas.	129
Figura 40. Imágenes recabadas de Ixoras.	129

Figura 41. Imágenes recabadas de Girasoles.....	130
Figura 42. Imágenes recabadas de Palmas.....	130
Figura 43. Página oficial de Python.....	133
Figura 44. Plataforma de Google Colab.....	133

Resumen

En la actualidad la visión por computadora se está utilizando para hacer reconocimiento de objetos como así también en el campo agrícola, por ende en este proyecto se realizó el estudio comparativo con algoritmos de aprendizaje automático en base a la identificación de plantas ornamentales teniendo en cuenta a las plantas más populares en los viveros de la ciudad de Milagro que son la ixoras, girasoles y palmeras, con el propósito de tener las distintas características de estos algoritmos midiendo sus diferentes parámetros como: Precisión, exactitud, sensibilidad, F1 (medida de precisión y exactitud) y tiempo de identificación; de forma que proporcione información necesaria para futuros proyectos a desarrollarse, el objetivo del trabajo es analizar los algoritmos de aprendizaje automático que identifican plantas ornamentales usando una metodología enfocada en los procesos de análisis de algoritmos con distintas etapas como: Etapas de recopilación de información de datos de entrada como la obtención de imágenes para obtener un dataset; procesamiento y transformación de datos en la cual se procesan las imágenes para su estudio extrayendo características únicas para su identificación; selección e implementación de los algoritmos de aprendizaje automático se realizó la selección de los algoritmos con los que se trabajó como los de redes neuronales convolucionales, K-nearest neighbors (KNN) y k-medias; evaluación y comparativa de los resultados obtenidos, etapa en la cual con los resultados obtenidos se pudo evaluar en base a distintos parámetros y decidir el algoritmo más óptimo.

Palabras clave: Algoritmo de aprendizaje automático, Redes Neuronales convolucionales (CNN), K-nearest neighbors (KNN), Python, Visión por computadora.

Abstract

Currently computer vision is being used to make object recognition as well as in the agricultural field, therefore in this project a comparative study was conducted with machine learning algorithms based on the identification of ornamental plants taking into account the most popular plants in the nurseries of the city of Milagro which are the ixoras, sunflowers and palm trees, in order to have the different characteristics of these algorithms by measuring their different parameters such as: Precision, accuracy, sensitivity, F1 (measurement of precision and accuracy) and identification time; in a way that provides necessary information for future projects to be developed, the objective of the work is to analyze machine learning algorithms that identify ornamental plants using a methodology focused on algorithm analysis processes with different stages such as: Input data information gathering stages such as obtaining images to obtain a dataset; data processing and transformation in which the images are processed for study extracting unique features for identification; selection and implementation of the machine learning algorithms, the selection of the algorithms with which we worked such as convolutional neural networks, K-nearest neighbors (KNN) and k-means; evaluation and comparison of the results obtained, a stage in which the results obtained were used to evaluate based on different parameters and decide the most optimal algorithm.

Keywords: machine learning algorithm, Convolutional Neural Networks (CNN), K-nearest neighbors (KNN), Python, Computer Vision.

1. Introducción

En la actualidad la tecnología avanza a pasos agigantados, gracias a su utilidad se puede desarrollar, implementar y realizar análisis de cualquier tipo de estudio, en este caso con el desarrollo continuo de nuevas tecnologías aplicando inteligencia artificial se puede trabajar en distintas áreas, de tal forma que en la realización del presente proyecto se realizó el estudio comparativo de algoritmos de aprendizaje automáticos donde a través de criterios como la exactitud, sensibilidad, precisión, tiempo utilizado para la detección y F1 se obtuvo resultados de eficiencia y eficacia de los mismos.

Considerando que los viveros ubicados en la ciudad de Milagro, realizan el cultivo de plantas ornamentales dedicándose a la comercialización del mismo; las plantas tienen distintos usos que va desde la decoración hasta la parte medicinal; por otra parte los algoritmos de aprendizaje automático para la identificación son difíciles reconocer a simple vista, saber cuál de ellos tiene mayor efectividad al momento de ser aplicados en función para identificar las distintas plantas; es por ello que en este proyecto se realizó el análisis de distintos algoritmos como el de redes neuronales convolucionales, KNN y k-medias donde a través de estos se pudo identificar las plantas ornamentales con la finalidad de aportar información necesaria para conocer la eficiencia y eficacia al momento de su aplicación donde se requiera este tipo de algoritmos.

De tal forma para realizar el estudio comparativo de los distintos algoritmos se utilizó el lenguaje de programación Python para el desarrollo de la codificación y las distintas librerías como TensorFlow, OpenCV, Keras entre otras, de las cuales se obtiene procesos como la adquisición de imágenes, procesamiento y segmentación, extracción de características, selección de características y clasificación para

realizar el análisis de las imágenes, además sirvió para la creación del dataset o conjunto de imágenes de plantas de los datos recabados del Vivero Belleza Natural para el entrenamiento de los algoritmos que se estudian.

1.1 Antecedentes del problema

Existe una gran cantidad de datos presente hoy en día alrededor del mundo en la cual el ser humano no es capaz de procesar toda esta información de manera manual es por ello que se busca el uso de herramientas que permitan el procesamiento de estas. Como expresa Zamorano (2018) sobre el uso de herramientas que permitan el almacenamiento, procesamiento y extracción de información es de suma importancia para obtener beneficios de un objetivo en específico, por tanto, estas herramientas son pocas utilizadas por las personas provocando que se tenga poco conocimiento al momento de obtener información útil para realizar un proyecto en específico, así mismo al querer obtener conocimiento sobre cualquier tema, la inteligencia artificial en la rama de Machine Learning o aprendizaje automático es importante para tomar decisiones con datos futuros proporcionados, también permite a las máquinas aprender con un entrenamiento adecuado. Es por ello, que se realizó el estudio de distintos algoritmos que ayuden a la identificación de la planta mediante el aprendizaje automático, a través de recopilación y procesamiento de imágenes.

Asimismo, el uso de algoritmos de aprendizaje automático para la identificación de plantas ornamentales es importante para el desarrollo de proyectos futuros. Aguilar y Campoverde (2020) explican que mediante el uso de redes neuronales convolucionales se puede realizar la clasificación de diferentes plantas realizando el entrenamiento respectivo de la aplicación mediante la detección o categorización de imágenes, este método ayudaría a optimizar el tiempo en reconocimiento de las

plantas o lo que se desea clasificar mediante estas. De manera que, se usó como uno de los algoritmos de estudios el de redes neuronales convolucionales para tener conocimiento si es uno de los más óptimos al momento de implementarlo en la identificación de las plantas ornamentales.

Además, los distintos algoritmos de aprendizaje automáticos utilizados en el presente proyecto, son usados para realizar el respectivo estudio comparativo de cada uno de ellos en el cual se toma en cuenta diferentes aspectos y parámetros a considerar las cuales son: el tiempo de identificación de la planta, exactitud, sensibilidad, precisión y F1, se realizó el análisis de estas características para conocer sobre que algoritmo es el más óptimo a usar en futuros proyectos.

1.2 Planteamiento y formulación del problema

A continuación, se describe el planteamiento y la formulación del problema con respecto a la realización del proyecto.

1.2.1 Planteamiento del problema

En la actualidad el estudio de diferentes algoritmos de aprendizaje automático para el conocimiento de cuál de ellos puede ser el más óptimo para la identificación de plantas ornamentales es poco usado al momento de crear un aplicativo o un sistema. El poco manejo de información sobre el grado de efectividad de cada algoritmo es uno de los problemas más comunes al momento de elegir o saber cuál es el más adecuado para utilizarlo en el área que se lo requiera para su implementación.

En consecuencia, mucho de estos algoritmos no son bien utilizados o aplicados de acuerdo a sus características para obtener su óptimo funcionamiento, de tal forma el estudio comparativo de algoritmos de aprendizaje automático realizado beneficia a futuras aplicaciones del mismo.

1.2.2 Formulación del problema

De igual manera para los desarrolladores de aplicaciones o sistemas, uno de los problemas comunes que se les presenta es el desconocimiento de ciertas características sobre que algoritmo sería el más óptimo para identificar a las plantas. Teniendo en consideración que los algoritmos de aprendizaje automático han tenido gran impacto hoy en día se busca obtener el grado de efectividad de los distintos algoritmos en base a sus funcionalidades; del mismo modo se han podido crear grandes inventos como los autos autónomos y los sistemas de detección de rostros entre otros gracias a la aplicación de estos algoritmos.

En base a lo mencionado anteriormente se realizó el estudio de tres algoritmos que ayuda a la identificación de plantas ornamentales mediante el uso de distintos algoritmos. De esta manera se formula la siguiente pregunta:

¿Cuál de los algoritmos de aprendizaje automático puede realizar la identificación de una planta ornamental de forma exacta, precisa en el menor tiempo y con mayor rendimiento?

1.3 Justificación de la investigación

El uso de algoritmos de aprendizaje automáticos para la identificación de plantas ornamentales conociendo sus distintos parámetros en base a su eficiencia y eficacia es una parte importante ya que ayuda a la hora de su aplicación o utilización en algún proyecto en el que se pueda aplicar algunos de ellos sabiendo sus capacidades respectivas, puesto que al conocer los parámetros respectivos de los distintos algoritmos de aprendizaje automáticos y haber realizado un estudio comparativo previo se tomaría una mejor decisión sobre cuál de ellos usar en su implementación.

Por eso se realizó el análisis de tres algoritmos que ayudan a la identificación de una planta ornamental, a su vez se toma en consideración tres plantas para la prueba de la propuesta tecnológica las cuales serán: girasol, Ixora y palmera, escogidas previamente del vivero “Belleza Natural” donde cultivan grandes cantidades de este tipo flora, de la cual se obtuvo una dataset con 1350 imágenes de las plantas antes mencionadas de forma que 1200 fueron tomadas para el entrenamiento de los algoritmos y 150 como validación, para la codificación de los algoritmos de aprendizaje automático para la identificación de plantas se usó el lenguaje de programación Python mediante el uso de sus librerías como tensorflow, OpenCV y Keras, que son las más utilizadas para aplicar clasificación y análisis de las imágenes; posterior a ello para la realización del entrenamiento se usó Google colab con el repositorio de la imágenes (dataset) subido en el drive la cual funciona como la base de datos. De tal forma que este proyecto ayude tanto a los estudiantes como desarrolladores que apliquen inteligencia artificial en específico algoritmos de aprendizaje automático para sus proyectos; por lo cual se realizó este estudio comparativo donde se codificarán los siguientes algoritmos:

Algoritmo de redes neuronales convolucionales

Algoritmo de KNN (K-nearest neighbor)

Algoritmo K-medias

1.4 Delimitación de la investigación

- **Espacio:** El proyecto se lo desarrolló con las imágenes tomadas del Vivero “Belleza Natural” de la ciudad de Milagro.
- **Tiempo:** El proceso que llevará a cabo el estudio de los distintos algoritmos y del proyecto tendrá una duración de 9 meses desde mayo del 2022 hasta enero del 2023.

- **Población:** Conjunto de imágenes de ixora (*Ixora coccinea*), girasol (*Helianthus annuus*) y palmera washingtonia (*Washingtonia robusta*).

1.5 Objetivo general

Comparar algoritmos de aprendizaje automático programándolos en Python para la verificación de su nivel de precisión, exactitud, tiempo, sensibilidad y F1 utilizado para la detección, en la identificación de una planta ornamental.

1.6 Objetivos específicos

- Analizar el funcionamiento de los algoritmos de aprendizaje automático que permite la identificación de imágenes mediante revisión bibliográfica para el establecimiento de parámetros que evaluarán para su comparación.
- Programar los algoritmos de aprendizaje automático mediante la utilización de Python y librerías para la obtención de criterios relacionados a estos.
- Establecer el conjunto de datos a utilizar mediante la captura de imágenes para el procesamiento y entrenamiento de los algoritmos de aprendizaje automático.
- Evaluar los criterios de exactitud, sensibilidad, precisión y F1 a través de la ejecución de los algoritmos para la comparación y establecimiento del algoritmo más óptimo que identifica plantas ornamentales.

1.7 Hipótesis

Al menos uno de los algoritmos de aprendizaje automático permitió la identificación de una planta ornamental de forma exacta, precisa, en el menor tiempo y con mayor rendimiento.

2. Marco teórico

2.1 Estado del arte

Emplear la visión por computadora para extraer información útil de imágenes y videos se está convirtiendo en una técnica clave para identificar cambios en las plantas. Mochida et al. (2018) explican que mediante el uso de visión por computadora para el mejoramiento de productividad de las plantas tuvieron que aplicar varios pasos que son el pre procesamiento, segmentación, extracción de características y la clasificación; el reprocesamiento es un paso preliminar del análisis de imágenes que tiene como objetivo organizar las propiedades de los datos para facilitar los pasos posteriores e incluso obtener resultados finales razonables.

En particular, cuando es apuntado a imágenes adquiridas en condiciones de campo, a diferencia de los entornos controlados, el pre procesamiento de imágenes contribuye a mejorar la calidad del procesamiento, la segmentación representa al primer paso importante para la extracción de información de objetos de datos de imágenes pre procesadas separando un conjunto de píxeles que incluyen objetos de interés en las imágenes que permiten la identificación y cuantificación automática de áreas correspondientes a órganos particulares en las plantas.

La extracción de características es un paso para crear un conjunto de características significativas e información no redundante que puede representar suficientemente las imágenes y en el método de clasificación se toma en cuenta dos aplicaciones principales: la clasificación taxonómica y clasificación de estados fisiológicos de plantas; la clasificación taxonómica basada en visión por computadora juega un papel esencial en el fenotipado de plantas para distinguir automáticamente las especies objetivo para el fenotipado de otras y la clasificación

de los estados fisiológicos de las plantas es la que permite obtener la descripción de estas, fisiologías, estados y calidades de las plantas, es por ello que la clasificación de plantas basado en la visión por computadora ha mostrado avances notables en el reconocimiento de plantas y la taxonomía de estas, para llegar al resultado de esta investigación se tuvo que usar varias herramientas de software que ayudaron al análisis de imágenes en el fenotipado de plantas, ayudando a facilitar la comunicación entre usuarios y desarrolladores, así mismo se usó un conjunto de datos tomados desde varias plataformas una de ellas fue Kaggle.

Concluyendo que el uso de estos métodos ha tenido un crecimiento significativo como un área multidisciplinaria que integra el conocimiento de la ciencia de las plantas, igualmente el fenotipado de plantas basadas en la visión ha mostrado avances notables en el reconocimiento de plantas y la clasificación taxonómica. Además, estos modelos ayudan al mejoramiento adicional de la precisión de reconocimiento en más tareas desafiantes, como la segmentación multietiqueta de múltiples órganos y especies en ambientes naturales.

En la actualidad se utilizan aplicativos para la identificación de distintas plantas usando algoritmos de aprendizaje automático que a través de estos se pueden estudiar e identificar objetos. Muñoz y Bolt (2021) realizaron un estudio para determinar las especies de plantas nativas en Chile por medio de redes neuronales convolucionales que les permitió la identificación de la flora a través de un aplicativo móvil, para ello trabajaron con recolección de datos o dataset, esto les permitió el almacenamiento de datos de distintas especies endémicas, nativas y exóticas, a través de un repositorio de imágenes obtenidas mediante una investigación profunda por plataformas web, donde utilizaron 100 imágenes por especie, teniendo como resultado 46 especies a través de entrenamientos y pruebas con un

aproximado de 95% de acierto, concluyendo que la aplicación tiene una alta probabilidad de éxito. De manera que se hizo uso de distintos algoritmos de aprendizaje automático basados en identificación de plantas ornamentales para la obtención de la eficiencia de los mismos.

El uso de la inteligencia artificial para la detección de la flora en se ha vuelto de suma importancia para la agilización de diferentes procesos. Cofre (2020) señala que el uso de distintas tecnologías que ayuden a la identificación de plantas nativas es importante ya que ayudaría a romper las limitantes que hay con las personas al momento de sus respectivas utilidades, es por ello que realizó un proyecto en la empresa Agrocotopaxi, que mediante el uso de la inteligencia artificial y técnicas de clasificación desarrollaron una aplicación móvil y un sistema web de las diferentes plantas nativas que se encontraban en la mini empresa, con el fin de dar a conocer los diferentes beneficios de las plantas.

Para la realización de este proyecto se usaron dos metodologías, la metodología SCRUM que fue usada para el desarrollo de la aplicación móvil y la metodología CRISP-DM para creación del sitio web mediante el software Django, Python, TensorFlow y Xamarin. El uso de todas estas herramientas permitió la creación de una aplicación móvil que se pudo adaptar en varias plataformas en la cual la predicción de la clasificación de las plantas nativas dio un resultado superior al 80%, esto fue posible gracias al uso de los algoritmos de redes neuronales convolucionales, en la cual tuvo como conclusión que el uso de la inteligencia artificial para realizar predicciones es óptima para los procesos de clasificación y que puede ser adaptada a varias tecnologías propuestas. Por esta razón se tomó en cuenta el algoritmo de redes neuronales convolucionales para su respectivo

estudio y comparación con los diferentes algoritmos planteados en la identificación de una planta ornamental.

De igual manera, para realizar un análisis de algoritmos se deben evaluar características relacionadas a su tiempo y espacio de ejecución, además de su correcta funcionalidad y desempeño, debido a que se evaluarán tres algoritmos de aprendizaje automático se cita el estudio realizado por Zamorano (2018), donde realizó un análisis comparativo con la finalidad de entender sus funciones, definir los criterios para hacer la comparación, mismos que comprenden características como: recursos del sistema, es decir todo lo relacionado a la memoria usada, tiempo de ajuste y tiempo necesario para ejecutar su función, también destaca la importancia de evaluar los valores de cada algoritmo teniendo en cuenta su exactitud, sensibilidad y precisión; para llevar a cabo este estudio, los parámetros de evaluación fueron elegidos gracias a métodos de validación de Scikit-Learn una biblioteca del lenguaje de programación Python, de esa forma se obtuvieron resultados con datos estandarizados y atendiendo a las métricas antes mencionadas que el mejor algoritmo fue KNeighborsClassifier teniendo como valores acertados un 96,78% de exactitud; un bajo error en probabilidades de 0,13 de log los y un bajo tiempo de ajuste de 5,35 s. Por lo tanto, mediante el presente proyecto se busca realizar un análisis similar siguiendo con las fases de obtención de adquisición, procesamiento, segmentación, extracción de características y reconocimiento e interpretación de los resultados, además de determinar el uso de almacenamiento y tiempo de cómputo para conocer el desempeño de cada ejecución.

Asimismo, las redes neuronales aplicadas en viveros ornamentales o en cualquier tipo de plantas existentes es de suma importancia ya que ayuda a la

identificación sea de la planta mismo o de sus diversas enfermedades que pueden atacarla. Vázquez, Ibarra, Flores y Moreno (2018) indican que la agricultura protegida ha tenido cada vez más acogida en México de acuerdo con la SAGARPA, en los últimos 15 años este tipo de agricultura ha tenido un crecimiento de 22, 461 hectáreas, para esto se realizaron 2 métodos importantes que son el Micro invernadero y la metodología de experimento; el micro invernadero es un lugar equipado con paredes de nylon transparente permitiendo monitorear el control de las diversas plantas cultivadas en dicho lugar, así mismo cuenta con sistema de sensores para el monitoreo de temperatura y humedad producido en el invernadero, este lugar cuenta con sistema de riego automatizado que permitirá el riego automático de las plantas; el siguiente método es el de metodología de experimento en la cual consiste en la división de 3 grupo de plantas para el estudio de sus características, cada planta contaba con un sensor de temperatura y humedad permitiendo que en cada planta se obtenga un estudio diferente sea cómo se comporta la planta o el nivel de agua que necesite cada una, este estudio permitió aplicar el aprendizaje mediante redes neuronales que le permitía diferenciar si las plantas estudiadas estaban en buen o mal estado, como resultado se logró implementar exitosamente un sistema de riego que permitiera un desarrollo pleno de las plantas criadas en el invernadero. Por lo que se decidió realizar un estudio comparativo de algoritmos de aprendizaje automáticos para la detección de una planta ornamental.

Es más, Ecuador es uno de los países con mayor cantidad de exportación de flores internacionalmente, tanto que se ha vuelto un sustento necesario para la economía del país. Sango, Chanaluisa y Llano (2021) explican que las ventas de flores en el año 2019 se ha registrado un crecimiento de 4.7% con respecto al año

2018 con un total de 874 millones de dólares, pero, así como existen ganancias también existen pérdidas, esto es debido a diversas plagas que dañan a las plantas y provocan el marchitamiento de las mismas, por lo cual implementaron redes neuronales en conjunto con Deep Learning para la identificación y control de plagas en las plantas cultivadas para la exportación, utilizando encuestas y entrevista que permitieron saber sobre el cuidado de las plantas y las diferentes plagas que afectan al cultivo para así poder combatir de manera rápida y que haya un bajo nivel de pérdida sea de las plantas o nivel económico, para el desarrollo del sistema se usó la metodología Mobile-D la cual permitió la creación de un sistema entendible, que sea eficaz y eficiente para el usuario, como resultado a base de imágenes captadas de las distintas flores se pudo identificar las diferentes plagas que dañaban los cultivos, esto ayudó a que se puedan controlar de manera rápida. En el presente proyecto se busca el estudio comparativo de diferentes algoritmos de aprendizaje automáticos para la detección de plantas ornamentales, para así poder identificar que algoritmo es el más eficaz y eficiente de los que se están realizando el estudio.

2.2 Bases teóricas

A continuación, se encontrarán los elementos a utilizar en el desarrollo mediante la presente propuesta tecnológica orientada a inteligencia artificial, así como los conceptos más importantes de nuestro objeto de estudio que es la identificación de una planta ornamental.

2.2.1 Vivero

A lo largo del tiempo se han cultivado plantas teniendo en cuenta su proceso en el correcto cuidado. Demaestri y Viale (2018) mencionan que los viveros producen plantas en envases de distinto tipo y tamaño donde para producir una planta de

calidad se debe reunir las siguientes condiciones, las cuales son: tener un desarrollo vigoroso, lo cual se puede observar por el grosor de las hojas y el color oscuro del follaje, abundante cabellera radical, que la raíz no este torcida, buena sanidad, una edad no mayor a dos años, a menor edad mayor posibilidad de adaptación entre otras, entonces puede decir que ha producido una planta fuerte y sana; por otra parte indican los principales tipos de suelo: arcilloso, arenoso, limoso y margoso o marga; donde dependiendo de su naturaleza química como la acidez, la alcalinidad o las sales minerales ayudara al desarrollo de la planta, ya que de este depende el drenaje y la fertilización. De manera que para el correcto proceso de plantación se debe tomar en cuenta la calidad y el tipo de suelo para tener una adaptación y crecimiento exitoso de la planta.

2.2.2 Plantas

Las plantas son seres vivientes creadas por la madre naturaleza con el objetivo de embellecer el ecosistema, tener un hogar significa que se podría plantar un jardín. Rubio y Fernández (2018) afirman que las plantas están presentes en la cotidianidad y resulta difícil ver una casa sin jardín o alguna planta, ya que estas armonizan el interior o exterior del hogar, pero para mantenerlas en perfecto estado necesitan información sobre los cuidados y donde plantarla, de forma que tenga un ambiente idóneo para desarrollarse de forma sana, es más las plantas son una fuente importante y siempre estarán presente para mantener el ecosistema en equilibrio y en constante crecimiento.

Las plantas ornamentales o también llamadas de ornato son una familia de plantas en concreto que se cultivan con el propósito de embellecer o decorar un lugar. Álvaro y Díaz (2020) declaran que en la actualidad las plantas ornamentales no son usadas solo para la decoración de jardines, calles, casas o balcones, sino

que ayudan también a mejorar el medio ambiental ya que ayudan a reducir el CO₂ de las ciudades y a tener un aire más limpio, este tipo de planta también ayuda al ser humano a reducir el estrés, aumentar la concentración y generar bienestar.

2.2.2.1 Ixoras

Las Ixoras, proveniente del nombre de una diosa de la india a la cual le ofrecen sus flores, es una de las plantas pequeñas más comunes en todo el entorno. Reyes-Pérez et al. (2018) Mencionan que estas son utilizadas para decorar jardines y parques, teniendo en cuenta que florecen en todo el año siendo de mayor demanda por todo el mundo por su vistoso muy colorido, además esta especie es de crecimiento lento lo cual con distintos sustratos se podrá tener un excelente desarrollo de la misma. Es por ello que se trabajará con esta planta ya que es una de las más comunes y de mayor demanda en la ciudad de Milagro.

2.2.2.2 Girasol

El girasol tiene distintos usos uno de ellos es el de decoración cabe destacar que de esta también se obtienen semillas y aceites comestibles, la planta es nombrada así ya que en su etapa de desarrollo se mueve en dirección al sol permitiéndole obtener todos los beneficios que este le brinda para su crecimiento. Osman et al. (2018) Mencionan que el girasol o (*Helianthus annuus* L.) en su nombre científico es uno de los 4 cultivos más importantes en el mundo ya que es una planta oleaginosa es decir que de su semilla se puede extraer aceite, también mencionan que gracias a su sistema radicular pueden resistirá a sequias ya que este le permite obtener agua de las capas más profundas de suelo. De igual manera teniendo en cuenta algunas de las características mencionadas con las que se trabajó para identificarla.

2.2.2.3 Palmeras

La palma blanca o washingtonia es un tipo de palmera perteneciente a la familia Arecaceae con características robustas y con un tamaño que podrían llegar a medir hasta unos 35 metros de altura dependiendo el año de vida; adaptable al tipo de suelos arcillosos y húmedos. Beluzan (2018) indica que esta palmera puede ser afectada como todas las palmeras de la familia Arecaceae por hongos teniendo en cuenta que los síntomas que se verían en la planta es el color amarillento en la mayoría de sus hojas o incluso manchas el cual provocaría el marchitamiento. De este modo se pueden ver algunos de las posibles causas que afecten a la planta, de dicha manera se tomó en cuenta las posibles afectaciones para el desarrollo del presente proyecto donde se obtuvo distintas características de dicha planta.

2.2.3 Visión por computadora

Los sistemas basados en visión por computadora o también llamado visión artificial ayudan al reconocimiento y comprensión de objetos. De acuerdo con Feng, Jiang, Yang, Du y Li (2019) explican que la visión por computadora tiene como objetivo clasificar, identificar y comprender automáticamente imágenes en clases predefinidas para el análisis de estas. Por tanto, se usó visión por computadora para la clasificación e identificación de imágenes de las plantas ubicadas en los diferentes viveros.

Además, la visión por computadora se ha venido usando desde la década de los 60 y a inicios de los 70. Es un sistema que se ha venido implementando en diversos campos con técnicas para el aprendizaje automático, gráficos, realidad virtual y aumentada, reconocimiento de patrones y para construcciones 3D (Kakani, Nguyen, Kumar, Kim, & Pasupuleti, 2020). Por consiguiente, la utilización de la visión de computadora para la identificación de varias plantas es de suma

importancia por lo cual en el presente proyecto se trabajó con el mismo, ya que este método se ha venido usando a los largos del tiempo dando buenos resultados.

2.2.4 Google Colab

El análisis de algoritmos conlleva a ejecutar un estudio basado en espacio y tiempo para detallar los parámetros que los caracterizan, existen herramientas que facilitan su entrenamiento, una de ellas es Google Colab o Colaboratory, según Aponte, Castañeda y Medina (2020), mencionan que esta plataforma es gratuita y ayuda a la realización de códigos en basados en el lenguaje de programación Python, permitiendo ejecutarlo en el navegador simulando algoritmos que utilizan redes neuronales artificiales, facilita el procesamiento de datos, debido a que una de sus ventajas es que se accede a él como un servicio de cuaderno que no utiliza configuración específica para utilizar recursos informáticos. Como se ha mencionado para realizar el análisis de los tres algoritmos sujetos a evaluación de su desempeño, Google Colab ha sido de gran utilidad ya que se aloja en Jupyter y brinda la facilidad de codificar con las librerías TensorFlow, Keras y OpenCV y MATLAB.

La evaluación de algoritmos debe ser precisa y estar definida por una secuencia de pasos que garantice su ejecución y desempeño, los códigos pueden ser almacenados y compartidos gracias a Google Colab, donde la información se puede encontrar en la nube y se puede acceder mediante un explorador de archivos, se recomienda crear varias copias que garanticen la integridad de los archivos mediante Jupyter notebook; debido a la necesidad de implementar tecnologías de inteligencia artificial se buscan softwares que garanticen el desarrollo de algoritmos y análisis de datos, su ventaja principal es que brinda acceso a GPU y TPU debidamente con almacenamiento en la nube de Google

Drive (Garrido, 2022). Con esta plataforma se facilitó la programación en equipo de los algoritmos, cada algoritmo se enfocó a la visión por computadora por lo cual se evaluaron desde el entrenamiento hasta el funcionamiento total que fue la detección de plantas ornamentales.

2.2.5 Algoritmos de aprendizaje automático

Los algoritmos de aprendizaje automático pueden poseer diferentes enfoques por lo cual es importante brindar un entrenamiento adecuado para que puedan identificar y correlacionar los datos ingresados, en este caso estuvieron orientados al reconocimiento de plantas, además de determinar su rendimiento funcional.

2.2.5.1 Redes neuronales convolucionales

Una de las partes más importante actualmente en identificación y aprendizaje automático que permite el entrenamiento del algoritmo a base de información recabada son las redes neuronales. De forma que Rivas y Mazón (2018) mencionan que las redes neuronales son un procesador de información modelada en los sistemas nerviosos biológicos donde a través de experiencias adquieren conocimiento, almacenando la información al igual forma como en el cerebro. Por lo que la utilización de este aporta en el desarrollo del algoritmo donde se identifican plantas ornamentales con el entrenamiento mediante la obtención de información como lo es la fotografía del cultivo.

El reconocimiento a base de características es un arduo trabajo y mucho más aplicarlos. Por lo que Artola (2019) indica que la red neuronal convolucionales es un tipo de red neuronal artificial la cual a través de la visualización imita al cerebro humano para identificar distintas características, en donde contiene capas especializadas para detectar propiedades y formas, de tal manera que se llega a especializar para reconocer una forma, rostro o silueta. Del mismo modo siendo

este un algoritmo altamente efectivo para la identificación se lo usó como uno de los algoritmos de aprendizaje automático para su comparación con otros.

Las redes neuronales convolucionales (CNN) son aquellas que trabajar con una gran cantidad de imágenes para dar mayores resultados ya que su estudio estaba basado en que cada neurona ensaya un píxel de la imagen. Andrade, Sinche y Hidalgo (2021) mencionan que es un tipo de red neuronal artificial profunda que obtiene este nombre de la operación matemática lineal entre matrices llamada convolución. Por esta razón las CNN son muchas más eficientes y rápidas para aplicar los métodos dentro de los datos que son procesados por las capas de entrada, capas de salida y ocultas que servirán para el entrenamiento del algoritmo de predicción que se va a estudiar.

El algoritmo de redes neuronales convolucionales para un correcto entrenamiento requiere de una gran cantidad de imágenes. Como señalan Chanampe et al. (2019), las redes neuronales convolucionales cuentan con una estrategia de entrenamiento que constan de cuatro técnicas, las cuales son: Pre-Training y Fine-Tuning, Data Augmentation y DropOut.

Pre-Training o el uso de redes pre-entrenadas consiste en la transferencia de aprendizaje, esta técnica se la lleva a cabo generalmente sobre imágenes naturales este método es uno de los más populares usados en las redes neuronales convolucionales ya que permite la aceleración de aprendizaje del algoritmo y Fine-Tuning es usado generalmente para ajustar los parámetros teniendo concordancia con las tareas que se le asignan al algoritmo como objetivo.

Data Augmentation o aumento de datos es una técnica usada de la forma más fácil para reducir la redundancia cuando se trabaja con imágenes como datos, donde esta técnica tiene de dos formas diferentes para mejorar los datos: La

primera forma es crear un desplazamiento de imagen horizontal y un reflejo horizontal y la segunda forma es el cambio de intensidad RGB en la imagen y DropOut también conocido como dilución es una técnica para regularizar y reducir el sobreajuste siempre y cuando se tenga un conjunto de datos para el entrenamiento del modelo.

De qué manera obtenemos un mejor funcionamiento del algoritmo CNN; este es uno de los algoritmos que ha contribuido principalmente al desarrollo de la visión por computadora a través del estudio de imágenes tomando entradas para categorizarlas obteniendo características únicas identificando líneas, curvas y formas complejas para identificar objetos tal como menciona Calderón y Hurtado (2019) que este tipo de redes está especializada en el procesamiento de datos con tipología malla, donde localiza patrones en imágenes para extraer características que se buscan, tiene capas con neuronas en 3 dimensiones como ancho, altura y profundidad, píxeles que van desde 0 hasta 255 con la escala RGB (red, green, blue) , usa un filtro para extraer características únicas llamado kernel, además se entrenan para realizar nuevas tareas a partir de redes ya existentes; Es por ello que el estudio y conocimiento más a profundidad del algoritmo beneficiaría a futuros trabajo otorgando información útil y veraz.

Es por ello, que es uno de los más usados y recomendados en todo tipo de identificación por su similitud con el cerebro humano y modo de aprendizaje a medida que más información tiene mejores resultados proporciona. De igual manera Santamaría, Zuñiga, Pineda, Somodevilla y Rossainz (2019) mencionan la importancia del mismo ya que en los últimos años en este campo de la IA es muy común encontrar app y programas que permitan reconocer algún objeto, persona o cosas y no solo eso, sino que en el campo medicinal y agrícola son utilizados

para identificar enfermedades entre otras cosas, el estudio de imágenes a través de este algoritmo es sumamente sorprendente funciona con cantidades moderadas de datos de capacidad (miles de imágenes etiquetadas) pudiéndose ejecutarlas en tan solo minutos, a modo que este algoritmo está potenciando el crecimiento para futuras tecnologías y modelos de reconocimiento, y por esta razón se lo compara con otros algoritmos similares.

2.2.5.2 Algoritmo K-medias

El algoritmo K medias o K-means es conocido por ser de clasificación no supervisada, se emplea cuando el objetivo es realizar particiones en un conjunto de datos, se caracteriza por tener un parámetro de entrada, el cual se denomina K, este puede llegar a indicar la cantidad de clústeres que se deben generar, su funcionamiento está basado en seleccionar el número de clúster, para luego seleccionar de forma aleatoria los K puntos, se continua con la comparación de distancia de los datos y el punto seleccionado, esto para asignar a el que se encuentre más cerca, este algoritmo brinda soluciones iterativas favoreciendo a los análisis que contienen conjuntos de datos regulares (Choez, 2022). Como se ha mencionado, este algoritmo destaca por ayudar a analizar conjuntos de datos y segmentarlos, de igual forma en el presente estudio, el algoritmo se desarrolló para realizar el procesamiento mediante visión por computadora y verificar su eficiencia en relación al tiempo, precisión, rendimiento y sensibilidad, al ser de tipo no supervisado no tuvo ninguna referencia o etiqueta de lo que se analizó.

El algoritmo K-means trabaja como un conjunto de datos que se describen con valores numéricos. Franco-Árcega et.al (2021) menciona que k-means es uno de los algoritmos más populares de agrupamiento en minería de datos; el punto central de este método es ir mejorando la posición la posición de los distintos objetos,

haciéndolos de interacción a interacción hasta que se encuentre la mejor división que hay entre objetos , esto quiere decir que se busca encontrar un grupo de objetos que sean parecidos entre sí, de la misma manera este algoritmo consta de un procedimiento para su ejecución que consiste en hacer el cálculo de distancia entre los objetos propuestos y los centros creados de cada grupo:

Usando la función euclidiana para medir la distancia más corta entre dos puntos:

$$d(x, y) = \sqrt{\sum_{i=1}^n (X_i - y_i)^2}$$

Esta ecuación indica que los objetos son representados por i valores de atributo y que todos son considerados para obtener la distancia de x y y .

Dónde: se identifican los centros creados de cada grupo; se asigna los objetos al grupo de centros más cercanos; se recalcula los centros ya creados considerando los objetos agregados anteriormente para así crear un centro general; si los centros calculados previamente en la interacción son iguales a los centros ejecutados anteriormente. K-medias, creado por MacQueen en 1967, se destaca como uno de los algoritmos más sencillos y ampliamente empleados. También conocido como K-medias, este algoritmo tiene como objetivo clasificar datos según su propia información, buscando minimizar una función de error cuadrático. Su fundamento radica en reducir la distancia interna entre los patrones de un grupo, garantizando así la localización de un mínimo local de la función. Cabe destacar que el resultado puede depender del punto de partida del algoritmo (Sánchez, 2021). Por tanto, el uso de este algoritmo para su estudio sería de gran importancia y beneficioso ya que es uno de los algoritmos más populares en cuanto a la identificación por medio de la minería de datos.

2.2.5.3 Algoritmo KNN (*K-nearest neighbor*)

El Algoritmo KNN (*K-nearest neighbor*) es un método simple basado en tecnología de clasificación de minería de datos del cual se establece un conjunto de entrenamiento. Potosi (2019) indica que el KNN es un clasificador multiclase donde a través de diferentes parámetros como la distancia métrica y el valor de k (número de vecinos más cercanos) se pueden obtener los cuales son considerados fundamentales dentro de este ya que el rendimiento está ligado al número k de vecinos más cercanos, donde también forma parte de un conjunto de entrenamiento con sus respectivas etiquetas que sirven como información para generar este modelo de clasificación, por ende se trabajará con este algoritmo KNN de clasificación donde a base de diferentes conjunto como muestras se obtuvo la clasificación de la distintas plantas para poder identificarlas y de tal forma medir su eficiencia y eficacia en comparación con los demás algoritmos. El algoritmo KNN consta de varias características que son:

El uso de la distancia euclidiana para la medición de la similitud entre tuplas de entrenamiento y las tuplas usadas en pruebas.

Este algoritmo es simple y fácil de aplicar así mismo proporciona una recomendación rápida y precisa al usuario.

Es uno de los clasificadores con mayor estabilidad a diferencia de otros métodos de minería de datos que existen, como uno de ellos los árboles de decisiones, esto a través de manejar gran cantidad de datos que son usados para el respectivo entrenamiento (Madariaga, Lao, Curra, & Martín, 2022).

El algoritmo K vecinos más cercanos (KNN) es un enfoque de aprendizaje automático supervisado basado en instancias. Puede ser aplicado para la clasificación de nuevas muestras o para hacer predicciones. Este algoritmo es de

fácil implementación y se utiliza ampliamente para resolver diversos problemas, tales como sistemas de recomendación, búsquedas semánticas y también la detección de anomalías. KNN es uno de los algoritmos de clasificación más populares, y su concepto se fundamenta en el principio de clasificar según la proximidad a los vecinos más cercanos (Guevara-Fernandez & Coral-Ygnacio, 2023).

Para ello, se debe calcular la distancia euclidiana con la siguiente fórmula:

$$dist(X_i, X_j) = \sqrt{(X_{i1} - X_{j1})^2 + (X_{i2} - X_{j2})^2 \dots (X_{ip} - X_{jp})^2}$$

Donde:

x_i : Es un clúster de entrada con características p ($x_{i1}, x_{i2}, \dots, x_{ip}$)

n : El número total de tuplas de entrada ($i = 1, 2, \dots, n$)

p : El número total de características ($j = 1, 2, \dots, p$)

La distancia euclidiana entre tuplas X_i y X_t ($t = 1, 2, \dots, n$)

De forma que la utilización de este algoritmo ayude ya que es un algoritmo clasificador que tiene como objetivo clasificar correctamente a través de un conjunto de datos iniciales por medio de la medición de distancia usando la fórmula euclidiana.

A demás, pese a que sirve esencialmente para clasificar datos a partir de la cercanía, también es utilizada para la identificación mediante esta misma técnica, pero a partir de datos de imágenes. Tobar-Díaz, Gao, Mas y Cambron-Sandoval (2023) mencionan que este algoritmo funciona bajo el supuesto de que los píxeles más cercanos en el espacio espectral pueden ser probablemente de la misma clase, siendo así que toma píxeles desconocidos y examina los píxeles disponibles que tengan un dominio espectral parecido, para darle un valor y a partir de este

valor define la clase que mejor le representa de un número determinado de vecinos cercanos; en conclusión, se lo utiliza al igual que otros algoritmos que permiten identificar imágenes a partir del estudio a profundidad de la misma, siendo esta uno de las razones por la cual se lo ha escogido para realizar este estudio comparativo.

2.2.6 Lenguaje de programación Python

Python es uno de los lenguajes de programación de alto nivel más utilizados para la creación de aplicaciones web y desarrollo de software. Valarezo, Honores, Gómez y Vincés (2018) mencionan que Python es un lenguaje de programación libre de propósito general, su código es interpretado por lo que podría ser relativamente lento, contiene gran variedad de librerías y es orientado a objetos. Por lo tanto, se usará este lenguaje para realizar el análisis de los algoritmos de redes neuronales para identificar plantas.

2.2.7 Librerías

Python contiene varias librerías que permiten la obtención de imágenes y analizar sus características a profundidad, es por ello que en el proceso de codificación de los algoritmos de aprendizajes automáticos antes mencionados se ha utilizado distintas librerías en la ejecución del proyecto como son:

2.2.7.1 *TensorFlow*

En el desarrollo de la inteligencia artificial y de algoritmos de aprendizaje automático la librería TensorFlow es muy importante ya que es una de las bibliotecas de código abierto y desarrollada por Google, de forma que se pueden satisfacer las necesidades a partir de redes neuronales artificiales donde permiten entrenar y construir redes para detectar patrones y razonamientos humanos. Quiroz, Valverde, Prieto y Apupalo Del Rosario (2020) mencionan la importancia y eficacia de la utilización de la librería TensorFlow en el aspecto de detección a partir

del entrenamiento mediante una red neuronal, la cual hace que esta librería pueda aprender y reconocer objetos en tiempo real implementada en algoritmos de aprendizaje automáticos, es por ello que en la codificación de nuestro algoritmo realizada en Python se usó la librería TensorFlow para el reconocimiento de una planta ornamental.

2.2.7.2 OpenCV

Detección de características con OpenCV la cual es parte de la vision artificial con gran gama de herramientas para resolver problemas en el procesamiento de imágenes. Sugue (2020) indica que esta es una biblioteca de vision artificial que que enlaza con python y desarrollada por Intel, de igual manera esta librería trabaja el procesamiento de imágenes en tiempo real, tambien es de codigo abierto y utilizada en distintos sistemas operativos como Mac OSX, Windows y Linux; abarcan una variedad de funciones aplicadas en distintas áreas como el reconocimiento facial, vision robotica y calibración de cámaras, de tal forma esta permite utilizar la cámara a travez de la interfaz de python para acceder a sus funciones. De esta forma se usó la librería openCV para poder obtener imágenes a traves de la cámara con la vision artificial y aplicada con python para identificar la planta ornamental ya que esta permite el analisis y procesamiento de imagenes.

2.2.7.3 Keras

El uso de la librería Keras dentro de lenguaje programación Python es uno de los más utilizados incorporada en el 2017 por TensorFlow para poder acceder a sus módulos. Torres (2020) menciona que Keras es una API de alto nivel que puede ser aplicada en redes neuronales convirtiéndola en el entorno perfecto para iniciarse en esta área; es de código abierto y trabaja con TensorFlow, CNTK o Thenos en el lenguaje de programación Python donde esta se convirtió en la API

integrada por tensorflow para construir y entrenar sus modelos. De este modo trabajamos con la librería Keras ya que es la que nos permite construir y entrenar nuestro modelo creado a través de tensorflow de forma que permitió el correcto funcionamiento de los algoritmos a comparar.

2.2.7.4 Sklearn

El uso de las librerías en Python puede ayudar a implementar los algoritmos gracias a la disponibilidad de estas; la librería sklearn o scikit-learn es una librería que proporciona las herramientas para aplicarlas en algoritmos de clasificación, regresión, clustering y reducción de dimensionalidad además presenta compatibilidad con otras librerías de Python como Numpy, matplotlib entre otras. De tal forma Douglass (2020) menciona algunos aspectos de las librerías Keras, tensorflow y sklearn dando a conocer que estas con Python son herramientas que facilitan la programación del usuario ya que tienen la capacidad de trabajar con modelos de aprendizaje profundo y automáticos de este modo se puede recalcar que el uso de esta librería dentro del lenguaje de programación Python tiene múltiples beneficios como trabajar con otros algoritmos para el análisis y la comparación, también es una de las bibliotecas de aprendizaje automático más popular dentro del mismo. De este modo se trabaja con sklearn ya que es gratuita y brinda técnicas para evaluar los resultados para llevar a cabo la comparativa y el análisis de los distintos algoritmos.

2.3 Marco legal

Para el estudio de los diferentes algoritmos de aprendizaje automático, se requiere conocer las distintas leyes, reglamentos, decretos, acuerdos y normas nacionales e internacionales vigentes que se relacionen con el proyecto a desarrollar, para esto se toman las siguientes leyes:

2.3.1 Constitución de la República del Ecuador

La importancia en el desarrollo de nuevas tecnologías radica en el cumplimiento de distintos aspectos como la Constitución de la República del Ecuador modificada en el año (2021) menciona que:

Art. 385.- El sistema nacional de ciencia, tecnología, innovación y saberes ancestrales, en el marco del respeto al ambiente, la naturaleza, la vida, las culturas y la soberanía, tendrá como finalidad: 1. Generar, adaptar y difundir conocimientos científicos y tecnológicos. 2. Recuperar, fortalecer y potenciar los saberes ancestrales. 3. Desarrollar tecnologías e innovaciones que impulsen la producción nacional, eleven la eficiencia y productividad, mejoren la calidad de vida y contribuyan a la realización del buen vivir (p. 158).

De forma que a base de este artículo menciona que las nuevas tecnologías son un gran aporte a la nación de forma que deben cumplir con los distintos aspectos fomentando nuevos conocimientos, es por ello, por lo que al momento de realizar el estudio y análisis de los distintos algoritmos se tomará en cuenta las características mencionadas en el presente artículo.

2.3.2 Ley de Propiedad Intelectual

Esta ley respalda el desarrollo del proyecto, por la cual se usa el siguiente artículo relacionado con los derechos a creaciones de autoría propia, de acuerdo como establece la Legislación Nacional - Ecuador (2014):

Art. 1.- El Estado reconoce, regula y garantiza la propiedad intelectual adquirida de conformidad con la ley, las decisiones de la Comisión de la Comunidad Andina y los convenios internacionales vigentes en el Ecuador.

La propiedad intelectual comprende:

1. Los derechos de autor y derechos conexos;
2. La propiedad industrial, que abarca, entre otros elementos, los siguientes:
 - a) Las invenciones;
 - b) Los dibujos y modelos industriales;
 - c) Los esquemas de trazado (topografías) de circuitos integrados;
 - d) La información no divulgada y los secretos comerciales e industriales;
 - e) Las marcas de fábrica, de comercio, de servicios y los lemas comerciales;
 - f) Las apariencias distintivas de los negocios y establecimientos de comercio;
 - g) Los nombres comerciales;
 - h) Las indicaciones geográficas; e,
 - i) Cualquier otra creación intelectual que se destine a un uso agrícola, industrial o comercial.

3. Las obtenciones vegetales. Las normas de esta Ley no limitan ni obstaculizan los derechos consagrados por el Convenio de Diversidad Biológica, ni por las leyes dictadas por el Ecuador sobre la materia (p.1).

Se toma en cuenta este artículo ya que manifiesta los derechos de los creadores intelectuales para salvaguardar proyectos propios.

2.3.3 Modelo de software libre

En la etapa de desarrollo de los algoritmos de visión por computadora se utilizó el lenguaje de programación Python y sus librerías TensorFlow, OpenCV y Keras, por lo cual se cita como fundamento al Gobierno Electrónico del Ecuador (2008):

Artículo 142.-Tecnologías libres

Se entiende por tecnologías libres al software de código abierto, los estándares abiertos, los contenidos libres y el hardware libre. Los tres primeros son considerados como Tecnologías Digitales Libres.

Se entiende por software de código abierto al software en cuya licencia el titular garantiza al usuario el acceso al código fuente y lo faculta a usar dicho software con cualquier propósito. Especialmente otorga a los usuarios, entre otras, las siguientes libertades esenciales:

La libertad de ejecutar el software para cualquier propósito;

La libertad de estudiar cómo funciona el software, y modificarlo para adaptarlo a cualquier necesidad. El acceso al código fuente es una condición imprescindible para ello;

La libertad de redistribuir copias; y,

La libertad de distribuir copias de sus versiones modificadas a terceros (p.1).

En base a la ley mencionada, se utilizaron plataformas de software libre para la codificación y entrenamiento de los algoritmos que se evaluarán.

3. Materiales y métodos

3.1 Enfoque de la investigación

Existen 3 tipos de enfoques: cualitativos, cuantitativos y mixtos en este caso el proyecto tiene un enfoque mixto.

3.1.1 Enfoque Mixto

El enfoque cualitativo ayuda a la recolección de datos sin medición numérica, este tipo de investigación se basa en el estudio del mundo real para así poder desarrollar una teoría. El enfoque cualitativo es la metodología en la cual se usan textos, discursos, imágenes, gráficos entre otras cosas, con el fin de comprender sobre el objeto estudiado (Nizama V & Nizama C, 2020). A base del enfoque cualitativo se obtuvo distintas características del objeto de estudio con la aplicación de técnicas de recolección de datos a través de entrevistas y fichas de observación con la finalidad de saber los tipos de plantas con mayor demanda que tienen los distintos viveros de la ciudad.

En cuanto al enfoque cuantitativo se caracterizan porque son objetivas y buscan comprobar una hipótesis, según Amaiquema, Vera, & Zumba (2019) mencionan que las investigaciones se basan en un conjunto de procesos sistemáticos que se aplican al objeto de estudio, gracias a este enfoque se puede generar un pensamiento que se fundamente en la representación numérica, y de esa forma realizar mediciones y evaluaciones de las variables o métricas que se estén analizando, además de fundamentarse en la estadística para la comprobación de la hipótesis. Para saber que una la formulación de una hipótesis es la correcta, su estructura debe estar compuesta por las unidades de observación, los sujetos u objetos y las variables o atributos que estarán susceptibles a medición de los cuales se esperan se relacionen para que se conforme la direccionalidad de la hipótesis.

Como se ha mencionado la hipótesis que se comprobó es que al menos uno de los algoritmos de aprendizaje automático permite la identificación de una planta ornamental de forma exacta, precisa, en el menor tiempo y con mayor rendimiento.

3.1.2 Tipo de investigación

Dentro de la realización de un proyecto como tal se pueden aplicar distintos tipos de investigación para llegar a su objetivo; como lo es la investigación descriptiva, aplicada y la investigación documental que en este caso se las usó.

3.1.2.1 Investigación descriptiva

Esta investigación se basa en la puntualización de características de la población que se está estudiando. Los resultados de la investigación descriptiva deben ser verídica, precisa y sistemática, en la cual la característica principal es que estos datos sean observables y verificables (Guevara, Verdesoto, & Castro, 2020). En la cual se ejecutó un análisis de las diferentes características tomadas mediante las entrevistas realizadas al dueño del vivero "Belleza natural".

3.1.2.2 Investigación aplicada

La investigación aplicada como tal, es la que permitirá resolver problemas o planteamientos específicos contribuyendo a la construcción de nuevos conocimientos. Nicomedes (2018) establece que:

Está orientada a resolver los problemas que se presentan en los procesos de producción, distribución, circulación, y consumo de bienes y servicios de cualquier actividad humana. Se denomina aplicadas; porque en base a investigación básica, pura o fundamental en las ciencias fácticas o formales se formulan problemas o hipótesis de trabajo para resolver los problemas de la vida productiva de la sociedad (p.3).

A base del conocimiento que adquirido usando este tipo de investigación permitió pasar del conocimiento teórico al práctico.

3.1.2.3 Investigación documental

La investigación documental es una de las técnicas de investigación cualitativa que permite la obtención, recopilación y selección de información de diferentes documentos, como: libros, revistas, constituciones entre otras. Este tipo de investigación también es llamada como una investigación bibliográfica, ya que es caracterizada por utilizar datos secundarios como fuente de información (Reyes-Ruiz & Carmona, 2020). Por consiguiente, se decidió usar este tipo de investigación para la realización del presente proyecto ya que gracias a esta se obtiene información relevante por medio de artículos científicos, libros o revistas.

3.1.3 Diseño de investigación

Para tomar en cuenta el tipo de investigación a usar se deben apreciar las diferentes características y necesidades que presenta el presente proyecto, es por ello, por lo que se implementó la investigación experimental.

3.1.3.1 Investigación Experimental

Este tipo de investigación cuenta con variables independiente pero el investigador observa de qué manera se desarrolla el fenómeno para así analizar y obtener información de este. Álvarez-Risco (2020) da a conocer que en la investigación experimental los datos obtenidos de dicha investigación se obtienen mediante la observación directa del hecho a investigar, en donde se realiza la manipulación de una sola variable y se espera la respectiva respuesta de otra, esta investigación se divide en tres tipos, las cuales son; la preexperimentales, casi experimentales y las experimentales puras. En definitiva, el uso de esta investigación para la creación del proyecto ayudó de manera considerable ya que gracias a este permitió analizar los valores obtenidos de las diferentes variables a través de la observación del funcionamiento de los distintos algoritmos.

3.2 Metodología

En el siguiente apartado se establece la metodología y fases que permitieron el desarrollo de los algoritmos, se siguió una metodología de desarrollo basada en un estudio de análisis comparativo que consta de pasos como mismos que servirán para la comparación y evaluación de resultados.

3.2.1 Variables de la investigación

Las variables de investigación conforman la parte medible del proyecto, estas variables contribuyen a responder las preguntas de la investigación favoreciendo al desarrollo de cada objetivo planteado.

3.2.1.1 Variable independiente

Los algoritmos: redes neuronales convolucionales, K-means y KNN.

El conjunto de imágenes de plantas: Ixoras, girasoles y palmas.

3.2.1.2 Variable Dependiente

3.2.1.2.1 Precisión

La precisión es un parámetro importante al querer medir la efectividad de un algoritmo, ya que la medición de este parámetro trata de la búsqueda del mínimo error y de resultados similares del algoritmo estudiado. Barrero (2019) indica que, para realizar el cálculo de la precisión, el número de resultados positivos se divide por el número de resultados positivos predichos por el clasificador, es importante que las muestras tomadas sean identificadas como positivas.

$$Precision = \frac{VerdaderoPositivo}{VerdaderoPositivo + FalsoPositivo}$$

True Positives: Verdadero Positivos

False Positives: Falso Positivos

3.2.1.2.2 *Exactitud*

Esta métrica mide el porcentaje en la que el modelo ha acertado, la exactitud o también llamada como accuracy con su nombre en inglés es una de las más usadas y preferidas. Mayer, Cáceres, Firman y Vera (2018) afirman que la exactitud es una de las métricas más usadas puesto que miden el porcentaje en que el modelo o el algoritmo ha acertado en la identificación, pero a la vez no la recomiendan, debido a que, se puede crear un algoritmo malo o sencillo y haga que parezca mejor de lo que está planteado.

$$exactitud = \frac{\text{Numero de predicciones correctas}}{\text{Numero total de muestras de entradas}}$$

Number of Correct Predictions: Número de predicciones correctas

Total number of predictions made: Número total de muestras de entrada

3.2.1.2.3 *Sensibilidad*

Métrica que indica la cantidad que el algoritmo de aprendizaje automático puede hacer la identificación del objeto. Cao y You (2020) argumentan que el Recall o también llamado sensibilidad es el número de elementos que se identifican de manera correcta como positivos, de manera que, para el cálculo de la sensibilidad se utilizará la siguiente formula:

$$sensibilidad = \frac{TP}{TP + FN}$$

TP: Positivos Verdaderos

FN: Negativos Falsos

3.2.1.2.4 *F1*

La puntuación F1 se usa para medir la precisión de una prueba. Zevallos, Onque, Canaza y Choqueneira (2022) declaran que el F1 indica que tan preciso es el clasificador en la cual trabaja con 0 y 1, estos valores dependen de cómo se vaya

ejecutando los clasificadores, cuando toma el valor de 0 es porque uno de los dos valores vale 0 y toma el valor de 1 cuando los dos valores toman el valor de 1, la cual se usa para combinar las medidas de precisión y sensibilidad (recall), esta métrica cuenta con la siguiente fórmula.

$$F1 = 2 * \frac{Precision * Exactitud}{Precision + Exactitud}$$

3.2.1.2.5 Tiempo utilizado para la detección

El Tiempo visto en la ejecución e identificación de los distintos algoritmos el cual depende de las características de la computadora donde sea ejecutado.

3.2.2 Fases de la investigación

Las fases investigativas son basadas en la metodología propuesta como se muestra a continuación ver Anexo 4. Metodología Propuesta, es importante mencionar que cada fase conllevó a realizar diferentes procesos de entrenamiento y codificación.

3.2.2.1 Metodología Propuesta

La metodología es creada para llevar los proyectos con un orden en donde por fases se van realizando los distintos procesos para ayudar a la realización del mismo. Gutiérrez (2019) utiliza un tipo de metodología creada por ellos para beneficiar y ser aplicada en estudios de comparación de distintos algoritmos, es por ello, por lo que se usó esta metodología ya que ayuda al análisis de distintos algoritmos de aprendizaje automático el cual se basa en la identificación de plantas ornamentales en la cual consta de las siguientes fases:

3.2.2.1.1 Etapa de recopilación de información de datos de entrada

En esta primera etapa se maneja la recolección de las imágenes para generar los dataset que posteriormente permiten el entrenamiento de los distintos

algoritmos mencionados en el proyecto, así mismo se realiza el procesamiento de las imágenes para realizar los debidos entrenamientos de manera rápida; para ello se realiza la el entrenamiento en Google colab ya que es un entorno web en donde se puede realizar el entrenamiento de manera rápida y sin usar mucho recurso del pc, de igual manera el dataset creado tiene que ser verificado ya que pueden ir imágenes en blanco o que no tengan relación con los datos seleccionados para realizar el entrenamiento.

3.2.2.1.2 Etapa de procesamiento y transformación de los datos al modelo espacio vectorial

En la segunda etapa se trabajó el procesamiento de las imágenes de manera más profunda para así proceder a transformar y limpiar los datos inmisarios que perjudiquen al futuro entrenamiento de los algoritmos, estos datos pueden ser imágenes o fotos que sean irrelevantes para el entrenamiento, de la misma manera que no tengan la calidad adecuada para el respectivo entrenamiento de los algoritmos.

3.2.2.1.3 Etapa de selección e implementación de los algoritmos de aprendizaje automático

Esta etapa consistió en la búsqueda de los diferentes algoritmos de aprendizaje automático que existen sean supervisados y no supervisados, para su respectiva selección y proseguir a la codificación de esos mediante su respectivo lenguaje de programación es Python y mediante las diferentes librerías como TensorFlow, OpenCV, Sklearn y Keras, se procede a realizar el estudio comparativo de cada uno de ellos.

3.2.2.1.4 Etapa de evaluación y comparación de los resultados obtenidos

En esta última etapa se procedió a realizar las distintas pruebas de los algoritmos usadas para el estudio comparativos en la identificación de una planta ornamental, en la cual se realizará el análisis de los distintos criterios planteados como son la precisión, exactitud, sensibilidad, F1 y tiempo utilizado para la detección, una vez comparados estos resultados se podría decir que algoritmo es el más óptimo para la identificación de plantas.

3.2.3 Recolección de datos

La recolección de datos fue sumamente necesaria para obtener la información necesaria para cumplir con cada una de las etapas del proyecto, se acudió a una serie de recursos bibliográficos y personal de trabajo de tres viveros del cantón Milagro como a continuación se detalla.

3.2.4.1. Recursos

Los recursos son parte fundamental para llevar a cabo el presente análisis comparativo, el conocimiento, la experiencia y la observación fueron gracias a los recursos humanos, por otro lado, los recursos materiales o tecnológicos están conformados por las distintas herramientas tanto de software como hardware.

3.2.3.1.1 Recursos Humanos

Estudiantes: Almeida Flores Lenin y Durango Devis Wilmer

Docente Guía

Empleados

Propietario o administrador

Clientes

3.2.3.1.2 Recursos Materiales

Computador

Utilitario de oficina

3.2.3.1.3 *Recursos tecnológicos*

Software

Celular Samsung A51 con versión de Android 12 y RAM de 4gb.

Language de programación Python

3.2.3.1.4 *Recursos bibliográficos*

Libros

Revistas

Artículos científicos

Google académico

3.2.3.1.5 *Presupuesto del proyecto*

En el presupuesto del proyecto se toma en cuenta varios aspectos importantes que ayudan en el desarrollo del presente proyecto, así mismo para el entrenamiento y creación de la base de datos del cual para se realizó una tabla que especifique los precios y el total a gastar como se muestra en el Anexo 1. Presupuesto del proyecto.

3.2.4.2. *Métodos y técnicas*

Los métodos y técnicas que se tomarán en cuenta para el desarrollo de la del presente proyecto son las siguientes:

3.2.3.1.6 *Método analítico*

El método analítico es la segmentación del problema que va de lo general a lo específico. Moscoso y Quiñonez (2018) mencionan que el método analítico fue de mucha ayuda para realizar su trabajo ya que les permitió separar, analizar e indagar su objeto de estudio en el documento realizado para luego integrarlos de manera holística. De tal forma el método analítico es importante para la realización del

presente proyecto gracias a que mediante este método se va a poder dividir los presentes problemas por partes para así poder analizarlos de forma individual y llegar a la solución óptima, esto provocaría una gran ventaja puesto a que se puede enfocar al estudio del problema mediante pequeñas partes, comprendiendo el problema detalladamente y con profundidad.

3.2.4 Análisis estadístico

La información base que permitió el desarrollo del estudio comparativo de los 3 algoritmos seleccionados, se tomó en cuenta la información recolectada en un vivero de la ciudad de milagro Vía Naranjito km 3 ½, ya que este permite que se recopile información y también la toma de fotos de las diferentes plantas, además se realizó la entrevista al dueño del vivero para la cual se elaboró las preguntas como se observa en el Anexo 2. Ficha de observación, relacionada con el tamaño de las plantas, su forma, así mismo las plantas más pedidas por tamaño (grande, mediana y pequeña); gracias a la aplicación de los algoritmos de aprendizaje automático y el análisis de las diferentes métricas a evaluar que fueron la precisión, sensibilidad, exactitud, F1 y tiempo de identificación del algoritmo se pudo concluir que el algoritmo de Redes Neuronales Convolucionales (CNN) es el más óptimo para futuras implementaciones ya que al realizar varias pruebas en diferentes imágenes tuvo índices de efectividad más altos que los otros dos algoritmos como se muestra en el Anexo 18. Cuadro comparativo, el cual muestra la tabla en donde se realizaron las diferentes pruebas con diversas fotos.

4. Resultados

4.1. Análisis del funcionamiento de los algoritmos de aprendizaje automático que permite la identificación de imágenes mediante revisión bibliográfica para el establecimiento de parámetros que evaluarán para su comparación.

Para la recolección de información del proyecto se utilizaron diferentes métodos de investigación como fuentes bibliográficas, que son libros, revistas científicas y proyectos de tesis de distintos autores ver Anexo 7. Autores que han usado algoritmos de aprendizaje automático., el cual muestra una tabla de distintos autores que han usado algoritmos de aprendizaje automático, así mismo, gracias a estas fuentes se pudo encontrar los parámetros que se necesitaban analizar de los diferentes algoritmos de aprendizaje automáticos para su estudio comparativo, de la misma forma se utilizó la técnica de investigación de la ficha de observación, esta técnica permitió conocer sobre las 3 plantas más populares en los viveros de la ciudad de Milagro.

Se realizaron las investigaciones pertinentes que ayuden a tomar las diferentes disposiciones para la evaluación de los distintos algoritmos a considerar en la propuesta tecnológica, para este análisis se toman varios criterios a considerar como lo son la exactitud, sensibilidad, precisión y F1, así mismo se investigó sobre los diferentes algoritmos existentes que estén dedicados al aprendizaje automático y considerar cuál de ellos son óptimos para la realización de estudio y su análisis ver Anexo 18. Cuadro comparativo el cuadro comparativo de los algoritmos de aprendizaje automático que se van a estudiar y comparar.

4.1.1. Observación

Se realizó una visita al vivero Belleza Natural ubicado en la vía Naranjito Km 3 ½ de la ciudad de Milagro, en la cual se pudo observar cuales eran las plantas más

compradas por los clientes que frecuentaban este lugar, de la misma forma se realizaron observaciones sobre las plantas ornamentales más vendidas como se observa en el Anexo 2. Ficha de observación por el tamaño de estas, es por ello que se tomaron en consideración para la identificación de las plantas ornamentales mediante algoritmos de aprendizaje automático, una planta pequeña, una mediana y una grande.

4.1.2. Entrevista

De la misma manera se realizó una entrevista a tres dueños de los diferentes viveros de la ciudad de Milagro para conocer sobre los tipos de plantas más vendidas para poder realizar la identificación de las tres plantas las cuales son el girasol, la ixora y la palmera como se muestra en el Anexo 3. Entrevista, así mismo se realizaron varias preguntas como cuales son las plantas más vendidas por el tamaño.

4.2. Programación de los algoritmos de aprendizaje automático mediante la utilización de Python y librerías para la obtención de criterios relacionados a estos.

En la realización de este objetivo se procedió con la codificación de los distintos algoritmos de aprendizaje automático ver desde el anexo Anexo 8. Codificación del algoritmo CNN hasta el anexo Anexo 15. Matriz de confusión del algoritmo K-Means que muestra el proceso de elaboración y prueba de los algoritmos; una vez obtenida la información necesaria de cómo utilizarlos para la identificación de las 3 plantas ornamentales, a través del lenguaje de programación Python y Google Colab ver Anexo 19. Recursos de las herramientas utilizadas donde se encuentran los recursos de cada herramienta utilizada, se ejecutaron los códigos generados y por medio del cual se pudieron utilizar las librería como OpenCV, tensorFlow, Keras

y Sklearn , entre otras que son las que permiten la obtención de las distintas características en el procesamiento de las imágenes, clasificación, predicción y detección, la cual se encuentran almacenando en drive de Google Colab como una base de datos para realizar el entrenamiento de los mismos. Como punto importante este proyecto no consta con interfases o manuales de usuario puesto que se lleva a cabo la programación de los diferentes algoritmos para su análisis comparativo con la finalidad de dar a conocer sus características de optimización y eficiencia para identificar plantas, así mismo este proyecto no va implementado en ninguna empresa en específico.

4.3. Establecimiento del conjunto de datos a utilizar mediante la captura de imágenes para el procesamiento y entrenamiento de los algoritmos de aprendizaje automático.

En este objetivo se realizó la captura de imágenes un celular de las tres distintas plantas que son la Ixora, la palma y el girasol, para esto se tuvieron que tomar distintas fotos de distintos ángulos, en el caso de las Ixoras se tuvieron que tomar fotos de los distintos colores existentes de estas plantas para que tengan una mayor exactitud al momento de identificar la planta al probar los distintos algoritmos, para el procesamiento y entrenamiento de imágenes de las plantas ver Anexo 20. Establecimiento de conjunto de imágenes el cual se realizó un archivo con dos carpetas una que constaba con las imágenes del entrenamiento que contiene 1200 fotos de las tres plantas y otra que consiste en la validación para el entrenamiento que consiste en 150 fotos, una vez realizada las fotos y las distintas carpetas estén creadas se realiza un archivo zip y se sube a la nube para tomarlo como una base de datos para el respectivo entrenamiento.

4.3.1. Procesamiento de imágenes

El procesamiento de imágenes se realizó mediante una serie de pasos para que estas puedan ser óptimas en el entrenamiento de los algoritmos de aprendizaje automático como se puede observar en el Anexo 5. Etapas del Procesamiento de imágenes donde muestra las etapas del procesamiento de imágenes.

Adquisición de imágenes o captura: En este paso se consiguió las imágenes mediante la realización de fotografías tomadas de las plantas, de la misma manera se sacaron varias imágenes de internet, tomando en cuenta que estén en una buena calidad, el fondo de estas imágenes o fotos, entre otras cosas.

Preprocesamiento: En esta etapa se realizó la corrección y mejora de calidad de las imágenes que fueron afectadas al momento de haber sido descargadas o en su caso haberlas tomadas a través de la cámara, esto con el fin de facilitar la identificación de las plantas y también el trabajo de las etapas siguientes.

Segmentación: Esta etapa fue de suma importancia ya que se separó las imágenes importantes de las que no tenían relevancia con el proyecto a realizar, por ejemplo, se eliminan imágenes que tengan fondos que puedan alterar el entrenamiento de los algoritmos y no permitan un correcto funcionamiento de estos.

Representación y descripción: Aquí se especificó el método para poder describir los datos y se seleccionó las características para extraer la información de las plantas, como por ejemplo, el color de las plantas, la forma que tienen y el tamaño de estas.

4.4. Evaluación de los criterios de exactitud, sensibilidad, precisión y F1 a través de la ejecución de los algoritmos para la comparación y establecimiento del algoritmo más óptimo que identifica plantas ornamentales.

En la ejecución de este objetivo se toma en cuenta los algoritmos codificados y evaluados en base a los distintos parámetros como exactitud, sensibilidad, precisión, F1 y el tiempo de identificación de la planta pero este último parámetro depende del rendimiento de la pc en que se lo ejecute, del cual uno de los 3 algoritmos de aprendizaje automático es el más óptimo para futuras aplicaciones y trabajos, teniendo como resultado que el algoritmo de redes neuronales convolucionales en base a lo realizado se concluye que este es el más óptimo tomando en cuenta sus características, como se muestra en el Anexo 18. Cuadro comparativo. En la cual en esta tabla se observa la comparación que se hace de los tres algoritmos para llegar a su debida conclusión de cual se debe usar para futuros proyectos.

5. Discusión

Con respecto a los diferentes algoritmos donde se realizar un análisis comparativo de ellos, es importante conocer sus diferentes limitantes y cuál sería el más óptimo de todos en base a sus capacidades. Se realizó un estudio comparativo de distintos algoritmos de aprendizaje automático para conocer el estado de suelo y su nivel de fertilidad para las plantaciones de banano, es por ello que se tomó 4 parcelas al azar, las cuales fueron clasificadas en zonas del 1 al 4, utilizaron un método supervisado, permitiendo la clasificación de los datos de estudio, la cual seleccionaron el algoritmos de árboles de decisión, las redes bayesianas, regresión logística entre otros algoritmos, para el estudio comparativo de estos algoritmos se tomaron varios parámetros como son el estadístico Kappa, error por media p-value, dando como resultado que el algoritmo más óptimo para la identificación del estado de nutriente del banano fue el de árboles de decisiones, dando a conocer que por medio del aprendizaje automático se pueden generar nuevos procedimientos permitiendo una toma de decisiones más efectiva (Vite, Carvajal, & Barrezueta, 2020). De igual forma el estudio realizado consiste en la identificación de tres plantas ornamentales, las cuales son la ixora, el girasol y la palmera, así mismo se toma en cuenta tres algoritmos para la identificación de estas que son el algoritmo de redes neuronales convolucionales, KNN y K-means donde fueron evaluados y comparados en distintos parámetros para obtener su eficiencia.

Además, se evaluó otro proyecto que tiene relación con la presente propuesta tecnológica que se está llevando a cabo. Sandoval (2018) indica que la cantidad de información generada día a día en internet en los diferentes medios existentes es de suma importancia para la toma de decisiones, es por ello que se realiza el

estudio comparativo de algoritmos de aprendizaje automático para la predicción de datos, para ello se tomaron varios algoritmos como los de clasificación y el algoritmo de regresión, las fases de desarrollo de los algoritmos constaron de 2 que eran la fase de entrenamiento en la cual tenían una cantidad considerable de datos que se separan por partes para su entrenamiento; y la fase de prueba, en esta fase los datos restantes se usan para realizar las pruebas necesarias para el correcto funcionamiento del algoritmo. De la misma manera los algoritmos usados en el proyecto tienen una fase de entrenamiento que se realiza para que estos puedan realizar la identificación de la planta de manera correcta; el uso de algoritmos para la realización de diferentes procesos facilita las tareas de los usuarios, ya que gracias al estudio comparativo de los algoritmos de aprendizaje automático se puede conocer sobre la eficiencia y eficacia estudiando sus distintos parámetros que me permitan conocer de sus funcionalidades y limitaciones de cada uno de ellos.

También, hoy en día es importante conocer sobre los diferentes lenguajes de programación y cuál es el más fácil de entender al momento de programar. Mirjalili y Raschka (2019) afirman que Python es uno de los lenguajes más usados al momento de usar algoritmos de aprendizaje automático, puesto que este lenguaje de programación es uno de los más intuitivos, fácil de manejar y es de código abierto, es decir no necesita de su compra para su uso. Por lo cual es importante conocer sobre los tipos de lenguajes de programación para poder aplicarlo en la realización del proyecto, es por ello que también se usó Python para la realización del mismo, puesto que es uno de los lenguajes de programación con mayores características que benefician en el proceso de desarrollo de los algoritmos por ser

fácil de entender, de código abierto e interpretado e incluso facilitando tiempo en el proceso de ejecución y también en agilizar los procesos que se vayan a realizar.

De modo que, el uso de librerías en el desarrollo de los distintos algoritmos de aprendizaje automático es uno de los implementos más usados por los programadores. Como indica Zamorano (2019) el uso de librerías aplicadas en proyectos donde utilicen algoritmos de aprendizaje automático son muy útiles puesto que ayuda a la agilización de procesos y al ahorro de tiempo en el desarrollo del mismo, una de las librerías más usadas es TensorFlow ya que es un sistema de aprendizaje automático que es usado a gran escala. Esta librería ayuda en los procesos del algoritmo de aprendizaje automático ya que facilita la creación e implementación de estos modelos.

De igual manera, el procesamiento de imágenes es un método usado en varios algoritmos de aprendizaje automático. En este artículo se describe cómo se puede usar el aprendizaje automático para procesar imágenes y clasificar objetos reciclables, en la cual para la realización de estos modelos usaron el lenguaje de programación Python, apoyado con el Framework de backend TensorFlow y la librería Keras, en este estudio se presentó un caso en el que se utiliza el aprendizaje profundo para clasificar diferentes tipos de objetos reciclables, en la cual obtuvieron varias ventajas y desventajas del mismo, así como las posibles aplicaciones futuras de esta tecnología (Salina, Osio, Cappelletti, & Morantes, 2021). De ahí que, con similitud a la propuesta planteada se usaron librerías como Keras, TensorFlow, para el procesamiento de imágenes de las distintas plantas como la ixora, el girasol y la palmera; ya que estas son librerías de alto nivel importante para el procesamiento y obtención de características de una imagen.

Además, este es un conjunto de técnicas aplicadas al mejoramiento de una imagen digital con el fin de mejorarla u obtener características específicas con la finalidad de análisis de la misma. Según Artola (2019) menciona que dentro de la clasificación de imágenes aplicadas en el algoritmo de CNN para identificar objetos el procesamiento de la imagen es de suma importancia ya que en base a esta se trabaja con píxeles de 0 a 225 para la obtención de características únicas de una gran cantidad de imágenes, de este modo se estudian las imágenes a profundidad; el cual tiene distintas etapas que va desde la selección de la imagen hasta la partición de la mismo en base a sus píxeles para el análisis. De este modo en la presente propuesta se ha trabajado con un dataset el cual se han estudiados a profundidad un conjunto de imágenes con la finalidad de obtener características únicas en la cual son aplicadas a cada uno de los algoritmos mostrando un óptimo funcionamiento.

A modo que, tener varios parámetros a estudiar para escoger que algoritmo sería el más óptimo para futuras implementaciones es importante. Como expresan Villalobos-Arias, Quesada-López, Martínez y Jenkins (2021) que existen varias técnicas de ajuste de hiperparámetros, como la validación cruzada, la búsqueda en cuadrícula y la optimización bayesiana, también se discuten distintas métricas usadas para evaluar la eficiencia de estas técnicas, como el error absoluto medio y el coeficiente de determinación, aunque haya una gran cantidad de técnicas de hiperparámetros disponibles, la elección de la técnica más adecuada dependerá de los datos que se vayan a utilizar, el algoritmo y los objetivos del proyecto a realizar. Por tanto, en el presente proyecto se usaron diferentes criterios para el estudio de los algoritmos las cuales son exactitud, sensibilidad, precisión, tiempo de identificación y F1.

Para concluir, la evaluación de los algoritmos forma parte importante para que puedan ser implementados de buena forma por los usuarios o desarrolladores. De forma que Ávila Pérez et.al (2020) mencionan la comparativa de los algoritmos usados en satélites los cuales cumplen con la función de identificar la cobertura boscosa a partir de imágenes satelitales donde utilizaron 4 algoritmos como Minimum distance classification (MDC), Maximum likelihood classification (MLC), Support vector machine (SVM) Y Neural net Classification (NNC), con el objetivo de comparar cuál de estos permite obtener mayor fiabilidad en la identificación de la cobertura boscosa del paisaje situado en Costa Rica, llegando a la conclusión de que los tres algoritmos MLC, SVM y NNC no tienen mayor diferencia pero al contrario de MDC que si tiene una gran diferencia y siendo este el de más baja fiabilidad, se recomienda utilizar cualquiera de los 3 anteriores. Con esto se llegó a la conclusión de lo importante que es medir o evaluar los distintos algoritmos en base a sus funciones para tener trabajos más óptimos y ayudar a futuros proyectos con información necesaria y útil para ser aplicadas en futuros trabajos, a modo que el análisis y estudio de algoritmos se realizan con la finalidad de beneficiar a futuros desarrolladores para aplicaciones de los mismo en un futuro proyectos.

6. Conclusión

En base a la propuesta tecnológica se llevó a cabo este proyecto teniendo como conclusión que:

Al momento de realizar la propuesta se consultó con varias fuentes bibliográficas como son los artículos de revista, tesis, entre otros, para conocer los distintos algoritmos de aprendizaje automático existentes y las métricas a comparar, mediante la revisión bibliográfica se pudo verificar que los algoritmos más utilizados para la identificación son KNN, K-Medias y Redes neuronales convolucionales (CNN), de la misma manera se escogieron 5 métricas a estudiar que consisten en la exactitud, sensibilidad, precisión, F1 y el tiempo de identificación de los algoritmos, así mismo se realizó una entrevista a dueños de viveros para conocer sobre las plantas más vendidas en su locales con la finalidad de determinar cuáles son las plantas que se usaran para la identificación.

Se realizó la programación en un entorno online llamado Google Colab en la cual se usó el lenguaje de programación Python para crear la estructura de los algoritmos, de la misma manera se usaron diversas librerías las cuales fueron OpenCV, TensorFlow, Keras y Sklearn, que ayudaron al entrenamiento del algoritmo, generar los gráficos de las métricas a comparar, la matriz de confusión y que permita la identificación de las plantas a través de la cámara.

Se obtuvieron los datos por medio de la recolección y creación de un dataset a base de toma de fotos de distintas plantas ornamentales las cuales son la ixora, girasol y palmera, esta dataset consta de 1350 imágenes que están divididas en 1200 imágenes para el entrenamiento de los algoritmos y 150 para la validación de estos, de la misma manera se tuvo que hacer una limpieza de imágenes que no servían para el entrenamiento de los algoritmos y no provoque fallos.

A través de la evaluación de los algoritmos de aprendizaje automático en base a parámetros como sensibilidad, exactitud, precisión, F1 y tiempo de identificación se pudo corroborar que dentro de estos 3 algoritmos el más óptimo para la identificación es el CNN en base a las características evaluadas.

7. Recomendación

En base a las conclusiones dadas anteriormente se recomienda:

Trabajar con algoritmos con funcionamiento similar para poder obtener mejores resultados en base a los datos de la evaluación de sus características.

Al momento de la programación utilice un lenguaje entendible y fácil usar de acuerdo a sus capacidades para una mejor ejecución eh entendimiento.

Para la obtención o captura de imágenes se puede trabajar con un dataset ya dados o se los puede crear, de modo que hay que tener que tener en cuenta la calidad de la imagen que va a obtener, las dimanaciones como alto y ancho de la imagen y sus pixeles para un óptimo trabajo.

En cuanto a la evaluación de los algoritmos se debe verificar o probarlos en distintas máquinas y entornos de desarrollo para un mejor análisis, ya que estas constan de varias características y pueden intervenir en la evaluación de los distintos parámetros en especial en el tiempo de identificación de las plantas.

Se recomienda que se use el algoritmo de redes neuronales convolucionales (CNN) para futuras implementaciones ya que gracias a las métricas de estudios comparadas dio como resultado que este algoritmo es el más óptimo.

Teniendo como base esta investigación, y utilizando las redes neuronales convolucionales se podría desarrollar un aplicativo móvil que permita la identificación de plantas que venden los viveros y la ubicación en la cual se podría adquirir.

8. Bibliografía

- Aguilar, J., & Campoverde, M. (2020). Clasificación de frutas basadas en redes neuronales convolucionales. *Polo del conocimiento*, 5(1), 3-22. Obtenido de <https://dialnet.unirioja.es/servlet/articulo?codigo=7436055>
- Álvarez-Risco, A. (2020). *Clasificación de las investigaciones*. Lima: Universidad de Lima, Facultad de Ciencias Empresariales y Económicas, Carrera de Negocios Internacionales. Obtenido de <https://hdl.handle.net/20.500.12724/10818>
- Álvaro, W. R., & Díaz, M. P. (29 de Mayo de 2020). Listado de plantas ornamentales urbanas de Tunja (Boyacá, Colombia). *MUTIS*, 10(1), 57-78. doi:<https://doi.org/10.21789/22561498.1602>
- Amaiquema, F., Vera, J., & Zumba, I. (2019). Enfoques para la formulación de la hipótesis en la investigación científica. *Scielo*. Obtenido de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1990-86442019000500354
- Andrade, H., Sinche, S., & Hidalgo, P. (2021). Modelo para detectar el uso correcto de mascarillas en tiempo real utilizando redes neuronales convolucionales. *Revista de Investigación en Tecnologías de la Información*, 111-120. Retrieved from <https://dialnet.unirioja.es/servlet/articulo?codigo=7741839>
- Aponte, A., Castañeda, E., & Medina, G. (2020). Diseño de un algoritmo para crear y entrenar una red neuronal que permita el reconocimiento de voz. *Universidad Nacional de Piura*, 47. Obtenido de <https://repositorio.unp.edu.pe/bitstream/handle/20.500.12676/2747/IMEC-APO-CAS-MED-2020.pdf?sequence=1&isAllowed=y>

Artola Moreno, Á. (2019). Clasificación de imágenes usando redes neuronales convolucionales en Python . Obtenido de <https://idus.us.es/handle/11441/89506>

Artola, Á. (2019). *Clasificación de imágenes usando redes neuronales convolucionales en Python*. Sevilla . Obtenido de <https://idus.us.es/bitstream/handle/11441/89506/TFG-2402-ARTOLA.pdf?sequence=1&isAllowed=y>

Ávila Pérez, Ortiz Malavassi, Soto Montoya, Vargas Solano, Aguilar Arias, & Miller Granados. (2020). Evaluación de cuatro algoritmos de clasificación de imágenes satelitales Landsat-8 y Sentinel-2 para identificar cobertura forestal en regiones altamente fragmentadas de Costa Rica. *Revista de Teledetección*, 37-49. doi:<https://doi.org/10.4995/raet.2020.13340>

Barbecho, E., & Zhindón, M. (2020). Diseño de un algoritmo de reconocimiento de placas vehiculares ecuatorianas usando redes neuronales convolucionales. *Journal of Science and Research*, 5(4), 76-86. doi:<https://doi.org/10.5281/zenodo.4147629>

Barrero, G. (junio de 2019). Comparación de la precisión estimada de diferentes algoritmos de aprendizaje automático aplicados al análisis de la calidad del Software desarrollado en IBM RPG. *Universidad Piloto de Colombia*. Obtenido de https://www.researchgate.net/publication/333660174_COMPARACION_DE_LA_PRECISION_ESTIMADA_DE_DIFERENTES_ALGORITMOS_DE_APRENDIZAJE_AUTOMATICO_APLICADOS_AL_ANALISIS_DE_LA_CALIDAD_DEL_SOFTWARE_DESARROLLADO_EN_IBM_RPG

- Beluzan, F. (2018). *Diagnóstico de enfermedades causadas por hongos en plantas de palmera y tipo palmera de uso ornamental, en un vivero de la Comunidad Valenciana*. UNIVERSITAT POLITÈCNICA DE VALÈNCIA , DEPARTAMENTO DE ECOSISTEMAS AGROFORESTALES, Valencia. Obtenido de [https://riunet.upv.es/bitstream/handle/10251/113888/Beluz%
c3%a1n%20-%20Diagn%
c3%b3stico%20de%20enfermedades%20causadas%20por%20hongos%20en%20plantas%20de%20palmera%20y%20tipo%20palmera%20de....pdf?sequence=1&isAllowed=y](https://riunet.upv.es/bitstream/handle/10251/113888/Beluz%c3%a1n%20-%20Diagn%c3%b3stico%20de%20enfermedades%20causadas%20por%20hongos%20en%20plantas%20de%20palmera%20y%20tipo%20palmera%20de....pdf?sequence=1&isAllowed=y)
- Bodero, E., Lopez, M., Congacha, A., Cajamarca, E., & Morales, C. (2020). Google Colaboratory como alternativa para el procesamiento de una red neuronal convolucional. *Espacios*, 41(7), 22. Obtenido de <https://www.revistaespacios.com/a20v41n07/20410722.html>
- Calderón, A., & Hurtado, H. (Julio - Diciembre de 2019). Machine learning en a deteccion de enfermedades en plantas. *Revista TIA (Tecnologia, Investigacion y Academia)*, 7(2), 56 - 62. Obtenido de <https://revistas.udistrital.edu.co/index.php/tia/article/view/15685/15932>
- Camacho, H., Bonfante, A., & Aparecida, C. (2020). Selección de documentos basados en centroides para la clasificación de patentes utilizando Word2Vec y KNN. *In Anais do XLVII Seminário Integrado de Software e Hardware*, 269-280. doi:<https://doi.org/10.5753/semish.2020.11335>
- Cao, K., & You, H. (27 de Noviembre de 2020). Fundamental Analysis Via Machine Learning. *SSRN*. Obtenido de https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3706532

- Cárdenas, P., Prado, D., Iglesias, B., Urdiales, R., Orellana, M., & Cedillo, P. (2021). Implementación del algoritmo K-means para clusterización de señales EEG durante la aplicación de una prueba Stroop. *Revista Tecnológica ESPOL*, 33(2), 172-188. doi:<https://doi.org/10.37815/rte.v33n2.847>
- Chanampe, H., Aciar, S., Vega, M., Molinari-Sotomayor, J. L., Carrascosa, G., & Loreface, A. (2019). *Modelo de redes neuronales convolucionales profundas para la clasificación de lesiones en ecografías mamarias*. Red de Universidades con Carreras en Informática. Buenos Aires: SEDICI. Obtenido de <http://sedici.unlp.edu.ar/handle/10915/76941>
- Choez, A. (2022). *dspace.utb.edu.ec/*. Obtenido de [dspace.utb.edu.ec/](http://dspace.utb.edu.ec/bitstream/handle/49000/11594/E-UTB-FAFI-SIST-000281.pdf?sequence=1&isAllowed=y): <http://dspace.utb.edu.ec/bitstream/handle/49000/11594/E-UTB-FAFI-SIST-000281.pdf?sequence=1&isAllowed=y>
- Cofre, M. A. (2020). *Aplicacion movil para la identificacion de las propiedades medicinales de las plantas nativas localizadas en la microempresa "AGROCOTOPAXI"*. Latacunga: Universidad Técnica de Cotopaxi. Obtenido de <http://repositorio.utc.edu.ec/bitstream/27000/6684/1/T-001487.pdf>
- Constitución de la República del Ecuador. (25 de Enero de 2021). *Defensa*. Obtenido de Defensa: https://www.defensa.gob.ec/wp-content/uploads/downloads/2021/02/Constitucion-de-la-Republica-del-Ecuador_act_ene-2021.pdf
- Demaestri, M., & Viale, S. (2018). *Como cultivar tus plantas*. Río Cuarto, Argentina: UniRío editora. doi:ISBN 978-987-688-311-5
- Douglass, M. (2020). Book Review: Hands-on Machine Learning with Scikit-Learn, Keras, and Tensorflow, 2nd edition by Aurélien Géron. *Springer*, 43, 1135-1136. doi:10.1007/s13246-020-00913-z

- Feng, X., Jiang, Y., Yang, X., Du, M., & Li, X. (2019). Computer vision algorithms and hardware implementations: A survey. *ElSevier*, 309-320. doi:<https://doi.org/10.1016/j.vlsi.2019.07.005>
- Franco-Árcega, A., Sobrevilla-Sólis, V., Gutiérrez-Sanchez, M. d., García-Islas, L. H., Suárez-Navarrete, A., & Rueda-Soriano, E. (2021). Sistema de enseñanza para la técnica de agrupamiento k-means. *Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI*, 9, 53-58. doi:<https://doi.org/10.29057/icbi.v9iEspecial.7384>
- Franco-Árcega, A., Sobrevilla-Sólis, V., Gutiérrez-Sánchez, M., García-Islas, L., Suárez-Navarrete, A., & Rueda-Soriano, E. (2021). Sistema de enseñanza para la técnica de agrupamiento k-means. *Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI*, 9(Especial), 53-58. doi:<https://doi.org/10.29057/icbi.v9iEspecial.7384>
- Garrido, A. (2022). Desarrollo de algoritmos para identificar alumnado con problemas de aprendizaje. *Universidad del País Vasco*, 80. Obtenido de https://addi.ehu.es/bitstream/handle/10810/58142/TFG_Andoni_Garrido.pdf?sequence=3&isAllowed=y
- Gobierno Electronico del Ecuador. (2008). www.gobiernoelectronico.gob.ec. Obtenido de www.gobiernoelectronico.gob.ec: <https://www.gobiernoelectronico.gob.ec/software-libre-y-software-publico-2/>
- González, H., & Ticona, U. (2019). Clustering, mediterraneidad y comercio internacional: aplicación empírica de los algoritmos Partitioning Around Medoids y K-means. *Revista Latinoamericana de desarrollo económico*(32), 95-129. Obtenido de http://www.scielo.org.bo/scielo.php?pid=S2074-47062019000200005&script=sci_arttext

- Guailazaca, C., & Hernandez, V. (2020). Clasificador de Productos Agrícolas para Control de Calidad basado en Machine Learning e Industria 4.0. *Revista Perspectivas*, 2(2), 21-28. doi:<https://doi.org/10.47187/perspectivas.vol2iss2.pp21-28.2020>
- Guevara, A. G., Verdesoto, A. A., & Castro, M. N. (2020). Metodologías de investigación educativa (descriptivas, experimentales, participativas, y de investigación-acción). *RECIMUNDO*, 4(3), 163-173. Obtenido de <https://www.recimundo.com/index.php/es/article/view/860/1363>
- Guevara-Fernandez, A., & Coral-Ygnacio, M. (2023). Sistema de recomendación de artículos de línea blanca basado en el algoritmo KNN. *Revista Científica De Sistemas E Informática*, 3(2), 1-18. doi:<https://doi.org/10.51252/rcsi.v3i2.557>
- Gutiérrez, A. (2019). *Metodología para la comparación de algoritmos de aprendizaje automático*. Azcapotzalco: Universidad Autónoma Metropolitana Unidad Azcapotzalco. Obtenido de http://zaloamati.azc.uam.mx/bitstream/handle/11191/6066/Metodologia_para_la_comparacion_de_algoritmos_Gutierrez_Rosales_2019.pdf?sequence=1&isAllowed=y
- Hernández, D., Gil, V., & Robledo, F. (2019). Una aplicación de seguridad electrónica para la autenticación de teléfonos Android basada en el análisis biométrico de la locomoción humana. *Revista INGENIERIA UC*, 26(2), 128-144. Obtenido de <https://www.redalyc.org/articulo.oa?id=70760276002>
- Kakani, V., Nguyen, V. H., Kumar, B. P., Kim, H., & Pasupuleti, V. R. (2020). A critical review on computer vision and artificial intelligence in food industry.

Journal of Agriculture and Food Research, 2, 1-12.
doi:<https://doi.org/10.1016/j.jafr.2020.100033>

Legislación Nacional - Ecuador. (10 de Febrero de 2014). *Gobierno electronico*.
Obtenido de Gobierno electronico:
<https://www.gobiernoelectronico.gob.ec/wp-content/uploads/2018/10/Ley-de-Propiedad-Intelectual.pdf>

Lozada-Portilla, W., Suarez-Barón, M., & Avendaño-Fernández, E. (2021).
Aplicación de redes neuronales convolucionales para la detección del tizón
tardío *Phytophthora infestans* en papa *Solanum tuberosum*. *Revista U.D.C.A
Actualidad & Divulgación Científica*, 24(2), 1-9.
doi:<https://doi.org/10.31910/rudca.v24.n2.2021.1917>

Lujano, E., Lujano, R., Huamani, J. C., & Lujano, A. (2023). Hydrological modeling
based on the KNN algorithm: An application for the forecast of daily flows of
the Ramis river, Peru. *Tecnología y Ciencias del Agua*, 14(2), 169-203.
doi:<https://doi.org/10.24850/j-tyca-14-02-05>

Madariaga, C., Lao, Y., Curra, D., & Martín, R. (2022). Empleo de algoritmos KNN
en metodología multicriterio para la clasificación de clientes, como sustento
de la planeación agregada. *Retos de la Dirección*, 16(1), 178-98. Obtenido
de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2306-91552022000100178

Mayer, M., Cáceres, M., Firman, A., & Vera, L. H. (2018). Desarrollo de algoritmos
de control de un sistema seguidor para la medición de las componentes de
la radiación solar. *EXTENSIONISMO, INNOVACIÓN Y TRANSPARENCIA
TECNOLÓGICA CLAVES PARA EL DESARROLLO*. Obtenido de
<https://revistas.unne.edu.ar/index.php/eitt/article/view/2887>

- Mirjalili, V., & Raschka, S. (2019). *Python Machine Learning*. Marcombo, S.A. Obtenido de https://books.google.com.mx/books?hl=es&lr=&id=5EtOEAAAQBAJ&oi=fnd&pg=PT5&dq=python+para+la+realizacion+de+algoritmos+de+aprendizaje+automatico&ots=eqK5SwUEL1&sig=MC6bUz0dFtyWSINc7_66_U8L2nQ#v=onepage&q&f=false
- Mochida, K., Koda, S., Inoue, K., Hirayama, T., Tanaka, S., Nishii, R., & Melgani, F. (2018). Computer vision-based phenotyping for improvement of plant productivity: a machine learning perspective. *GigaScience*, 8(1), 1-12. doi:<https://doi.org/10.1093/gigascience/giy153>
- Moscoso, V., & Quiñonez, E. (2018). Proyecto integrador de saberes, evidencia del resultado de aprendizaje. *INNOVA*, 3(3), 84-94. Obtenido de <https://dialnet.unirioja.es/servlet/articulo?codigo=6368499>
- Muñoz, I., & Bolt, A. (2021). Diseño y desarrollo de aplicación móvil para la clasificación de flora nativa chilena utilizando redes neuronales convolucionales. *AtoZ: novas práticas em informação e conhecimento*, 1-13. doi:<http://dx.doi.org/10.5380/atoz.v11i0.81419>
- Nicomedes, E. N. (2018). *Tipos de Investigación*. Lima: Universidad Santo Domingo de Guzmán. Obtenido de <http://repositorio.usdg.edu.pe/handle/USDG/34>
- Nizama V, M., & Nizama C, L. M. (2020). El enfoque cualitativo en la investigación jurídica, proyecto de investigación cualitativa y seminario de tesis. 38(2), 69-90. doi:<https://doi.org/10.24265/voxjuris.2020.v38n2.05>
- Núñez, F., Cruz, G., Tolentino, H., Tomás, V., & Felipe, A. (2021). Aplicación del Algoritmo K means para la Tipificación de Trastornos Psicológicos mediante la escala DASS -21 en Alumnos, durante el aislamiento en tiempos de

Covid19. *Ciencia Huasteca Boletín Científico De La Escuela Superior De Huejutla*, 9(17), 12-20. doi:<https://doi.org/10.29057/esh.v9i17.6528>

Núñez, F., Hernández, J. F., Felipe, A. M., & Tomás, V. T. (2019). Aplicación del algoritmo k-means como técnica de minería de datos para determinar el nivel de autoestima en los alumnos universitarios mediante la escala de Rosenberg. *Ciencia Huasteca Boletín Científico De La Escuela Superior De Huejutla*, 7(14), 9-17. doi: <https://doi.org/10.29057/esh.v7i14.4428>

Osman, A., Brunet-Salazar, E., Barreda-Valdés, A., Colás-Sánchez, A., González-Aguilar, D., & Chacón-Iznaga, A. (2018). Efecto de FitoMas-E sobre el crecimiento de *Helianthus annuus* L. cv. CIAP JE- 94 en periodo poco lluvioso. *Scielo*, 45(4). Obtenido de http://scielo.sld.cu/scielo.php?pid=S0253-57852018000400012&script=sci_arttext&tlng=en

Pascasio, J., Mela, R., Vélez, M. L., & Rangel, J. C. (2022). Implementación de redes neuronales para la clasificación de desechos dentro de un cesto inteligente. *Centros: Revista Científica Universitaria*, 11(1), 229-245. Obtenido de <https://revistas.up.ac.pa/index.php/centros/article/view/2542>

Potosi, R. (2019). *Rendimiento de Algoritmos de Clasificación en el Reconocimiento de Actividades de personas*. Universidad tecnica de Norte, Ibarra. Obtenido de <http://repositorio.utn.edu.ec/handle/123456789/9154>

Quintero, C., Merchán, F., Cornejo, A., & Sánchez, J. (2018). Uso de Redes Neuronales Convolucionales para el Reconocimiento Automático de Imágenes de Macroinvertebrados para el Biomonitorio Participativo. *Knowledge E*, 585-596. doi:<https://doi.org/10.18502/keg.v3i1.1462>

- Quiroz, M., Valverde, G., Prieto, J., & Apupalo Del Rosario, L. (Julio - Septiembre de 2020). Detección de personal no autorizado en el departamento de TI utilizando redes neuronales convolucionales en tiempo real con Raspberry Pi 3 B+. *Revista Ciencia e Investigación*, 5(3), 49-60. Obtenido de <https://zenodo.org/record/3926937#.Y1q8TnbMJdg>
- Reyes-Pérez, Luna-Murillo,, Espinoza-Coronel, Lara-Capistrán, Díaz-Rivera, & Hernández-Montiel. (Agosto de 2018). Efecto del sustrato y reguladores del crecimiento sobre el enraizamiento de tres variedades de Ixoras (*Ixora coccinea* L.) en invernadero. *Agro Productividad*, 11(8), 145 - 148. doi:<https://doi.org/10.32854/agrop.v11i8.1110>
- Reyes-Ruiz, L., & Carmona, F. A. (2020). *La investigación documental para la comprensión ontológica del objeto de estudio*. Barranquilla. Obtenido de <https://hdl.handle.net/20.500.12442/6630>
- Rivas, W., & Mazón, B. (2018). *Redes neuronales artificiales aplicadas al reconocimiento de patrones*. Machala: UTMACH. Obtenido de https://www.researchgate.net/profile/Bertha-Mazon-Olivo/publication/327703478_Capitulo_1_Generalidades_de_las_redes_neuronales_artificiales/links/5b9fe3c0299bf13e6038a1d8/Capitulo-1-Generalidades-de-las-redes-neuronales-artificiales.pdf
- Rosales, G., Gutierrez, X., & Hayashida, A. (2022). Aplicación de machine learning para campañas de marketing en la banca comercial. *Interfases*, 187-200. doi: <https://doi.org/10.26439/interfases2022.n016.5953>
- Rubio Coll, M., & Fernández Terricabras, M. (2018). *El Gran libro de las Plantas de Interior*. España: De Vecchi. Obtenido de https://books.google.es/books?hl=es&lr=&id=te1_DwAAQBAJ&oi=fnd&pg=

PT4&dq=plantas+ornamentales++libro+..2018&ots=nxul-
 WNzKF&sig=y4g9Q2KWInr-
 8CmfSk5gBaGXeHQ#v=onepage&q=plantas%20ornamentales%20%20libr
 o%20..2018&f=false

Salina, M., Osio, J. R., Cappelletti, M. A., & Morantes, M. (2021). Aprendizaje automático aplicado al procesamiento de imágenes para la clasificación de objetos reciclables. *XXVII Congreso Argentino de Ciencias de la Computación (CACIC)*, (págs. 31-40). Obtenido de <http://sedici.unlp.edu.ar/handle/10915/130312>

Sánchez, R. (2021). Clasificación no supervisada de imágenes médicas y minería de datos. Algoritmo S3 vs K-medias. *Revista Cubana de Investigaciones Biomédicas*, 40. Obtenido de http://scielo.sld.cu/scielo.php?pid=S0864-03002021000200006&script=sci_arttext&tlng=en

Sancho, F. (14 de Diciembre de 2020). *Fernando Sancho Caparrini*. Obtenido de Fernando Sancho Caparrini: <http://www.cs.us.es/~fsancho/>

Sandoval, L. (19 de Julio de 2018). Algoritmos de aprendizaje automatico para análisis y predicción de datos. *ITCA-FEPADE*, págs. 36-40. Obtenido de http://redicces.org.sv/jspui/bitstream/10972/3626/1/Art6_RT2018.pdf

Sango, B., Chanaluisa, C., & Llano, A. (2021). Sistema inteligente para el análisis de plagas en plantas rosas Freedom aplicando redes neuronales. *RECIMUNDO*, 5(2), 344-357. doi:10.26820/recimundo/5.(2).abril.2021.344-357

Santamaría, L., Zuñiga, S., Pineda, I., Somodevilla, M., & Rossainz, M. (2019). Reconocimiento de genes en secuencias de ADN por medio de imágenes.

- Res. Comput. Sci*, 808-812. Obtenido de https://sci2s.ugr.es/caepia18/proceedings/docs/CAEPIA2018_paper_78.pdf
- Santos, D., Dallos, L., & Gaona-García, P. (2020). Algoritmos de rastreo de movimiento utilizando técnicas de inteligencia artificial y machine learning. *Información Tecnológica*, 31(3), 23-38. doi:<http://dx.doi.org/10.4067/S0718-07642020000300023>
- Sarmiento-Ramos, J. L. (2020). Aplicaciones de las redes neuronales y el deep. *Revista UIS Ingenierías*, 19(4), 1-18. Obtenido de <https://www.redalyc.org/journal/5537/553768213002/553768213002.pdf>
- Serna, E. (2021). *Desarrollo e innovación de ingeniería* (Vol. 2). (6, Ed.) Medellín: Instituto Antioqueño de Investigación. Obtenido de https://www.researchgate.net/profile/Edgar-Duarte-Forero-2/publication/354786561_Configuracion_de_una_red_de_prestadores_de_servicios_en_salud_para_zonas_urbanas_utilizando_herramientas_de_programacion_matematica/links/6152640c154b3227a8b3df43/Configuraci
- Smarandache, F., & Leyva, M. (2019). *Neutrosophic Computing and Machine Learning*. Mexico: University of New Mexico. Obtenido de https://books.google.com.ec/books?id=XNcrEAAAQBAJ&printsec=copyright&hl=es&source=gbs_pub_info_r#v=onepage&q&f=false
- Sugue, J. (2020). *Análisis de algoritmos de detección de características de OpenCV en Raspberry Pi*. Universidad de Sevilla, Escuela Técnica superior de Ingeniería. Sevilla: Departamento de Teoría de la señal y comunicaciones. Obtenido de <https://biblus.us.es/bibing/proyectos/abreproy/93163/fichero/TFG-3163+VICENTE+SUGUE%2C+JAVIER+JES%C3%9AS+DE.pdf>

- Tobar-Díaz, R., Gao, Y., Mas, J., & Cambron-Sandoval, V. (2023). Classification of land use and land cover through machine learning algorithms: a literature review. *Revista Teledetección*, 1-19. doi:<https://doi.org/10.4995/raet.2023.19014>
- Torres, J. (2020). *Python Deep Learning: Introduccion practica con keras t TensorFlow* 2. MARCOMBO. Obtenido de <https://books.google.es/books?hl=es&lr=&id=aExOEAAAQBAJ&oi=fnd&pg=PT14&dq=articulo+cientifico+libreria+keras+..2019&ots=5HM1U70CvB&sig=vrmXe30ifAtbQMGIImm9EeCu2vWg#v=onepage&q&f=false>
- Valarezo, M., Honores, J., Gómez, A., & Vincés, L. (2018). Comparación de tendencias tecnológicas en aplicaciones web. *3C Tecnología. Glosas de Innovación aplicadas a la pyme*, 7(3), 28-49. doi:<http://dx.doi.org/10.17993/3ctecno.2018.v7n3e27.28-49/>
- Vázquez, M., Ibarra, M., Flores, F., & Moreno, H. (2018). Redes neuronales aplicadas al control de riego usando instrumentación y análisis de imágenes para un microinvernadero aplicado al cultivo de Albahaca. *Research in Computing Science*, 93-103. Obtenido de https://www.rcs.cic.ipn.mx/2018_147_5/Redes%20neuronales%20aplicadas%20al%20control%20de%20riego%20usando%20instrumentacion%20y%20analisis%20de%20imagenes.pdf
- Venegas, R. (2021). Aplicaciones de inteligencia artificial para la clasificación automatizada de propósitos comunicativos en informes de ingeniería. *Signos*, 54(107), 942-970. doi:<http://dx.doi.org/10.4067/S0718-09342021000300942>

Villalobos-Arias, L., Quesada-López, C., Martínez, A., & Jenkins, M. (2021).

Técnicas de ajuste de hiperparámetros de algoritmos de aprendizaje automático para la estimación de esfuerzo: un mapeo de literatura. *Revista Ibérica de Sistemas e Tecnologías de Informação*(E42), 305-318. Obtenido de

<https://www.proquest.com/openview/447edae31d1463bb40fc755a5eca13d1/1?pq-origsite=gscholar&cbl=1006393>

Vite, H., Carvajal, H., & Barrezueta, S. (1 de Enero de 2020). Aplicación de

algoritmos de aprendizaje automático para clasificar la fertilidad de un suelo bananero. *Conrado*, págs. 15-19. Obtenido de

http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1990-86442020000100015

Zamorano, J. (2018). Comparativa y Análisis de Algoritmos de Aprendizaje

Automático para la Predicción del Tipo Predominante de Cubierta Arbórea.

E-Prints Complutense, 133. Obtenido de

[https://eprints.ucm.es/id/eprint/48800/1/Memoria%20TFM%20Machine%20Learning_Juan_Zamorano_para_difundir%20\(2\).pdf](https://eprints.ucm.es/id/eprint/48800/1/Memoria%20TFM%20Machine%20Learning_Juan_Zamorano_para_difundir%20(2).pdf)

Zamorano, J. (2018). *Comparativa y análisis de algoritmos de aprendizaje*

automático para la predicción del tipo predominante de cubierta arbórea.

Madrid. Obtenido de

[https://eprints.ucm.es/id/eprint/48800/1/Memoria%20TFM%20Machine%20Learning_Juan_Zamorano_para_difundir%20\(2\).pdf](https://eprints.ucm.es/id/eprint/48800/1/Memoria%20TFM%20Machine%20Learning_Juan_Zamorano_para_difundir%20(2).pdf)

Zamorano, J. (2019). *Comparación y análisis de métodos de clasificación con las*

bibliotecas scikit-learn y TensorFlow en Python. Málaga: Universidad de

Málaga. Obtenido de

<https://riuma.uma.es/xmlui/bitstream/handle/10630/19037/ZamoranoruizjuanMemoria.pdf?sequence=1&isAllowed=y>

Zevallos, A., Onque, S., Canaza, A., & Choqueneira, P. (30 de Septiembre de 2022). Modelo predictivo de la potabilidad del agua mediante un árbol de decisión en Inteligencia Artificial. *Revista Innovación y Software*, 3(2), págs. 121-131. Obtenido de

<https://revistas.ulasalle.edu.pe/innosoft/article/view/72/83>

9. Anexos

9.1 Anexo 1. Presupuesto del proyecto

Tabla 1. Costo de los materiales

N	Ítem	Cantidad	Precio	Total
1	Cámara kodak PIXPRO AZ421 con calidad de 16.15 MP 1/2 .3 CCD	1	\$175,00	\$175,00
2	Equipos de oficina	1	\$5,00	5,00
3	Celular Samsung A51 con versión de Android 12 y RAM de 4gb	1	\$150,00	\$150,00
4	Computadora	1	\$250,00	\$250,00
5	Desarrollo	2	\$200,00	\$400,00
Total				\$980,00

Materiales a utilizar para el desarrollo del proyecto costo estimado a gastar.
Almeida y Durango, 2023

9.2 Anexo 2. Ficha de observación



**UNIVERSIDAD AGRARIA DEL ECUADOR
FACULTAD DE CIENCIAS AGRARIAS
INGENIERIA EN CIENCIAS DE LA COMPUTACIÓN**

Nombre del vivero: Belleza natural

Dirección: Milagro, Vía Naranjito Km 3 ½

Objetivo: Describir los tipos de plantas existentes mediante la observación directa en los viveros para el desarrollo del proyecto.

Ficha de observación

Tabla 2. Ficha de observación de los procesos de identificación de las plantas

Ficha N°	1
Proceso a observar	Identificación de los tipos de plantas existentes en el vivero.
Objetivo	Tamaño de las diferentes plantas. Describir los tipos de plantas existentes mediante la observación directa en los viveros para el desarrollo del proyecto.
Fecha	25/06/2022 14:30:01 p.m.
Aspectos	Hallazgos
Tamaños de plantas más compradas	El tamaño de las plantas más compradas mediante la observación fueron plantas pequeñas y medianas
Plantas más compradas.	Las plantas más compradas fueron la Ixora, las palmas y suculentas.

Tipo de publicidad utilizado para la identificación de vivero.	El tipo de publicidad que usan para la identificación del vivero son las redes sociales como Facebook.
Referencias que siguieron los clientes para llegar al lugar.	La referencia que siguieron los clientes para llegar al lugar fue mediante Facebook ya que ahí estaba la localización del vivero.
Cantidad de clientes que llegaron en el tiempo que duró la observación.	La cantidad de clientes que llegaron al vivero fueron pocas, en total llegaron 2 personas.

Aspectos observados en el tipo y tamaño de las plantas
Almeida y Durango, 2022

Análisis: A través de la observación directa realizada a un viveros que sirvió para la identificación de las plantas, así mismo para saber sobre el tipo, el tamaño y sobre las plantas más vendidas en este lugar, de la misma manera se observaron las plantas que las personas más preguntaban las cuales fueron tres y cada una por su tamaño, la ixora como una planta pequeña, el girasol como la mediana y la palmera como una planta grande, estas tres plantas se tomarán como referencia para realizar la identificación a través de los tres algoritmos de aprendizaje automático.

9.3 Anexo 3. Entrevista



**UNIVERSIDAD AGRARIA DEL ECUADOR
FACULTAD DE CIENCIAS AGRARIAS
INGENIERIA EN CIENCIAS DE LA COMPUTACIÓN**

Entrevistados: Isabel de Carrasco, Washington Paredes y Ana Tenensaray

Entrevistadores: Almeida flores Antonio Lenin y Durango Devis Wilmer Javier

Objetivo: Identificar las plantas más vendidas en los viveros mediante la entrevista para utilizar dentro del estudio comparativo

Preguntas

1. ¿Cuáles son las 3 plantas más solicitadas en el vivero?

Señora Isabel de Carrasco: “Son rosas, palmeras e Ixoras”.

Señor Washington Paredes: “Tengo de todo tipo de plantas, pero las que más se venden son las Ixoras, palmas y cactus”.

Señora Ana Tenensaray: “Las plantas más solicitadas por los clientes son las Ixoras, las palmas y los girasoles”.

2. ¿Cuál es la planta de tamaño pequeña más vendida?

Señora Isabel de Carrasco: “La planta de tamaño pequeño que más se vende es la Ixora”

Señor Washington Paredes: “La planta más vendida en tamaño pequeño son las suculentas”.

Señora Ana Tenensaray: “La planta más vendida en tamaño pequeño es la Ixora”

3. ¿Cuál es la planta de tamaño mediano más vendida?

Señora Isabel de Carrasco: “La palmera es una de las plantas más vendidas en tamaño mediano.”

Señor Washington Paredes:” De tamaño mediano la más vendida es la palma Washingtonia”.

Señora Ana Tenensaray: “La planta de tamaño mediano la más vendida es el girasol”.

4. ¿Cuál es la planta de tamaño grande más vendida?

Señora Isabel de Carrasco: “La palmera también es una de las plantas más vendidas en plantas de tamaño grande”.

Señor Washington Paredes: “La planta más vendida de tamaño grande es la palma fénix”.

Señora Ana Tenensaray: “La planta más vendida de tamaño grande es la palma”.

5. ¿Ha escuchado usted sobre algoritmos de aprendizaje automático para la identificación de plantas y cree que sería importante usarlos?

Señora Isabel de Carrasco: “No he escuchado sobre los algoritmos pero sería importante usar uno de los algoritmos que identifiquen plantas ya que la tecnología cada vez va avanzando”.

Señor Washington Paredes: “Si he escuchado sobre aplicaciones o sitios web que usan ese tipo de algoritmos, yo creo que si sería importante implementar esos algoritmos que ayuden a la identificación de diversas plantas que existen”.

Señora Ana Tenensaray: “Si he escuchado respecto a los algoritmos, yo creo que serían necesarios para la identificación de plantas ya que cada vez la tecnología avanza y las personas se rigen más por lo que dice su teléfono celular o la computadora”.

Análisis de la entrevista: Luego de realizar la entrevista se pudo observar que se venden mucho tipo de plantas en los viveros, pero también se preguntó sobre las plantas más vendidas dependiendo del tamaño en este caso la planta grande que es la palmera, la mediana que es el girasol y la pequeña que es la ixora, de la misma manera se hizo una pregunta sobre si conocían o han escuchado sobre los algoritmos de aprendizaje automático y 2 de los 3 entrevistados dijeron que sí y

que sería de suma importancia implementarlos en aplicativos o sitios web que ayuden a la identificación de las plantas ya que poco a poco la tecnología va avanzando y la gente se va actualizando.

9.4 Anexo 4. Metodología Propuesta



Figura 1. Metodología propuesta para desarrollar los algoritmos. Almeida y Durango, 2023

9.5 Anexo 5. Etapas del Procesamiento de imágenes

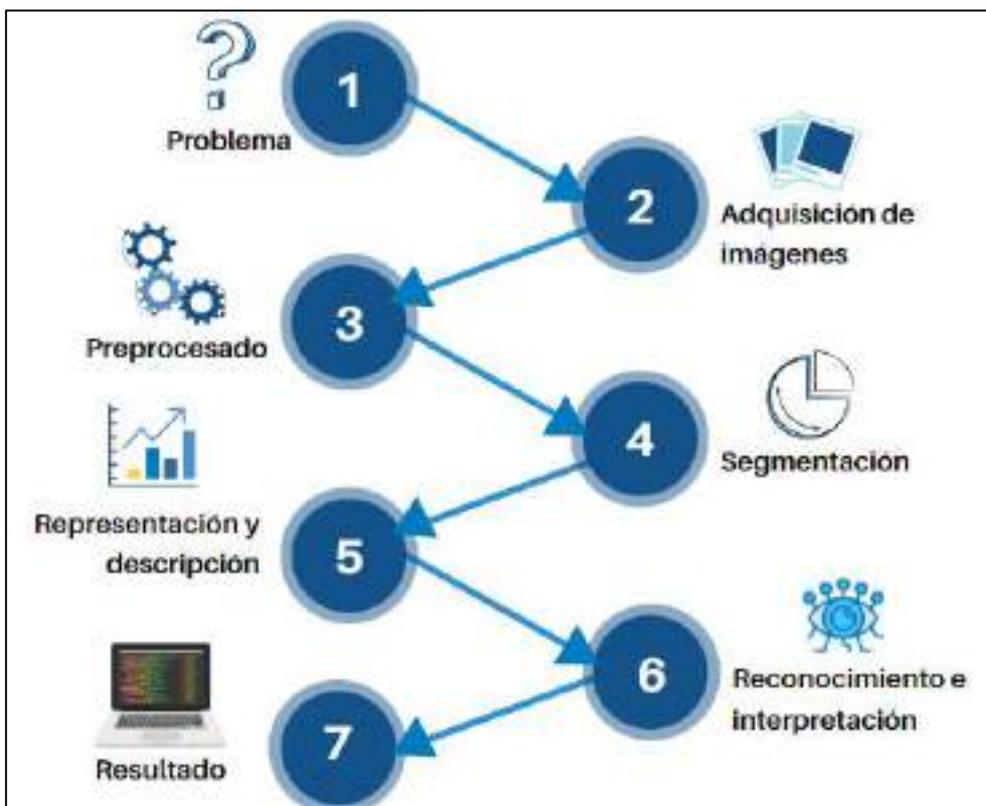


Figura 2. Etapas fundamentales del procesamiento de imágenes. Elaboración propia a partir de la fuente: Galárraga, 2018

9.6 Anexo 6. Funcionamiento de los algoritmos K medias, CNN y KNN

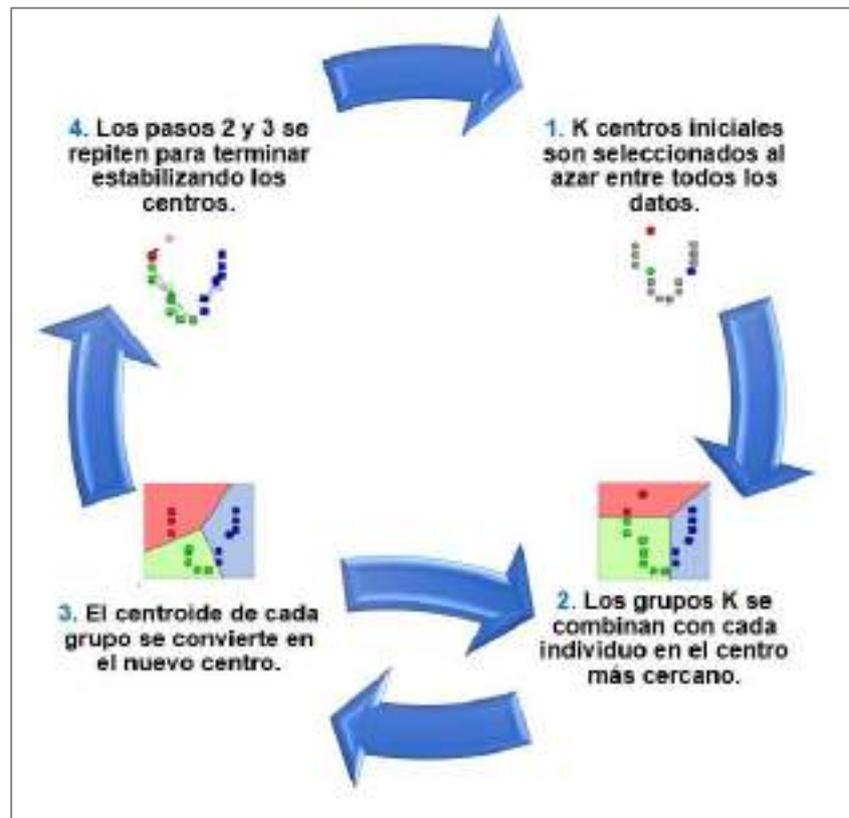


Figura 3. Funcionamiento del algoritmo K means.
Elaboración propia a partir de la fuente: Sancho, 2020

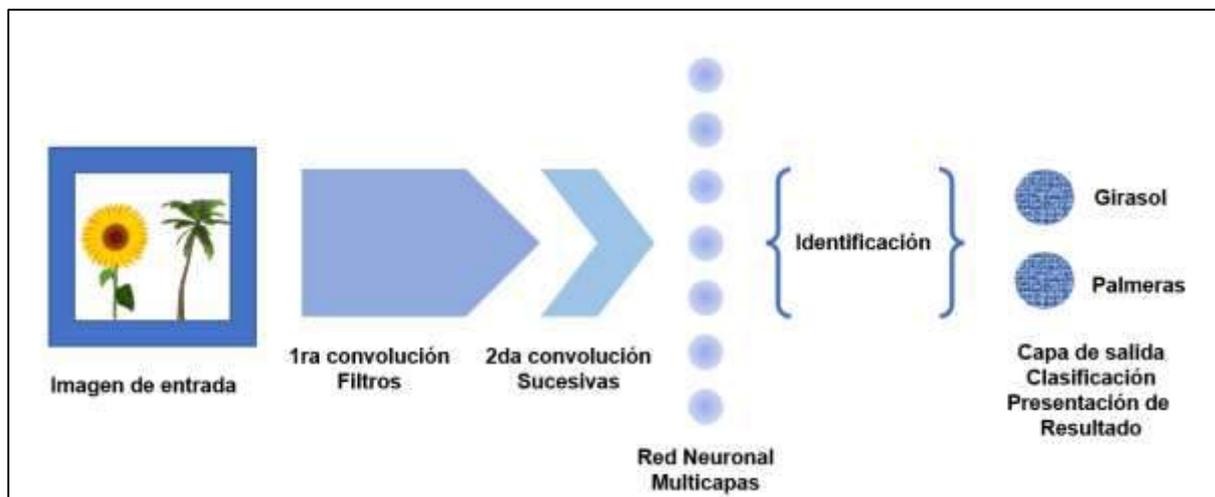


Figura 4. Arquitectura de las fases de una red neuronal convolucional.
Elaboración propia a partir de la fuente: Paredes, 2021

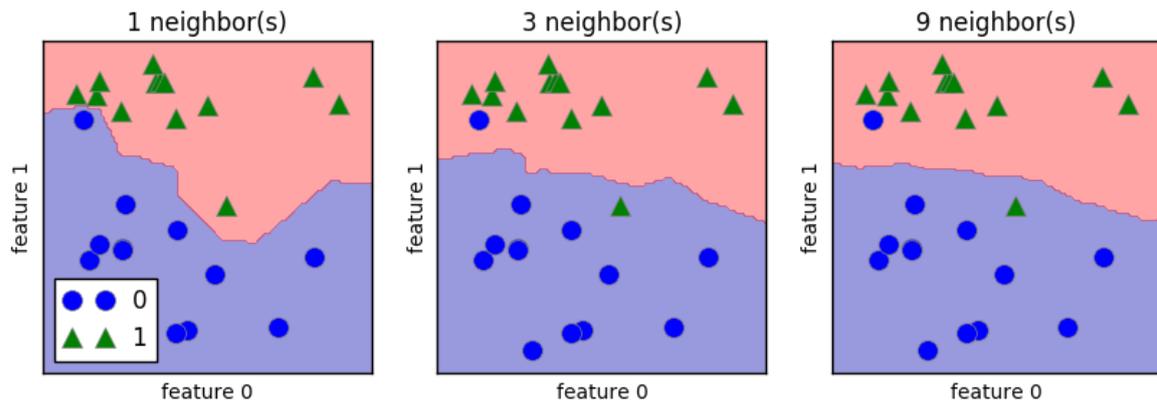


Figura 5. K ejemplos del conjunto de vecinos más cercanos en el algoritmo KNN Sancho, 2020

9.7 Anexo 7. Autores que han usado algoritmos de aprendizaje automático.

En esta tabla se observa diferentes autores que han usado algoritmos de aprendizaje automático para diferentes aplicaciones en la vida diaria y profesional, en este caso se buscaron autores que hayan usado los siguientes algoritmos que son KNN, K-means y CNN que sirvieron para el estudio y análisis de la propuesta tecnológica.

Tabla 3. Diferentes Autores que implementaron algoritmos de aprendizaje automático.

Autores-Año	Titulo	Tipo de documento	Algoritmos utilizados para visión por computador
Lujano E, Lujano R, Huamani y Lujano A (2023)	Modelado hidrológico basado en el algoritmo KNN: una aplicación para el pronóstico de caudales diarios del río Ramis, Perú.	Artículo científico	KNN
Santos, Dallos y Gaona-García (2020)	Algoritmos de rastreo de movimiento utilizando técnicas de inteligencia artificial y machine learning.	Artículo científico	KNN, ORB
Rosales, Gutierrez y Hayashida (2022)	Aplicación de machine learning para campañas de marketing en la banca comercial	Artículo científico	KNN, ANN, naive Bayes, Decision tree
Núñez, Cruz, Tolentino, Tomás y Felipe (2021)	Aplicación del Algoritmo K means para la Tipificación de Trastornos Psicológicos mediante la escala DASS -21 en Alumnos, durante el aislamiento en tiempos de Covid19	Artículo científico	K-means
Venegas (2021)	Aplicaciones de inteligencia artificial para la clasificación automatizada de propósitos comunicativos en informes de ingeniería	Artículo científico	KNN
González y Ticona (2019)	Clustering, mediterraneidad y comercio internacional: aplicación empírica de los algoritmos Partitioning Around Medoids y K-means	Artículo científico	K-means
Hernández, Gil y Robledo (2019)	Una aplicación de seguridad electrónica para la autenticación de teléfonos Android basada en el análisis biométrico de la locomoción humana	Artículo científico	KNN
Cárdenas et al. (2021)	Implementación del algoritmo K-means para clusterización de señales EEG durante la aplicación de una prueba Stroop	Artículo científico	K-means

Autores-Año	Titulo	Tipo de documento	Algoritmos utilizados para visión por computador
Franco-Árcega et al. (2021)	Sistema de enseñanza para la técnica de agrupamiento k-means	Artículo científico	K-means
Brbecho y Zhindón (2020)	Diseño de un algoritmo de reconocimiento de placas vehiculares ecuatorianas usando redes neuronales convolucionales	Artículo científico	CNN
Bodero, Lopez, Congacha, Cajamarca y Morales (2020)	Google Colaboratory como alternativa para el procesamiento de una red neuronal convolucional	Artículo científico	CNN
Camacho, Bonfante y Aparecida (2020)	Selección de documentos basados en centroides para la clasificación de patentes utilizando Word2Vec y KNN	Artículo científico	KNN
Guallazaca y Hernandez (2020)	Clasificador de Productos Agrícolas para Control de Calidad basado en Machine Learning e Industria 4.0	Artículo científico	KNN
Núñez, Hernández, Felipe y Tomás (2019)	Aplicación del algoritmo k-means como técnica de minería de datos para determinar el nivel de autoestima en los alumnos universitarios mediante la escala de Rosenberg	Artículo científico	K-means
Smarandache y Leyva (2019)	Neutrosophic Computing and Machine Learning	Libro	K-means
Sarmiento-Ramos (2020)	Aplicaciones de las redes neuronales y el deep learning a la ingeniería biomédica	Artículo científico	CNN
Lozada-Portilla, Suarez-Barón y Avendaño-Fernández (2021)	Aplicación de redes neuronales convolucionales para la detección del tizón tardío Phytophthora infestans en papa Solanum tuberosum	Artículo científico	CNN

Autores-Año	Titulo	Tipo de documento	Algoritmos utilizados para visión por computador
Quintero, Merchán, Cornejo y Sánchez (2018)	Uso de Redes Neuronales Convolucionales para el Reconocimiento Automático de Imágenes de Macroinvertebrados para el Biomonitorio Participativo	Artículo científico	CNN
Serna (2021)	Desarrollo e Innovación en Ingeniería	Libro	CNN
Pascasio, Mela, Vélez y Rangel (2022)	Implementación de redes neuronales para la clasificación de desechos dentro de un cesto inteligente	Artículo científico	CNN

Artículos científicos y libros sobre la implementación de algoritmos de aprendizaje automático.
Almeida y Durango, 2023

9.8 Anexo 8. Codificación del algoritmo CNN

```
- CONEXION DE LA CARPETA DE GOOGLE DRIVE PARA SER USADA EN COLAB

[ ] from google.colab import drive
   drive.mount('/content/drive')

Mounted at /content/drive

PASO 1: Instalar todos los requerimientos y librerías que serán usadas para correr el
algoritmo.

[ ] #!ENV PYTHON3 and
   git clone https://github.com/ultralytics/yolov5 # clone repo
   cd yolov5
   pip install -qr requirements.txt # install dependencies
   pip install -q roboflow

import torch
import os
from IPython.display import Image, clear_output # to display images

print(f"Setup complete. Using torch {torch.__version__} [{"torch.cuda.get_device_properties(0).name if torch.cuda.is_available() else 'CPU'}]")
```

Figura 6. Conexión con Google Drive para que los archivos alojados en la nube puedan ser usados y permitan la ejecución del algoritmo.

Almeida y Durango, 2023

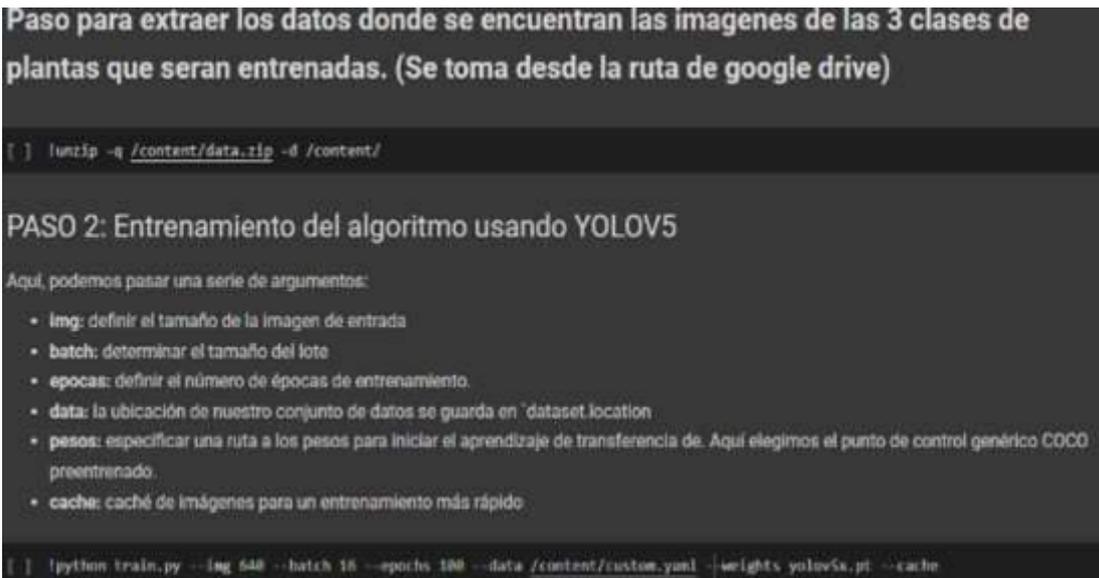


Figura 7. Extracción del archivo data.zip donde se encuentran alojadas las imágenes de los tres tipos de plantas para su identificación y debido a su entrenamiento usando YOLOV5.

Almeida y Durango, 2023

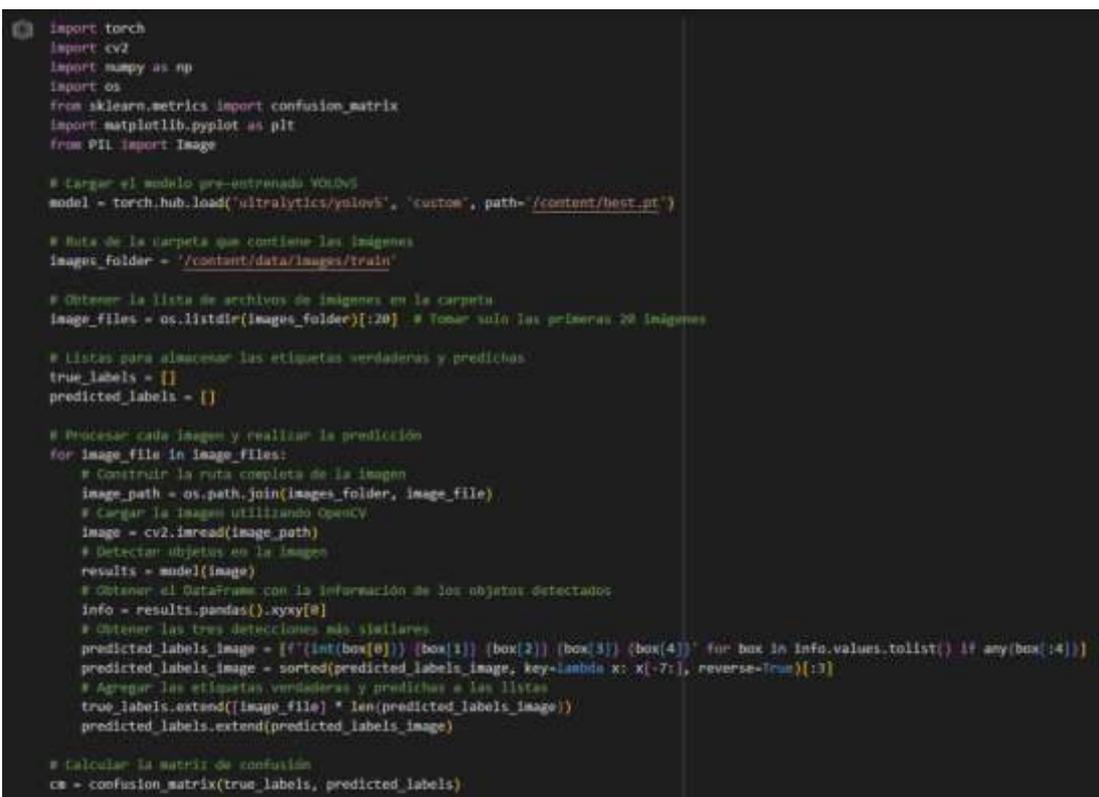


Figura 8. Carga de modelo Pre-entrenado de YOLOV5, importación y ejecución de las librerías para calcular y presentar la matriz de confusión.

Almeida y Durango, 2023

```

# Mostrar la matriz de confusión
plt.figure(figsize=(8, 6))
plt.imshow(cm, interpolation='nearest', cmap='Blues')
plt.title('Matriz de Confusión')
plt.colorbar()
plt.xlabel('Etiqueta Predicha')
plt.ylabel('Etiqueta Real')

# Guardar el gráfico de la matriz de confusión como una imagen
plt.savefig('/content/matriz_confusion.png')

# Cargar la imagen de la matriz de confusión
imagen_confusion = Image.open('/content/matriz_confusion.png')
imagen_confusion.show()

```

Figura 9. Mostrar la matriz de confusión la cual se guardará la imagen para poder ser descargada.

Almeida y Durango, 2023

```

import torch
import cv2
import numpy as np
import os
from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score
import matplotlib.pyplot as plt
from PIL import Image
plt.close("all")

# Cargar el modelo pre-entrenado YOLOv5
model = torch.hub.load('ultralytics/yolov5', 'custom', path='/content/best.pt')

# Ruta de la carpeta que contiene las imágenes
images_folder = '/content/data/images/train'

# Obtener la lista de archivos de imágenes en la carpeta
image_files = os.listdir(images_folder)

# Limitar la cantidad de imágenes a procesar
num_images_to_process = 10
image_files = image_files[:num_images_to_process]

# Listas para almacenar los índices de rendimiento y la matriz de confusión
precisions = []
recalls = []
f1_scores = []
accuracies = []

# Procesar cada imagen y calcular los índices de rendimiento
for image_file in image_files:
    # Construir la ruta completa de la imagen
    image_path = os.path.join(images_folder, image_file)
    # Cargar la imagen utilizando OpenCV
    image = cv2.imread(image_path)
    # Detectar objetos en la imagen
    results = model(image)
    # Obtener el DataFrame con la información de los objetos detectados
    info = results.pandas().xyxy[0]
    # Obtener las tres detecciones más similares
    predicted_labels = [f'{int(box[0])} {box[1]} {box[2]} {box[3]} {box[4]}' for box in info.values.tolist() if any(box[:4])]
    predicted_labels = sorted(predicted_labels, key=lambda x: x[-7], reverse=True)[:3]

```

Figura 10. Importar las librerías numpy, sklearn y matplotlib para calcular las métricas de medición del algoritmo y poderlas presentar mediante un gráfico.

Almeida y Durango, 2023

```

# Calcular los índices de rendimiento para esta imagen
precision = precision_score(predicted_labels, predicted_labels, average='micro')
recall = recall_score(predicted_labels, predicted_labels, average='micro')
f1_score_value = f1_score(predicted_labels, predicted_labels, average='micro')

# Calcular la exactitud
true_labels = [f'{int(box[0])} {box[1]} {box[2]} {box[3]} {box[4]}' for box in info.values.tolist()]
accuracy = accuracy_score(true_labels, predicted_labels)

# Agregar los índices de rendimiento y la exactitud a las listas
precisions.append(precision)
recalls.append(recall)
f1_scores.append(f1_score_value)
accuracies.append(accuracy)

# Calcular los índices de rendimiento promedio
average_precision = np.nanmean(precisions)
average_recall = np.nanmean(recalls)
average_f1_score = np.nanmean(f1_scores)
average_accuracy = np.nanmean(accuracies)

average_precision = max(0.0, min(1.0, round(average_precision, 1)))
average_recall = max(0.0, min(1.0, round(average_recall, 1)))
average_f1_score = max(0.0, min(1.0, round(average_f1_score, 1)))
average_accuracy = max(0.0, min(1.0, round(average_accuracy, 1)))

# Imprimir los resultados
print(f'Precisión promedio: {average_precision:.4f}')
print(f'Sensibilidad promedio: {average_recall:.4f}')
print(f'F1-Score promedio: {average_f1_score:.4f}')
print(f'Exactitud promedio: {average_accuracy:.4f}')

# Crear el gráfico de barras con los índices de rendimiento promedio

# Crear una lista con los nombres de los índices
indices = ['Precisión', 'Sensibilidad', 'F1-Score', 'Exactitud']

# Crear una lista con los valores promedio de los índices
valores = [average_precision, average_recall, average_f1_score, average_accuracy]

```

Figura 11. Impresión de los resultados de los parámetros calculados.
Almeida y Durango, 2023

```

# Crear un gráfico de barras
plt.bar(indices, valores)
# Agregar los valores encima de cada barra en el nuevo gráfico utilizando annotate
for i, v in enumerate(valores):
    plt.annotate(f'{v:.4f}', xy=(i, v), xytext=(i, v + 0.01), ha='center', va='bottom')

# Guardar el gráfico como una imagen
plt.savefig('/content/indices_rendimiento.png')

# Ruta de la imagen guardada
image_path = '/content/indices_rendimiento.png'

img = Image.open(image_path)
img = img.convert('RGB')
img

```

Figura 12. Crear grafica de barras de los parámetros calculados y que permita guardar la imagen para su descarga.
Almeida y Durango

```

import torch
import cv2
import numpy as np
import pandas as pd
from google.colab.patches import cv2_imshow
from IPython.display import display, Javascript
from google.colab.output import eval_js
import base64
import sys

# LECTURA DEL MODELO
model = torch.hub.load('ultralytics/yolov5', 'custom', path='/content/best.pt')

def capture_frame():
    js = Javascript('''
        async function captureFrame() {
            const div = document.createElement('div');
            const video = document.createElement('video');
            video.style.display = 'block';
            const stream = await navigator.mediaDevices.getUserMedia({video: true});
            document.body.appendChild(div);
            div.appendChild(video);
            video.srcObject = stream;
            await video.play();
            div.appendChild(document.createElement('hr'));
            const canvas = document.createElement('canvas');
            canvas.style.display = 'block';
            div.appendChild(canvas);
            canvas.width = video.videoWidth;
            canvas.height = video.videoHeight;
            canvas.getContext('2d').drawImage(video, 0, 0);
            stream.getVideoTracks()[0].stop();
            div.remove();
            return canvas.toDataURL('image/jpeg');
        }
    ''')
    display(js)
    data = eval_js('captureFrame()')
    binary = base64.b64decode(data.split(',')[1])
    image = np.asarray(bytearray(binary), dtype='uint8')
    frame = cv2.imdecode(image, cv2.IMREAD_COLOR)
    return frame

```

Figura 13. Código para la detección de la imagen.
Almeida y Durango, 2023

```

while True:
    # Capturar un frame de la cámara
    frame = '/content/ixora.png' # se puede probar mejor subiendo una img y poner directamente la ruta '/content/photo.png'

    # Detectar objetos en el frame
    results = model(frame)

    # Obtener el Dataframe con la información de los objetos detectados
    info = results.pandas().xyxy[0]
    print(info)

    # Mostrar el frame con los objetos detectados
    cv2_imshow(np.squeeze(results.render()))

    # LECTURA DE TECLADO
    break

cv2.destroyAllWindows()

```

Figura 14. Código para la captura de la imagen y la clasificación sobre el tipo de planta mostrada.
Almeida y Durango, 2023

9.9 Anexo 9. Matriz de confusión del algoritmo CNN

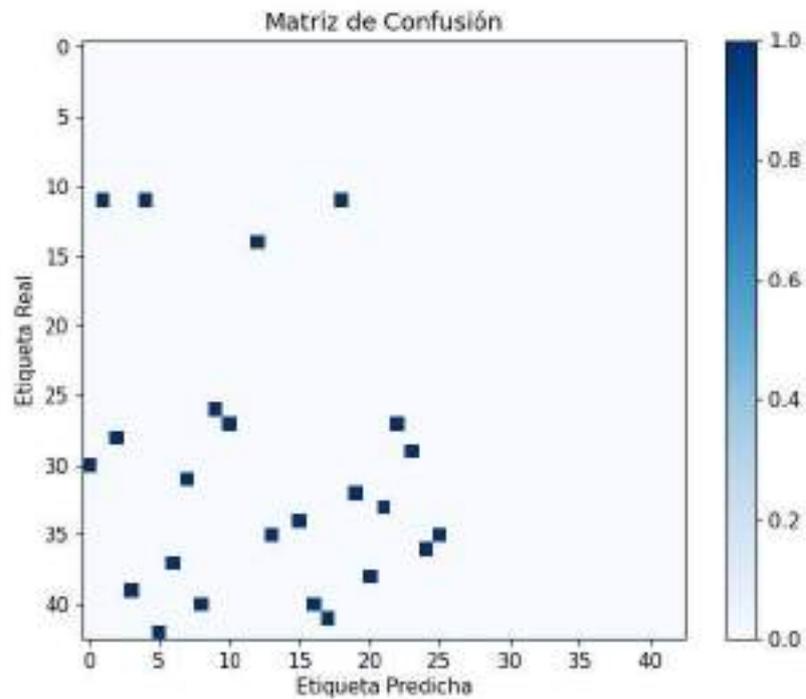


Figura 15. Matriz de confusión del algoritmo CNN
Almeida y Durango, 2023

9.10 Anexo 10. Métricas de estudio del algoritmo CNN

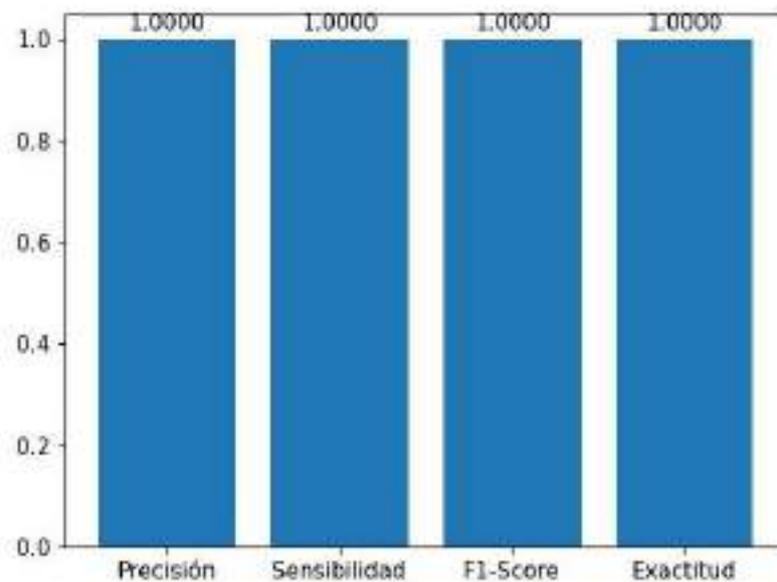
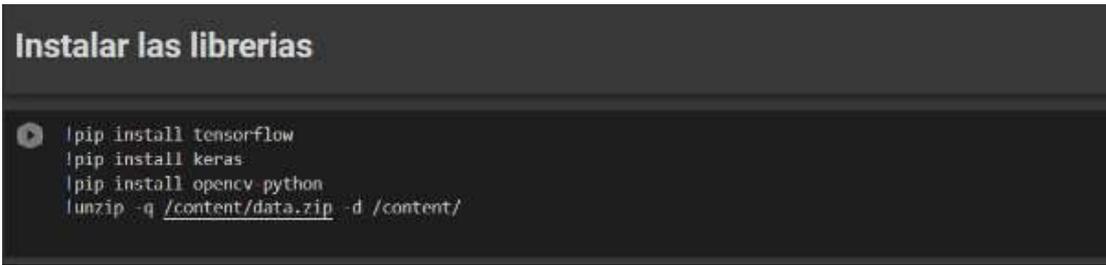


Figura 16. Métricas de medición algoritmo CNN
Almeida y Durango, 2023

9.11 Anexo 11. Codificación del algoritmo KNN



```
Instalar las librerías

!pip install tensorflow
!pip install keras
!pip install opencv-python
!unzip -q /content/data.zip -d /content/
```

Figura 17. Instalación de librerías para el funcionamiento del algoritmo KNN. Almeida y Durango, 2023



```
+ Cargar el dataset

! from google.colab import files

# Carga el archivo ZIP que contiene el dataset
uploaded = files.upload()

# Extrae el contenido del archivo ZIP
import zipfile
import io
zf = zipfile.ZipFile(io.BytesIO(uploaded['data.zip']), 'r')
zf.extractall()

# Note: upload widget Upload widget is only available when the colab has been executed in the current browser session. Please rerun this cell to enable
using data.zip to data.zip
```

Figura 18. Carga y extracción del dataset de imágenes de las plantas para el entrenamiento del algoritmo. Almeida y Durango, 2023

· Código knn

```

import cv2
import numpy as np
import os
from keras.applications.resnet_v2 import ResNet50V2, preprocess_input
from keras.models import Model
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix
import matplotlib.pyplot as plt

# Cargar modelo ResNet50V2 preentrenado
base_model = ResNet50V2(weights='imagenet')
model = Model(inputs=base_model.input, outputs=base_model.get_layer('avg_pool').output)

# Configuración del clasificador KNN
k = 3
knn = KNeighborsClassifier(n_neighbors=k)

# Preprocesamiento y extracción de características de imágenes de entrenamiento
train_dir = 'data/images/train/'
train_files = os.listdir(train_dir)
train_features = []
train_labels = []
for file in train_files:
    label = file.split('-')[0] # Obtener etiqueta a partir del nombre de archivo
    img = cv2.imread(os.path.join(train_dir, file))
    img = cv2.resize(img, (224, 224))
    img = img.astype('float32')
    img /= 255
    img = np.expand_dims(img, axis=0)
    img = preprocess_input(img)
    features = model.predict(img)
    features = features.reshape(features.shape[0], -1)
    train_features.append(features)
    train_labels.append(label)
train_features = np.vstack(train_features)

```

Figura 19. . Ejecución y entrenamiento del algoritmo KNN por medio de las imágenes extraídas del dataset y presentar las gráficas de la matriz de confusión y métricas calculadas por medio de la librería llamada sklearn. Aleida y Durango, 2023

```

# Entrenamiento de KNN
knn.fit(train_features, train_labels)

# Clasificación de imágenes de validación
val_dir = 'data/images/val/'
val_files = os.listdir(val_dir)
true_labels = []
predicted_labels = []
for file in val_files:
    label = file.split('-')[0] # Obtener etiqueta a partir del nombre de archivo
    true_labels.append(label)
    img = cv2.imread(os.path.join(val_dir, file))
    img = cv2.resize(img, (224, 224))
    img = img.astype('float32')
    img /= 255
    img = np.expand_dims(img, axis=0)
    img = preprocess_input(img)
    features = model.predict(img)
    features = features.reshape(features.shape[0], -1)
    prediction = knn.predict(features)[0]
    predicted_labels.append(prediction)
    with open(os.path.join('data/labels/val/', os.path.splitext(file)[0] + '.txt'), 'w') as f:
        f.write(prediction)

# Filtrar las etiquetas seleccionadas en los datos verdaderos y predichos
selected_labels = ['flores', 'sunflower', 'sunflowers']
selected_labelss = ['0', '1', '2']
filtered_true_labels = []
filtered_predicted_labels = []
for true_label, predicted_label in zip(true_labels, predicted_labels):
    if true_label in selected_labels and predicted_label in selected_labelss:
        filtered_true_labels.append(true_label)
        filtered_predicted_labels.append(predicted_label)

# Calcular la matriz de confusión con las etiquetas filtradas
cm = confusion_matrix(filtered_true_labels, filtered_predicted_labels, labels=selected_labels)

# Configuración de la figura
plt.figure(figsize=(8, 6))

```

Figura 20. Validación y clasificación de las imágenes obtenidas del dataset para el entrenamiento del algoritmo obtenidas a través de etiquetas a partir del nombre de la planta.

Almeida y Durango, 2023

```

# Muestra la matriz de confusión
plt.imshow(cm, interpolation='nearest', cmap=plt.cm.Blues)
plt.title('Matriz de Confusión')
plt.colorbar()

# Añadir etiquetas a los ejes
tick_marks = np.arange(len(selected_labels))
plt.xticks(tick_marks, selected_labels, rotation=45)
plt.yticks(tick_marks, selected_labels)

# Añadir anotaciones en los cuadros
thresh = cm.max() / 2.
for i in range(cm.shape[0]):
    for j in range(cm.shape[1]):
        plt.text(j, i, format(cm[i, j], 'd'), ha="center", va="center", color="white" if cm[i, j] > thresh else "black")

# Ajustar los márgenes de la figura
plt.tight_layout()

# Muestra la figura
plt.show()

# Cálculo de métricas
accuracy = accuracy_score(true_labels, predicted_labels)
precision = precision_score(true_labels, predicted_labels, average='weighted')
recall = recall_score(true_labels, predicted_labels, average='weighted')
f1 = f1_score(true_labels, predicted_labels, average='weighted')

# Gráficas de métricas
labels = ['Exactitud', 'Precisión', 'Sensibilidad (Recall)', 'F1-score']
metrics = [accuracy, precision, recall, f1]

plt.bar(labels, metrics)
plt.ylabel('Valor')
plt.title('Métricas del clasificador VNN')

# Agregar los valores encima de cada barra
for i, v in enumerate(metrics):
    plt.text(i, v, str(round(v, 2)), ha='center', va='bottom')

plt.show()

```

Figura 21. Presentación de grafica de la matriz de confusión y las métricas evaluadas del algoritmo las cuales son la exactitud, sensibilidad, precisión y F1.

Almeida y Durango, 2023

```

display(js)
data = eval_js('takePhoto({})'.format(quality))
binary = base64.b64decode(data.split(',')[1])
with open(filename, 'wb') as f:
    f.write(binary)
return filename

# Función para convertir la respuesta de JavaScript a una imagen OpenCV

def js_to_image(filename):
    img = cv2.imread(filename)
    return img

# Capturar una imagen de la cámara del navegador
filename = 'photo.jpg'
data_url = take_photo(filename)

# Convertir la imagen a formato OpenCV
img = js_to_image(data_url)

# Mostrar la imagen
cv2.imshow('img')

# Preprocesamiento y extracción de características de la imagen
img = cv2.resize(img, (224, 224))
img = img.astype('float32')
img /= 255
img = np.expand_dims(img, axis=0)
img = preprocess_input(img)
features = model.predict(img)
features = features.reshape(features.shape[0], -1)

```

Figura 22. Captura de la imagen para convertirla en formato OpenCV y pueda ser identificada por el algoritmo.
Almeida y Durango, 2023

```

# Clasificación con KNN
prediction = knn.predict(features)[0]

# Imprimir resultado
if prediction == 'ixora':
    plant_name = 'Ixora'
elif prediction == 'sunflower' or prediction == 'sunflowers':
    plant_name = 'Girasol'
elif prediction == 'palm':
    plant_name = 'Palmera'
else:
    plant_name = 'No identificado'

print("La imagen pertenece a la planta:", plant_name)

# Liberar cámara y destruir ventanas
cv2.destroyAllWindows()

```

Figura 23. Presentar el tipo de planta que está identificando por medio de la imagen tomada.
Almeida y Durango, 2023

9.12 Anexo 12. Matriz de confusión del algoritmo de KNN

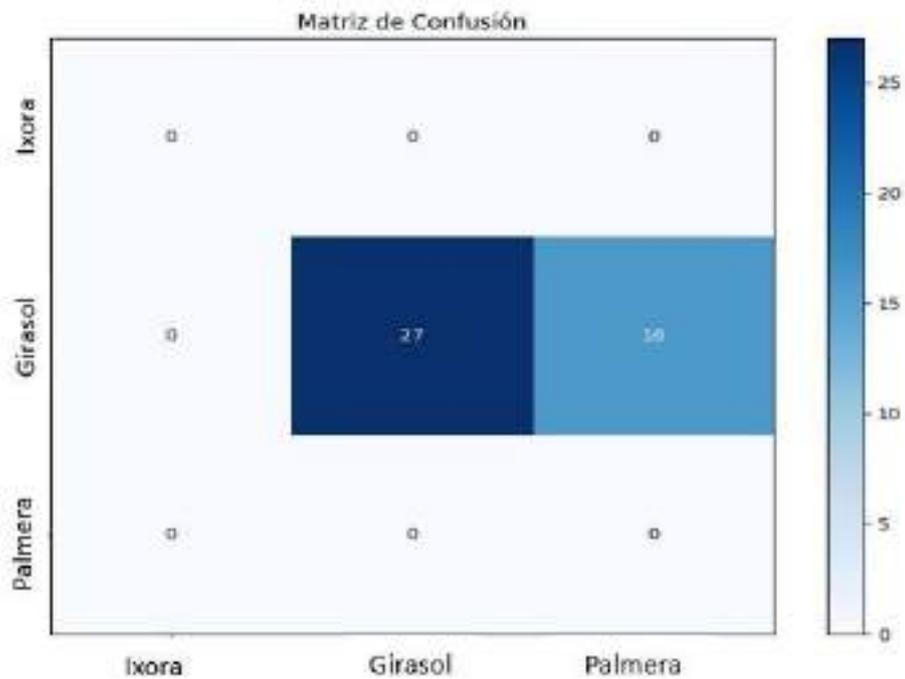


Figura 24. Matriz de confusión del algoritmo de KNN
Almeida y Durango, 2023

9.13 Anexo 13. Métricas de estudio del algoritmo KNN

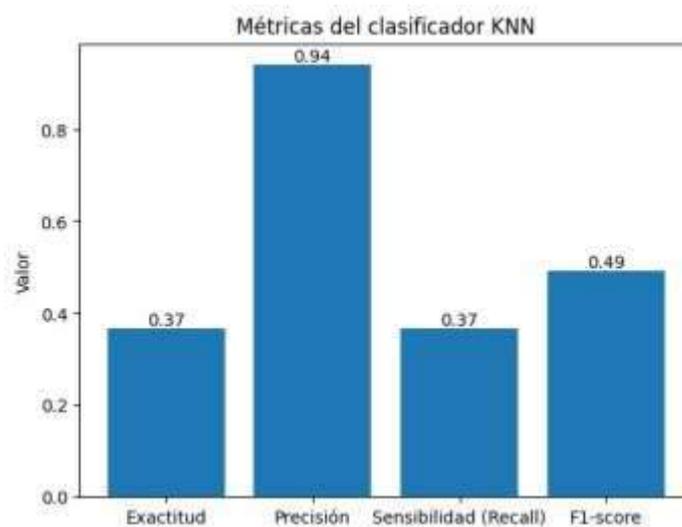


Figura 25. Métricas de medición algoritmo KNN.
Almeida y Durango, 2023

9.14 Anexo 14. Codificación del algoritmo K-Means

```
✓ [1] !pip install tensorflow
      !pip install keras
      !pip install opencv-python
```

Figura 26. Instalación de librerías para el funcionamiento del algoritmo K-Means.

Almeida y Durango, 2023

```
✓ [2] from google.colab import files
      # Carga el archivo ZIP que contiene el dataset
      uploaded = files.upload()

      # Extrae el contenido del archivo ZIP
      import zipfile
      import io
      zf = zipfile.ZipFile(io.BytesIO(uploaded['data.zip']), "r")
      zf.extractall()
```

Figura 27. Carga y extracción del dataset de imágenes de las plantas para el entrenamiento del algoritmo.

Almeida y Durango, 2023

- Código kmeans

```

import cv2
import numpy as np
import os
from keras.applications.vgg16 import preprocess_input
from keras.applications.vgg16 import VGG16
from keras.models import Model
from sklearn.cluster import KMeans
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix
from sklearn.preprocessing import LabelEncoder
import matplotlib.pyplot as plt
import seaborn as sns

# Cargar modelo VGG16 preentrenado
base_model = VGG16(weights='imagenet')
model = Model(inputs=base_model.input, outputs=base_model.get_layer('fc1').output)

# Configuración de K-Medias
kmeans = KMeans(n_clusters=3, init='k-means++', random_state=42)

# Preprocesamiento y extracción de características de imágenes de entrenamiento
train_dir = 'data/images/train/'
train_files = os.listdir(train_dir)
train_features = []
train_labels = []
for file in train_files:
    label = file.split('-')[0] # Obtener etiqueta a partir del nombre de archivo
    img = cv2.imread(train_dir + file)
    img = cv2.resize(img, (224, 224))
    img = img.astype('float32')
    img /= 255
    img = np.expand_dims(img, axis=0)
    img = preprocess_input(img)
    features = model.predict(img)
    features = features.reshape(features.shape[0], -1)
    train_features.append(features)
    train_labels.append(label)
train_features = np.vstack(train_features)

```

Figura 28. Ejecución y entrenamiento del algoritmo K-Means por medio de las imágenes extraídas del dataset y presentar las gráficas de la matriz de confusión y métricas calculadas por medio de la librería llamada sklearn.

Almeida y Durango, 2023

```

# Entrenamiento de K-Medias
kmeans.fit(train_features)

# Clasificación de imágenes de validación
val_dir = 'data/images/val/'
val_files = os.listdir(val_dir)
true_labels = []
predicted_labels = []
for file in val_files:
    label = file.split('.')[0] # Obtener etiqueta a partir del nombre de archivo
    true_labels.append(label)
    img = cv2.imread(val_dir + file)
    img = cv2.resize(img, (224, 224))
    img = img.astype('float32')
    img /= 255
    img = np.expand_dims(img, axis=0)
    img = preprocess_input(img)
    features = model.predict(img)
    features = features.reshape(features.shape[0], -1)
    prediction = kmeans.predict(features)[0]
    if prediction == 0:
        plant_name = 'icora'
    elif prediction == 1:
        plant_name = 'sunflower'
    else:
        plant_name = 'palm'
    predicted_labels.append(plant_name)
    with open('data/labels/val/' + os.path.splitext(file)[0] + '.txt', 'w') as f:
        f.write(plant_name)

# Codificación de etiquetas
label_encoder = LabelEncoder()
label_encoder.fit(true_labels)
true_labels_encoded = label_encoder.transform(true_labels)
predicted_labels_encoded = label_encoder.transform(predicted_labels)

# Cálculo de métricas
accuracy = accuracy_score(true_labels_encoded, predicted_labels_encoded)
precision = precision_score(true_labels_encoded, predicted_labels_encoded, average='weighted')
recall = recall_score(true_labels_encoded, predicted_labels_encoded, average='weighted')
f1 = f1_score(true_labels_encoded, predicted_labels_encoded, average='weighted')

```

Figura 29. Clasificación de las imágenes, validación, y presentación de grafica de la matriz de confusión y las métricas evaluadas del algoritmo las cuales son la exactitud, sensibilidad, precisión y F1. Almeida y Durango, 2023

```

# Matriz de confusión
cm = confusion_matrix(true_labels_encoded, predicted_labels_encoded)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title('Matriz de Confusión')
plt.xlabel('Etiqueta Predicha')
plt.ylabel('Etiqueta Real')
plt.show()

# Gráficas de métricas
labels = ['Exactitud', 'Precisión', 'Sensibilidad (Recall)', 'F1-score']
metrics = [accuracy, precision, recall, f1]

plt.bar(labels, metrics)
plt.ylabel('Valor')
plt.title('Métricas del clasificador K-Medias')

# Agregar los valores encima de cada barra
for i, v in enumerate(metrics):
    plt.text(i, v, str(round(v, 2)), ha='center', va='bottom')

plt.show()

```

Figura 30. Presentación de grafica de la matriz de confusión y las métricas evaluadas del algoritmo las cuales son la exactitud, sensibilidad, precisión y F1.

Almeida y durango, 2023

```

[ ] # Importar las librerías necesarias
import io
import base64
import numpy as np
import cv2
from google.colab.patches import cv2_imshow
from IPython.display import display, Javascript
from google.colab.output import eval_js
from tensorflow.keras.applications.resnet50 import preprocess_input

# Luego puedes utilizar la función preprocess_input() en tus imágenes

# Función para capturar una imagen de la cámara del navegador
def take_photo(filename='photo.jpg', quality=0.8):
    js = Javascript('''
        async function takePhoto(quality) {
            const div = document.createElement('div');
            const video = document.createElement('video');
            video.style.display = 'block';
            const stream = await navigator.mediaDevices.getUserMedia({video: true});
            document.body.appendChild(div);
            div.appendChild(video);
            video.srcObject = stream;
            await video.play();
            div.appendChild(document.createElement('hr'));
            const canvas = document.createElement('canvas');
            canvas.style.display = 'block';
            div.appendChild(canvas);
            canvas.width = video.videoWidth;
            canvas.height = video.videoHeight;
            canvas.getContext('2d').drawImage(video, 0, 0);
            stream.getVideoTracks()[0].stop();
            div.remove();
            return canvas.toDataURL('image/jpeg', quality);
        }
    ''')

```

Figura 31. Importar las librerías necesarias que permitan abrir la cámara y la captura de la imagen a través de la cámara del navegador.

Almeida y Durango, 2023

```
display(js)
data = eval_js('takePhoto({})'.format(quality))
binary = base64.b64decode(data.split(',')[1])
with open(filename, 'wb') as f:
    f.write(binary)
return filename

# Función para convertir la respuesta de JavaScript a una imagen OpenCV
def js_to_image(filename):
    img = cv2.imread(filename)
    return img

# Capturar una imagen de la cámara del navegador
filename = take_photo()

# Cargar la imagen capturada en OpenCV
img = cv2.imread(filename)

# Mostrar la imagen
cv2_imshow(img)

# Convertir la imagen a escala de grises
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Mostrar la imagen en escala de grises
cv2_imshow(gray)

# Aplicar el filtro Canny para detectar bordes
edges = cv2.Canny(gray, 100, 200)

# Mostrar la imagen con bordes detectados
cv2_imshow(edges)

data = take_photo()
# get OpenCV format image
frame = js_to_image(data)

print(frame.shape)
```

Figura 32. Cambiar la imagen a escalas de grises y aplicar la detección por bordes para la identificación de la planta.
Almeida y Durango, 2023

```

# Preprocesamiento y extracción de características de la imagen
img = cv2.resize(frame, (224, 224))
img = img.astype('float32')
img /= 255
img = np.expand_dims(img, axis=0)
img = preprocess_input(img)
features = model.predict(img)
features = features.reshape(features.shape[0], -1)

# Clasificación de la imagen
prediction = kmeans.predict(features)[0]

if prediction == 0:
    plant_name = 'Es una Ixora'
elif prediction == 1:
    plant_name = 'Es un Girasol'
else:
    plant_name = 'Es una Palmera'
print('Clasificación '+plant_name)

# Liberar cámara y destruir ventanas
cv2.destroyAllWindows()

```

Figura 33. Presentar el tipo de planta que está identificando por medio de la imagen capturada, tomando en cuenta las características de la imagen como su color y su forma.
Almeida y Durango, 2023

9.15 Anexo 15. Matriz de confusión del algoritmo K-Means

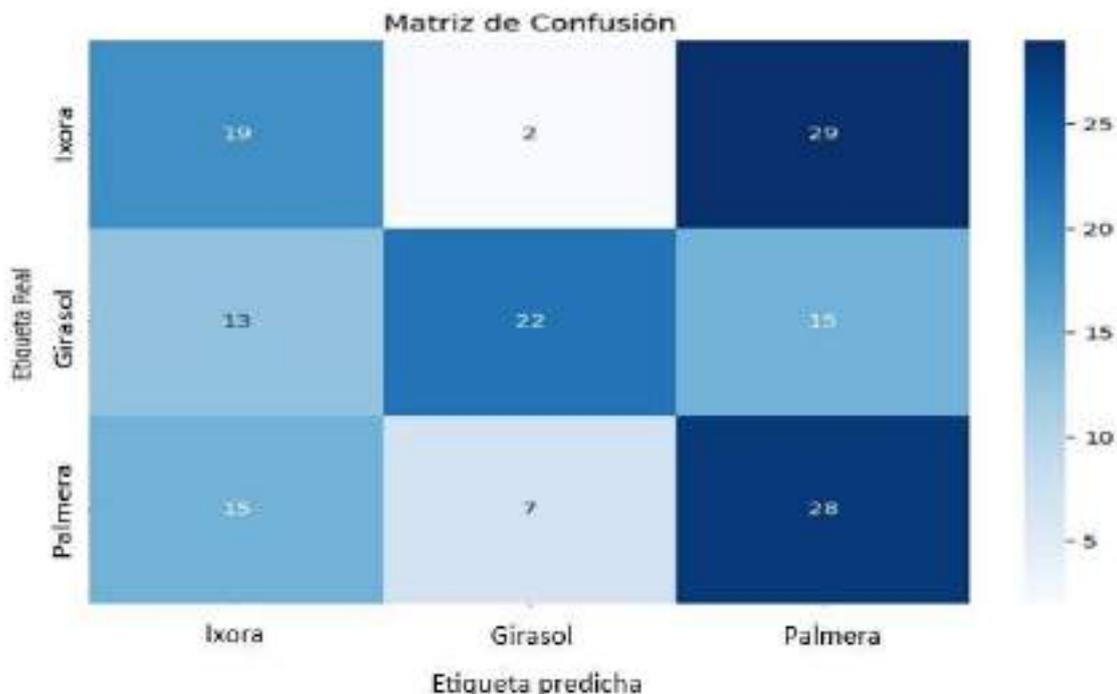


Figura 34. Matriz de confusión del algoritmo de K-means
Almeida y Durango, 2023

9.16 Anexo 16. Métricas de estudio del algoritmo K-Medias

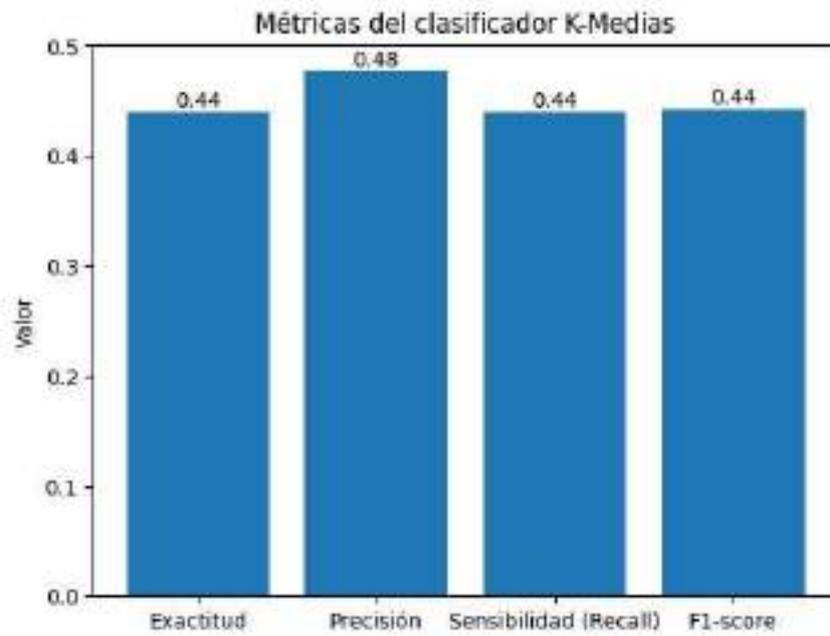


Figura 35. Métricas de medición algoritmo K-Means
Almeida y Durango, 2023

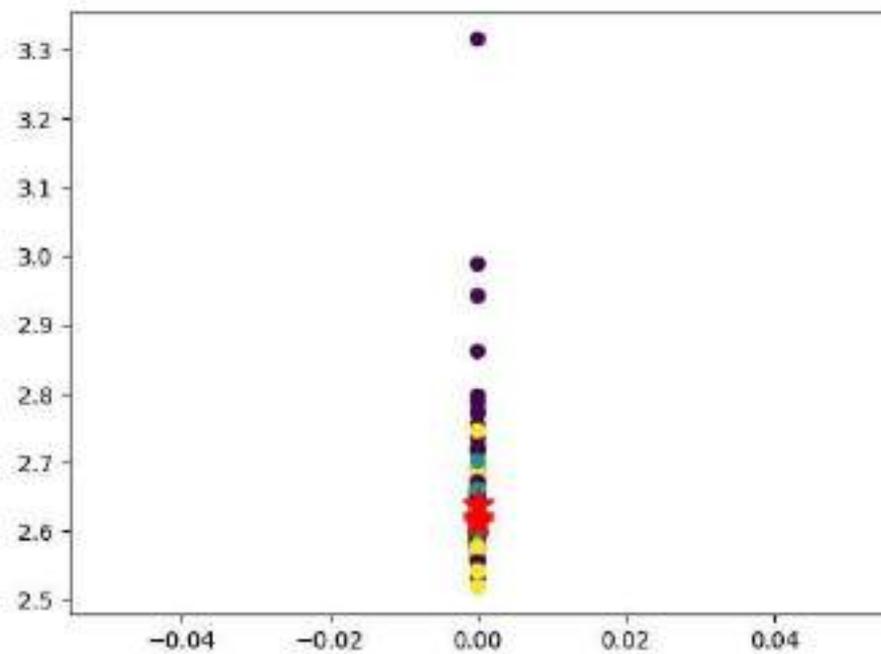


Figura 36. Datos por cada Cluster algoritmo Kmeans
Almeida y Durango, 2023

9.17 Anexo 17. Grafica comparativa de las métricas calculadas

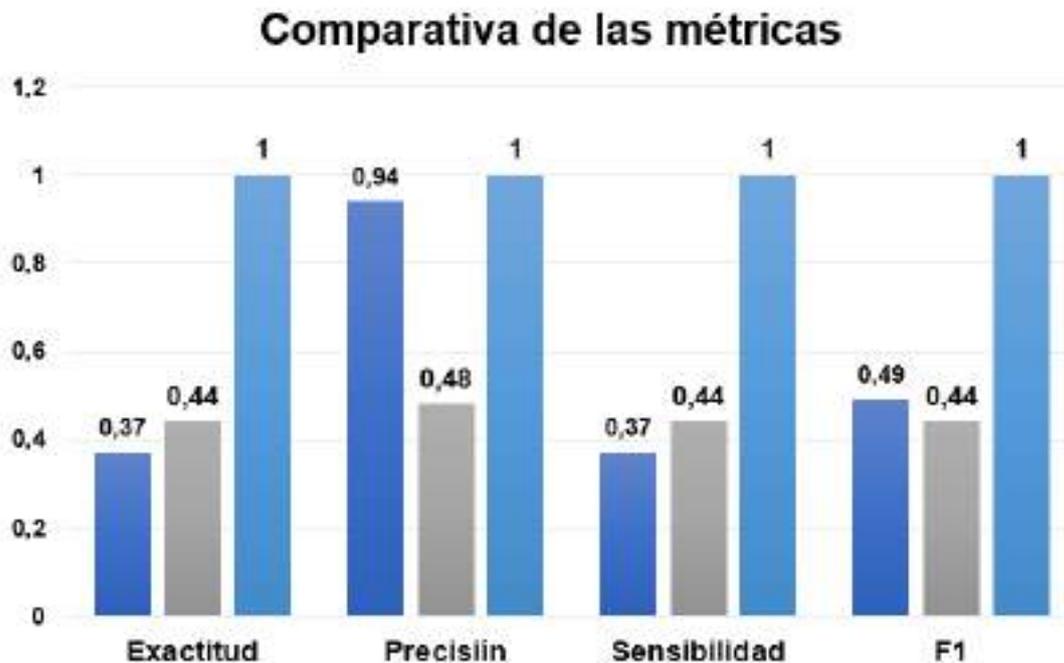


Figura 37. Gráfico de barras comparativo de las diferentes métricas calculadas de los algoritmos KNN, K-Medias y CNN.

Almeida y Durango, 2023

9.18 Anexo 18. Cuadro comparativo

En la tabla se observa la comparación que se hace de los tres algoritmos para llegar a su debida conclusión de cual se debe usar para futuros proyectos en la cual refleja un porcentaje de los parámetros estudiados como la precisión, exactitud, sensibilidad, F1 y el tiempo de la identificación de las plantas y también el tipo de aprendizaje como supervisados o no supervisados

Tabla 4. Comparativa de algoritmos de aprendizaje automático.

Foto	Planta	Tipo de aprendizaje	algoritmo	Exactitud	Sensibilidad	Precisión	F1	Tiempo
ixora-1.jpg	Ixora	supervisado	KNN	0.37	0.37	0.94	0.49	2s
ixora-1.jpg	Ixora	No supervisado	K-Means	0.44	0.44	0.48	0.44	1s
ixora-1.jpg	Ixora	supervisado	Red neuronal convolucional (CNN)	0.85	0.85	0.95	0.73	1s
ixora-2.jpg	Ixora	supervisado	KNN	0.38	0.38	0.85	0.41	3s
ixora-2.jpg	Ixora	No supervisado	K-Means	0.63	0.63	0.65	0.63	3s
ixora-2.jpg	Ixora	supervisado	Red neuronal convolucional (CNN)	0.80	0.85	0.96	0.79	2s
ixora-3.jpg	Ixora	supervisado	KNN	0.45	0.44	0.78	0.53	3s
ixora-3.jpg	Ixora	No supervisado	K-Means	0.50	0.68	0.47	0.50	1s
ixora-3.jpg	Ixora	supervisado	Red neuronal convolucional (CNN)	0.90	0.97	0.97	0.96	1s
ixora-4.jpg	Ixora	supervisado	KNN	0.63	0.70	0.71	0.55	2s
ixora-4.jpg	Ixora	No supervisado	K-Means	0.44	0.44	0.48	0.44	1s
ixora-4.jpg	Ixora	supervisado	Red neuronal convolucional (CNN)	0.85	0.91	0.95	0.73	1s
ixora-5.jpg	Ixora	supervisado	KNN	0.37	0.37	0.94	0.49	2s
ixora-5.jpg	Ixora	No supervisado	K-Means	0.66	0.78	0.50	0.66	3s
ixora-5.jpg	Ixora	supervisado	Red neuronal convolucional (CNN)	0.78	0.87	0.95	0.73	2s
ixora-6.jpg	Ixora	supervisado	KNN	0.55	0.55	0.94	0.60	2s
ixora-6.jpg	Ixora	No supervisado	K-Means	0.50	0.68	0.67	0.50	1s
ixora-6.jpg	Ixora	supervisado	Red neuronal convolucional (CNN)	0.79	0.88	0.95	0.79	2s

Foto	Planta	Tipo de aprendizaje	algoritmo	Exactitud	Sensibilidad	Precisión	F1	Tiempo
ixora-7.jpg	Ixora	supervisado	KNN	0.87	0.68	0.91	0.56	2s
ixora-7.jpg	Ixora	No supervisado	K-Means	0.60	0.60	0.48	0.69	2s
ixora-7.jpg	Ixora	supervisado	Red neuronal convolucional (CNN)	0.85	0.91	0.96	0.94	2s
ixora-8.jpg	Ixora	supervisado	KNN	0.70	0.56	0.97	0.73	3s
ixora-8.jpg	Ixora	No supervisado	K-Means	0.75	0.87	0.54	0.75	2s
ixora-8.jpg	Ixora	supervisado	Red neuronal convolucional (CNN)	0.85	0.91	0.96	0.94	1s
ixora-9.jpg	Ixora	supervisado	KNN	0.73	0.68	0.91	0.56	1s
ixora-9.jpg	Ixora	No supervisado	K-Means	0.35	0.35	0.40	0.38	4s
ixora-9.jpg	Ixora	supervisado	Red neuronal convolucional (CNN)	0.87	0.90	0.94	0.80	2s
ixora-10.jpg	Ixora	supervisado	KNN	0.55	0.64	0.80	0.59	3s
ixora-10.jpg	Ixora	No supervisado	K-Means	0.45	0.38	0.44	0.38	4s
ixora-10.jpg	Ixora	supervisado	Red neuronal convolucional (CNN)	0.97	0.97	0.97	0.97	1s
ixora-11.jpg	Ixora	supervisado	KNN	0.45	0.50	0.69	0.49	2s
ixora-11.jpg	Ixora	No supervisado	K-Means	0.38	0.38	0.78	0.38	2s
ixora-11.jpg	Ixora	supervisado	Red neuronal convolucional (CNN)	0.89	0.94	0.94	0.80	2s
ixora-12.jpg	Ixora	supervisado	KNN	0.45	0.38	0.78	0.51	2s
ixora-12.jpg	Ixora	No supervisado	K-Means	0.40	0.40	0.55	0.45	2s
ixora-12.jpg	Ixora	supervisado	Red neuronal convolucional (CNN)	0.87	0.90	0.94	0.87	2s
ixora-13.jpg	Ixora	supervisado	KNN	0.55	0.57	0.55	0.42	2s

Foto	Planta	Tipo de aprendizaje	algoritmo	Exactitud	Sensibilidad	Precisión	F1	Tiempo
ixora-13.jpg	Ixora	No supervisado	K-Means	0.52	0.63	0.47	0.32	2s
ixora-13.jpg	Ixora	supervisado	Red neuronal convolucional (CNN)	0.90	0.97	0.97	0.81	2s
ixora-14.jpg	Ixora	supervisado	KNN	0.37	0.37	0.94	0.38	2s
ixora-14.jpg	Ixora	No supervisado	K-Means	0.45	0.52	0.42	0.55	2s
ixora-14.jpg	Ixora	supervisado	Red neuronal convolucional (CNN)	0.87	0.88	0.87	0.82	1s
ixora-15.jpg	Ixora	supervisado	KNN	0.54	0.45	0.84	0.42	1s
ixora-15.jpg	Ixora	No supervisado	K-Means	0.45	0.49	0.62	0.55	2s
ixora-15.jpg	Ixora	supervisado	Red neuronal convolucional (CNN)	0.97	0.82	0.92	0.83	1s
ixora-16.jpg	Ixora	supervisado	KNN	0.32	0.37	0.45	0.35	3s
ixora-16.jpg	Ixora	No supervisado	K-Means	0.45	0.45	0.62	0.49	3s
ixora-16.jpg	Ixora	supervisado	Red neuronal convolucional (CNN)	0.80	0.84	0.92	0.83	1s
ixora-17.jpg	Ixora	supervisado	KNN	0.47	0.45	0.78	0.42	3s
ixora-17.jpg	Ixora	No supervisado	K-Means	0.55	0.37	0.78	0.42	3s
ixora-17.jpg	Ixora	supervisado	Red neuronal convolucional (CNN)	0.95	0.95	0.95	0.95	1s
ixora-18.jpg	Ixora	supervisado	KNN	0.37	0.37	0.78	0.37	3s
ixora-18.jpg	Ixora	No supervisado	K-Means	0.45	0.35	0.68	0.32	3s
ixora-18.jpg	Ixora	supervisado	Red neuronal convolucional (CNN)	0.89	0.83	0.95	0.82	2s
ixora-19.jpg	Ixora	supervisado	KNN	0.38	0.45	0.55	0.40	2s
ixora-19.jpg	Ixora	No supervisado	K-Means	0.38	0.38	0.45	0.38	3s

Foto	Planta	Tipo de aprendizaje	algoritmo	Exactitud	Sensibilidad	Precisión	F1	Tiempo
ixora-19.jpg	Ixora	supervisado	Red neuronal convolucional (CNN)	0.89	0.72	0.93	0.85	2s
ixora-20.jpg	Ixora	supervisado	KNN	0.54	0.54	0.62	0.58	3s
ixora-20.jpg	Ixora	No supervisado	K-Means	0.38	0.41	0.45	0.42	3s
ixora-20.jpg	Ixora	supervisado	Red neuronal convolucional (CNN)	0.92	0.92	0.94	0.92	1s
sunflower-1.jpg	Girasol	supervisado	KNN	0.37	0.37	0.94	0.49	2s
sunflower-1.jpg	Girasol	No supervisado	K-Means	0.44	0.44	0.48	0.44	1s
sunflower-1.jpg	Girasol	supervisado	Red neuronal convolucional (CNN)	0.85	0.85	0.95	0.73	1s
sunflower-2.jpg	Girasol	supervisado	KNN	0.40	0.40	0.90	0.50	2s
sunflower-2.jpg	Girasol	No supervisado	K-Means	0.41	0.41	0.50	0.41	2s
sunflower-2.jpg	Girasol	supervisado	Red neuronal convolucional (CNN)	0.90	0.92	0.92	0.93	1s
sunflower-3.jpg	Girasol	supervisado	KNN	0.37	0.37	0.78	0.37	3s
sunflower-3.jpg	Girasol	No supervisado	K-Means	0.45	0.35	0.68	0.32	3s
sunflower-3.jpg	Girasol	supervisado	Red neuronal convolucional (CNN)	0.89	0.83	0.95	0.82	2s
sunflower-4.jpg	Girasol	supervisado	KNN	0.38	0.45	0.55	0.40	2s
sunflower-4.jpg	Girasol	No supervisado	K-Means	0.38	0.38	0.45	0.38	3s
sunflower-4.jpg	Girasol	supervisado	Red neuronal convolucional (CNN)	0.89	0.72	0.93	0.85	2s
sunflower-5.jpg	Girasol	supervisado	KNN	0.42	0.37	0.64	0.39	2s
sunflower-5.jpg	Girasol	No supervisado	K-Means	0.45	0.45	0.52	0.43	4s
sunflower-5.jpg	Girasol	supervisado	Red neuronal convolucional (CNN)	0.97	0.95	0.94	0.97	1s

Foto	Planta	Tipo de aprendizaje	algoritmo	Exactitud	Sensibilidad	Precisión	F1	Tiempo
sunflower-6.jpg	Girasol	supervisado	KNN	0.37	0.49	0.90	0.37	3s
sunflower-6.jpg	Girasol	No supervisado	K-Means	0.41	0.41	0.48	0.42	4s
sunflower-6.jpg	Girasol	supervisado	Red neuronal convolucional (CNN)	0.80	0.87	0.82	0.82	3s
sunflower-7.jpg	Girasol	supervisado	KNN	0.55	0.57	0.55	0.42	2s
sunflower-7.jpg	Girasol	No supervisado	K-Means	0.52	0.63	0.47	0.32	2s
sunflower-7.jpg	Girasol	supervisado	Red neuronal convolucional (CNN)	0.90	0.97	0.97	0.81	2s
sunflower-8.jpg	Girasol	supervisado	KNN	0.37	0.37	0.78	0.37	3s
sunflower-8.jpg	Girasol	No supervisado	K-Means	0.45	0.35	0.68	0.32	3s
sunflower-8.jpg	Girasol	supervisado	Red neuronal convolucional (CNN)	0.89	0.83	0.95	0.82	2s
sunflower-9.jpg	Girasol	supervisado	KNN	0.40	0.40	0.80	0.49	2s
sunflower-9.jpg	Girasol	No supervisado	K-Means	0.37	0.38	0.38	0.37	3s
sunflower-9.jpg	Girasol	supervisado	Red neuronal convolucional (CNN)	0.75	0.64	0.78	0.70	2s
sunflower-10.jpg	Girasol	supervisado	KNN	0.39	0.40	0.67	0.39	2s
sunflower-10.jpg	Girasol	No supervisado	K-Means	0.32	0.34	0.65	0.33	2s
sunflower-10.jpg	Girasol	supervisado	Red neuronal convolucional (CNN)	0.79	0.63	0.92	0.87	1s
sunflower-11.jpg	Girasol	supervisado	KNN	0.62	0.74	0.74	0.63	3s
sunflower-11.jpg	Girasol	No supervisado	K-Means	0.55	0.55	0.60	0.55	2s
sunflower-11.jpg	Girasol	supervisado	Red neuronal convolucional (CNN)	0.90	0.87	0.92	0.88	1s
sunflower-12.jpg	Girasol	supervisado	KNN	0.37	0.49	0.90	0.37	3s

Foto	Planta	Tipo de aprendizaje	algoritmo	Exactitud	Sensibilidad	Precisión	F1	Tiempo
sunflower-12.jpg	Girasol	No supervisado	K-Means	0.41	0.41	0.48	0.42	4s
sunflower-12.jpg	Girasol	supervisado	Red neuronal convolucional (CNN)	0.80	0.87	0.82	0.82	3s
sunflower-13.jpg	Girasol	supervisado	KNN	0.55	0.57	0.55	0.42	2s
sunflower-13.jpg	Girasol	No supervisado	K-Means	0.52	0.63	0.47	0.32	2s
sunflower-13.jpg	Girasol	supervisado	Red neuronal convolucional (CNN)	0.90	0.97	0.97	0.81	2s
sunflower-14.jpg	Girasol	supervisado	KNN	0.41	0.43	0.43	0.55	3s
sunflower-14.jpg	Girasol	No supervisado	K-Means	0.36	0.46	0.36	0.42	2s
sunflower-14.jpg	Girasol	supervisado	Red neuronal convolucional (CNN)	0.89	0.78	0.92	0.69	s
sunflower-15.jpg	Girasol	supervisado	KNN	0.42	0.37	0.64	0.39	2s
sunflower-15.jpg	Girasol	No supervisado	K-Means	0.45	0.45	0.52	0.43	4s
sunflower-15.jpg	Girasol	supervisado	Red neuronal convolucional (CNN)	0.97	0.95	0.94	0.97	1s
sunflower-16.jpg	Girasol	supervisado	KNN	0.73	0.68	0.91	0.56	1s
sunflower-16.jpg	Girasol	No supervisado	K-Means	0.35	0.35	0.40	0.38	4s
sunflower-16.jpg	Girasol	supervisado	Red neuronal convolucional (CNN)	0.87	0.90	0.94	0.80	2s
sunflower-17.jpg	Girasol	supervisado	KNN	0.45	0.44	0.78	0.53	3s
sunflower-17.jpg	Girasol	No supervisado	K-Means	0.50	0.68	0.47	0.50	1s
sunflower-17.jpg	Girasol	supervisado	Red neuronal convolucional (CNN)	0.90	0.97	0.97	0.96	1s
sunflower-18.jpg	Girasol	supervisado	KNN	0.65	0.45	0.65	0.39	2s
sunflower-18.jpg	Girasol	No supervisado	K-Means	0.54	0.50	0.60	0.42	1s

Foto	Planta	Tipo de aprendizaje	algoritmo	Exactitud	Sensibilidad	Precisión	F1	Tiempo
sunflower-18.jpg	Girasol	supervisado	Red neuronal convolucional (CNN)	0.94	0.87	0.95	0.90	1s
sunflower-19.jpg	Girasol	supervisado	KNN	0.42	0.37	0.64	0.39	2s
sunflower-19.jpg	Girasol	No supervisado	K-Means	0.45	0.45	0.52	0.43	4s
sunflower-19.jpg	Girasol	supervisado	Red neuronal convolucional (CNN)	0.97	0.95	0.94	0.97	1s
sunflower-20.jpg	Girasol	supervisado	KNN	0.62	0.74	0.74	0.63	3s
sunflower-20.jpg	Girasol	No supervisado	K-Means	0.55	0.55	0.60	0.55	2s
sunflower-20.jpg	Girasol	supervisado	Red neuronal convolucional (CNN)	0.90	0.87	0.92	0.88	1s
palm-1.jpg	Palmera	supervisado	KNN	0.38	0.45	0.78	0.54	1s
palm-1.jpg	Palmera	No supervisado	K-Means	0.45	0.40	0.41	0.47	2s
palm-1.jpg	Palmera	supervisado	Red neuronal convolucional (CNN)	0.87	0.93	0.90	0.82	1s
palm-2.jpg	Palmera	supervisado	KNN	0.42	0.49	0.52	0.52	1s
palm-2.jpg	Palmera	No supervisado	K-Means	0.52	0.40	0.78	0.37	3s
palm-2.jpg	Palmera	supervisado	Red neuronal convolucional (CNN)	0.81	0.93	0.92	0.80	2s
palm-3.jpg	Palmera	supervisado	KNN	0.42	0.37	0.64	0.39	2s
palm-3.jpg	Palmera	No supervisado	K-Means	0.45	0.45	0.52	0.43	4s
palm-3.jpg	Palmera	supervisado	Red neuronal convolucional (CNN)	0.97	0.95	0.94	0.97	1s
palm-4.jpg	Palmera	supervisado	KNN	0.55	0.57	0.55	0.42	2s
palm-4.jpg	Palmera	No supervisado	K-Means	0.52	0.63	0.47	0.32	2s
palm-4.jpg	Palmera	supervisado	Red neuronal convolucional (CNN)	0.90	0.97	0.97	0.81	2s

Foto	Planta	Tipo de aprendizaje	algoritmo	Exactitud	Sensibilidad	Precisión	F1	Tiempo
palm-5.jpg	Palmera	supervisado	KNN	0.45	0.47	0.55	0.42	3s
palm-5.jpg	Palmera	No supervisado	K-Means	0.62	0.63	0.67	0.52	2s
palm-5.jpg	Palmera	supervisado	Red neuronal convolucional (CNN)	0.80	0.67	0.87	0.71	1s
palm-6.jpg	Palmera	supervisado	KNN	0.38	0.45	0.55	0.40	2s
palm-6.jpg	Palmera	No supervisado	K-Means	0.38	0.38	0.45	0.38	3s
palm-6.jpg	Palmera	supervisado	Red neuronal convolucional (CNN)	0.89	0.72	0.93	0.85	2s
palm-7.jpg	Palmera	supervisado	KNN	0.45	0.37	0.68	0.45	2s
palm-7.jpg	Palmera	No supervisado	K-Means	0.60	0.64	0.63	0.60	1s
palm-7.jpg	Palmera	supervisado	Red neuronal convolucional (CNN)	0.87	0.90	0.72	0.91	2s
palm-8.jpg	Palmera	supervisado	KNN	0.38	0.45	0.55	0.40	2s
palm-8.jpg	Palmera	No supervisado	K-Means	0.38	0.38	0.45	0.38	3s
palm-8.jpg	Palmera	supervisado	Red neuronal convolucional (CNN)	0.89	0.72	0.93	0.85	2s
palm-9.jpg	Palmera	supervisado	KNN	0.73	0.68	0.91	0.56	1s
palm-9.jpg	Palmera	No supervisado	K-Means	0.35	0.35	0.40	0.38	4s
palm-9.jpg	Palmera	supervisado	Red neuronal convolucional (CNN)	0.87	0.90	0.94	0.80	2s
palm-10.jpg	Palmera	supervisado	KNN	0.37	0.37	0.94	0.49	2s
palm-10.jpg	Palmera	No supervisado	K-Means	0.44	0.44	0.48	0.44	1s
palm-10.jpg	Palmera	supervisado	Red neuronal convolucional (CNN)	0.85	0.85	0.95	0.73	1s
palm-11.jpg	Palmera	supervisado	KNN	0.37	0.37	0.78	0.37	3s

Foto	Planta	Tipo de aprendizaje	algoritmo	Exactitud	Sensibilidad	Precisión	F1	Tiempo
palm-11.jpg	Palmera	No supervisado	K-Means	0.45	0.35	0.68	0.32	3s
palm-11.jpg	Palmera	supervisado	Red neuronal convolucional (CNN)	0.89	0.83	0.95	0.82	2s
palm-12.jpg	Palmera	supervisado	KNN	0.70	0.68	0.64	0.71	2s
palm-12.jpg	Palmera	No supervisado	K-Means	0.42	0.38	0.41	0.41	3s
palm-12.jpg	Palmera	supervisado	Red neuronal convolucional (CNN)	0.95	0.94	0.94	0.95	1s
palm-13.jpg	Palmera	supervisado	KNN	0.35	0.36	0.69	0.44	2s
palm-13.jpg	Palmera	No supervisado	K-Means	0.48	0.63	0.55	0.46	2s
palm-13.jpg	Palmera	supervisado	Red neuronal convolucional (CNN)	0.90	0.93	0.92	0.92	1s
palm-14.jpg	Palmera	supervisado	KNN	0.55	0.57	0.55	0.42	2s
palm-14.jpg	Palmera	No supervisado	K-Means	0.52	0.63	0.47	0.32	2s
palm-14.jpg	Palmera	supervisado	Red neuronal convolucional (CNN)	0.90	0.97	0.97	0.81	2s
palm-15.jpg	Palmera	supervisado	KNN	0.47	0.45	0.78	0.42	3s
palm-15.jpg	Palmera	No supervisado	K-Means	0.55	0.37	0.78	0.42	3s
palm-15.jpg	Palmera	supervisado	Red neuronal convolucional (CNN)	0.90	0.93	0.94	0.93	1s
palm-16.jpg	Palmera	supervisado	KNN	0.38	0.38	0.78	0.40	3s
palm-16.jpg	Palmera	No supervisado	K-Means	0.37	0.40	0.70	0.42	3s
palm-16.jpg	Palmera	supervisado	Red neuronal convolucional (CNN)	0.89	0.90	0.91	0.94	1s
palm-17.jpg	Palmera	supervisado	KNN	0.39	0.45	0.78	0.42	2s
palm-17.jpg	Palmera	No supervisado	K-Means	0.38	0.37	0.48	0.49	4s

Foto	Planta	Tipo de aprendizaje	algoritmo	Exactitud	Sensibilidad	Precisión	F1	Tiempo
palm-17.jpg	Palmera	supervisado	Red neuronal convolucional (CNN)	0.91	0.94	0.87	0.92	2s
palm-18.jpg	Palmera	supervisado	KNN	0.39	0.49	0.78	0.40	2s
palm-18.jpg	Palmera	No supervisado	K-Means	0.32	0.38	0.80	0.62	2s
palm-18.jpg	Palmera	supervisado	Red neuronal convolucional (CNN)	0.79	0.95	0.94	0.83	1s
palm-19.jpg	Palmera	supervisado	KNN	0.42	0.49	0.52	0.52	1s
palm-19.jpg	Palmera	No supervisado	K-Means	0.52	0.40	0.78	0.37	3s
palm-19.jpg	Palmera	supervisado	Red neuronal convolucional (CNN)	0.81	0.93	0.92	0.80	2s
palm-20.jpg	Palmera	supervisado	KNN	0.38	0.40	0.90	0.54	3s
palm-20.jpg	Palmera	No supervisado	K-Means	0.43	0.43	0.55	0.49	3s
palm-20.jpg	Palmera	supervisado	Red neuronal convolucional (CNN)	0.90	0.93	0.94	0.93	1s

Aspectos observados de los algoritmos en base a distintos parámetros para la identificación de una planta ornamental.
Almeida y Durango, 2023

9.19 Anexo 19. Recursos de las herramientas utilizadas

Tabla 5. Herramientas y los recursos usados para la ejecución y realización de los algoritmos.

Herramientas	Recursos
Google Colab	12 GB de Ram
	107.7 GB de Disco
Python	Versión 3

Aspectos observados sobre los recursos necesarios para la ejecución de los algoritmos e identificación de las plantas
Almeida y Durango, 2023

9.20 Anexo 20. Establecimiento de conjunto de imágenes

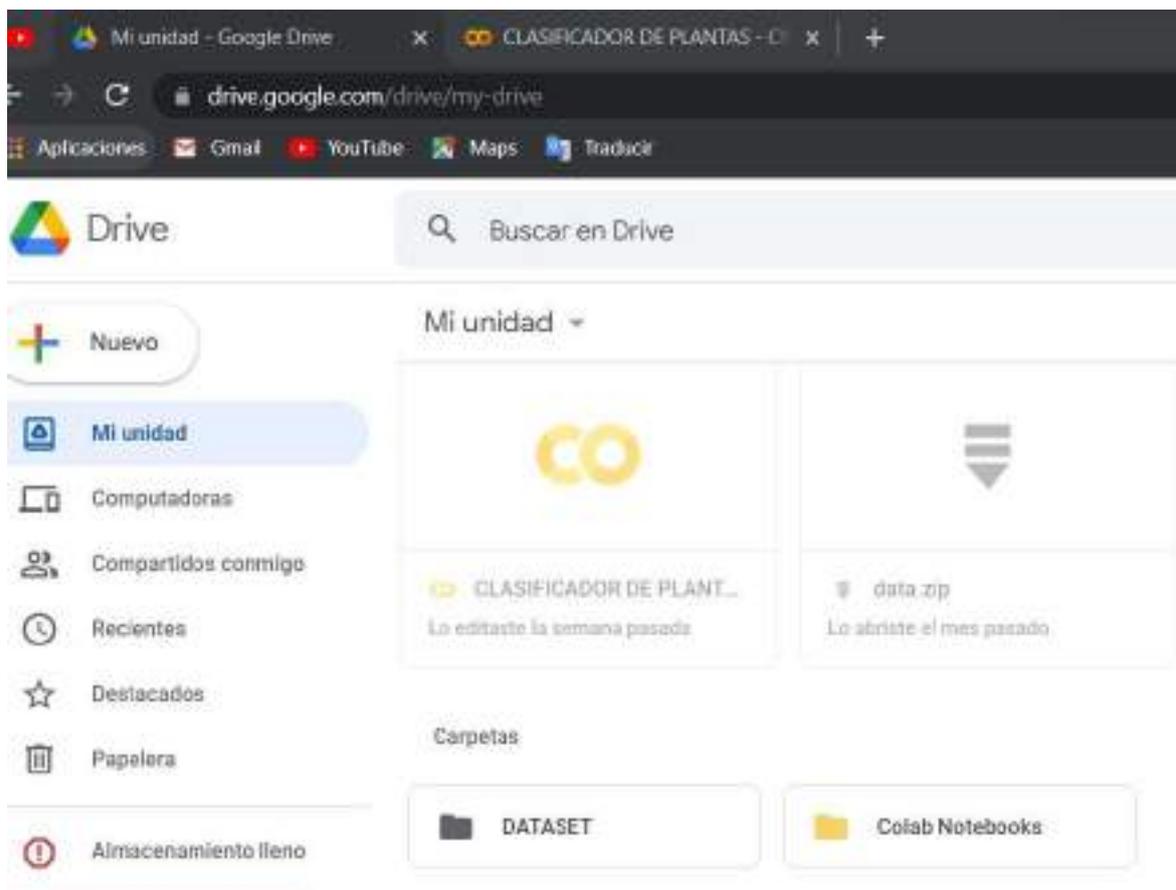


Figura 38. Almacenamiento de conjunto de imágenes para el entrenamiento de los algoritmos.

Almeida y Durango, 2023

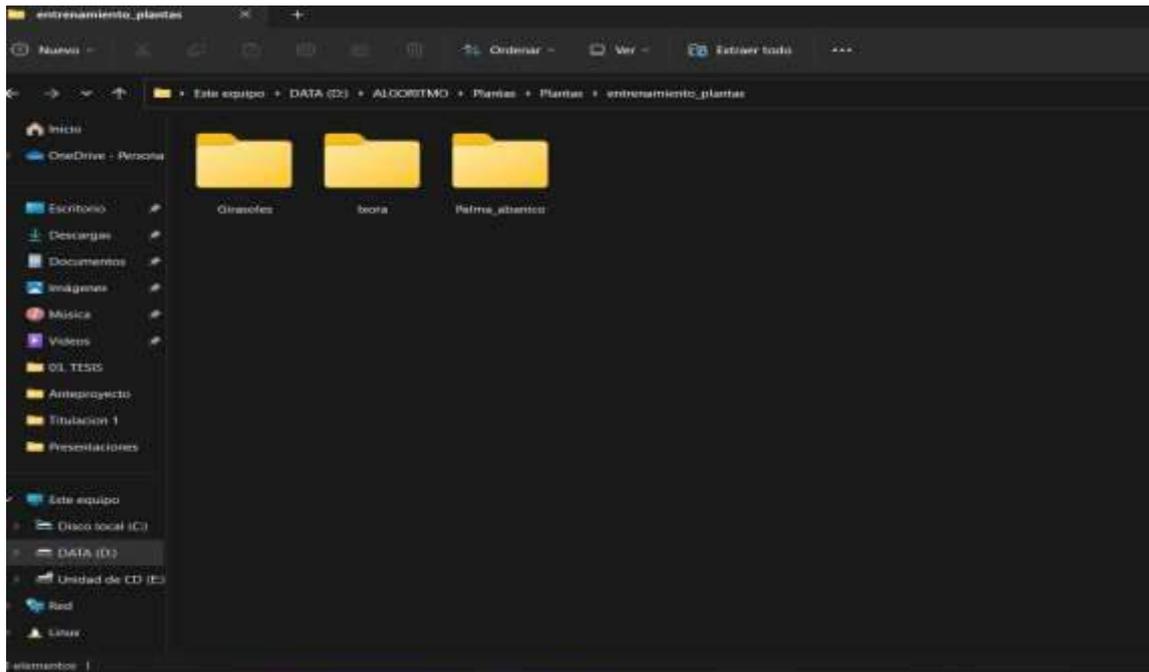


Figura 39. Carpetas de plantas.
Almeida y Durango, 2023

Nombre	Tipo	Tamaño comprimido	Protegido	Tamaño	Relación
0a3e19853f	Archivo.JPG	10 KB	No	10 KB	3%
0a81e53a09	Archivo.JPG	16 KB	No	16 KB	1%
0a8631ebc1	Archivo.JPG	20 KB	No	20 KB	1%
0aef60c92d	Archivo.JPG	15 KB	No	15 KB	3%
0aee6a0b8f	Archivo.JPG	14 KB	No	14 KB	3%
0beb16625f	Archivo.JPG	15 KB	No	15 KB	3%
0c2fb8717a	Archivo.JPG	26 KB	No	26 KB	1%
0c6d94a940	Archivo.JPG	18 KB	No	18 KB	2%
0c9b1eb77d	Archivo.JPG	16 KB	No	16 KB	1%
0cd157889e	Archivo.JPG	24 KB	No	24 KB	1%
0da99085b9	Archivo.JPG	18 KB	No	18 KB	3%
0e0643e3bd	Archivo.JPG	21 KB	No	22 KB	2%
0fa631353c	Archivo.JPG	17 KB	No	18 KB	4%
0fbc83cfed	Archivo.JPG	23 KB	No	23 KB	3%
0fdbec9fa7	Archivo.JPG	14 KB	No	14 KB	3%
001ba6c41c	Archivo.JPG	36 KB	No	36 KB	1%
1a42790dfd	Archivo.JPG	13 KB	No	14 KB	3%
1aac202c57	Archivo.JPG	13 KB	No	13 KB	2%
1ad2aee7d5	Archivo.JPG	15 KB	No	15 KB	3%

Figura 40. Imágenes recabadas de Ixoras.
Almeida y Durango, 2023

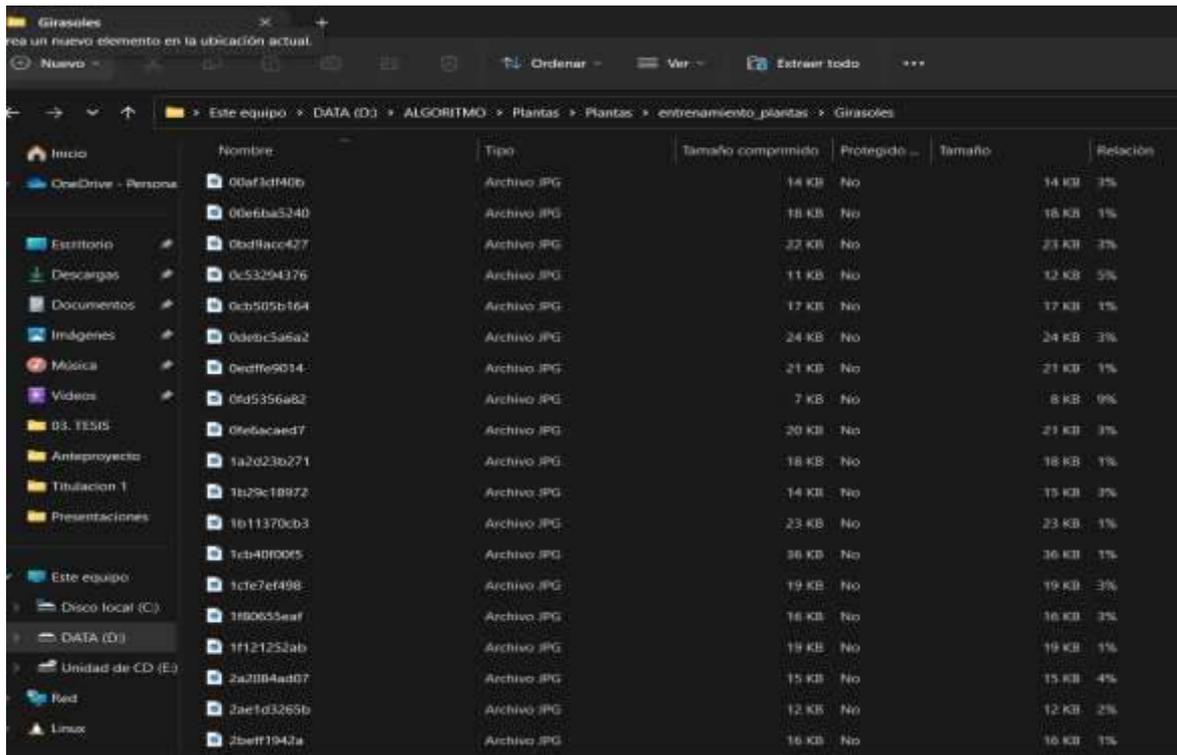


Figura 41. Imágenes recabadas de Girasoles.
Almeida y Durango, 2023

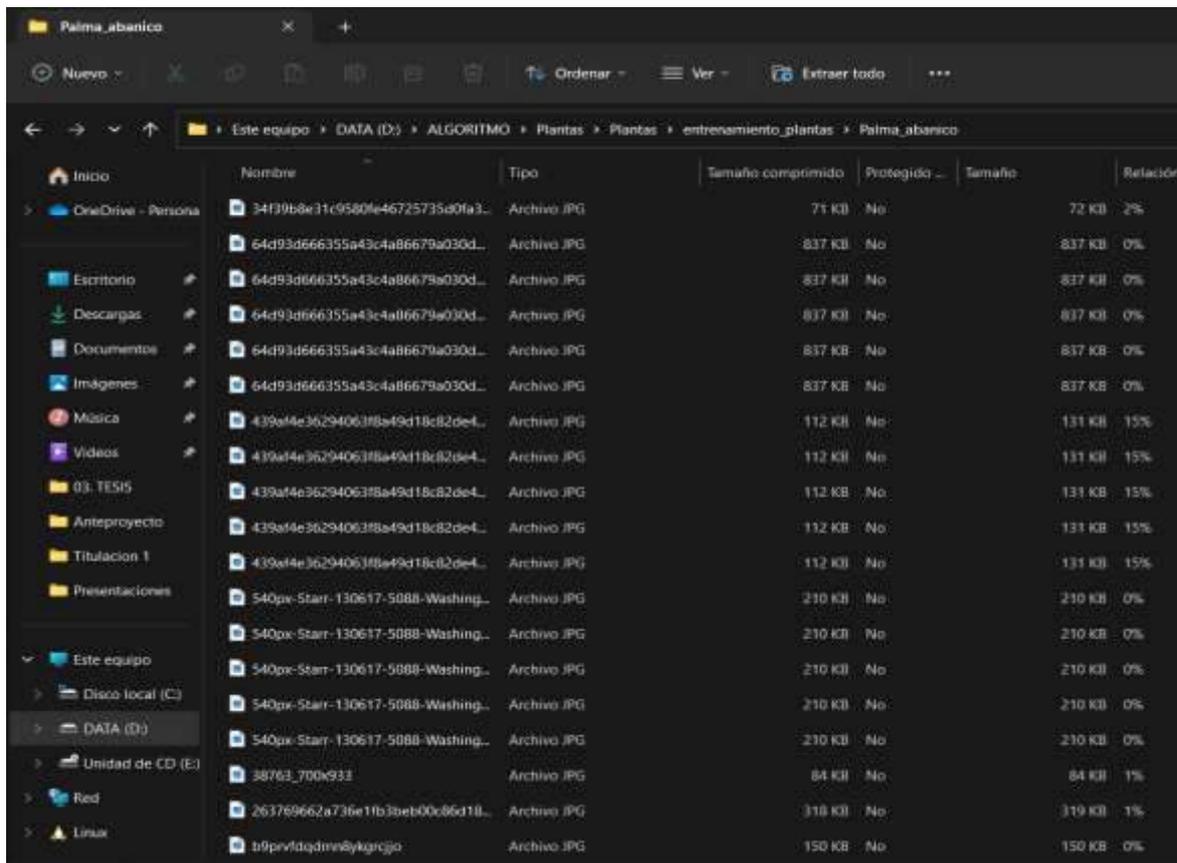


Figura 42. Imágenes recabadas de Palmas.
Almeida y Durango, 2023

9.21 Anexo 21. Manual Técnico



UNIVERSIDAD AGRARIA DEL ECUADOR FACULTAD DE CIENCIAS AGRARIAS INGENIERIA EN CIENCIAS DE LA COMPUTACION

Manual Técnico

Objetivo: Indicar el uso correcto de los algoritmos de aprendizaje automático de manera descriptiva mediante las herramientas utilizadas para la creación y ejecución de estos.

1. Aspecto técnico

Este manual fue desarrollado con el fin de brindar información importante sobre las herramientas que se usaron para la creación del proyecto, para así poder dar un correcto mantenimiento, nuevas instalaciones a futuro, en ese manual técnico estarán todas las herramientas que se utilizaron en los algoritmos de aprendizaje automático, con la finalidad de explicar a los usuarios que deseen realizar alguna modificación en los algoritmos.

2. Herramientas utilizadas en el desarrollo

Aquí se explicarán las diversas herramientas informáticas se usaron para el desarrollo de los algoritmos.

2.1 Python

Python es un lenguaje de programación más usados por la mayoría de programadores, ya que es utilizado mayormente en aplicaciones, el desarrollo

de sistemas web, entre otros. Existen muchas versiones de Python una más funcional que otra, pero para el desarrollo de los distintos algoritmos se utiliza la versión de Python 3.8, ya que esta versión de Python es compatible con las librerías usadas para el desarrollo de los distintos algoritmos.

2.2 Google Colab

Este es un producto creado por Google que es gratuito, permitiendo a cualquier usuario escribir y ejecutar códigos de Python en el navegador, esta plataforma tiene un gran beneficio ya que ayuda a ejecutar tareas de aprendizaje automático, ciencia de datos; es por ello que se utilizara esta plataforma web para ingresar los códigos para realizar el entrenamiento de los distintos algoritmos de aprendizaje automático a estudiar.

3. Instalación de programas

Se detallará los programas y herramientas usados para el desarrollo del proyecto. Para poder trabajar en los códigos se deberá descargar e instalar Python 3.8 en la cual se puede descargar de manera gratuita en la página oficial

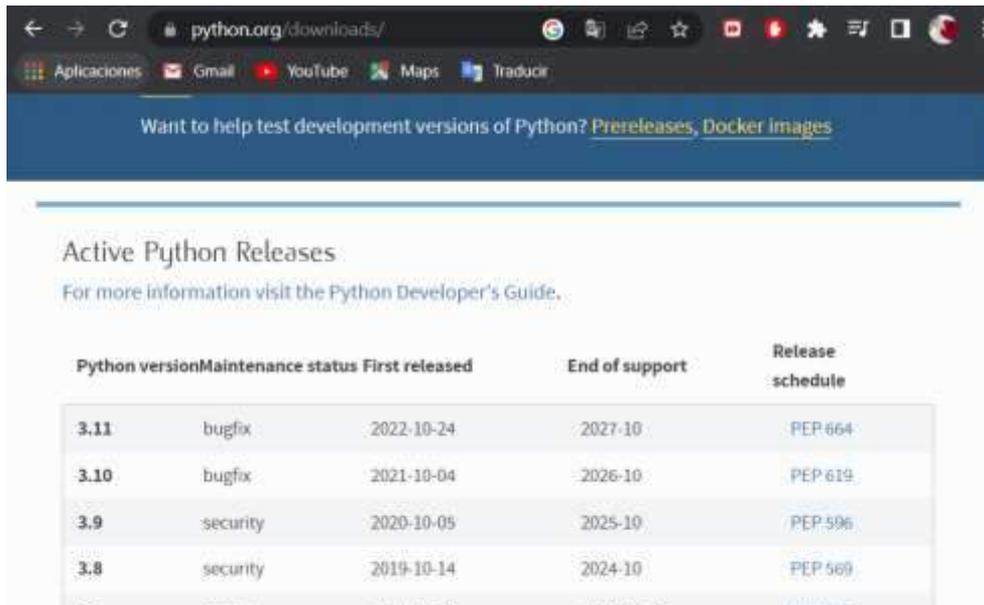


Figura 43. Página oficial de Python.
Almeida y Durango, 2023

Una vez instalado Python se procederá a descargar e instalar las siguientes herramientas, como es PyCharm. El sitio web del lenguaje de programación utilizado para el desarrollo de los algoritmos es: <https://www.python.org/>, así mismo la página oficial para la ejecución y el entrenamiento de los algoritmos es: <https://colab.research.google.com/>, hay que tomar en cuenta que tanto como el lenguaje de programación, como las plataformas utilizadas sean descargadas en las versiones requeridas para la ejecución de los algoritmos. Por último, se abre la

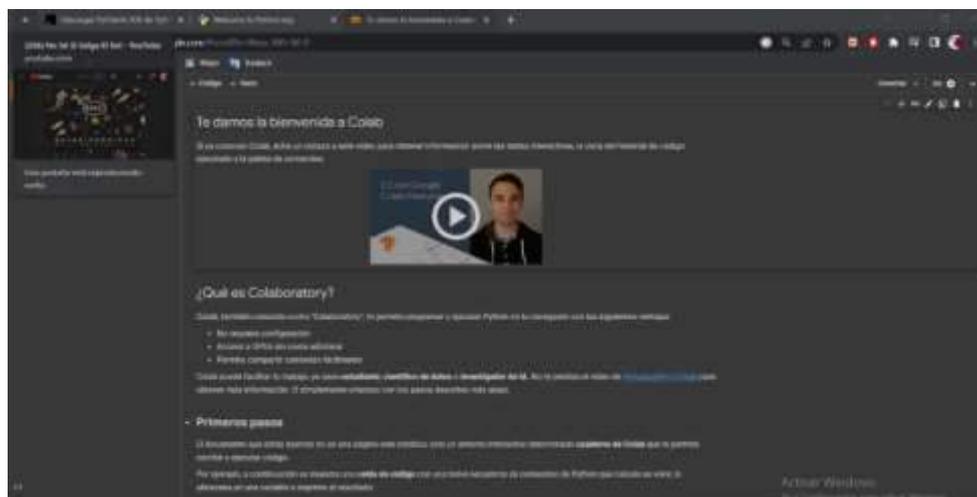


Figura 44. Plataforma de Google Colab
Almeida Y Durango, 2023

plataforma de Google Colab, esta plataforma trabaja en la web así que no se necesitará ser descargada, ni instalada en el computador.

4. Requisitos mínimos para la ejecución de los algoritmos

Los requisitos mínimos para que los algoritmos puedan ejecutarse son:

Sistema operativo: Windows 10

Procesador: Intel ® Core i5

Memoria: 8 GB

Disco duro: 128 Gb

Resolución de pantalla: 1024 x 600

Periféricos: Teclado, Mouse y Cámara