

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

CENTRO UNIVERSITARIO UAEM TEXCOCO



DOCTORADO EN CIENCIAS DE LA COMPUTACIÓN

“Algoritmos para identificar plantas de maíz para automatizar la fertilización en México”

T E S I S

Que para obtener el grado de Doctor en Ciencias de la Computación

Presenta:

Marcos Yamir Gómez Ramos

Director de Tesis: Dr. José Sergio Ruiz Castilla
Co-director de Tesis: Dr. Jair Cervantes Canales
Tutor: Dr. Farid García Lamont

Índice general

Índice de figuras	4
Índice de tablas	6
1. Introducción	9
1.1. Planteamiento del Problema	9
1.2. Antecedentes	11
1.3. Justificación	11
1.4. Hipótesis	11
1.5. Objetivos	12
1.5.1. Objetivo General	12
1.5.2. Objetivos Específicos	12
1.6. Contribuciones	12
2. Estado del Arte	14
3. Metodología	22
3.1. Conceptos Básicos sobre Inteligencia Artificial	22
3.2. Metodología Implementada	26
3.2.1. Captura de las imágenes de plantas de maíz y maleza para elaboración de dataset propio.	26
3.2.2. Características y muestras de las imágenes contenidas en el Dataset	27
3.3. Herramientas Matemáticas	28
3.3.1. Segmentación	28
3.3.2. Extracción de características.	30
3.3.3. Características Texturales	30
3.3.4. Características Cromáticas	32
3.4. Algoritmos de Clasificación utilizados durante las pruebas.	33
3.4.1. Clasificadores más utilizados en ML	33
3.4.2. Naive Bayes	33
3.4.3. Random Forest	34
3.4.4. SVM	34
3.4.5. K-Nearest Neighbors	35
3.4.6. Backpropagation	35
3.4.7. Redes Neuronales Convolucionales	36
3.5. Herramientas Computacionales	37
3.5.1. Clasificación	37
3.5.2. Segmentación	41

3.5.3. Extracción de características	41
3.5.4. Espacios de color	42
3.6. Inteligencia Artificial	44
3.6.1. Aprendizaje Automático o Machine Learning	44
3.6.2. Aprendizaje Profundo o Deep Learning	46
4. Etapa Experimental	47
4.1. Búsqueda de Datasets de Plantas de Maíz	47
4.2. Revisión de Aplicaciones Móviles	48
4.2.1. PlantNet	48
4.2.2. ArbolApp	49
4.2.3. Plantifier	51
4.3. Recolección de imágenes propias	53
4.4. Métricas para evaluar una clasificación de Aprendizaje Automático	55
4.5. Técnicas clásicas de reconocimiento	56
4.5.1. Segmentación	56
4.5.2. Extracción de características	57
4.5.3. Clasificación	59
4.6. Redes Neuronales Convolucionales	64
4.6.1. Arquitectura de una CNN	68
5. Resultados	70
5.1. Técnicas clásicas	70
5.2. Redes Neuronales Convolucionales	74
6. Discusiones	80
6.1. Técnicas clásicas vs CNN	80
6.2. Comparación con otros trabajos relacionados	81
7. Conclusiones	82
8. Publicaciones	83
Referencias	86

Índice de figuras

3.1. Metodología implementada en la Primera etapa (Publicada en RCS)	26
3.2. Metodología implementada en la Segunda etapa (Publicada en IJCOPI)	26
3.3. Imágenes de maíz y maleza.	27
3.4. Segmentación de imágenes de maíz.	28
3.5. Segmentación de imágenes de maleza.	29
3.6. Espacio de color RGB	42
3.7. Espacio de color HSV	43
3.8. Espacio de color L a b	43
3.9. Relación de la Inteligencia Artificial con Aprendizaje Automático y Aprendizaje Profundo	44
4.1. Imágenes encontradas en el Data Set de Kaggle. A partir de kaggle.com	47
4.2. Imágenes propias tomadas en una plantación de maíz. Elab. propia	48
4.3. Pantallas de Clasificación. A partir del uso de la aplicación PlantNet.	49
4.4. Pantallas de Búsqueda. A partir del uso de la aplicación PlantNet.	50
4.5. Inicio y Menú. A partir del uso de la aplicación ArbolApp.	50
4.6. Similitudes. A partir del uso de la aplicación ArbolApp.	51
4.7. Inicio y Selección. A partir del uso de la aplicación Plantifier.	51
4.8. Registro. A partir del uso de la aplicación Plantifier.	52
4.9. Ubicación de los lugares donde se recolectaron las imágenes en México. Google Maps.	53
4.10. Muestra de Dataset de imágenes de maíz de los 3 estados de México.	54
4.11. Muestra de Dataset de imágenes de maleza de los 3 estados de México.	54
4.12. Matriz de Confusión	55
4.13. Finalización de corrida y generación de archivo.	59
4.14. Clasificación con el algoritmo de Naive Bayes.	60
4.15. Clasificación con el algoritmo de Random Forest.	61
4.16. Clasificación con el algoritmo de SVM.	62
4.17. Clasificación con el algoritmo de vecino más cercano.	63
4.18. Clasificación con el algoritmo de Backpropagation.	64
4.19. Arquitectura de una Red Neuronal Convolutiva. Pro Deep Learning with TensorFlow.	68
4.20. Metodología implementada en Tercera etapa CNN + SVM en ARTIIS 2021.	69
5.1. Gráfica mostrando los métodos para segmentar y sus resultados. Elaboracion propia en Python	71
5.2. Gráfica de resultados con la mejor precisión. Elaboracion propia en Python	74
5.3. Tabla comparativa de Resultados.	74
5.4. Curva ROC. Elaboracion propia en Python	75

5.5. Valor del area bajo la curva.	75
5.6. Resultados en las pruebas usando CNN y CNN+SVM.	76
5.7. Comparación de resultados obtenidos vs el método propuesto.	77
5.8. Matriz de confusión y demás métricas de evaluación.	78
5.9. Gráfica de precisión en el entrenamiento.	78
5.10. Gráfica de pérdida en el entrenamiento.	79
5.11. Gráfica del area bajo la curva.	79
8.1. Primer página del primer artículo publicado.	83
8.2. Primer página del segundo artículo publicado.	84
8.3. Primer página del tercer artículo publicado.	85

Índice de tablas

3.1. Características de imágenes del Dataset.	27
3.2. Características extraídas.	32
5.1. Resultados de porcentajes con el método PCA.	70
5.2. Resultado de porcentajes con segmentación Otsu.	71
5.3. Resultados con método PCA y características cromáticas.	71
5.4. Resultados con segmentación Otsu y características cromáticas.	72
5.5. Resultados con segmentación Otsu y con la combinación de características texturales y cromáticas.	72
5.6. Resultados con método PCA y con la combinación de características texturales y cromáticas.	72
5.7. Resultados con la red neuronal convolucional (CNN)	73
5.8. Resultados Validación Cruzada.	73

Resumen

La presente investigación se enfoca en buscar y proponer una solución al problema del agricultor mexicano, dicho problema consiste en cómo se realiza la fertilización en sembradíos de maíz en México. La fertilización manual es un proceso lento, tedioso y poco efectivo; las desventajas pueden ser el mal manejo del fertilizante, aplicación errónea y desperdicio; derivado de esto, se puede decir que la fertilización es un proceso costoso para el agricultor.

Se busca responder a la hipótesis sobre si *Los algoritmos de inteligencia artificial aplicados en la agricultura mejorarán los procesos al distinguir entre plantas de maíz y maleza, apoyando a la toma de decisiones de los agricultores*. Siendo el objetivo principal de la investigación: *Identificar plantas de maíz para diferenciarlas de la maleza usando Visión Artificial a partir de un Dataset de imágenes*, evaluando el método propuesto a través de un estudio cuantitativo.

La metodología implementada incluye un Dataset propio, se emplearon técnicas de clasificación tradicionales (clásicas), los cuales son: Naive Bayes, Random Forest, K-nn, SVM y Backpropagation, arrojando un porcentaje máximo del 98.98 % de precisión realizando pre-procesamiento y segmentación de imágenes con Otsu y el método PCA. También se emplearon las redes neuronales convolucionales de forma individual y en combinación con las SVM a través de un algoritmo híbrido, estas arrojaron un porcentaje máximo de 99.72 % de Exactitud y 99.88 % de Precisión.

Los resultados generados han arrojado que el modelo propuesto en cada una de las etapas de experimentación da respuesta positiva y comprueba que la hipótesis que se planteó es *Verdadera*.

Se plantean nuevos retos derivados de los experimentos y los resultados generados, tales como: aumentar la cantidad de imágenes, realizar más tareas de experimentación con nuevos algoritmos como Adaboost, Gboost y XGBoost entre otros; y facilitar el Dataset en algún repositorio para futuras investigaciones.

Palabras clave: Clasificación, Segmentación, Extracción de características, Redes neuronales convolucionales, Máquinas de soporte vectorial, Algoritmo híbrido.

Abstract

This research is focused on finding and proposing a solution to the problem of the Mexican farmer, this problem consists of how fertilization is carried out in corn fields in Mexico. Manual fertilization is a slow, tedious and ineffective process; the disadvantages can be fertilizer mismanagement, misapplication and waste; derived from this, it can be said that fertilization is a costly process for the farmer.

It seeks to answer the hypothesis on whether *Artificial intelligence algorithms applied in agriculture will improve processes by distinguishing between corn plants and weeds, supporting farmers' decision making*. Being the main objective of the research: *To identify corn plants to differentiate them from weeds using Artificial Vision from a Dataset of images*, evaluating the proposed method through a quantitative study.

The implemented methodology includes a proprietary Dataset, traditional (classical) classification techniques were used, which are: Naive Bayes, Random Forest, K-nn, SVM and Backpropagation, yielding a maximum percentage of 98.98% accuracy by performing pre-processing and segmentation of images with Otsu and the PCA method. Convolutional neural networks were also used individually and in combination with SVMs through a hybrid algorithm, which yielded a maximum percentage of 99.72% Accuracy and 99.88% Precision.

The results generated have shown that the model proposed in each of the experimental stages gives a positive response and proves that the hypothesis is *True*.

New challenges derived from the experiments and the results generated, such as: increase the number of images, perform more experimentation tasks with new algorithms such as Adaboost, Gboost and XGBoost among others; and make the Dataset available in a repository for future research.

Keywords: Classification, Segmentation, Feature extraction, Convolutional neural networks, Support vector machines, Hybrid algorithm.

Capítulo 1

Introducción

1.1. Planteamiento del Problema

México es un país del continente Americano, ubicado en la parte sur de América del Norte. Muestra un territorio semidesértico, con clima árido en la parte norte del país mientras que en la parte sur y suroeste tiene un clima frío y templado; al sur-sureste hay bosques tropicales. Los principales cultivos que se cultivan en México son: maíz, caña de azúcar, aguacate, chile verde, tomate, sorgo y frijol (MSC, 2019). La variedad de climas, suelos y ecosistemas en México lo convierten en un país ideal para cultivar una gran variedad de productos.

El maíz es el cultivo representativo de México por su importancia económica, social y cultural; siendo el cultivo más abundante del país, es un grano que aporta nutrientes al organismo humano y ha sido parte de la dieta del pueblo mexicano durante siglos; el rendimiento del cultivo es mayor en el ciclo primavera-verano; se divide principalmente en dos: maíz blanco y maíz amarillo; aunque se tienen registradas 59 variedades criollas de maíz en México (Oleary, 2016). El maíz blanco se distribuye mayormente pensando en el consumo humano, mientras que la producción de maíz amarillo se distribuye para la industria.

La producción de esta semilla, según los datos de 2019 fue de 31.2 millones de toneladas en una superficie sembrada de 8 millones de hectáreas, hubo un crecimiento con respecto al año 2017 ya que la producción en ese año fue de 27.8 millones de toneladas, mientras que la superficie sembrada en el mismo año fue de 7.5 millones de hectáreas, gran parte del territorio nacional es propicio para la producción por lo que en los 32 estados de la República Mexicana se produce Maíz en grano; México en el 2017 exportó a 17 países y ocupa el 8° lugar en producción mundial de maíz. (ASERCA, 2019)

El problema del agricultor mexicano en el proceso de fertilización surge al hacerlo de forma manual, ya que se realiza de manera lenta debido a que se debe recorrer surco por surco aplicando el fertilizante, invirtiendo 8 horas aproximadamente para completar la tarea en sembradíos de aproximadamente una hectárea y eso no garantiza que la fertilización se haga de manera efectiva. El fertilizador, que es la persona que se encarga de realizar la tarea, además debe cargar el fertilizante que se aplicará, ya sea en polvo o una mezcla de agua con algún tipo de fertilizante, estimando 200 litros en promedio por hectárea.

Al fertilizar de forma manual existen deficiencias como son: mal manejo de producto, mala fertilización debido a que el fertilizador no es preciso en la cantidad aplicada en cada planta, que se desperdicia fertilizante donde no hay nada sembrado por rociar o aplicar de forma continua y que

puede resultar costoso porque se requieren pagar n jornadas de trabajo por hectárea; por lo que la solución propuesta es mejorar el proceso de fertilización y hacerlo a través de un dispositivo con algoritmos de inteligencia artificial que fertilice de manera eficaz, optimizando tiempo y minimizando costos.

Los daños de los fertilizantes por su composición química son innegables, ya que causan efectos nocivos sobre el medio ambiente, su uso presenta demasiados riesgos de daño ambiental, principalmente la contaminación de las aguas subterráneas o mantos acuíferos sobre el que se aplican. Es importante mencionar que un *fertilizante*: aporta nutrientes al suelo, la finalidad es aumentar el contenido de nutrientes beneficiosos y favorecer el crecimiento de las plantas. Debe enfatizarse: *No todos los fertilizantes son iguales y no todos los fertilizantes son buenos para el suelo y el medio ambiente.* (Silva, Álvarez Cobelas & González, 2017)

Los productores agrícolas se han enfocado en agregar nitrógeno, fósforo y potasio a los cultivos, y aunque esta práctica fue descubierta a mediados del siglo XIX, continuaron practicándola. La contaminación por fertilizantes ocurre cuando los fertilizantes se aplican en cantidades mayores de las que las plantas pueden absorber, cuando son arrastrados por el agua o el viento de la superficie del suelo antes de que puedan ser absorbidos. La contaminación de los acuíferos por agroquímicos y residuos es uno de los problemas más graves en la mayoría de los países desarrollados y está aumentando en muchos países en desarrollo.

Estudios realizados en Reino Unido han calculado que se absorben entre 50 y 60 kg de nitrógeno por hectárea al año en los cultivos y que el 58 % de los acuíferos se contaminan con nitratos provenientes de la agricultura. (Silva et al., 2017)

El nitrógeno es el fertilizante que más necesitan los cultivos de maíz y otros tipos de plantas para crecer, también es uno de los insumos más costosos en la producción agrícola. "Un agricultor de maíz en Sinaloa, México puede llegar a gastar 1700 dolares por hectárea y de esto, un 40 a 60 % es lo que gasta en fertilizantes; de este porcentaje, hasta un 80 % corresponde al gasto por el nitrógeno". (Silva et al., 2017)

1.2. Antecedentes

El maíz requiere un buen manejo de la fertilidad del suelo, principalmente los llamados híbridos que requieren grandes cantidades de fertilizante para lograr altos rendimientos. Tradicionalmente los agricultores realizan dos fertilizaciones: la primera aplicación durante la siembra del maíz y la segunda cuando la planta tiene 50 cm de altura. La mayoría de los agricultores lo realiza de forma manual, es decir, depositando un puñado de fertilizante al pie de la planta o con una bomba tipo mochila de 15 litros; otros pocos lo hace por medio de aspersores agrícolas para tractor, cuatro en línea que realizan la tarea haciendo más rápida esta opción, pero aun con deficiencias.

1.3. Justificación

El maíz necesita una cierta cantidad de elementos minerales para crecer; las deficiencias de los cultivos ocurren cuando ciertos nutrientes minerales son deficientes o se encuentran en exceso. Se aconseja contar con un suelo rico en potasio y fósforo; también que contenga nitrógeno en mayor cantidad, especialmente durante el período vegetativo.

La fertilización lleva tiempo y dedicación; como lo hemos mencionado anteriormente, aun así, no se garantiza la correcta aplicación y por lo tanto el aumento de cultivo puede verse afectado. Por lo que la propuesta es: detectar e identificar plantas de maíz y diferenciarlas de la maleza usando inteligencia artificial, tomando en cuenta que la mata = *conjunto de plantas de maíz* y una planta de maíz = *un solo tallo*.

1.4. Hipótesis

Los algoritmos de inteligencia artificial aplicados en la agricultura mejorarán los procesos al distinguir entre plantas de maíz y maleza, apoyando a la toma de decisiones de los agricultores, es decir: realizarían tareas específicas y tomarán medidas en la planta haciendo un uso eficiente de fertilizantes y pesticidas a través de intervenciones más específicas, ayudando a la agricultura de precisión para obtener una mayor producción de maíz.

1.5. Objetivos

1.5.1. Objetivo General

Identificar plantas de maíz para diferenciarlas de la maleza usando Visión Artificial a partir de un data set de imágenes.

1.5.2. Objetivos Específicos

- Obtener un data set de imágenes aéreas de plantas de maíz y de maleza.
- Determinar las características de las plantas de maíz para identificarla.
- Implementar algoritmos de inteligencia artificial para clasificación de maíz y maleza.
- Obtener resultados.

1.6. Contribuciones

La presente investigación se desarrolló de la siguiente manera: En el capítulo 1, encontramos la introducción y el planteamiento del problema que consiste en cómo el agricultor mexicano realiza la fertilización a sembradíos de maíz. La fertilización manual es un proceso lento, tedioso y poco efectivo. Las desventajas pueden ser el mal manejo del fertilizante, aplicación errónea y desperdicio del fertilizante.

En el capítulo 2, revisamos el estado del arte. En el Estado del arte se encontraron investigaciones publicadas con la temática relacionada o parecida a nuestra investigación. En el Estado del arte, se encontró información relevante, que sirvió de base para los experimentos diseñados para este trabajo. En los trabajos consultados, se encontró que han usado diferentes tipos de métodos utilizados para la extracción de características y varias técnicas de clasificación de imágenes usando diferentes algoritmos. En el estado del arte, encontramos que las SVM se usaron en el 40 %, el método Otsu en el 20 %, Backpropagation en el 10 % y KNN en el 10 %, y en el restante 20 % se encontraron trabajos con Perceptrón, CNN, Naive Bayes, K-means, árboles de decisión, Fuzzy C-means y cadenas de Markov.

En el capítulo 3, se encuentra el marco teórico donde se presentan los conceptos usados sobre inteligencia artificial como los siguientes, clasificación, segmentación de imágenes y aprendizaje automático. Por otro lado, los algoritmos más utilizados y sus métricas de evaluación. finalmente, aprendizaje profundo, redes neuronales convolucionales y su arquitectura.

El capítulo 4, nosotros incluimos la metodología que consiste, en primer lugar, la búsqueda de Datasets que contienen imágenes de maíz y maleza. En segundo lugar, la revisión de apps móviles para estudiar cómo realizan la identificación de tipos de plantas. En tercer lugar, nosotros generamos un Dataset con imágenes de maíz y maleza de tres estados del país, en este caso fueron: Hidalgo, Puebla y San Luis Potosí. En cuarto lugar, se llevaron a cabo pasos para cada una de las etapas para experimentar, generar resultados y publicar los resultados obtenidos, como se puede apreciar en el diagrama de la metodología. En quinto lugar, se utilizaron algoritmos en Matlab para el pre-procesamiento, extracción de características texturales, cromáticas e híbridas y segmentación

de las imágenes. También, se usaron algoritmos en Weka para la clasificación de las imágenes en las primeras dos etapas. En las etapas restantes, se experimentó con algoritmos de Redes Neuronales Convolucionales (CNN) y Python para la generación de resultados. Este último algoritmo, se combinó con un algoritmo de Máquina de soporte Vectorial (SVM). en este caso, este algoritmo híbrido usa una CNN como extractor de características y la SVM como clasificador.

En el capítulo 5 mostramos los resultados, después de llevar a cabo nuestra experimentación. 1) El mejor porcentaje se obtuvo al hacer uso del método Otsu en la segmentación, además de la extracción de características de textura. En este caso, se usó una Red Backpropagation como clasificador generando como resultado un 98.98 % de Accuracy y un 99 % de precisión. Agregar que, en esta etapa se hizo pre-procesamiento de imágenes. 2) El mejor porcentaje se obtuvo al hacer uso del método Otsu en la segmentación, extracción de características híbridas (texturales y cromáticas) y una Red Backpropagation como clasificador, generando como resultado un 98.89 % de Accuracy y un 98.9 % de precisión. Por otro lado, se agregó el resultado de una CNN. La CNN generó un Accuracy de 97 % y un 96 % de precisión. Finalmente, se realizó una validación cruzada con cinco K-folds y se obtuvo un 95 % de Accuracy y un 93 % de precisión. 3) Para esta etapa, se usaron algoritmos en Python, los cuales arrojaron el mejor porcentaje al combinar con una CNN y una SVM. El modelo híbrido (CNN + SVM) generó un porcentaje de efectividad del 99.72 % de Accuracy y un 99.88 % de Precisión, Este resultado se comparó con 15 experimentos de otros autores y se concluyó que superaba a 13 de los resultados mostrados. La comparación se muestra en una tabla comparativa.

En el capítulo 6, encontramos las conclusiones. Las conclusiones indican que, el uso de algoritmos tradicionales como Naive Bayes, Random Forest, K-nn, SVM y Backpropagation arrojaron buenos resultados en los experimentos. Agregar que, las imágenes requieren de un pre-procesamiento lo que conlleva a más recursos y tiempo computacional. Por otro lado, con la CNN se ha obtenido excelentes resultados en menor tiempo y sin hacer uso de pre-procesamiento de imágenes. Las recomendaciones para ampliar o continuar con el tema de investigación consiste en aumentar la cantidad de imágenes tanto de maíz como de maleza. Las alternativas son: con Data Augmentation o bien obtener imágenes de más estados de la República Mexicana. Se pueden realizar nuevos experimentos con algoritmos como Adaboost, Gboost y XGBoost que son usados en la actualidad para mejorar el rendimiento de los algoritmos de aprendizaje.

Capítulo 2

Estado del Arte

En el trabajo de (Yang, 2000) *Reconocimiento y clasificación de hierba y maíz aplicando redes neuronales artificiales a las imágenes* abordan la problemática de diferenciar las plantas en los cultivos de maíz de las malas hierbas. Con su trabajo buscan desarrollar un modelo de red neuronal artificial (ANN) usando el modelo backpropagation. Los colores asociados con los megapíxeles de la imagen se utilizaron como datos de entrada, con esto se asume que el modelo podrá utilizar otra información como la forma contenida en la imagen. Logrando los siguientes resultados, los píxeles originales de las imágenes tomadas en el campo son de 756 x 504 y luego se disminuyeron a 100 x 100 píxeles, las imágenes representan una sola mata o planta, ya sea de maíz o de maleza; se usaron 40 imágenes de maíz y 40 imágenes de maleza para el entrenamiento de la red neuronal. La tasa de éxito para reconocimiento de la planta de maíz fue de 100% y para la maleza de 80%; se consideró satisfactorio dada la limitada cantidad de datos utilizados para el entrenamiento y las limitaciones del hardware de computadora utilizado. Por lo tanto concluyeron que un sistema de reconocimiento de malezas basado en una red neuronal artificial puede ser utilizado en la aplicación precisa de herbicidas en campos agrícolas.

En la investigación de (Lanlan & You-xian, 2009) *Reconocimiento de maleza/maíz por máquinas de soporte vectorial usando características de textura* resolvió el problema de encontrar un clasificador para identificar malezas en milpas al inicio del crecimiento, haciendo uso de una SVM (Support Vector Machine). Se empleó un clasificador de SVM para reconocer las malezas y plántulas de maíz, la segmentación de la imagen se realizó mediante transformación, las imágenes se utilizaron para obtener la textura de la maleza y el maíz; estas características de textura fueron utilizadas en el procedimiento de clasificación. Los resultados muestran que los clasificadores SVM con diferentes selecciones de características pueden identificar con éxito malezas en cultivos de maíz con un 100% de precisión.

En el estudio de (Ramírez, 2012) *Desarrollo de un método de procesamiento de imágenes para la discriminación de maleza en cultivos de maíz* aborda el problema que es identificar y distinguir malezas en los plantíos de maíz. Con su trabajo desarrolló e implementó un método basado en el análisis de imágenes para identificar malezas, cultivos y suelos presentes en las plantaciones de maíz. Se usó una base de datos de 200 imágenes de las cuales 100 fueron para entrenar el sistema y las otras 100 para la validación del mismo. La primera etapa fue segmentación, la segunda etapa se dividió en 3 secciones, primero el análisis estructural wavelet, segundo extracción de características y tercero la clasificación de maleza o planta de maíz gracias a una combinación de técnicas de segmentación fundamentadas en redes neuronales (perceptrones multicapa) y detección directa. Los resultados muestran la eficacia del método con una tasa de clasificación del 94.77%.

El investigador (Muñoz, 2011) en su trabajo *Reconocimiento de malezas por medio de matrices de coocurrencia, dimensión fractal y redes neuronales artificiales* aborda el problema del reconocimiento automático de malezas en épocas tempranas de un cultivo como consecuencia de la necesidad de regular el uso de herbicidas y sus efectos secundarios no deseados. Con su trabajo busca implementar un método de clasificación automático de hojas de maleza asociadas al cultivo de maíz, por medio del uso combinado de matrices de coocurrencia y de la dimensión fractal. Cuatro grupos de malezas son estudiadas, en conjunto con uno de hojas de maíz de menos de un mes de gestación. Por cada grupo son objeto de análisis 20 imágenes, tomando 10 para el entrenamiento de una red neuronal artificial en configuración backpropagation, y 10 para su validación posterior. El porcentaje de aciertos sobre errores es determinado con base en los falsos negativos y falsos positivos del sistema. Hasta la fecha no existen resultados.

En la publicación de (Nowakowski et al., 2011) *Proceso de identificación de daños en granos de maíz y cebada utilizando análisis de imagen neuronal* abordan el problema de identificación de daños mecánicos a las cariopses de grano basadas en fotografías digitales. Con su trabajo buscan identificar un conjunto de características universales que ayudan a distinguir las cariopses sanas de las dañadas. Como resultado de este estudio, se ha realizado una red neuronal artificial de un tipo de perceptron multicapa cuya capacidad de identificación está cerca de la humana. La tasa de error global promedio aproximada del modelo es del 4,5% y la tasa de error promedio aproximada de los humanos es del 2%.

En el trabajo de investigación de (Zaforas, 2017) *Inteligencia Artificial como servicio: reconocimiento de imágenes*” aborda el problema de detección y clasificación de imágenes. Busca abordar proyectos que integren técnicas avanzadas de Inteligencia artificial; en el caso de las técnicas de reconocimiento de imágenes, una alternativa muy interesante es apoyarnos en los modelos pre-entrenados que las principales plataformas cloud nos ofrecen. Se analizaron los servicios que ofrecen AWS (Servicios web de Amazon), Google Cloud y Microsoft Azure. Como resultado del estudio la mejor plataforma es AWS, pero no se descartan las otras dos debido a que cumplen características diferentes, es decir, pueden servir para reconocer otro tipo de imágenes.

El autor (Baruffaldi, 2016) en su trabajo *Redes neuronales adversarias para el reconocimiento de malezas* dedica su investigación para resolver el problema de la identificación de malezas para implementar la fumigación selectiva en campos de cultivo; junto con las técnicas de Deep Learning, donde los datos de entrenamiento son imágenes con la presencia de cultivo y maleza. El sistema de visión propuesto está basado en Convolutional Neural Networks (CNN). Se utilizó la técnica de Generative Adversarial Networks (GAN) para hacer un pre-entrenamiento no supervisado del modelo de modo de explotar la gran cantidad de imágenes que se obtienen a partir de secuencias de video y se utiliza una SVM en la última capa de la red entrenada sobre datos etiquetados utilizando Data Augmentation para lograr mayor robustez. El resultado fue la identificación de malezas en un 91% con SVM y un 84% con K-means (método de agrupamiento).

En la investigación de (Padilla, 2016) *Robot móvil terrestre para la aplicación de agroquímicos en invernaderos controlados* se encontró que el problema del uso de una plataforma móvil automática está siendo desarrollado para apoyar las tareas agrícolas precisas. Se desarrollo una plataforma móvil automatizada que puede dividirse en varios componentes: de dirección, sistemas de detección, de control de flujo y sistema de seguridad. Como resultado: la técnica de control más adecuada para el manejo de motores de corriente continua de baja potencia, es el control proporcional, dado

que no se requiere controlar el arranque de los mismos, más si su velocidad de funcionamiento. Y esto se lo realiza mediante el uso de ancho de pulsos configurados digitalmente en el controlador.

En el trabajo de (Baez, 2018) *Los robots: un nuevo paradigma en la agricultura* se aborda la problemática de un paradigma en las personas que no les gusta utilizar tecnología en sus cultivos agrícolas; se basa en el trabajo focalizado, micro trabajos, pequeñas máquinas trabajando en red y conectadas al Big Data; proporcionan incrementos de productividad en las explotaciones más pequeñas o en cultivos específicos. Como resultado se generará más producción por hectárea, se generarán nuevos puestos tecnológicos en la industria agrícola y se hará un uso eficiente de los agroquímicos.

Se encontró que (Martínez, 2017) en su trabajo *Reconocimiento de imágenes con Redes Convolucionales en C* se refiere a cómo crear una red neuronal artificial, específicamente una red convolucional para el reconocimiento de símbolos. Es una red supervisada, formada a partir de un conjunto muy grande de símbolos proporcionados por el conjunto de datos MNIST. Como resultado, una vez que la red ha sido entrenada, debería poder clasificar un símbolo (ya sea del MNIST o del propio autor) con alta confianza, es decir, del 90%.

En su trabajo (Haug et al., 2014) propuso un enfoque de visión artificial para distinguir los cultivos de zanahoria de las malas hierbas, logrando una precisión del 93.8%. Las imágenes se adquirieron con una cámara monocular multiespectral que mira hacia abajo en el espectro visible rojo R e infrarrojo cercano (NIR). El suelo se elimina aplicando los Índices de Vegetación de Diferencia Normalizada (NDVI), a cada píxel en la imagen de entrada. Se selecciona un umbral en el espacio NDVI utilizando el método de Otsu y se enmascaran todos los píxeles con valores NDVI más pequeños que el umbral.

Varios autores, entre ellos (Hlaing & Khaing, 2014a) desarrollaron un método para distinguir entre cultivos y malas hierbas con una tasa de clasificación errónea del 33.3%. Los valores absolutos de verde menos rojo y verde menos azul se calculan en cada píxel de la imagen. Si ambos valores son mayores que un valor umbral, el píxel se clasifica como planta, de lo contrario, el píxel se clasifica como fondo. Los píxeles de las plantas se evalúan con una función de decisión proporcionada por un clasificador para discriminar entre cultivos y malezas.

En su investigación (Tewari et al., 2014) diseñó y desarrolló un microcontrolador para la aplicación de herbicidas específicos del sitio con una técnica de detección automática de malezas en imágenes de campos de cultivo. Para la detección de malezas, cada píxel de la imagen se procesa de la siguiente manera. Cuando la intensidad del color verde es mayor que las intensidades del color rojo y azul, se supone que el píxel es verde.

El número total de píxeles verdes se divide por el tamaño de la imagen, Este valor representa el porcentaje de malezas en la imagen. Informaron una eficiencia de deshierbe del 90%.

En el artículo de (Wei et al., 2014) se propone un método automatizado para separar frutos maduros de un campo con agricultura compleja obteniendo una precisión del 95%; basado en un algoritmo mejorado que se fundamenta en el umbral de Otsu utilizando una nueva función en el espacio de color OHTA.

La investigación de (Choi et al., 2015) introdujo un nuevo algoritmo de extracción de línea de guía para mejorar la precisión de navegación de los robots de desmalezado en los arrozales. El error

del ángulo de inclinación para el centro de la línea de guía es inferior a 1° , sin embargo, aumenta gradualmente con el tamaño de la planta. Las filas de arroz se detectan a partir de imágenes adquiridas con una cámara equipada con un filtro de paso de banda IR (>795 nm). Cada imagen se filtra con un filtro mediano para eliminar el ruido, después de eso, la imagen se procesa con el método de Otsu para obtener la imagen binaria de las filas de arroz.

El investigador (Torres-Sánchez et al., 2015) propone un método automatizado para la detección de vegetación en cultivos cuyas plantas tienen consistencia herbácea en la parte aérea, haciendo uso de un algoritmo de análisis de imágenes que se basa en objetos a través del método Otsu. Las imágenes fueron capturadas con dos sensores (una cámara convencional visible y una cámara multispectral) montadas en un vehículo aéreo no tripulado y adquiridas sobre los campos de maíz, girasol y trigo. El error de clasificación de la vegetación varía entre 0% y 10%.

El trabajo de (Jiang et al., 2015) describe un algoritmo de detección de hileras de cultivos basado en múltiples ROI para imágenes agronómicas (trigo, maíz y soja), informaron una precisión del 93%. La detección de la fila de cultivo se realiza sobre la vegetación, una vez que se ha segmentado de acuerdo con el proceso de índices de color de la vegetación.

En la propuesta de (Meng et al., 2015) se desarrolló un sistema agrícola para el desmalezado entre hileras de cultivos. Las imágenes se procesaron en el espacio HSI para detectar vegetación (los píxeles se consideran vegetación si su valor de Hue está entre 120 y 160). Desde la detección verde se realiza la detección de filas de cultivo. El error promedio para la localización de la línea de cultivo es inferior a 2,7 cm, con un valor de error máximo de 6,8 cm. La eficiencia depende de la segmentación de la vegetación, sin embargo, no se proporcionan los resultados de la segmentación.

El autor (Guijarro et al., 2015) introdujo una estrategia para distinguir entre el suelo y las plantas verdes en las imágenes de los campos de maíz con una precisión del 92.09%. La metodología incluye: estimación de wavelets discretas, extracción de descriptores de textura y un proceso de umbral. Durante el proceso de umbralización, se extraen píxeles con un valor verde espectral que es 10% más alto, en relación con los otros dos componentes. Luego, un experto humano eliminó los falsos positivos manualmente.

En el artículo de (Balasubramaniam & Ananthi, 2016) proporciona un método para segmentar regiones deficientes en nutrientes en imágenes de diferentes cultivos que contienen píxeles faltantes. Alcanzaron un valor de precisión entre 92% y 99%, no se muestran los resultados de segmentación de vegetación y su método se basó en un algoritmo intuitivo de fuzzy C-means.

En el proyecto de investigación de (Kazmi et al., 2015) se presentó una metodología para detectar cardo en campos de remolacha azucarera con una precisión de hasta el 97%, mediante el uso de una función de clasificación construida con la distancia de Mahalanobis y el análisis discriminante lineal. Los datos de entrada son la combinación de varios índices de vegetación de color. Tiene otra propuesta similar para la detección de malezas en imágenes de hojas escaneadas mediante el uso del esquema Bag-Of-Words con K-Nearest Neighbours (KNN) y Máquina de soporte de vectores (SVM). Los datos de entrada son la combinación de CVI logrando una precisión de clasificación general superior al 99%.

En la obra de (Ye et al., 2015) introdujo un método probabilístico de campo aleatorio de Markov superpíxel. Se aplicó para la detección de cultivos bajo fuertes cambios de iluminación. El método

produce el valor medio más alto de 92.29 % con la desviación estándar más baja de 4.65 %.

La aportación de (Cheng & Matson, 2015) propone un método para la identificación de malezas y arroz basado en análisis de características de imágenes y técnicas basadas en aprendizaje automático que logran una precisión del 98.8 %. Para clasificar la maleza del arroz, primero se aplica el buscador de puntos de Harris para ubicar los puntos de interés, luego se extraen múltiples características que posteriormente se clasifican utilizando un algoritmo basado en aprendizaje automático (árboles de decisión).

Un trabajo por demás importante es el de (Parvathi et al., 2015) quien desarrolló un método de segmentación de la vegetación en imágenes de cultivos de betabel y maíz con precisión del 87 %. Su método se basa en un modelo bayesiano simple que utiliza las propiedades de los espacios de color RGB y HSV. En este trabajo, las características utilizadas para entrenar el modelo se extrajeron manualmente.

La aportación de (Montalvo Martínez et al., 2016) proporcionó un método basado en componentes principales y métricas de CVI para dividir el suelo de las plantas, con un error de 8.07 %. Las imágenes de verdad real se construyeron semiautomáticamente procesando los canales de color RGB y con la ayuda de un experto humano.

Referente a los tallos del maíz, se ha encontrado en los estudios revisados que solo se mencionan en los títulos a los tallos como un complemento (Chua, 2016), (Suzuki et al., 2016), (Pueschel et al., 2013); son pocas las investigaciones que involucran directamente estudios sobre los tallos, en los cuales se encuentran los siguientes: (Muñoz, 2011) ha encontrado que existen enfermedades en tallos de los manzanos; (Chen et al., 2016) realizó una comparativa proteómica de las hojas encontradas en diferentes posiciones del tallo de las plántulas de maíz y (Liang et al., 2020) realizó una detección de frutas y tallos de litchi.

Existen varias características que ayudan a identificar plantas y malezas, entre las que se incluyen: textura, color y forma (Cervantes et al., 2018). Algunas investigaciones actuales en las cuales su enfoque ha sido centrarse en las características de histogramas y wavelet (Chau et al., 2017), representación dispersa (feng Wang et al., 2019), madurez de las hojas, movimientos rotativos y calibración (Imanov & Alzouhbi, 2018), utilizando SVM (Niño et al., 2019), CNN (Zheng et al., 2019), FCN (Zhang et al., 2019), KNN (Imanov & Alzouhbi, 2018) como clasificadores con los mejores resultados.

Otros investigadores como (Aakif & Khan, 2015) propuso clasificar las plantas según sus hojas mediante la identificación de características morfológicas y descriptores de Fourier, utilizando redes neuronales artificiales (ANN por sus siglas en inglés) en combinación con Backpropagation como clasificador; (Cervantes et al., 2018) utilizó técnicas de selección de características mediante la eliminación de propiedades que producen ruido en las imágenes, lo que facilita su uso en pequeños conjuntos de datos; (Cervantes et al., 2017) Utilizó técnicas para extraer características geométricas sin variación a escala, rotación y traslación, de textura y color. Utilizando el clasificador bayesiano, el algoritmo Backpropagation y el clasificador SVM (Dong et al., 2019) propuso un modelo de reconocimiento de características basado en la red de cápsulas y (García et al., 2016) de igual forma propuso caracterizar frutas empleando descriptores de Fourier y momentos de Hu como descriptores complementarios, la principal contribución del artículo es la caracterización del color.

Algunos estudios muy completos sobre la segmentación de imágenes por características del color, como el estudio de (Garcia-Lamont et al., 2018), se basan en las características más relevantes del color, es decir, la intensidad y cromaticidad; también menciona los espacios de color más utilizados en obras relacionadas y caracteriza estos espacios comparando los pros y los contras de su uso. (Jalili et al., 2016) propuso el uso de un algoritmo genético para reducir el ruido con la característica de dimensionalidad. También (Zhao et al., 2015) propone describir las características de la imagen por forma, tipo de margen y geometría debido a que estas características tienen morfologías comunes en hojas de muchas especies diferentes.

Hoy en día las ciencias computacionales se han dedicado a resolver problemas en el entorno en el que vivimos, sus derivaciones como el procesamiento digital de imágenes y el aprendizaje automático son áreas fundamentales para la obtención de resultados destacados, convirtiéndose así en técnicas fundamentales o métodos para reconocimiento de imágenes a través de visión artificial.

Actualmente se ha implementado el uso de algoritmos de visión artificial en múltiples campos incluido la agricultura, las aplicaciones en este sector son variadas pues van desde detección de frutas (Garcia et al., 2016), detección de plantas (Cervantes et al., 2017), reconocimiento de plantas a partir de hojas (Niño et al., 2019), clasificación de enfermedades en los cultivos (Zhang et al., 2019) y reconocimiento de malezas (feng Wang et al., 2019). Esto impacta directamente la producción del cultivo reduciendo los costos de los productores, por lo que las ciencias computacionales han hecho grandes aportes intentando dar solución a este problema, sin embargo y pese al esfuerzo la mejor solución no ha sido presentada.

Algunos investigadores que se encargan del estudio y clasificación de las plantas, realizan la detección mediante métodos oculares, considerando procedimientos que no son exactos, sin embargo, en la literatura existen trabajos enfocados al reconocimiento de plantas, logrando su objetivo con propuestas propias usando técnicas de aprendizaje profundo, usando redes neuronales convolucionales (CNN) por sus siglas en inglés (Jiao et al., 2019) (Zheng et al., 2019).

En investigaciones anteriores, existen contribuciones en el campo de la segmentación en imágenes de color y escala de grises, considerada un área de grandes oportunidades que aún se estudia rigurosamente, tanto para entornos controlados como los no controlados; por lo que se han desarrollado revisiones en profundidad de los trabajos relacionados con la segmentación de imágenes en color (Garcia-Lamont et al., 2018) y algunos otros trabajos como el de (feng Wang et al., 2019) han segmentado con redes FCN totalmente convolucionales.

La revisión de literatura nos ha mostrado que se ha desarrollado la detección de enfermedades en diferentes plantas, usando técnicas de extracción de características GWT y clasificando con máquinas de vectores de soporte (SVM) (Prasad et al., 2012). Sumado a esto, la ola de implementación de aprendizaje profundo ha tomado mucha fuerza buscando resolver problemas de distintas disciplinas; se ha hecho uso de las redes neuronales CNN para la detección de plagas y enfermedades en las plantas de jitomate (Fuentes et al., 2017) y en algunos casos las técnicas de aprendizaje profundo como las de aprendizaje automático se han fusionado con el mismo propósito (Shijie et al., 2017); sin dejar atrás el desarrollo de sistemas robóticos en conjunto con métodos de visión artificial (Schor et al., 2016).

El desarrollo de esta investigación se ha destacado por el bajo costo computacional comparado con otros como (Zhang et al., 2018) (Durmus et al., 2017) ya que para el entrenamiento y pruebas

utilizan hardware adicional y completamente dedicado.

Una serie de estudios que incluyen enfoques de aprendizaje profundo han reivindicado el rendimiento de vanguardia en un número considerable de tareas. En las que se incluyen, la clasificación de imágenes, el procesamiento del lenguaje natural, el reconocimiento del habla, clasificación de símbolos y textos, además de la clasificación de plantas, maleza y enfermedades en diversos cultivos.

El trabajo de (Silva et al., 2017) presentan una herramienta para el reconocimiento de marcas de ganado que utiliza CNN y SVM. En los experimentos, los autores utilizaron 12 marcas de ganado y un conjunto de 540 imágenes y la precisión obtenida en el experimento alcanzó el 93.28 %. El mismo autor (Silva & Welfer, 2019) en otra investigación realizó dos experimentos alcanzando índices de 93.11 % y 95.34 %, respectivamente, en el reconocimiento de marcas de ganado.

Los investigadores (Niu & Suen, 2012) presentan un algoritmo que utiliza CNN y SVM para resolver el problema del reconocimiento de textos con una tasa de reconocimiento del 94,40 %. (Agarap, 2017) propone un modelo híbrido capaz de reconocer símbolos de la base de Datos MNIST con una precisión del 91.86 %.

Existen excelentes precisiones como el propuesto por (Basly et al., 2020) para reconocimiento de actividad humana usando una CNN pre-entrenada con 1,2 millones de imágenes de alta resolución del subconjunto de entrenamiento de clasificación ILSVRC2015 del conjunto de datos ImageNet. Lo cual generó un resultado de 99.92 % de Accuracy; es importante mencionar que usaron una red pre entrenada con más de 14 millones de imágenes, debido a esto fue que sus pruebas han arrojado un excelente porcentaje.

El uso del aprendizaje profundo también se describe en la investigación realizada por (P. Constante, 2016), quienes utilizaron una red neuronal de tres capas con entrada a través de backpropagation; utilizaron este método para clasificar fresas y obtuvieron resultados de reconocimiento de 92.5%(Garcia et al., 2016) clasifican 20 clases de frutas comunes en México, las frutas mejores clasificadas en los experimentos fueron: piña y limón con un porcentaje de efectividad del 97.68 % para ambas.

(Cervantes et al., 2018) Menciona que los métodos actuales de reconocimiento involucran algoritmos avanzados para medir las características de forma y textura de los objetos presentes en las imágenes, ya que brindan una gran cantidad de información para su clasificación. (Yang, 2000) Resuelve el problema de distinguir las plantas de maíz de la maleza implementando una red neuronal artificial (ANN), mediante el uso del algoritmo Backpropagation que tiene una tasa de detección del 100 % en maíz y del 80 % en maleza, es decir, obtiene una media del 90 %.

(Hlaing & Khaing, 2014b) Desarrollaron un método para distinguir entre plantas y malas hierbas con una tasa de clasificación del 66.7%. (E.G. et al., 2020)usaron técnicas tradicionales de procesamiento de imágenes para la detección de enfermedades y plagas en la planta de jitomate, donde el algoritmo SVM fue el que mejor resultado arrojó con un 93.86 % de efectividad.

En algunos casos la identificación de los cultivos y las malas hierbas fue posible gracias a la técnica SVM. El objetivo era identificar cultivos, suelos y malas hierbas. La identificación de cultivos y malas hierbas permite, sin duda, diversas aplicaciones futuras. La propuesta alcanzó un 94,3 % de precisión en la identificación de tres categorías: cultivos, suelos y malezas. La solución podría

utilizarse en vehículos, tractores autónomos ([Campos et al., 2017](#)) y robots.

En la literatura encontramos que existen métodos parecidos al propuesto, como el de ([Campos et al., 2017](#)) que clasifica cultivo, suelo y maleza con un porcentaje de precisión del 94.3 %; ([Jiang et al., 2020](#)) clasificó maíz y maleza con una efectividad de 96.51 %. ([Li et al., 2020](#)) clasificó maíz, manzana y uva con una precisión del 94 %.

En la literatura revisada, existen múltiples métodos de extracción de características y técnicas de clasificación de imágenes utilizando diferentes algoritmos. Actualmente, vemos que la visión artificial es la tecnología clave para identificar con precisión las plantas de maíz y distinguirlas de las malezas para poder hacer una fertilización completa, haciéndola de manera automática y precisa. ([Gómez et al., 2020](#))

Capítulo 3

Metodología

En la actualidad existen universidades, gobiernos, empresas y muchas otras instituciones que invierten grandes cantidades de dinero en proyectos de investigación relacionados con la *Inteligencia Artificial*, ya que existen muchos ejemplos que ofrecen los resultados esperados por alguna de estas compañías en los ámbitos de mercado de consumo, lucha contra spam, empresas automovilísticas, usos militares y agrícolas.

La idea principal es que, aunque existe una gran diferencia entre los sistemas naturales y los artificiales, hasta cierto punto de abstracción, ambos pueden describirse como sistemas que tratan con objetos abstractos a través de un conjunto de reglas. Las aplicaciones más frecuentes de la inteligencia artificial incluyen áreas como la robótica, el análisis y procesamiento de imágenes, además del procesamiento automatizado de textos. (Iglesias, 2013)

Por ahora, no se trata de obtener los datos, porque están fácilmente disponibles; el reto es poder extraer información de los datos, es decir, darle sentido y sacar conclusiones útiles.

3.1. Conceptos Básicos sobre Inteligencia Artificial

La inteligencia artificial (IA) es una disciplina académica relacionada con la teoría de la computación cuyo objetivo es emular algunas de las facultades intelectuales humanas en sistemas artificiales. Con inteligencia humana nos referimos típicamente a procesos de percepción sensorial (visión, audición, etc.) y a sus consiguientes procesos de reconocimiento de patrones, por lo que las aplicaciones más habituales de la IA son el tratamiento de datos y la identificación de sistemas. (Iglesias, 2013)

Aprendizaje automático: Esta es una condición necesaria para que una entidad artificial sea considerada inteligente. Si una máquina es incapaz de aprender cosas nuevas, le resultará difícil adaptarse a su entorno, un requisito de todos los seres inteligentes.

Visión artificial: A través de la cámara, la computadora puede ver el mundo a su alrededor, pero tomar la imagen es una cosa y comprenderla es otra. La visión artificial busca que los sistemas inteligentes puedan interpretar su entorno a partir de imágenes.

Manipulación de objetos: Utiliza visión artificial o (alternativamente algún tipo de sensor) que le permite a la computadora saber dónde está el objeto, qué forma tiene, hacerse una idea de su peso, etc. Además, debe ser capaz de aplicar la presión correcta para no dañar el objeto y saber

qué movimientos realizar para llevar el objeto a su nuevo destino. (Serrano, 2016)

Sistema de visión y procesamiento de imágenes: Está formado por una serie de subsistemas que actúan sobre una escena para interpretar una característica destacable.

La materia prima del procesamiento y la visión, es una imagen, considerada una representación del mundo físico que contiene información relevante, recopilada mediante un proceso de muestreo generalmente por medios electrónicos. Para obtener una imagen digital se requiere un proceso que incluye captura, muestreo, cuantificación y codificación.

Procesamiento digital de imágenes: Es la manipulación de imágenes por computadora, el tipo de operación realizada coincide a nivel de procesos que construyen el sistema de procesamiento de una imagen (bajo nivel, medio nivel y alto nivel).

Imágenes a color: El modelo de color RGB se basa en una combinación de los colores primarios rojo (Red), verde (Green) y azul (Blue). El origen de este patrón está en la tecnología de la televisión y puede considerarse como la representación básica del color en ordenadores, cámaras digitales y escáneres, así como en el almacenamiento de imágenes. La mayoría del software para el procesamiento de imágenes utiliza este modelo para la representación interna del color. (Cuevas et al., 2010)

Extracción y selección de atributos: Suponiendo que se quiere analizar un conjunto de datos para resolver algún problema. Antes de proceder con el análisis de datos, es necesario encontrar una forma adecuada de representarlos. Una buena presentación de datos es esencial, porque los datos que a menudo son redundantes o contienen información errónea no ayudarán a la prueba.

La simplificación de datos también se conoce como reducción de dimensionesz tiene como objetivo eliminar aspectos redundantes como las correlaciones entre variables (redundancia) o las fluctuaciones aleatorias (ruido). Estos aspectos se agrupan en lo que solemos llamar ruido, es decir, todos aquellos factores que no representan información relevante del proceso que estaremos estudiando. En cambio, llamaremos a las "señales" factores que contribuyen de manera relevante a los datos. Entonces, en general, la siguiente identidad se mantendrá $Datos = señal + ruido$

Extracción de características de imágenes: La visión por computadora es el estudio y diseño de artefactos computacionales que procesan y entienden las imágenes. Estos artefactos a veces emplean aprendizaje automático. Se revisan algunas técnicas básicas utilizadas en visión artificial para representar imágenes en máquinas con problemas de aprendizaje.

Tareas de aprendizaje automático: Dos de las tareas de aprendizaje automático supervisado más comunes son la clasificación y regresión. En las tareas de clasificación, el programa debe aprender a predecir valores para las variables de respuesta de una o más variables explicativas. Es decir, el programa debe predecir la categoría, clase o etiqueta más probable para observaciones. Las aplicaciones de clasificación incluyen la predicción si el precio de una acción subirá o bajará, o decidir si un artículo de noticias pertenece a la política o a la sección de ocio. En problemas de regresión, el programa debe predecir el valor de una variable de respuesta. Algunos de los ejemplos de problemas de regresión incluyen predecir las ventas de un nuevo producto, o el salario de un trabajo según su descripción. Similar a la clasificación, los problemas de regresión requieren un aprendizaje supervisado.

Tareas de aprendizaje no supervisado: Una tarea de aprendizaje no supervisado común es descubrir grupos de observaciones relacionadas, llamados clústeres, dentro de los datos de entrenamiento. Esta tarea, denominada agrupación o análisis de agrupaciones, asigna observaciones a grupos de modo que las observaciones dentro de los grupos sean más similares entre sí basándose en alguna medida de similitud de lo que son con las observaciones en otros grupos. La agrupación en clústeres se utiliza a menudo para explorar un conjunto de datos. Por ejemplo, dada una colección de críticas de películas, un algoritmo de agrupación podría descubrir conjuntos de críticas positivas y negativas. El sistema no podrá etiquetar los clústeres como "positivos" o "negativos" sin supervisión, solo tendrá conocimiento de que las observaciones agrupadas son similares entre sí en cierta medida. Una aplicación común de la agrupación en clústeres es descubrir segmentos de clientes dentro de un mercado para un producto. Entendiendo qué atributos son comunes a grupos particulares de clientes, los especialistas en marketing pueden decidir qué aspectos de sus campañas necesitan ser enfatizados. La agrupación en clústeres también es utilizada por servicios de radio por internet; por ejemplo, dada una colección de canciones, un algoritmo de agrupación tal vez pueda agrupar las canciones según sus géneros. Usando diferentes medidas de similitud, el mismo algoritmo de agrupamiento podría agrupar las canciones por sus claves o por los instrumentos que contienen.

La **reducción de la dimensionalidad** es otra tarea común de aprendizaje no supervisado. Algunos de los problemas pueden contener miles o incluso millones de variables explicativas, que pueden hacer computacionalmente costosas trabajar con ellas. Además, la capacidad del programa para generalizar puede reducirse si algunas de las variables explicativas capturan ruido o son irrelevantes para la relación subyacente. La reducción de la dimensionalidad es el proceso de descubrir las variables explicativas que dan cuenta de los mayores cambios en la variable de respuesta. La reducción de dimensionalidad también se puede utilizar para visualizar datos. Es fácil de visualizar un problema de regresión, por ejemplo: como predecir el precio de una casa desde su tamaño; el tamaño de la casa se puede trazar en el eje x del gráfico, y el precio de la casa se puede trazar en el eje y. De igual forma, es fácil visualizar el precio de la vivienda. El problema de regresión surge cuando se agrega una segunda variable explicativa. El número de los baños de la casa podrían trazarse en el eje z, por ejemplo; sin embargo, miles de variables explicativas se vuelven imposibles de visualizar.

Datos Desbalanceados: Se refiere a una situación en la que el número de observaciones no es el mismo para todas las clases de un Dataset usado para clasificación. Hay varios métodos de muestreo que han sido diseñados para tratar datos desbalanceados, los cuales pueden ser agrupados en cuatro categorías: *Submuestreo*, *Sobremuestreo*, *Generación de datos sintéticos* y *aprendizaje sensible al costo*. Todos estos métodos modifican la proporción de las clases y el tamaño del Dataset original. Debido a esto, lo que puede provocar un conjunto de datos desbalanceado al clasificar es un *Desequilibrio en las clases* ya que, los clasificadores cometen errores y podrían realizar una predicción incorrecta si no obtienen la cantidad de información necesaria.

Validación cruzada: Cross-validation es una técnica utilizada para determinar los resultados de un análisis estadístico y asegurar que sean diferentes de la partición entre datos de entrenamiento y de prueba. Consiste en la repetición y cálculo de la media aritmética obtenida de las medidas de evaluación sobre diferentes particiones. Se utiliza en entornos donde el objetivo principal es pronosticar y estimar la precisión de un modelo que se debe realizar en la práctica.

K-Folds: En la validación cruzada, los datos de muestra se dividen en k *subconjuntos* , uno de los subconjuntos se utiliza como datos de prueba y el resto como datos de entrenamiento. El proceso de validación cruzada es repetido durante k *iteraciones* , con cada uno de los posibles subconjuntos de datos de prueba. Finalmente se realiza la media aritmética de los resultados de cada iteración para obtener un único resultado.

¿Por qué usar la validación cruzada garantiza mejores resultados?: La validación cruzada es una forma de predecir el ajuste de un modelo a un conjunto de datos de prueba hipotético, cuando el conjunto de datos de prueba claramente no está disponible. Por lo que *garantiza la calidad del modelo y la posibilidad de generalizarlo.*

3.2. Metodología Implementada

En este apartado, se muestra el sistema propuesto que permite la identificación precisa de plantas de maíz y maleza en los cultivos, implementando técnicas de extracción de características y utilizando algoritmos de inteligencia artificial que contribuyen a la eficiencia en la aplicación y uso del fertilizante en los plantíos de maíz en México, reduciendo la pérdida económica de los agricultores y pretendiendo evitar la contaminación de suelos y mantos acuíferos. El método adoptado está representado por tres módulos, segmentación, extracción de características y clasificación, tal como lo sugiere (E.G. et al., 2020).

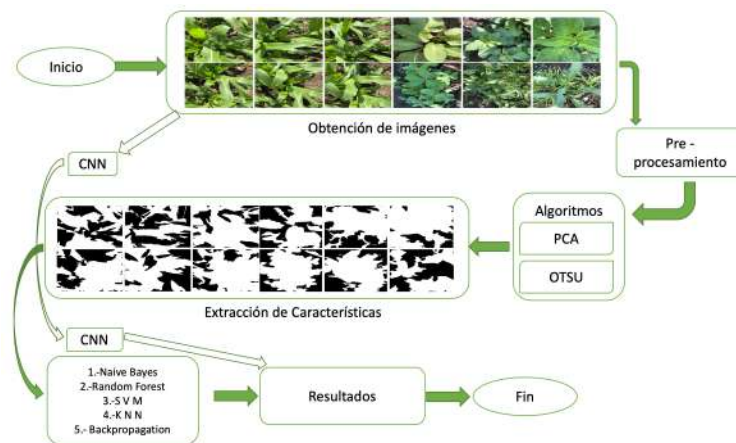


Figura 3.1: Metodología implementada en la Primera etapa (Publicada en RCS) (Gómez et al., 2020)

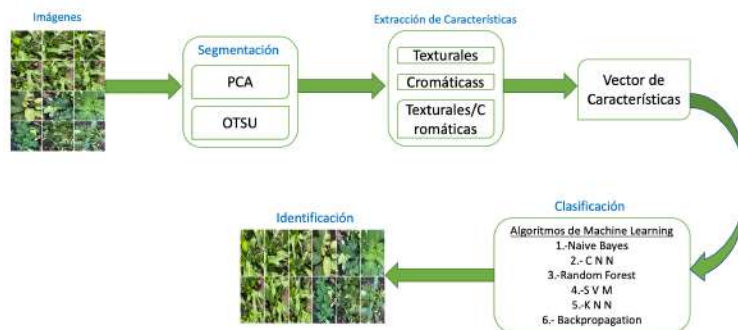


Figura 3.2: Metodología implementada en la Segunda etapa (Publicada en IJCOPI) (Gómez et al., 2021)

3.2.1. Captura de las imágenes de plantas de maíz y maleza para elaboración de dataset propio.

Como se mencionó, lo complicado en esta fase de experimentación ha sido encontrar un conjunto de datos específico sobre plantas de maíz con las características necesarias para los experimentos a realizar, los conjuntos de datos que se encuentran en la web no coinciden o las imágenes no son suficientes para los experimentos que se buscan realizar. Por lo que se ha decidido crear un dataset propio, que contiene por el momento 1480 imágenes de alta resolución de las cuales se pretende

obtener la información necesaria para realizar las primeras pruebas en el software Matlab y en el lenguaje de programación Python.

Todas las fotos fueron tomadas al aire libre; para las primeras pruebas se obtuvieron 1480 fotografías, la captura se realizó con un teléfono móvil iPhone 8 plus con cámara de 12 megapíxeles y se usó un selfie stick como ayuda para una mejor toma aérea.

Las características completas del iPhone 8 plus son las siguientes:

Pantalla Retina HD 5.5”, Pantalla widescreen LCD Multi-Touch de 5.5 pulgadas (diagonal) con tecnología IPS, Resolución de 1920 x 1080 pixeles a 401 ppi, Relación de contraste 1300:1 (normal). Pantalla True Tone, Pixeles dual-domain para ángulos de visualización amplios, Zoom de pantalla, Alcance fácil, Sistema de dos cámaras de 12 MP: gran angular y teleobjetivo. Gran angular: apertura de 1.8 y Teleobjetivo: apertura de 2.8 Modo Retrato. Iluminación de Retrato con cinco efectos (Luz Natural, Luz de Estudio Fotográfico, Luz al Contorno, Reflector, Reflector B y N). Estabilización óptica de imagen. HDR Automático para fotos. Estabilización automática de imagen. Captura de imagen en formatos HEIF y JPEG. Grabación de video 4K a 24 cps, 30 cps o 60 cps. Grabación de video HD de 1080p a 30 cps o 60 cps. Grabación de video HD de 720p a 30 cps. Estabilización óptica de imagen para video. Zoom óptico de 2x para acercar; zoom digital de hasta 6x. Flash LED True Tone. Compatibilidad con video en cámara lenta de 1080p a 120 cps o 240 cps. Video en cámara rápida con estabilización. Estabilización cinemática de video (1080p y 720p). Video con autoenfoco continuo. Toma fotos de 8 MP mientras graba videos 4K. Zoom de reproducción. Grabación de video en formatos HEVC y H.264. Cámara FaceTime HD de 7 MP. Grabación de video HD de 1080p a 30 cps. Retina Flash. HDR Automático para fotos. Estabilización automática de imagen. Modo Ráfaga. Chip A11 Bionic Neural Engine. Sistema Operativo iOS 13.

3.2.2. Características y muestras de las imágenes contenidas en el Dataset

Clase	Tipo	Cantidad	Anchura	Altura	Formato digital
1	Maíz	500	768	1024	JPG
2	Maleza	500	768	1024	JPG

Tabla 3.1: Características de imágenes del Dataset.



Figura 3.3: Imágenes de maíz y maleza.

3.3. Herramientas Matemáticas

3.3.1. Segmentación

Las imágenes de maíz y maleza fueron segmentadas empleando los siguientes pasos: 1) Cálculo de niveles de gris en alto contraste a partir de una combinación lineal óptima de componentes de color RGB (Gonzalez et al., 2004) (Sonka et al., 2010); 2) Estimación del límite óptimo ejecutando el algoritmo de segmentación de borde adaptativo *Otsu* (Gonzalez et al., 2004) (Sonka et al., 2010) y una fase del análisis de componentes principales *PCA* (Schor et al., 2016); 3) Procedimientos morfológicos para llenar los espacios vacíos posibles en la imagen segmentada (Gonzalez et al., 2004) (Sonka et al., 2010). Todo esto para obtener la mejor segmentación, incluso con cambios en las condiciones generales de iluminación. Al segmentar la imagen, el sistema puede usar solo la región de las hojas, determinar su contorno y calcular los atributos por extracción de características. (Cervantes et al., 2017) Cuando una imagen se segmenta con éxito, el sistema utiliza la región de interés y calcula los atributos mediante la extracción de características, en este trabajo se utilizan las características texturales y cromáticas.

Comparativa de la imagen original y la imagen segmentada de la planta de maíz

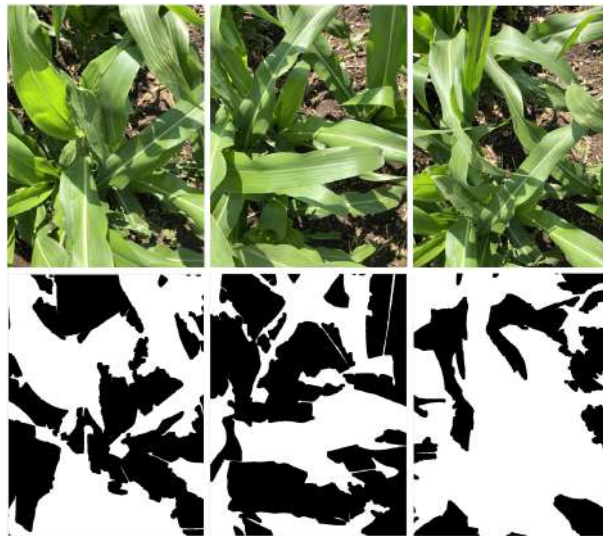


Figura 3.4: Segmentación de imágenes de maíz.

Comparativa de la imagen original y la imagen segmentada de la maleza

Explicación de como se utilizó el algoritmo de Otsu para segmentar:

Paso 1: Computar el histograma normalizado de la imagen.

- $f(x, y)$ = imagen $M \times N$;
- $\{0, 1, 2, \dots, L - 1\}$ = niveles de gris de la imagen;
- n_i = número de píxeles con nivel de intensidad i ;
- $p_i = \frac{n_i}{MN}$ = probabilidad de que ocurra el nivel de intensidad i .

Paso 2: Computar las sumas acumulativas $P_1(k)$, $k = 0, 1, \dots, L - 1$.

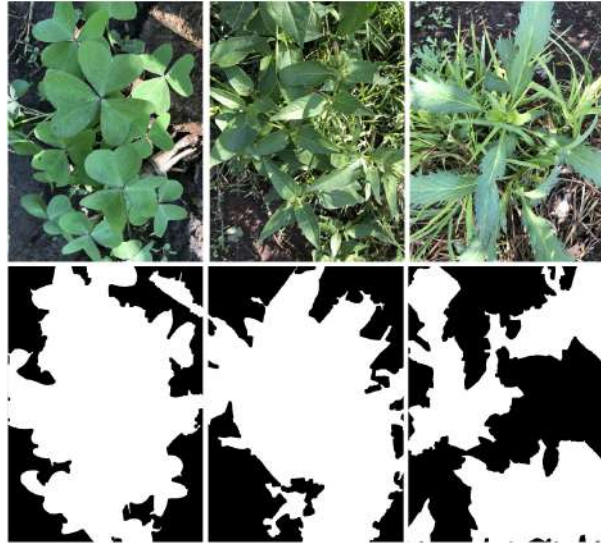


Figura 3.5: Segmentación de imágenes de maleza.

$$P_1(k) = \sum_{i=0}^k p_i \quad (3.1)$$

Si fijásemos un valor umbral T en el nivel de intensidad $k(T = k)$, dividiríamos el conjunto de píxeles de la imagen en dos clases:

$C1$ = píxeles con nivel de intensidad en $[0, k]$

$C2$ = píxeles con nivel de intensidad en $[k + 1, L - 1]$

Entonces, $P_1(k)$ indica la probabilidad de que un pixel pertenezca a la clase C_1 (probabilidad de que ocurra la clase

Paso 3: Computar las medias acumulativas $m(k)$, $k = 0, 1, \dots, L - 1$.

$$m(k) = \sum_{i=0}^k ip_i \quad (3.2)$$

Paso 4: Computar la media global m_G .

$$m_G = \sum_{i=0}^{L-1} ip_i \quad (3.3)$$

Paso 5: Computar la varianza entre clases.

$$(\sigma_B)^2(k) = \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]} \quad (3.4)$$

Paso 6: Maximizar la varianza anterior.

$$k^* = \text{Max}(\sigma_B)^2(k) \text{ donde } (0 \leq k \leq L - 1) \quad (3.5)$$

Este paso persigue el objetivo original, extender los niveles de intensidad de las dos clases tanto como sea posible. Si el máximo no es único, definir k^* como la media de todos los máximos posibles.

Paso 7: Obtener la medida de separabilidad.

$$\mu^* = \frac{(\sigma_B)^2(k)}{(\sigma_G)^2} \quad (3.6)$$

3.3.2. Extracción de características.

La obtención de características se considera la base de los algoritmos de aprendizaje automático; la implementación correcta de los métodos de extracción identifica los descriptores utilizados para distinguir el maíz y las malezas.

Como se ha podido revisar en el apartado de revisión de la literatura, los métodos más usados en la obtención de características son: extracción de características texturales, cromáticas y geométricas. La imagen tiene muchas características que, cuando se extraen, se agregarán a un vector que contiene todas estas características. En este apartado, la imagen se representa a través de un conjunto de valores numéricos potencialmente discriminatorios; por lo tanto, las características obtenidas no tienen variación de escala, rotación y traslación, lo que permite al clasificador reconocer objetos sin importar los diferentes tamaños, posiciones y orientaciones.

Cuando se obtienen las características más adecuadas de las plantas de maíz y de la maleza, se utilizarán 6 clasificadores diferentes para realizar pruebas y producir resultados. Desarrollando así un análisis comparativo del método de extracción de características texturales, cromáticas y un sistema que combina ambas características (híbrido); utilizando ambos tipos de segmentación PCA y Otsu que miden el rendimiento del sistema con clasificadores de aprendizaje automático en el conjunto de datos utilizado para el experimento.

En total se obtuvieron 357 características en cada imagen, de las cuales 84 fueron texturales y 273 cromáticas, la extracción de características se realizó de la siguiente manera:

3.3.3. Características Texturales

Los rasgos de textura que dan lugar a una propiedad pueden ser áspera, rugosa, lisa o suave, entre otras. La textura es invariable al movimiento porque repite un patrón en una superficie; por lo tanto, ahí se explica por qué la percepción visual de la textura es independiente de la posición visual. Los rasgos texturales se extraen de las superficies, para este experimento en específico se extraen de las hojas de maíz y las malas hierbas; para ello, se ha implementado el algoritmo de Haralick (Haralick et al., 1973) haciendo uso de matrices de coincidencia en escala de grises.

A continuación, se reseñan los descriptores de la forma en que Haralick hace uso de ellos:

$$f_1 = \sum_i \sum_j [p(i, j)^2] \quad (3.7)$$

$$f_2 = \sum_{n=0}^{N_g-1} n^2 \left\{ \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j)}{|i-j|=n} \right\} \quad (3.8)$$

$$f_3 = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} [ijp(i, j) - \mu_x \mu_y]}{\sigma_x \sigma_y} \quad (3.9)$$

$$f_4 = \sum_i \sum_j (i - \mu_x)^2 p(i, j) \quad (3.10)$$

$$f_5 = \sum_i \sum_j \frac{1}{1 + (i - j)^2} p(i, j) \quad (3.11)$$

$$f_6 = \sum_{i=2}^{2N_g} i P_{x+y}(i) \quad (3.12)$$

$$f_7 = \sum_{i=2}^{2N_g} (i - f_8)^2 P_{x+y}(i) \quad (3.13)$$

$$f_8 = - \sum_{i=2}^{2N_g} P_{x+y}(i) \log\{P_{x+y}(i)\} \quad (3.14)$$

$$f_9 = - \sum_i \sum_j p(i, j) \log\{p(i, j)\} \quad (3.15)$$

$$f_{10} = \sum_{i=0}^{N_g-1} (i - f_8)^2 P_{x-y}(i) \quad (3.16)$$

$$f_{11} = - \sum_{i=0}^{N_g-1} P_{x-y}(i) \log\{P_{x-y}(i)\} \quad (3.17)$$

$$f_{12} = \frac{HXY - HXY1}{\text{máx}\{HX, HY\}} \quad (3.18)$$

$$f_{13} = (1 - e^{[-2(HXY2-HXY)]})^{\frac{1}{2}} \quad (3.19)$$

$$f_{14} = (\text{Segundo valor mas grande de } Q)^{\frac{1}{2}} \quad (3.20)$$

Donde $p(i, j)$ = determina el valor de la matriz de co-ocurrencia en la posición (i, j) . N_g = Niveles de intensidad detallados en la región segmentada de la imagen. $p_x(i)$ = i ésima entrada en la matriz de probabilidad de co-ocurrencia inicial obtenida sumando las filas en $p(i, j)$. $p_y(j)$ = j ésima entrada a la matriz de probabilidad de co-ocurrencia inicial obtenida sumando las columnas en $p(i, j)$. R = Cantidad total de componentes en la matriz de co-ocurrencia. μ_x, μ_y = Media de P_x y P_y . σ_x, σ_y = Desviación estándar de P_x y P_y . HX, HY = Entropía de P_x y P_y . El vector de características texturales X_t obtenido puede ser representado como:

$$Xt = [x_1, x_2, \dots, x_{85}] \quad (3.21)$$

$$Xt = [x_{Rlbp}, x_{RH}, x_{Glbp}, x_{GH}, x_{Blbp}, x_{BH}] \quad (3.22)$$

Donde $x_{Rlbp}, x_{Glbp}, x_{Blbp}$ representan las características (LBP) (chen He & Wang, 1990) (Wang & He, 1990) obtenidas en el canal de color R, G y B respectivamente, x_{RH}, x_{GH} y x_{BH} representan las características texturales de Haralick obtenidas en los canales R, G y B respectivamente.

Los descriptores utilizados por Haralick extraen 84 características en total, 28 por cada canal de color RGB ($28 \times 3 = 84$). Se describen a continuación:

1.- Segundo momento angular	2.-Contraste
3.- Correlación	4.-Suma de cuadrados
5.-Momento de diferencia inversa	6.-Suma promedio
7.-Suma de varianza	8.-Suma de entropía
9.-Entropía	10.-Diferencia de varianzas
11.-Diferencia de entropías	12.-Medida de correlación I
13.-Medida de correlación II	14.-Máx. de coeficiente de correlación.

Tabla 3.2: Características extraídas.

Se extrae el promedio y un rango por eso es el doble, es decir, dos valores por cada característica.

3.3.4. Características Cromáticas

Las características de color aportan mucha información y se pueden extraer partiendo de un espacio de color específico, se obtienen partiendo de tres canales primarios como RGB, valor de saturación de tonalidad HSV y escala de grises entre otros; localizando descriptores mediante diferentes algoritmos, considerando: momentos de Hu, descriptores de Fourier, transformada discreta de coseno DCT y características de Gabor. Los momentos invariantes de Hu (Hu, 1962) como descriptores incorporan datos de la variable de color de la región de interés. La DCT utiliza transformaciones de base y funciones de coseno de diferentes longitudes de onda. Una particularidad de la DCT en relación con la transformada discreta de Fourier DFT es la limitación al uso de coeficientes reales. Las características de Gabor (Gabor, 1946) se considera una técnica robusta utilizada para la extracción de características en imágenes, siendo una técnica híbrida compuesta por el núcleo de transformación de Fourier en una función gaussiana.

Se utilizó el algoritmo de *Gabor* con el cual se extraen un total de 201 características en total, 67 por cada canal de color RGB ($67 \times 3 = 201$).

De igual forma se utilizaron *Momentos de Hu* con intensidad para extraer 21 características en total, 7 por cada canal de color RGB ($7 \times 3 = 21$).

También se extrajeron características *DCT* (Transformada discreta del coseno) las cuales fueron 12 en total, 4 por cada canal de color RGB ($4 \times 3 = 12$).

Se extrajeron 24 características con los *Descriptores de Fourier*, 8 por cada canal de color RGB ($8 \times 3 = 24$).

Finalmente se extrajeron 15 características de contraste, 5 por cada canal de color RGB ($5 \times 3 = 15$).

3.4. Algoritmos de Clasificación utilizados durante las pruebas.

En esta sección, se han utilizado algoritmos de aprendizaje automático para clasificar imágenes de plantas de maíz y maleza; identificando solo dos clases diferentes, por lo que podemos decir que se hace una clasificación binaria y se mide el rendimiento de cada uno de ellos. Los clasificadores utilizados en Matlab y la plataforma de software para el aprendizaje automático Weka (Waikato Environment for Knowledge Analysis) son: Backpropagation (Propagación hacia atrás o Retropropagación), SVM, KNN, Random Forest y Naive Bayes.

3.4.1. Clasificadores más utilizados en ML

Clasificadores más usados en Machine Learning:

- Algoritmos de Regresión
- Algoritmos Bayesianos
- Algoritmos de Agrupación
- Algoritmos de Árbol de Decisión
- Algoritmos de Redes Neuronales
- Algoritmos de Reducción de Dimensión (PCA)
- Algoritmos de Aprendizaje Profundo (CNN)

3.4.2. Naive Bayes

Los clasificadores bayesianos se fundamentan en la teoría de decisión bayesiana, el principio de Bayes proporciona una metodología básica para resolver problemas de clasificación de modelos cuando se conocen las distribuciones de probabilidad de los patrones. El clasificador bayesiano utiliza un enfoque probabilístico para asignar una clase a una instancia. ([John & Langley, 1995](#))

Los algoritmos basados en el teorema de Bayes, en particular Naive Bayes, los valores de precisión del estimador numérico se eligen a partir del análisis de los datos de entrenamiento; asimismo, este método es esencial para resolver dilemas de clasificación de instancias, cuando se conoce la distribución de probabilidad de los mismos. Un clasificador bayesiano utiliza una orientación probabilística para asignar la clase a una instancia. Sea C la clase de un objeto, que pertenece a un conjunto de m clases (C_1, C_2, \dots, C_m) y X_k es objeto con k características $X_k = [x_1, x_2, \dots, x_k]$, para este caso, el patrón de características define una clase específica. A continuación, el algoritmo calcula la siguiente probabilidad condicional $p(C_i|X_k)$ haciendo uso de la regla de Bayes:

$$p(C_i|X_k) = \frac{p(X_k|C_i)p(C_i)}{p(X_k)}, i = 1, 2, \dots, m. \quad (3.23)$$

En la ecuación 4.23 $p(X_k|C_i)$, $p(C_i)$ y $p(X_k)$ calculados a partir de los datos de entrenamiento. Asimismo, basándose en el teorema de Bayes, para una nota específica (X_k), la clase a la que pertenece viene dada por la máxima probabilidad posterior:

$$f(X_k) = \arg \max_i p(C_i|X_k) \quad (3.24)$$

3.4.3. Random Forest

Los bosques aleatorios, es un algoritmo que está constituido por los clasificadores de árboles de decisión, cada árbol depende de los valores de un vector aleatorio muestreado de forma independiente que tiene la misma distribución para todos los árboles del bosque. El error general de los bosques converge a un límite a medida que aumenta el número de árboles en el bosque. Cuando un modelo se generaliza y falla depende de la fuerza de cada árbol en el bosque y la correlación entre ellos. (Breiman, 2001)

3.4.4. SVM

La máquina de vectores de soporte es una técnica de clasificación ampliamente utilizada en los últimos años, el punto principal de SVM es usar el kernel cuando se trabaja con conjuntos no lineales, no hay mínimos locales, la solución depende de un pequeño conjunto de datos, y el poder discriminante del modelo se obtiene optimizando las amplitudes de separación entre clases, estas características permiten a la SVM obtener resultados muy competitivos comparado con otros clasificadores. (Vapnik, 1999) La SVM descrita como un algoritmo se denota a continuación. Suponiendo que un grupo de datos de entrenamiento X , se expresa como:

$$(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \dots, (\mathbf{x}_n, \mathbf{y}_n) \quad (3.25)$$

i.e. $X = \{x_i, y_i\}_{i=1}^n$ donde $x_i \in \mathbb{R}^d$ y $y_i \in (+1, -1)$. El entrenamiento de una SVM resuelve un problema de programación cuadrática, como se muestra a continuación:

$$\max_{\alpha_i} -\frac{1}{2} \sum_{i,j=1}^l \alpha_i y_i \alpha_j y_j \mathbf{K} \langle x_i, x_j \rangle + \sum_{i=1}^l \alpha_i \quad (3.26)$$

$$\text{subject to: } \sum_{i=1}^l \alpha_i y_i = 0, C \geq \alpha_i \geq 0, i = 1, 2, \dots, l$$

donde $C > 0$, $\alpha_i = [\alpha_1, \alpha_2, \dots, \alpha_l]^T$, $\alpha_i \geq 0, i = 1, 2, \dots, l$ son coeficientes que pertenecen a x_i, y_i con α_i no nulos que se llaman Vectores de Soporte; \mathbf{K} representa una función. SVM es un clasificador lineal, es decir, realiza una clasificación entre dos conjuntos de datos a través de la construcción

de una recta que divide dos clases. Cuando esto no sucede, se emplea una función llamada *Kernels*, que transforma el espacio de entrada en un espacio dimensional, donde los conjuntos pueden ser linealmente separables después de la transformación. Sin embargo, la selección de una función se limita a aquellas que satisfacen las condiciones de Mercer (Vapnik, 1999). S Es el conjunto de vectores de soporte obtenido tras el entrenamiento, por lo que el hiperplano óptimo viene dado por:

$$\sum_{i \in S} (\alpha_i y_i) K(\mathbf{x}_i, \mathbf{x}_j) + b = 0 \quad (3.27)$$

y la función de decisión óptima se define como:

$$f(x) = \text{sign}\left(\sum_{i \in S} (\alpha_i y_i) K(\mathbf{x}_i, \mathbf{x}_j) + b\right) \quad (3.28)$$

donde $\mathbf{x} = [x_1, x_2, \dots, x_l]$ son datos de entrada, α_i y y_i son los multiplicadores usados en el método de Lagrange. Un objeto nuevo x puede clasificarse mediante la ecuación 4.28. El vector \mathbf{x}_i se aplica en forma de producto escalar. Hay un multiplicador de Lagrange α para cada punto de entrenamiento. Cuando se encuentra el límite del margen en el hiperplano, sólo se satisfacen los puntos más cercanos al hiperplano $\alpha > 0$. Estos puntos son los vectores de soporte.

3.4.5. K-Nearest Neighbors

El algoritmo del vecino más cercano, separa un nuevo punto en el conjunto de datos a través de la distancia euclidiana, encontrando las k distancias más próximas al objeto a clasificar, después, la clase del punto más cercano en el conjunto de datos se asigna por mayoría de votos. (Aha et al., 1991) Como primera instancia, se calculan las distancias del nuevo punto a cada objeto en el conjunto de datos, la distancia euclidiana de un punto a a un punto b se calcula como sigue:

$$d(P_0, P_1) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (3.29)$$

Posteriormente, se localizan las k distancias más cercanas al nuevo punto, finalmente, se asigna la clase del punto más cercano en el conjunto de datos por mayoría de votos.

3.4.6. Backpropagation

Los humanos, para resolver problemas de la vida diaria, toman conocimientos previos adquiridos a partir de la experiencia de algún área específica, de la misma forma, las redes neuronales artificiales, recolectan información sobre problemas resueltos para construir modelos o sistemas que puedan tomar decisiones de forma automática. Los diversos enlaces entre neuronas forman un sistema adaptativo cuyos pesos se actualizan a través un algoritmo de aprendizaje particular. Dentro de las redes neuronales artificiales, uno de los algoritmos de aprendizaje más utilizados

es Backpropagation. Dicho algoritmo realiza el proceso de aprendizaje y clasificación en cuatro puntos; inicialización de ponderaciones, propagación hacia adelante, propagación hacia atrás y actualización de ponderaciones. (Rumelhart et al., 1986)

Para que el proceso de aprendizaje se lleve a cabo, el algoritmo de retropropagación cambia iterativamente los pesos entre las neuronas, disminuyendo el error cuadrático entre la salida óptima y la obtenida con los pesos actualizados. Cada uno de los casos del conjunto de entrenamiento $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ se usa para ajustar los pesos en la red. Al presentarse un ejemplo, el indicador se propaga hacia adelante en la red hasta obtener la salida. La salida de la j -ésima unidad oculta se calcula como:

$$o_{nj}^h = f_j^h(\text{net}_{nj}^h) = \frac{1}{1 + \exp(-\text{net}_{nj}^h)} \quad (3.30)$$

donde $\text{net}_{nj}^h = \sum w_{ji}^h x_{ni} + \theta_j^h \cdot w_{ji}^h$ es el peso del enlace de la i -ésima neurona de entrada a la j -ésima neurona oculta. θ_j^h y f_j^h representan el sesgo y la función de activación de la j -ésima neurona oculta. Así, la salida de la k -ésima neurona está representada por:

$$o_{nk}^o = f_k^o(\text{net}_{nk}^o) = \frac{1}{1 + \exp(-\text{net}_{nk}^o)} \quad (3.31)$$

donde los superíndices h y o se refieren a los valores en las capas ocultas y de salida respectivamente. El error entre la salida actual y la salida óptima se calcula para adaptar los pesos con $E_n = \frac{1}{2} \sum_{k=1}^C (t_{nk} - o_{nk}^o)$. El procedimiento de ajuste se obtiene mediante el método del gradiente descendiente para disminuir la dimensión del error. El procedimiento se aplica primero a los pesos de la capa de salida y se propaga hacia atrás por medio de la red, hasta que los pesos de la primer capa se han ajustado $\Delta w_{kj}^o = -\eta \frac{\partial E_n}{\partial w_{kj}^o}$ and $\Delta w_{ji}^h = -\eta \frac{\partial E_n}{\partial w_{ji}^h}$. Este procedimiento se utiliza para cada caso en el Dataset, hasta que se cumpla un criterio para detenerse.

3.4.7. Redes Neuronales Convolucionales

Una **Red Neuronal** (Neural Network) al igual que una regresión lineal o un modelo SVM, es una función matemática:

$$y = f_{NN}(x) \quad (3.32)$$

La función f_{NN} tiene una forma particular: es una **función anidada**. Las **capas** mencionadas en la redes neuronales funcionan de la siguiente manera, para una red neuronal de 3 capas que devuelve un escalar, f_{NN} es de la forma:

$$y = f_{NN}(x) = f_3(f_2(f_1(x))) \quad (3.33)$$

En la ecuación anterior, f_1 y f_2 son funciones vectoriales de la siguiente forma:

$$f_l(z) \stackrel{\text{def}}{=} g_l(W_l z + b_l) \quad (3.34)$$

donde l se llama el índice de capa y puede abarcar desde 1 hasta n número de capas. La función g_l se conoce como **función de activación**. Es una función fija, generalmente no lineal, elegida por el analista de datos antes de que se inicie el entrenamiento. Los parámetros W_l (una matriz) y b_l (un vector) se aprenden para cada capa utilizando el descenso por gradiente que optimiza una función de coste particular.

Una **red neuronal convolucional** (CNN, por sus siglas en inglés) es un tipo especial de red neuronal de avance (FFNN, por sus siglas en inglés) que reduce significativamente el número de parámetros en una red neuronal profunda con muchas unidades sin sacrificar demasiado la calidad del modelo. Las CNN han encontrado aplicaciones en el procesamiento de imágenes y texto, donde superan a muchos de los mejores resultados establecidos previamente. (Burkov, 2019)

Si se entrena una red neuronal para que reconozca regiones que contengan la misma información así como los bordes, este conocimiento permitirá a la red neuronal predecir el objeto representado en la imagen, fue lo que se buscó hacer con la identificación de maíz y maleza en este trabajo.

3.5. Herramientas Computacionales

3.5.1. Clasificación

Clasificación: La clasificación es un cometido en el reconocimiento de imágenes, en la que a un individuo se le etiqueta según algunos atributos característicos; en el entendido de que un individuo es cualquier tipo de entidad. La clasificación desde el punto de vista del reconocimiento de patrones consta de tres etapas importantes: definición de clases, representación de la información como atributos y aprendizaje por algoritmos.

Métodos basados en modelos probabilísticos.- Naive Bayes es la representación más sencilla de algoritmos basados en probabilidad. Se fundamenta en el teorema de Bayes. Clasifica nuevos patrones $x = (x_1, \dots, x_m)$ asignándoles la clase k que maximiza la probabilidad condicional de la clase en función de la secuencia de atributos de datos observados.

Métodos basados en distancias.- Estos métodos clasifican nuevos patrones en función de medidas de similitud o distancia. Esto se realiza comparando nuevos ejemplos con un conjunto (uno para cada clase) de prototipos y asignando la clase del prototipo más cercano, o encontrando una base de ejemplos más cercanos. La generalización se posterga hasta que se clasifiquen nuevos ejemplos, es por eso que se llama aprendizaje perezoso.

Este algoritmo, trabaja de la forma más sencilla, almacenando todos los ejemplos en la memoria durante el aprendizaje y clasifica los nuevos ejemplos en función de las clases de los k ejemplos más cercanos (por eso también se le llama basado en memoria, en instancias, en casos o ejemplos). Para obtener el conjunto de k vecinos más cercanos, calculamos la distancia entre el ejemplo que se clasificará $x = (x_1, \dots, x_m)$ y la totalidad de los ejemplos almacenados $x = (x_i1, \dots, x_im)$. La distancia Euclídea o Euclidiana es una de las más utilizadas en este método.

Clasificadores lineales.- Las máquinas de soporte vectorial o máquinas de vectores de soporte (SVM) es un algoritmo que mejora un clasificador lineal al encontrar un hiperplano mejor que el hiperplano generado por un clasificador lineal simple. Las SVM son un algoritmo de optimización

que elige el hiperplano con límite máximo entre todos los hiperplanos posibles, para separar los ejemplos positivos de los negativos (un hiperplano con la misma distancia entre positivos y negativos). Describir el hiperplano a partir de los llamados vectores de soporte. Suelen ser los puntos más próximos al hiperplano y los puntos que lo definen.

Backpropagation.- También llamada de Retropropagación, Es un tipo de red neuronal artificial de aprendizaje supervisado, organizada en capas que pueden tener un número variable de neuronas, la particularidad con la que cuenta este algoritmo es que cada neurona está conectada a todas las neuronas de la capa anterior, excepto a las neuronas de la primer capa. Estas neuronas de la primer capa se denominan neuronas de entrada y las neuronas de la última capa se denominan neuronas de salida. Obviamente, la cantidad de entradas y salidas depende del tipo de problema que se quiera resolver. Las redes se pueden organizar en varias capas, las capas existente entre las neuronas de entrada y las de salida se denominan capas ocultas; esto permite más de una capa de neuronas ocultas.

El tipo de problema que se pueda resolver dependerá del número de capas ocultas en la red. Sin capas ocultas, se pueden resolver problemas linealmente separables. Con una sola capa oculta es posible resolver problemas de descomposición con curvas; y con dos capas ocultas es posible resolver problemas con separación arbitraria, por lo que en principio no tiene sentido usar más de dos capas ocultas en una red. Generalmente la separación se representa en un plano bidimensional, pero podría ampliarse a cualquier cantidad de dimensiones (esto depende de la cantidad de neuronas en la capa inicial o de entrada).

Random Forest.-Un bosque aleatorio es un conjunto de árboles de decisión. Utiliza bagging para corregir la tendencia de los árboles de decisión al sobreajuste. Al crear muchos árboles entrenados en sub-muestras aleatorias de las muestras y características aleatorias de los datos, se reduce la varianza. Debido a que el entrenamiento se basa en sub-muestras de datos, el bosque aleatorio puede evaluar el error OOB y evaluar el rendimiento. También pueden rastrear la importancia de la característica promediando la importancia de la característica sobre todos los árboles. La intuición para entender el bagging proviene de un ensayo de 1785 del Marqués de Condorcet. La esencia es que, si está creando un jurado, debe agregar a cualquiera que tenga una probabilidad superior al 50% de emitir el veredicto correcto y luego promediar sus decisiones. Cada vez que agregue otro miembro (y su proceso de selección es independiente de los demás), obtendrá un mejor resultado. La idea con bosque aleatorio es crear un "bosque" de árboles de decisión entrenados en diferentes columnas de los datos de entrenamiento. Si cada árbol tiene una probabilidad superior al 50% de una clasificación correcta, debe incorporar su predicción. El bosque aleatorio ha sido una herramienta excelente tanto para la clasificación como para la regresión.

Reducción de dimensionalidad: La reducción de la dimensionalidad está motivada por varios problemas. Primero, se puede utilizar para mitigar los problemas causados por la maldición de dimensionalidad. En segundo lugar, la reducción de dimencionalidad se puede utilizar para comprimir datos mientras se minimiza la cantidad de información que se pierde. En tercer lugar, comprender la estructura de datos con cientos de dimensiones puede ser difícil; datos con solo dos o tres dimensiones se pueden visualizar fácilmente. Extracción y pre-procesamiento de características, los problemas que involucran los datos de alta dimensión pueden verse afectados por la maldición de la dimensionalidad. Como las dimensiones de un conjunto de datos aumenta, el número de muestras necesarias para que un estimador generalice aumenta exponencialmente. Adquirir datos tan grandes puede ser inviable en algunas aplicaciones y aprender de grandes conjuntos de

datos requiere más memoria y poder de procesamiento. Además, la escasez de datos a menudo aumenta con sus dimensiones. Puede que se vuelvan más difíciles de detectar instancias similares en el espacio de alta dimensión ya que todos los casos son igualmente escasos. El análisis de componentes principales (PCA), también conocido como la transformada de Karhunen-Loeve, es una técnica utilizada para buscar patrones en datos de alta dimensión. PCA es comúnmente utilizado para explorar y visualizar conjuntos de datos de alta dimensión. También se puede utilizar para comprimir datos y procesarlos antes de que otro estimador los utilice. PCA reduce un conjunto de variables de alta dimensión posiblemente correlacionadas con una dimensión inferior y un conjunto de variables sintéticas no correlacionadas linealmente llamadas componentes principales. Los datos de dimensiones inferiores conservarán la mayoría de la varianza de los datos originales como sea posible. PCA reduce las dimensiones de un conjunto de datos proyectando los datos en un subespacio de dimensiones inferiores. Por ejemplo, un conjunto de datos bidimensionales podría ser reducido al proyectar los puntos en una línea; cada instancia en el conjunto de datos entonces estará representado por un solo valor en lugar de un par de valores. Un espacio tridimensional y su conjunto de datos se podría reducir a dos dimensiones proyectando las variables en un plano.

En general, un conjunto de datos de n dimensiones se puede reducir proyectando el conjunto de datos en un sub-espacio k -dimensional, donde k es menor que n . Más formalmente PCA se puede utilizar para encontrar un conjunto de vectores que abarquen un sub-espacio lo que minimiza la suma del cuadrado y los errores de los datos proyectados. Esta proyección retendrá la mayor proporción de varianza del conjunto de datos original.

Red Neuronal: Se trata de un modelo informático formado por un conjunto de unidades denominadas neuronas artificiales, que se interconectan para transmitir señales. La información de entrada pasa a través de la red neuronal (donde se somete a varias operaciones) produciendo valores de salida. Cada neurona está conectada a otra neurona por enlaces, en estos enlaces el valor de salida de la neurona anterior se multiplica por un valor de peso; estos pesos de enlace pueden aumentar o inhibir el estado de activación de las neuronas vecinas. De la misma forma, a la salida de una neurona, puede existir una función límite o umbral que ajuste el valor del resultado, o imponga un límite que no debe ser superado antes de transmitir a otra neurona.

Filtro: Es una de las principales formas de funcionamiento en el procesamiento de imágenes digitales, se pueden utilizar para diferentes propósitos pero en cualquier caso, el resultado por píxel depende del píxel de su entorno. Algunos de sus objetivos son: Suavizar la imagen, eliminar ruido, realzar la imagen y detectar bordes.

Filtro de convolución: Este es el espacio real (x, y) , donde se puede representar como una matriz cuadrada o rectangular (matriz convolucional) de tamaño (m, n) mucho menor que la imagen original. La "matriz de convolución" se mueve sobre la imagen de manera que el elemento focal de la matriz convolucional coincide con cada píxel de la imagen. En cada posición, se multiplica por su valor el valor de cada píxel de la imagen que coincide con la posición de un elemento de la matriz de convolución. El píxel de la imagen, que coincide con el elemento focal de la matriz de convolución, se sustituye por la suma de los productos.

Estandarización de datos: Muchos estimadores funcionan mejor cuando están capacitados en conjuntos de datos estandarizados. Los datos estandarizados tienen media cero y varianza unitaria. Una variable explicativa con la media cero se centra en el origen; su valor medio es cero. Un vector de características tiene varianza unitaria cuando las varianzas de sus características son todas del

mismo orden de magnitud. Por ejemplo, suponga que un vector de características codifica dos variables explicativas. Los primeros valores de la primera variable van de cero a uno. Los valores de la segunda variable explicativa van de cero a 100. La segunda característica debe ser escalado a un rango más cercano a 0,1 para que los datos tengan varianza unitaria. Si una característica de la varianza es de magnitud mayor que las varianzas de las otras características, ésta puede dominar el algoritmo de aprendizaje y evitar que aprenda de la otras variables. Algunos algoritmos de aprendizaje también convergen al parámetro óptimo más lentamente cuando los datos no están estandarizados. El valor de una variable explicativa se puede estandarizar o normalizar, descontando la media de esa variable y fraccionando la diferencia por la desviación estándar de esa variable.

Normalización de datos: Para que funcionen mejor los algoritmos de Machine Learning usados, se deben normalizar las "variables o datos" que ingresan al algoritmo; esto quiere decir, comprimir o extender los valores de la variable para que estén dentro de un rango específico. Puede ser: escalado de variables (Feature Scaling o MinMax Scaler) y escalado estandar (Standar Scaler).

3.5.2. Segmentación

Segmentación: La segmentación de imágenes divide la imagen en las partes que la constituyen, en la medida de la subdivisión donde las regiones u objetos de interés están aislados. Los algoritmos de segmentación se fundamentan en cualquiera de las dos propiedades básicas de los valores de nivel de gris: "discontinuidad o similitud." entre los niveles de gris de los píxeles vecinos.

En caso de discontinuidad, la imagen se segmenta en función de los cambios repentinos en el nivel de gris y se puede realizar la detección de puntos aislados, la detección de líneas y bordes; y en similitud, las imágenes se dividen en función de encontrar áreas con valores similares, de acuerdo con criterios predefinidos, se realizan crecimiento de área y umbralización.

Segmentación Otsu: Es un procedimiento basado en los cambios de similitud entre el fondo y el objeto, por uso de borde adaptativo que es una técnica de detección de bordes. Usa técnicas estadísticas para resolver el problema, en particular utiliza el umbral de varianza que es una medida de la dispersión de valores (dispersión de valores de gris).

Método PCA: Principal Component Analysis es un algoritmo de Machine Learning no supervisado, el objetivo no es estudiar o predecir una variable, sino estudiar todas las variables que conforman nuestro conjunto de datos. Es la proyección de un sistema de n dimensiones a un nuevo espacio con un número menor de dimensiones; es una técnica de reducción de características que reduce la dimensionalidad de un problema con respecto al número de variables implicadas en él. El *componente principal* es una combinación de las variables originales que tiene mayor varianza, para obtener los componentes principales se realiza un proceso iterativo donde se descompone linealmente las variables originales en nuevas componentes, la variable original está multiplicada por unos factores denominados loadings (cargas o pesos), estos factores están normalizados a 1 y así es como se obtienen o se calculan las componentes principales.

3.5.3. Extracción de características

Consiste en extraer características de las distintas regiones de interés (ROI), no todas las características obtenidas tienen la misma importancia, el uso de demasiadas características puede aumentar la complejidad del clasificador y reducir su rendimiento.

Principales Características que se extraen de una imagen: Las primeras son las características de intensidad (color o cromáticas), donde se utilizan los valores de gris para obtenerlas; en segundo lugar están las características geométricas, de forma o morfológicas, éstas se calculan a partir del área y borde de la región de interés (área, perímetro, convexidad, momentos de primer, segundo y tercer orden, etc.)

Vector de características: La información de la imagen consta de una cuadrícula de píxeles y los valores de color de cada píxel, el vector de características es el que contiene esta información; son objetos determinados por una serie de puntos que pueden ser modificados para generar una nueva imagen diferente a la original.

3.5.4. Espacios de color

Un modelo de aprendizaje automático transforma sus datos de entrada en resultados significativos, es un proceso que se *aprende* a partir de la exposición a datos conocidos en las entradas y salidas de problemas anteriores. Por lo tanto, el dilema central en el aprendizaje automático y el aprendizaje profundo es transformar los datos de manera significativa: en otras palabras, aprender representaciones valiosas de los datos de entrada, muestras o ejemplos que nos acerquen a la salida esperada. Una representación, en esencia, es un modo opuesto de ver los datos, de codificarlos o representarlos. Por ejemplo, una imagen en color se puede codificar en formato RGB (rojo, verde y azul) o en formato HSV (matíz, saturación y valor): estas son dos formas diferentes de representar los mismos datos. Algunas de las tareas que podrían resultar difíciles con una representación pueden ser más sencillas con otra. Los modelos de aprendizaje automático consisten en encontrar representaciones adecuadas para sus datos de entrada, es decir, transformar los datos para ser más aptos para la tarea en cuestión, por ejemplo: un proyecto de clasificación. (Chollet, 2018)

El propósito de un espacio de color es facilitar la distinción de los colores dentro de un sistema coordinado tridimensional, o formar un subespacio dentro del mismo sistema donde un punto único represente y diferencie a cada color. En la literatura, los espacios de color para el trabajo con imágenes son los siguientes: RGB, HSV (tono, saturación, valor), HSI (tono, saturación, intensidad), $L * a * b$, $L * u * v$, YUV e YCbCr. (Garcia-Lamont et al., 2018)

Espacio de color RGB: Rojo, verde y azul (Red, Green and Blue), es la constitución del color expresando la intensidad de los colores primarios de la luz. El modelo de color RGB es el modelo utilizado en todos los sistemas que forman imágenes a través de rayos de luz, ya sea emitiéndolos o recibéndolos. Los componentes de color son las señales roja, verde y azul; cada color se transmite de forma independiente y aislado del resto. De esta manera, no hay pérdida en el procesamiento de imágenes porque los colores primarios continúan existiendo tal como fueron transmitidos.

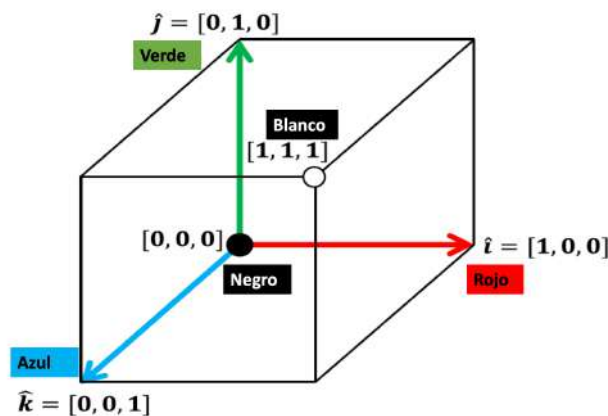


Figura 3.6: Espacio de color RGB
(Garcia-Lamont et al., 2018)

Espacio de color HSV: Matíz, saturación y valor (Hue, Saturation and Value) Define un patrón de color basado en sus componentes Matíz.- Se interpreta como un grado de ángulo donde el posible valor se encuentra del 0 a 360. Saturación.- Se representa como la longitud del eje de brillantez Negro-Blanco; donde los valores están del 0 a 100% Valor.-Representa la elevación en el

eje Blanco-Negro; donde los valores posibles se encuentran del 0 al 100 %

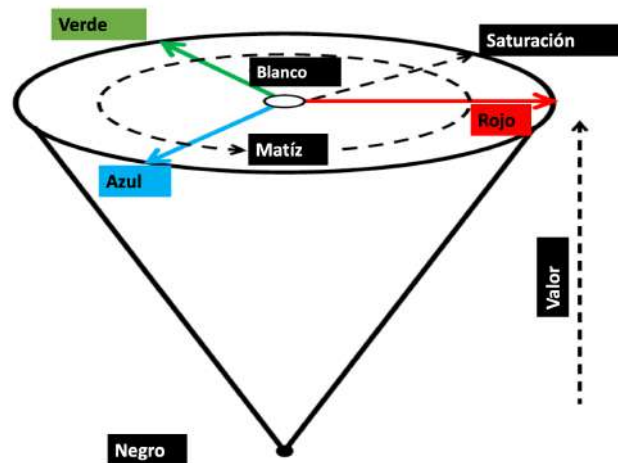


Figura 3.7: Espacio de color HSV
(García-Lamont et al., 2018)

Espacio de color $L^*a^*b^*$: Es la abreviatura de dos espacios de color distintos, el más conocido es CIELAB y el otro es HunterLAB. Los espacios de color tienen relación en intenciones y propósitos, pero son distintos. Ambos espacios se derivan del espacio principal (CIE 1931 XYZ), pero, CIELAB se calcula usando la raíz cúbica y HunterLAB se calcula usando la raíz cuadrada. Las siglas LAB se refieren al espacio de color en tres dimensiones, en donde L^* es la iluminación de negro a blanco, a^* es de rojo a verde y b^* la de azul a amarillo.

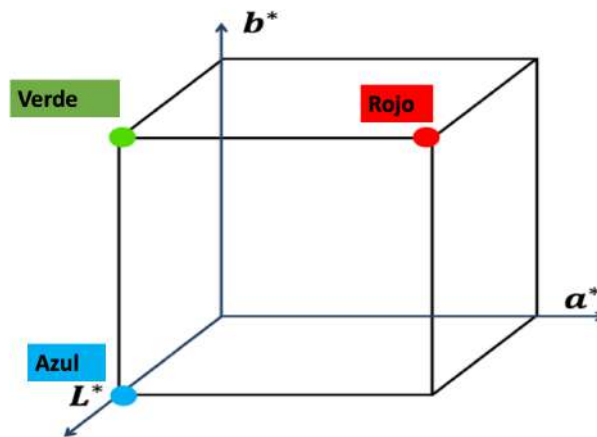


Figura 3.8: Espacio de color L a b
(García-Lamont et al., 2018)

3.6. Inteligencia Artificial

La inteligencia artificial vio la luz en la década de los 50s, cuando un grupo pequeño de pioneros informáticos comenzaron a preguntarse si se podría hacer que las computadoras actuaran como humanos, es decir, si podían "pensar". Una breve definición del tema sería: el intento de automatizar tareas intelectuales normalmente realizadas por humanos. Como tal, la IA es un área que incluye el aprendizaje automático y el aprendizaje profundo.

Durante mucho tiempo, múltiples expertos creyeron que se podía lograr una inteligencia artificial a nivel humano si los programadores creaban un conjunto de reglas lo suficientemente grande para manipular el conocimiento. A este enfoque se le conoce como IA simbólica y fue el paradigma de la inteligencia artificial dominante desde la década de 1950 hasta finales de 1980, alcanzando su punto máximo durante el auge de los sistemas expertos en esa misma década. Mientras que, en la IA simbólica, el objeto demostró ser suficiente para resolver problemas lógicos bien definidos, no funcionó de la forma esperada cuando había que definir reglas explícitas para resolver problemas más complejos y difusos como la clasificación de imágenes, el reconocimiento de voz y la traducción de lenguas. (Chollet, 2018)

En la figura 3.4 podemos observar una imagen donde se muestra el aprendizaje automático, el aprendizaje profundo y cómo se relacionan con la inteligencia artificial.

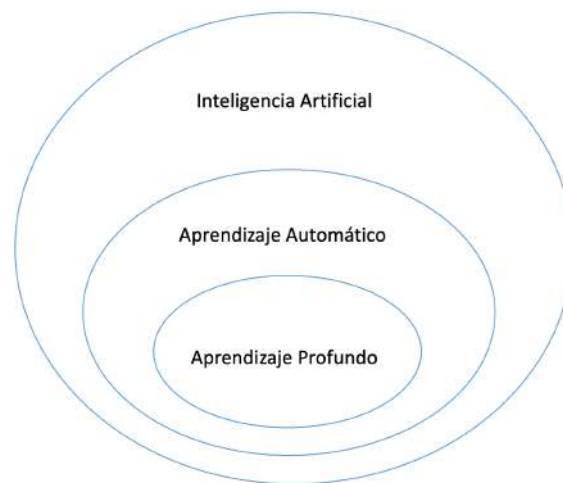


Figura 3.9: Relación de la Inteligencia Artificial con Aprendizaje Automático y Aprendizaje Profundo

(Chollet, 2018)

3.6.1. Aprendizaje Automático o Machine Learning

Turing citó a Ada Lovelace en su histórico artículo de 1950 "Computing Machinery and Intelligence" (Maquinaria de computación e inteligencia), mientras meditaba sobre los ordenadores de propósito general y si estos tenían la capacidad de aprender y realizar cosas originales a partir de estos aprendizajes, y concluyó que sí.

El aprendizaje automático surge de esta pregunta: ¿podría un ordenador ir más allá de "lo que sabemos ordenar que haga aprender por sí mismo cómo realizar una labor determinada? ¿Puede una computadora sorprendernos? En lugar de que los programadores elaboren reglas de procesamiento de datos a mano, ¿podría un ordenador aprender automáticamente estas reglas observando los datos?

Esta cuestión abre la puerta a un nuevo paradigma de programación. En la programación clásica, el paradigma de la IA simbólica, los seres humanos introducen reglas que se interpretan con un programa y los datos deben procesarse de acuerdo con esas reglas, y de ahí salen las respuestas. Con el aprendizaje automático, los humanos introducen datos y las respuestas que se esperan de los datos, y de ahí salen las reglas. Estas reglas se pueden aplicar a nuevos datos para generar respuestas originales.

Un sistema de aprendizaje automático se entrena en lugar de programarse explícitamente. Se le presentan muchos ejemplos relacionados con un objetivo, y encuentra un sistema estadístico en estos ejemplos que, finalmente, permite al sistema elaborar reglas para automatizar la tarea. Aunque el aprendizaje automático no empezó a crecer hasta la década de 1990, se convirtió rápidamente en el subcampo más productivo, popular y exitoso de la IA, una preferencia potenciada por la disponibilidad de hardware más rápido y Datasets más grandes.

El aprendizaje automático está fuertemente relacionado con las estadísticas matemáticas, pero se diferencia de ésta en varios aspectos importantes. A diferencia de la estadística, el aprendizaje automático tiende a tratar con conjuntos de datos grandes y complejos (como un conjunto de datos de millones de imágenes, cada una de ellas compuesta por decenas de miles de píxeles) para los que el análisis estadístico clásico, como el análisis bayesiano, sería poco práctico.

Como resultado, el aprendizaje automático, y especialmente el aprendizaje profundo, presenta comparativamente poca teoría matemática, quizás demasiado poca y está orientado a la ingeniería. Es una disciplina práctica en la que las ideas se demuestran empíricamente con más frecuencia que teóricamente.

Un modelo de aprendizaje automático transforma sus datos de entrada en resultados significativos, un proceso que se *aprende* a partir de la exposición a ejemplos conocidos de entradas y salidas. Por lo tanto, el problema central en el aprendizaje automático y el aprendizaje profundo es transformar los datos de manera significativa: en otras palabras, aprender representaciones útiles de los datos de entrada a mano, representaciones que nos acerquen a la salida esperada. Todos los algoritmos de aprendizaje automático consisten en encontrar automáticamente esas transformaciones que convierten los datos en representaciones más útiles para una tarea determinada.

Finalmente, esto es lo que hace el aprendizaje automático, técnicamente: buscar representaciones útiles de algunos datos de entrada, dentro de un espacio predefinido de posibilidades, utilizando la guía de una señal de retroalimentación. Esta sencilla idea permite resolver una gama extraordinariamente amplia de tareas intelectuales, desde el reconocimiento del habla hasta la conducción autónoma de automóviles.

3.6.2. Aprendizaje Profundo o Deep Learning

El aprendizaje profundo es un subcampo específico del aprendizaje automático: una nueva forma de aprender representaciones a partir de los datos que enfatiza en el aprendizaje de capas sucesivas de representaciones cada vez más representativas. El término *profundo* en el aprendizaje profundo no se refiere a ningún tipo de conocimiento más profundo obtenido a través del enfoque; más bien, representa la idea de sucesivas capas de representación. El número de capas que contribuyen a un modelo de datos se denomina profundidad del modelo. El aprendizaje profundo moderno por lo general, involucra docenas o incluso cientos de clases de representación consecutivas, todas las cuales se aprenden automáticamente cuando se exponen a los datos de entrenamiento. En el aprendizaje profundo, estas representaciones en capas se aprenden casi siempre a través de modelos conocidos como redes neuronales, estructurados en capas literales apiladas unas sobre otras. (Chollet, 2018)

En el aprendizaje profundo se suele utilizar el algoritmo denominado **propagación hacia atrás** (Backpropagation) para actualizar los valores de los parámetros en las redes neuronales. Este algoritmo es eficiente para calcular gradientes en las redes neuronales utilizando la regla de la cadena. Durante el descenso por gradiente, los parámetros de la red neuronal se actualizan de forma proporcional a la derivada parcial de la función de coste con respecto al parámetro en cuestión en cada iteración del entrenamiento. (Burkov, 2019)

Las funciones de activación tradicionales, tales como la función tangente hiperbólica, tienen gradientes en el rango $(0, 1)$, y la propagación hacia atrás calcula los gradientes según la regla de la cadena. En una red de n capas, esto tiene el efecto de multiplicar n de estos números pequeños para calcular los gradientes de las capas anteriores, lo que significa que el gradiente disminuye exponencialmente con n .

Sin embargo, las implementaciones actuales de los algoritmos de aprendizaje de redes neuronales permiten entrenar efectivamente redes neuronales muy profundas (cientos de capas). Esto debido a varias mejoras combinadas, entre ellas las funciones ReLU y las LSTM (memoria a corto plazo), y a técnicas tales como saltar conexiones, usadas en redes neuronales residuales, además de otras modificaciones avanzadas del algoritmo de descenso por gradiente.

El aprendizaje profundo describe al entrenamiento de redes neuronales que contiene más de dos capas (sin contar la salida) utilizando el moderno conjunto de herramientas algorítmicas y matemáticas, independientemente de cuán profunda sea la red neuronal. Muchos problemas se pueden resolver con redes neuronales que tienen 2 o 3 capas entre las capas de entrada y salida; así como en el método propuesto de este trabajo. Las capas que no pertenecen a las capas de entrada o salida suelen denominarse **capas ocultas**.

Así que esto es el aprendizaje profundo, técnicamente: la capacidad de aprender representaciones de datos en varias etapas. Es una idea sencilla, pero resulta que dispositivos muy simples, suficientemente escalados, pueden acabar pareciendo mágicos.

Capítulo 4

Etapa Experimental

4.1. Búsqueda de Datasets de Plantas de Maíz

Se revisaron varios sitios web con data sets como: *Plant Image*, *Google Dataset* y *Kaggle*, en este último se encontró información con solo 75 imágenes necesarias para el proyecto. Además de esto se tuvo acceso a una plantación de maíz que servirá para tomar fotos y realizar pruebas para las siguientes entregas.

Dataset de plantas de Maíz



Figura 4.1: Imágenes encontradas en el Data Set de Kaggle. A partir de kaggle.com

Plantación de Maíz

DATASET PLANTACIÓN DE MAÍZ (imágenes propias)



Figura 4.2: Imágenes propias tomadas en una plantación de maíz. Elab. propia

4.2. Revisión de Aplicaciones Móviles

Se hizo la revisión de algunas aplicaciones móviles que prometían la identificación automática de diferentes tipos y especies de plantas, flores y hortalizas, se tomaron fotografías a dos tipos diferentes de plantas que se usaron de ejemplo, fue el mismo procedimiento en las tres aplicaciones revisadas, estas fueron:

4.2.1. PlantNet

Descripción: Es una aplicación para la recopilación, anotación y recuperación de plantas.

Funcionamiento: La aplicación permite identificar los tipos de plantas, pero tarda un poco porque debe comparar con la base de datos botánica con la que cuenta, no hace una identificación exacta a la primera ya que realiza una comparación para obtener el nombre botánico de dicha planta, si la planta tiene suficientes imágenes o ilustraciones en la base de datos, sugiere imágenes que podrían ser iguales a la que se tomó desde la cámara. : La aplicación se encuentra en la siguiente dirección electrónica. <https://identify.plantnet.org>



Figura 4.3: Pantallas de Clasificación. A partir del uso de la aplicación PlantNet.

4.2.2. ArbolApp

Descripción: Es una aplicación móvil gratuita dirigida a todo el público y se basa en la investigación científica realizada por el Real Jardín Botánico del CSIC. Sus contenidos incluyen 143 especies descritas en 122 fichas; los árboles descritos se encuentran en su mayoría en Andorra, Portugal, España y las Islas Baleares.

Funcionamiento: La aplicación no identifica instantáneamente las plantas, mediante un pequeño cuestionario intuitivo va desechando los tipos que no coinciden con las respuestas que el usuario elige, así continua hasta llegar a alguna parecida. Cada categoría cuenta con un mapa de distribución, una descripción breve y varias fotografías. La aplicación se encuentra en la siguiente dirección electrónica. <https://apps.apple.com/mx/app/arbollapp/id936600048>

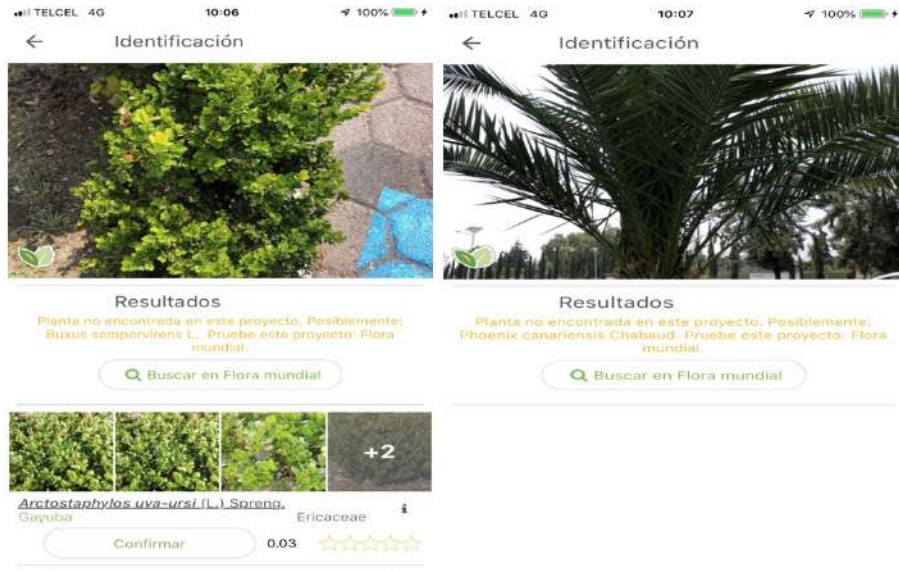


Figura 3. Resultados de búsqueda y sugerencia de similitud.

Figura 4. Resultados sin identificación.

Figura 4.4: Pantallas de Búsqueda. A partir del uso de la aplicación PlantNet.

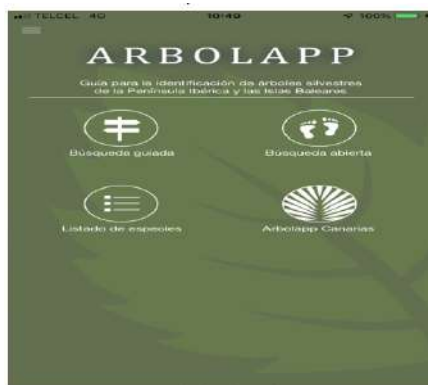


Figura 5. Inicio de la Aplicación



Figura 6. Menú de ArbolApp

Figura 4.5: Inicio y Menú. A partir del uso de la aplicación ArbolApp.



Figura 7. Similitud de hojas.



Figura 8. Similitud de frutos.

Figura 4.6: Similitudes. A partir del uso de la aplicación ArbolApp.

4.2.3. Plantifier

Descripción: Es una herramienta gratuita de reconocimiento de plantas; te permite subir una foto de una planta desconocida, los usuarios de la comunidad *mygarden.org* tratarán de reconocerla lo antes posible y darán una respuesta a tu pregunta. Fácil y sin cargo.

Funcionamiento: La aplicación no identifica instantáneamente las plantas, sugiere enviar la foto a un experto que podría ayudarnos con la identificación, hay que hacer un registro previo para poder acceder a ese servicio. La aplicación se localiza en la siguiente dirección web. <https://www.plantifier.com>



Figura 9. Inicio de plantifier



Figura 10. Selección de foto de la planta

Figura 4.7: Inicio y Selección. A partir del uso de la aplicación Plantifier.

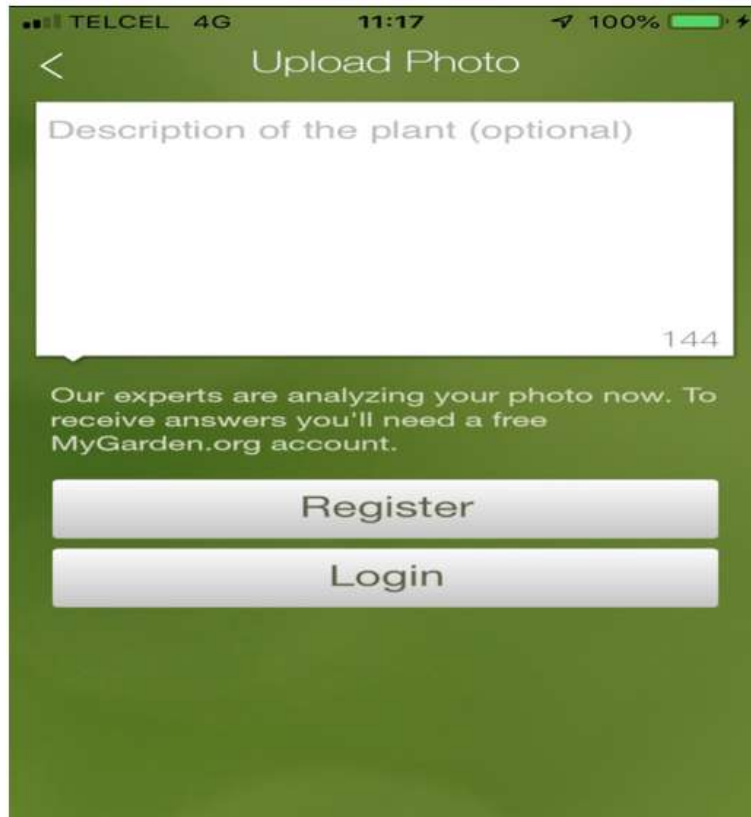


Figura 11. Registro para recepción de ayuda

Figura 4.8: Registro. A partir del uso de la aplicación Plantifier.

4.3. Recolección de imágenes propias

Las imágenes que sirvieron para los primeros experimentos y que dieron paso a las publicaciones que se mencionadas en el punto 4.1, fueron obtenidas de un plantío experimental en el municipio de Metztitlán en el estado de Hidalgo, se ubica geográficamente entre los paralelos 20° 36 de latitud norte y 98° 46 longitud oeste, a una altitud de 1,320 metros sobre el nivel del mar. El Municipio tiene un clima templado-subcálido, el cual registra una temperatura anual de 20,2°C., y una precipitación pluvial de 437 milímetros por año. El período de lluvias es de junio a septiembre. Sus cultivos principales son: maíz, con una superficie sembrada de 3,575 hectáreas, papa con 360 hectáreas, frijol con 1,750 hectáreas, calabaza con 510 hectáreas, chile verde con 301 hectáreas, tomate verde con 288 hectáreas y jitomate con 176 hectáreas. Todo lo mencionado anteriormente, favorece la plantación de maíz dos veces por año debido a la situación geográfica, clima y cercanía con fuentes acuíferas.

Se recolectaron 1480 imágenes para la primera fase de experimentación, fueron tomadas con un teléfono móvil iPhone 8 plus con cámara de 12 megapíxeles con autoenfoco, completando así el Data Set con el que se trabajó.

La recolección de imágenes se realizó el día 2 de marzo del 2020 de las 11 de la mañana a la 1 de la tarde, se usó para las muestras todas las plantas de maíz y maleza que se encontraron en 10 filas o surcos, cada fila de unos 100 metros de longitud.

La experimentación y para publicaciones futuras, se han utilizado las imágenes del conjunto de datos propio y que fueron recolectadas en fechas y lugares diferentes; las correspondientes al estado de Hidalgo fueron recogidas en el municipio de Francisco I. Madero en el mes de marzo del 2020, las del estado de San Luis Potosí fueron recogidas en el mes de mayo del 2021 en el municipio de Rio Verde y las del estado de Puebla fueron tomadas durante el mes de junio del 2021 en el municipio de Aquixtla, en total se adquirieron 4167 para los experimentos realizados, de las cuales el 80 % de ellas sirvió para entrenamiento y el 20 % de testing, por lo que se puede observar el Dataset está desbalanceado pero eso no afectó los resultados de la clasificación, como lo podrán verificar más adelante. El Dataset de maíz y maleza contiene todas las imágenes RGB tomadas desde una altura aproximada de 1 m. como lo recomienda (Lameski et al., 2017). Teniendo en cuenta que el número de imágenes es medianamente pequeño (ya que existen experimentos con Datasets de menos de 100 imágenes y hasta de 1000 imágenes, citarlos); y además el tamaño de la imagen es grande, se disminuyó la resolución de las imágenes grandes originales (768 x 1024) en imágenes más pequeñas de (150 x 150).

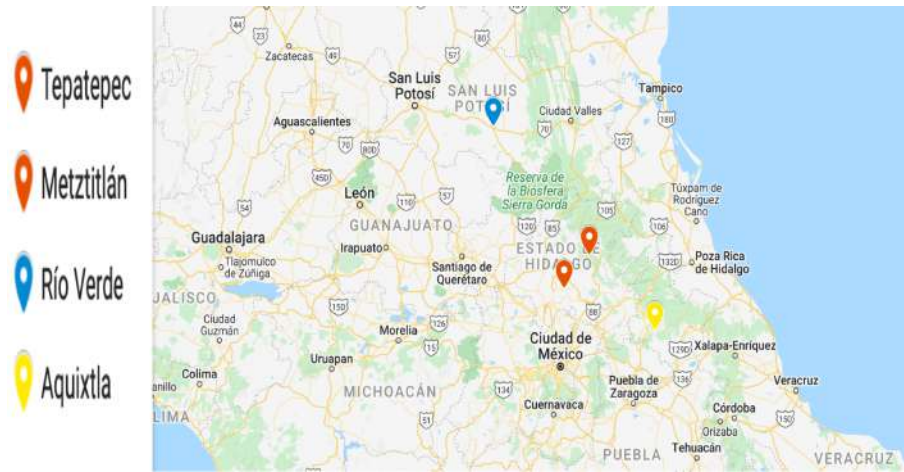


Figura 4.9: Ubicación de los lugares donde se recolectaron las imágenes en México. Google Maps.



Figura 4.10: Muestra de Dataset de imágenes de maíz de los 3 estados de México.

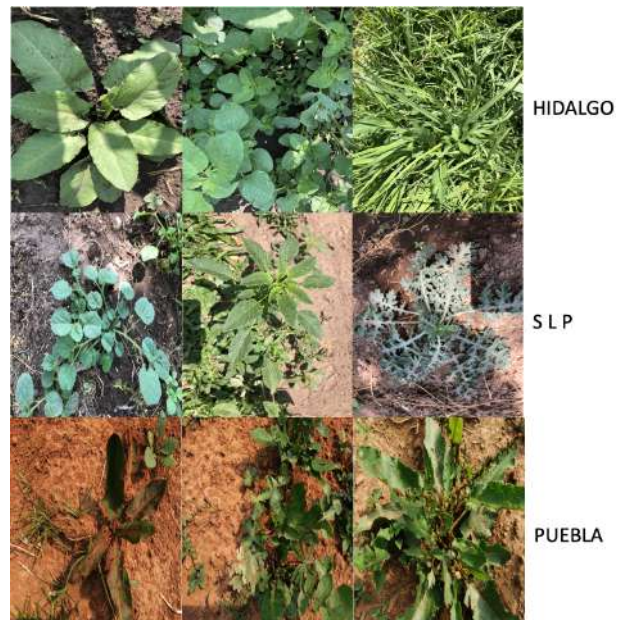


Figura 4.11: Muestra de Dataset de imágenes de maleza de los 3 estados de México.

4.4. Métricas para evaluar una clasificación de Aprendizaje Automático

		Resultado de la Predicción		Total
		p	n	
Valor Actual	p'	Verdadero Positivo	Falso Negativo	P'
	n'	Falso Positivo	Verdadero Negativo	N'
Total		P	N	

Figura 4.12: Matriz de Confusión
([Harrison, 2019](#))

Métricas aplicadas a la Matriz de Confusión para evaluar una clasificación:

1. Accuracy → Exactitud
2. Precision → Precisión
3. Sensitivity o Recall → Sensibilidad
4. Especificity → Especificidad
5. F1 Score o F-Measure → Precision + Sensibilidad en una sola métrica

Accuracy: Es el porcentaje total de elementos clasificados correctamente.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (4.1)$$

Precision: Este es el número de elementos correctamente identificados como positivos del número total de elementos identificados como positivos.

$$Precision = \frac{TP}{TP + FP} \quad (4.2)$$

Sensitivity o Recall: Este es el número de elementos correctamente identificados como positivos del número total de elementos positivos verdaderos.

$$Recall = \frac{TP}{TP + FN} \quad (4.3)$$

F1 Score o F-Measure: Es una medida muy acogida porque abrevia la precisión y sensibilidad en una sola métrica. (Precision y Recall)

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4.4)$$

Specificity: Es la cantidad de elementos identificados correctamente como negativos fuera del número total de negativos.

$$Specificity = \frac{TN}{TN + FP} \quad (4.5)$$

4.5. Técnicas clásicas de reconocimiento

(aquí explicas que herramientas utilizas para la segmentación, extracción de características y de clasificación)

En su mayoría, los algoritmos implementados en la primera y segunda etapa de estudio están codificados para su uso en Matlab R2019 a excepción de la red neuronal convolucional que está codificado en el lenguaje de programación Python 3.5; para la tercera y cuarta etapa se utilizó solamente Python en su versión 3.5 dado que la precisión fue mejor que en las experimentaciones anteriores. Para este trabajo se ha utilizado un equipo informático portátil con las siguientes características: MacBook Pro con un procesador Intel Core i5 de 2.6 GHz con 8 Gigabytes de memoria Ram.

Para la implementación de la CNN utilizada, se requirió de dos conjuntos de datos, en nuestro caso se utilizaron 800 imágenes para el entrenamiento y 200 imágenes para la validación. Finalmente, se realizó una validación cruzada con 5 configuraciones de imágenes arrojando una eficiencia del 97 % en el primer y segundo experimento que dieron lugar a las publicaciones del primer y segundo artículo respectivamente. Para el tercer experimento se utilizó una metodología híbrida, utilizando la CNN como extractor de características y la SVM como clasificador arrojando un porcentaje del 98 % en la precisión.

4.5.1. Segmentación

Pseudocódigo utilizado en Otsu

Función de Segmentación por el método Otsu de una imagen en escala de grises.

1.- Se lee la imagen de entrada.

2.- Se convierte a escala de grises de 8 bits.

3.- Se realiza la segmentación con el método Otsu.

- Esta función requiere el Toolbox de procesamiento de imágenes de Matlab.

4.- Se guarda la imagen segmentada del objeto. 5.- Se muestra o imprime la imagen segmentada (binaria).

Pseudocódigo utilizado en el método PCA

Función de Segmentación de un objeto con fondo homogéneo utilizando componentes principales.

1.- Se lee la imagen de entrada.

2.- Se realiza la segmentación con el método PCA.

- El valor de la frontera debe ser entre -1 y 1.
- Donde los valores positivos dilatan la segmentación.
- Los valores negativos erosionan la segmentación.

3.- Se guarda la imagen binaria (segmentada) del objeto.

4.- Se muestra o imprime la imagen segmentada (binaria).

4.5.2. Extracción de características

Pseudocódigo utilizado en la extracción de características Texturales

Función de extracción de características por medio de los descriptores texturales de Haralick.

1.- Se lee la imagen de entrada.

2.- Se realiza la segmentación.

3.- Se extraen los descriptores texturales de Haralick (Tabla 3.2).

- Momento angular
- Contraste
- Correlación
- Suma de cuadrados
- Momento de diferencia inversa
- Suma promedio
- Suma de entropía
- Suma de varianza

- Entropía
- Diferencia de varianzas
- Diferencia de entropías
- Medida de correlación (2)
- Máximo coeficiente de correlación

4.- Se genera un vector por imagen con esas características y 28 elementos.

5.- Se almacena el vector de 14 características.

Pseudocódigo utilizado en la extracción de características Cromáticas

Función de extracción de características cromáticas por medio de características de Gabor, momentos de Hu, características DCT y descriptores de Fourier.

- 1.- Se lee la imagen de entrada.
- 2.- Se realiza la segmentación.
- 3.- Se extraen las características de Gabor.
- 4.- Se extraen los momentos de Hu de la imagen.
- 5.- Se extraen las características DCT de la imagen.
- 6.- Se calculan los descriptores de Fourier de la imagen.
- 7.- Se extraen las características de contraste de la imagen.
- 8.- Se genera un vector por imagen con todas las características encontradas.
- 9.- Se almacena el vector de características.

Pseudocódigo utilizado en la extracción de características Híbridas (Texturales y Cromáticas)

Función de extracción de características híbridas, que es la combinación de características texturales y cromáticas.

- 1.-Se lee la imagen de entrada.
- 2.-Se realiza la segmentación.
- 3.-Se extraen los descriptores texturales de Haralick.

- Momento angular
- Contraste
- Correlación
- Suma de cuadrados
- Momento de diferencia inversa
- Suma promedio
- Suma de entropía
- Suma de varianza
- Entropía

- Diferencia de varianzas
- Diferencia de entropías
- Medida de correlación (2)
- Máximo coeficiente de correlación

4.-Se extraen las características de Gabor.

5.-Se extraen los momentos de Hu de la imagen.

6.-Se extraen las características DCT de la imagen.

7.-Se calculan los descriptores de Fourier de la imagen.

8.- Se extraen las características de contraste de la imagen.

9.-Se genera un vector por imagen con todas las características encontradas.

10.-Se almacena el vector de características.

Ejemplo de la terminación de la corrida en Matlab y generación del archivo `.arff`

Para correr las pruebas con los algoritmos, se generó un archivo con la extensión `.arff` desde Matlab, posteriormente se utilizó en WEKA para conocer los resultados de la clasificación de imágenes de maíz y maleza.

Ejemplo de la finalización de una corrida en Matlab

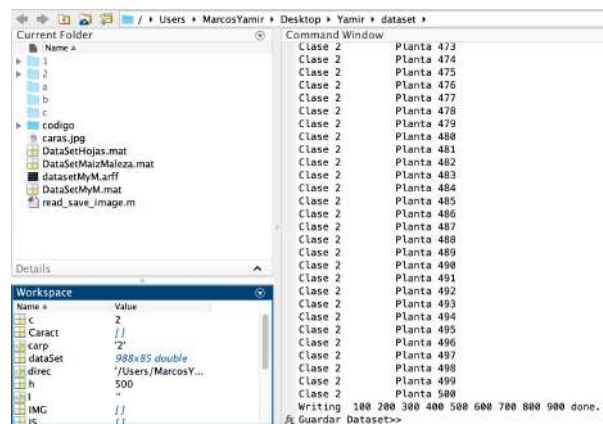


Figura 4.13: Finalización de corrida y generación de archivo.

Como ya se ha comentado, el software Weka permite realizar tareas como: pre-procesamiento, clasificación y visualización de datos, además de aplicación de filtros, clustering, búsqueda de asociaciones y selección de atributos. Nosotros nos enfocamos en la clasificación, la cual se puede visualizar en la siguiente sección (4.8).

4.5.3. Clasificación

Pseudocódigo utilizado en Weka para clasificación con Naive Bayes

- 1.- Se abre el software Weka
- 2.- Se visualiza su explorador

- 3.- Se selecciona abrir archivo .arff (el generado en Matlab)
- 4.- Se selecciona la pestaña “Clasificación”
- 5.- Pulsamos sobre el botón “choose” para elegir el clasificador a utilizar, en este caso Naive Bayes
- 6.- Pulsamos el botón “cross validation” para elegir el número de particiones con las cuales se realizará una validación cruzada, en este caso elegimos el valor 10 que tiene por default.
- 7.- Todos los valores de los demás parámetros del clasificador se dejan por default en Weka
- 8.- Se presiona el botón “start” para iniciar la clasificación
- 9.- Una vez terminada la clasificación, el explorador muestra los resultados de las instancias correctamente clasificadas y las clasificadas incorrectamente, además de valores de algunas métricas importantes para evaluar la clasificación:

- Precision
- Recall
- F-measure
- Matriz de confusión

- 10.- Se guardan los valores generados en la clasificación.

Ejemplo de la clasificación con Naive Bayes en Weka

Resultado arrojado por Weka en la clasificación con Naive Bayes

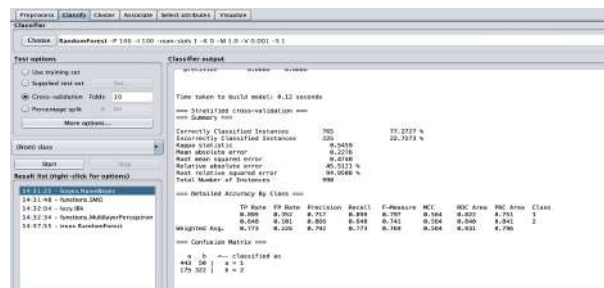


Figura 4.14: Clasificación con el algoritmo de Naive Bayes.

Pseudocódigo utilizado en Weka para clasificación con Random Forest

- 1.- Se abre el software Weka
- 2.- Se visualiza su explorador
- 3.- Se selecciona abrir archivo .arff (el generado en Matlab)
- 4.- Se selecciona la pestaña “Clasificación”
- 5.- Pulsamos sobre el botón “choose” para elegir el clasificador a utilizar, en este caso Random Forest
- 6.- Pulsamos el botón “cross validation” para elegir el número de particiones con las cuales se realizará una validación cruzada, en este caso elegimos el valor 10 que tiene por default.
- 7.- Todos los valores de los demás parámetros del clasificador se dejan por default en Weka
- 8.- Se presiona el botón “start” para iniciar la clasificación
- 9.- Una vez terminada la clasificación, el explorador muestra los resultados de las instancias correctamente clasificadas y las clasificadas incorrectamente, además de valores de algunas métricas importantes para evaluar la clasificación:

- Precision
- Recall
- F-measure
- Matriz de confusión

10.- Se guardan los valores generados en la clasificación.

Ejemplo de la clasificación con Random Forest en Weka

Resultado arrojado por Weka en la clasificación con Random Forest

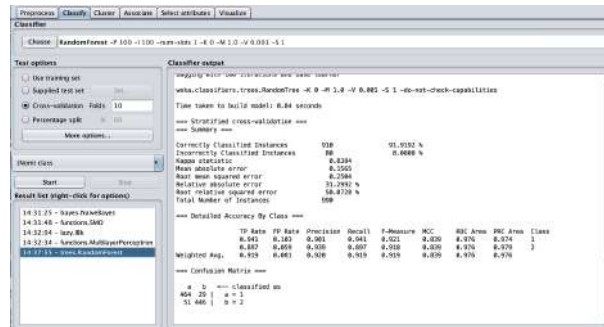


Figura 4.15: Clasificación con el algoritmo de Random Forest.

Pseudocódigo utilizado en Weka para clasificación con SVM

- 1.- Se abre el software Weka
- 2.- Se visualiza su explorador
- 3.- Se selecciona abrir archivo .arff (el generado en Matlab)
- 4.- Se selecciona la pestaña “Clasificación”
- 5.- Pulsamos sobre el botón “choose” para elegir el clasificador a utilizar, en este caso SVM
- 6.- Pulsamos el botón “cross validation” para elegir el número de particiones con las cuales se realizará una validación cruzada, en este caso elegimos el valor 10 que tiene por default.
- 7.- Todos los valores de los demás parámetros del clasificador se dejan por default en Weka
- 8.- Se presiona el botón “start” para iniciar la clasificación
- 9.- Una vez terminada la clasificación, el explorador muestra los resultados de las instancias correctamente clasificadas y las clasificadas incorrectamente, además de valores de algunas métricas importantes para evaluar la clasificación:

- Precision
- Recall
- F-measure
- Matriz de confusión

10.- Se guardan los valores generados en la clasificación.

Ejemplo de la clasificación con SVM en Weka

Resultado arrojado por Weka en la clasificación con SVM

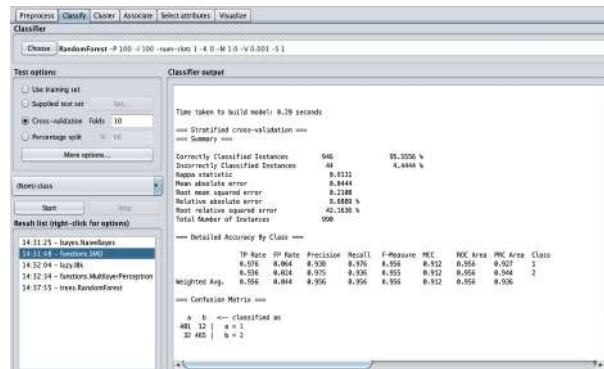


Figura 4.16: Clasificación con el algoritmo de SVM.

Pseudocódigo utilizado en Weka para clasificación con KNN

- 1.- Se abre el software Weka
- 2.- Se visualiza su explorador
- 3.- Se selecciona abrir archivo .arff (el generado en Matlab)
- 4.- Se selecciona la pestaña “Clasificación”
- 5.- Pulsamos sobre el botón “choose” para elegir el clasificador a utilizar, en este caso KNN
- 6.- Pulsamos el botón “cross validation” para elegir el número de particiones con las cuales se realizará una validación cruzada, en este caso elegimos el valor 10 que tiene por default.
- 7.- Todos los valores de los demás parámetros del clasificador se dejan por default en Weka
- 8.- Se presiona el botón “start” para iniciar la clasificación
- 9.- Una vez terminada la clasificación, el explorador muestra los resultados de las instancias correctamente clasificadas y las clasificadas incorrectamente, además de valores de algunas métricas importantes para evaluar la clasificación:

- Precision
- Recall
- F-measure
- Matriz de confusión

- 10.- Se guardan los valores generados en la clasificación.

Ejemplo de la clasificación con KNN en Weka

Resultado arrojado por Weka en la clasificación con vecino más cercano

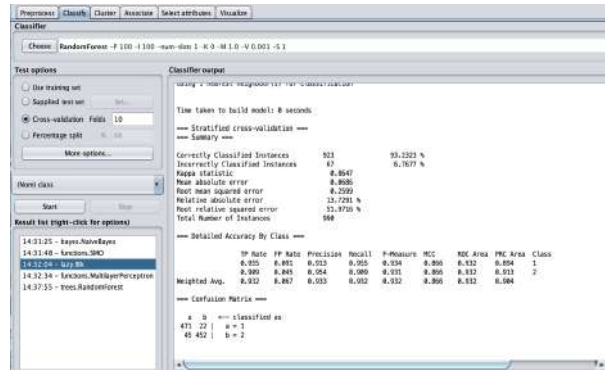


Figura 4.17: Clasificación con el algoritmo de vecino más cercano.

Pseudocódigo utilizado en Weka para clasificación con Backpropagation

- 1.- Se abre el software Weka
- 2.- Se visualiza su explorador
- 3.- Se selecciona abrir archivo .arff (el generado en Matlab)
- 4.- Se selecciona la pestaña “Clasificación”
- 5.- Pulsamos sobre el botón “choose” para elegir el clasificador a utilizar, en este caso Backpropagation
- 6.- Pulsamos el botón “cross validation” para elegir el número de particiones con las cuales se realizará una validación cruzada, en este caso elegimos el valor 10 que tiene por default.
- 7.- Todos los valores de los demás parámetros del clasificador se dejan por default en Weka
- 8.- Se presiona el botón “start” para iniciar la clasificación
- 9.- Una vez terminada la clasificación, el explorador muestra los resultados de las instancias correctamente clasificadas y las clasificadas incorrectamente, además de valores de algunas métricas importantes para evaluar la clasificación:

- Precision
- Recall
- F-measure
- Matriz de confusión

- 10.- Se guardan los valores generados en la clasificación.

Ejemplo de la clasificación con Backpropagation en Weka

Resultado arrojado por Weka en la clasificación con Backpropagation

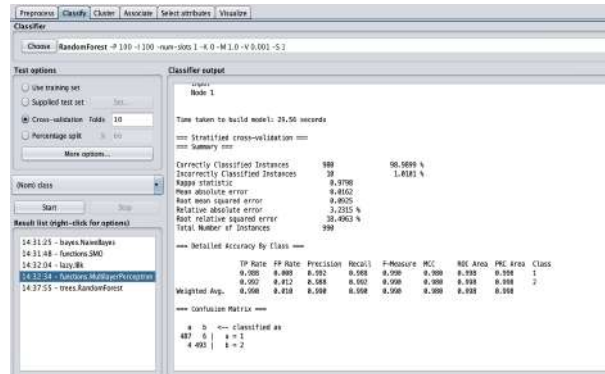


Figura 4.18: Clasificación con el algoritmo de Backpropagation.

Parámetros e Hiperparámetros

En las respectivas pruebas se utilizaron los parámetros e hiper-parámetros que vienen por default en el software Weka, los cuales se mencionan a continuación para cada uno de los algoritmos mencionados.

NAIVE BAYES batchSize =100, numDecimalPlaces=2 y Cross-validation =10 folds.

RANDOM FOREST batchSize =100, batchSize =100, maxDepth =0, numDecimalPlaces=2, numExecutionSlots= 1, numFeatures = 0, numIterations = 100, seed = 1 y Cross-validation =10 folds.

KNN knn=1, batchSize =100, numDecimalPlaces=2, windowsSize =0 y Cross-validation =10 folds.

SVM batchSize =100, c=1.0, epsilon = 1.0E-12, numDecimalPlaces=2, numFolds=-1, random-seed=1, toleranceParameter= 0.001 y Cross-validation =10 folds.

BACKPROPAGATION batchSize =100, hiddenLayers =a, learningRate=0.3, momentum= 0.2, numDecimalPlaces=2, seed=0, trainingTime=500, validationSetSize=0, validationThreshold=20 y Cross-validation =10 folds.

4.6. Redes Neuronales Convolucionales

Una red neuronal convolucional es un tipo de red neuronal artificial supervisada que procesa sus capas imitando la corteza visual del ojo humano para identificar diferentes características de la entrada que lo ayudan a reconocer objetos; se utiliza principalmente en las técnicas de clasificación de texto, discursos y audio, además de su uso frecuente en imágenes. Las redes neuronales convolucionales están inspiradas en la red de perceptrón multicapa. La CNN permite la convolución a través de un filtro en la matriz de píxeles de cada flujo de imágenes hasta cubrir toda la imagen, utilizando una función de activación ReLU y un pooling (agrupación). Como resultado se genera una matriz nueva a la cual se le puede aplicar nuevamente una convolución. Es posible repetir el procedimiento, es decir, se pueden aplicar varias convoluciones; por lo tanto, de aquí es donde toma el nombre de aprendizaje profundo. Después de aplicar convoluciones, obtenemos una red totalmente conectada que se puede clasificar usando una función de activación como softmax para

las multiclasas. (Pattanayak, 2017) En este trabajo de investigación se ha utilizado Python como lenguaje de programación, TensorFlow para desarrollar y entrenar modelos, además de incluir las librerías Keras, Numpy, Matplotlib, Pandas y Sklearn.

Pseudocódigo utilizado para clasificación con CNN

1.- Se importan las librerías a utilizar

- Keras
- Tensorflow
- Matplotlib
- Numpy
- Pandas
- Sklearn

2.- Se leen las imágenes del directorio

- Se define un tamaño para las imágenes (150x150)
- Se manejan los datos en forma de lista
- Definimos las etiquetas, 1 para maíz y 0 para maleza (clasificación binaria)
- Se guarda la información en un arreglo

3.-Se define el 80 % de las imágenes para entrenamiento y el 20 % restante para validación

4.-Se normalizan los datos (es un cambio de escala del rango original de los datos, se busca que los valores estén dentro del rango 0 y 1, ya que si no se realiza es posible que los datos no converjan)

5.-Se construye el modelo de la red neuronal convolucional

- 3 capas de convolución
- 3 capas Pool
- normalización y droops entre ellas
- activación Relu
- Flatten para la red completamente conectada
- Capa densa 'Sigmoid' para la clasificación binaria (en estos casos funciona mejor que softmax)
- Se compila el modelo con los siguientes valores:
 - loss='binary _ crossentropy'
 - optimizer='rmsprop'
 - metrics=['accuracy']

6.-Se imprime un resumen del modelo de la CNN antes de realizar la corrida

7.-Se entrena el modelo (se elige la cantidad de épocas) y al final se guarda el modelo en el equipo de computo

8.-Una vez que termina de entrenar el modelo se grafican:

- la precisión y la pérdida en el entrenamiento
- la precisión y la pérdida en la validación

9.-Se muestran las gráficas de precisión y pérdida

10.-Se carga el modelo entrenado guardado en el paso 6

11.-Se evalúa el modelo y muestra el Accuracy

12.-Se muestra la matriz de confusión y los valores de la clasificación:

- Exactitud
- Precisión
- F1-score
- MCC
- Sensitividad
- Especificidad

13.-Se grafica la curva ROC

14.-Se muestra el valor del área bajo la curva (AUC).

Pseudocódigo utilizado para clasificación con CNN + SVM (Híbrida)

1.- Se importan las librerías a utilizar

- Keras
- Tensorflow
- Matplotlib
- Numpy
- Pandas
- Sklearn

2.- Se leen las imágenes del directorio

- Se define un tamaño para las imágenes (150x150)
- Se manejan los datos en forma de lista
- Definimos las etiquetas, 1 para maíz y 0 para maleza (clasificación binaria)
- Se guarda la información en un arreglo

3.-Se define el 80 % de las imágenes para entrenamiento y el 20 % restante para validación

4.-Se normalizan los datos (es un cambio de escala del rango original de los datos, se busca que los valores estén dentro del rango 0 y 1, ya que si no se realiza es posible que los datos no converjan)

5.-Se construye el modelo de la red neuronal convolucional

- 3 capas de convolución
- 3 capas Pool
- normalización y droops entre ellas
- activación Relu
- Flatten para la red completamente conectada
- Capa densa 'Sigmoid' para la clasificación binaria (en estos casos funciona mejor que softmax)
- Se compila el modelo con los siguientes valores:
 - loss='squared _ hinge'
 - optimizer='Adam'
 - metrics=['accuracy']

6.-Se imprime un resumen del modelo de la CNN antes de realizar la corrida

7.-Se entrena el modelo (se elige la cantidad de épocas) y al final se guarda el modelo en el equipo de computo

8.-Una vez que termina de entrenar el modelo se grafican:

- la precisión y la pérdida en el entrenamiento
- la precisión y la pérdida en la validación

9.-Se muestran las gráficas de precisión y pérdida

10.-Se carga el modelo entrenado guardado en el paso 6

11.-Se evalúa el modelo y muestra el Accuracy

12.-Se muestra la matriz de confusión y los valores de la clasificación:

- Exactitud
- Precisión
- F1-score
- MCC
- Sensitividad
- Especificidad

13.-Se grafica la curva ROC

14.-Se muestra el valor del área bajo la curva (AUC).

4.6.1. Arquitectura de una CNN

Las representaciones aprendidas por un algoritmo de aprendizaje profundo se pueden observar en la figura 3.6 donde una red de varias capas de profundidad transforma una imagen de maíz o maleza para reconocer de qué tipo se trata.

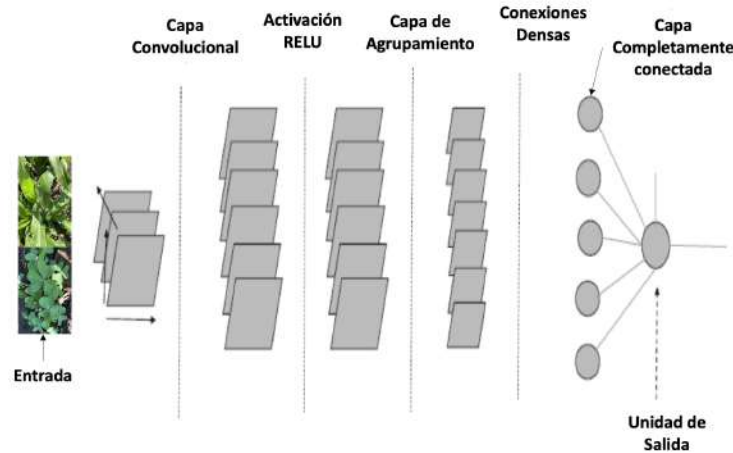


Figura 4.19: Arquitectura de una Red Neuronal Convolutiva. Pro Deep Learning with TensorFlow.

Como se puede verificar en la figura 3.6, la red modifica la imagen de maíz o maleza en representaciones que son cada vez más distintas de la imagen original y cada vez con más información sobre el resultado final. Se puede pensar en una red profunda como una operación de separación de información en varios pasos en la que la información atraviesa filtros sucesivos y se vuelve progresivamente más pura (es decir, más útil para una tarea).

El aprendizaje automático consiste en asignar entradas (como imágenes) a objetivos (como la etiqueta "maíz o maleza"), lo que se hace observando diversos ejemplos de imágenes y objetivos. También, las redes neuronales profundas asignan la entrada a este objetivo a través de una secuencia profunda de transformaciones de datos simples llamados capas y estas transformaciones en los datos se aprenden mediante la exposición a ejemplos. Ahora veamos cómo se produce este aprendizaje, concretamente.

La especificación de lo que hace una capa con sus datos de entrada almacenados en los pesos de la capa, que es esencialmente un conjunto de números. Técnicamente, diríamos que la transformación implementada por una capa es parametrizada por sus pesos. En este contexto, aprender significa encontrar un conjunto de valores para los pesos de todas las capas de la red, de modo que la red asigne correctamente las entradas de las muestras a sus objetivos relacionados. Pero la cuestión es que la red neuronal profunda puede contener millones de parámetros. Encontrar el valor correcto para cada uno de ellos puede parecer una tarea de enormes proporciones, especialmente si se modifica el valor de un parámetro afecta al comportamiento de todos los demás parámetros.

Para tener control de algo, primeramente se debe poder observar y medir. Si se quiere controlar la salida de la red neuronal, hay que ser capaz de medir lo lejos que está esa salida de lo que se deseaba. Este es el objetivo de la *función de pérdida* de la red, también llamada *función objetivo*.

La función de pérdida toma los pronósticos de la red y el objetivo real (lo que usted quería que la red produzca) y calcula una puntuación de distancia, que captura lo bien que lo ha hecho la red en el ejemplo que se está usando.

El truco básico en el aprendizaje profundo es utilizar este puntaje como una señal de retroalimentación para ajustar ligeramente el valor de los pesos, de manera que se reduzca la puntuación de pérdida para el ejemplo actual. Este ajuste es el trabajo del optimizador, que implementa el algoritmo *Backpropagation* (Retropropagación): es el algoritmo focal en el aprendizaje profundo.

Al principio, los pesos de la red se asignan a valores aleatorios, por lo que la red se limita a aplicar una serie de cambios aleatorios. Obviamente, su resultado dista mucho de lo que se espera de manera ideal, y por consecuencia la puntuación de las pérdidas es muy alta. Pero, con cada caso que la red procesa, los pesos se van ajustando de a poco hacia la dirección correcta, y la puntuación de pérdidas va disminuyendo. Este es el bucle de entrenamiento, que, insistiendo suficientes veces (normalmente decenas de iteraciones a lo largo de miles de ejemplos), genera valores de peso que minimizan la función de pérdida. Una red con una pérdida mínima es aquella en la que las salidas se acercan al máximo al objetivo, que es una red entrenada. Nuevamente, se trata de un mecanismo sencillo que, una vez escalado, acaba pareciendo mágico.

Ejemplo de arquitectura de una CNN + SVM (Híbrida)

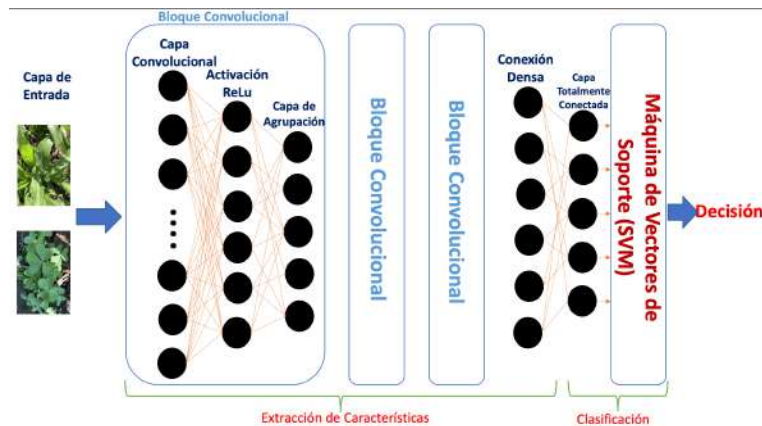


Figura 4.20: Metodología implementada en Tercera etapa CNN + SVM en ARTIIS 2021.
(Ramos et al., 2021)

Capítulo 5

Resultados

5.1. Técnicas clásicas

En este apartado se muestran los resultados obtenidos en las pruebas; se presentan los porcentajes de precisión para el reconocimiento de maíz y malezas implementando diferentes algoritmos de clasificación y los métodos de segmentación utilizados; además de comparar resultados entre ellos. Las *medidas de desempeño* y sus abreviaturas se muestran en las tablas de la siguiente forma: Accuracy=Acc, Precisión=P, Especificidad=E, Sensibilidad=S, Recall=R, F-measure=F-m y Coeficiente de Correlación de Mattew=MCC.

En la Tabla 5.1 se pueden distinguir los resultados obtenidos al implementar el método PCA y comparando los algoritmos de clasificación utilizados; Backpropagation con 98.78 % es el porcentaje de mejor precisión al extraer características texturales.

Clasificador	Acc	P	E	S	R	F-m	MCC
Naive Bayes	78.44 %	0.803 %	0.658 %	0.909 %	0.784 %	0.781 %	0.587 %
Random Forests	93.11 %	0.933 %	0.90 %	0.961 %	0.931 %	0.931 %	0.864 %
SVM	93.21 %	0.935 %	0.894 %	0.969 %	0.932 %	0.932 %	0.867 %
KNN	93.62 %	0.938 %	0.904 %	0.967 %	0.936 %	0.936 %	0.874 %
CNN	96 %	0.96 %	0.92 %	1.0 %	0.96 %	0.96 %	0.954 %
Backpropagation	98.78 %	0.988 %	0.985 %	0.989 %	0.988 %	0.988 %	0.976 %
Promedio	92.19 %	0.926 %	0.876 %	0.965 %	0.921 %	0.921 %	0.853 %

Tabla 5.1: Resultados de porcentajes con el método PCA.

En la Tabla 5.2 se pueden distinguir los resultados obtenidos al implementar el método de segmentación Otsu y comparando los algoritmos de clasificación utilizados; Backpropagation con 98.98 % es el porcentaje de mejor precisión al extraer características texturales.

Clasificador	Acc	P	E	S	R	F-m	MCC
Naive Bayes	77.3 %	0.792 %	0.647 %	0.898 %	0.773 %	0.769 %	0.564 %
Random Forests	91.9 %	0.92 %	0.897 %	0.941 %	0.919 %	0.919 %	0.839 %
SVM	95.6 %	0.956 %	0.935 %	0.975 %	0.956 %	0.956 %	0.912 %
KNN	93.2 %	0.933 %	0.909 %	0.955 %	0.932 %	0.932 %	0.866 %
Backpropagation	98.98 %	0.99 %	0.991 %	0.987 %	0.99 %	0.99 %	0.98 %
Promedio	92.16 %	0.925 %	0.883 %	0.959 %	0.921 %	0.921 %	0.852 %

Tabla 5.2: Resultado de porcentajes con segmentación Otsu.

En la siguiente gráfica se observan los mejores resultados de esta investigación; mostrando en el eje x los algoritmos implementados que son: Backpropagation, SVM, KNN, Naive Bayes, Random Forest y CNN; y en el eje de coordenadas y se puede ver la precisión de cada uno de los clasificadores mencionados. La barra en color verde corresponde a las pruebas ejecutadas y a los resultados conseguidos con el método *PCA* y la barra en color amarillo corresponde a las pruebas ejecutadas y a los resultados conseguidos con el método de segmentación *Otsu*.

En la siguiente figura se puede verificar que el método de segmentación *Otsu* y el clasificador *Backpropagation* obtuvieron 98.98 % siendo la precisión más alta.

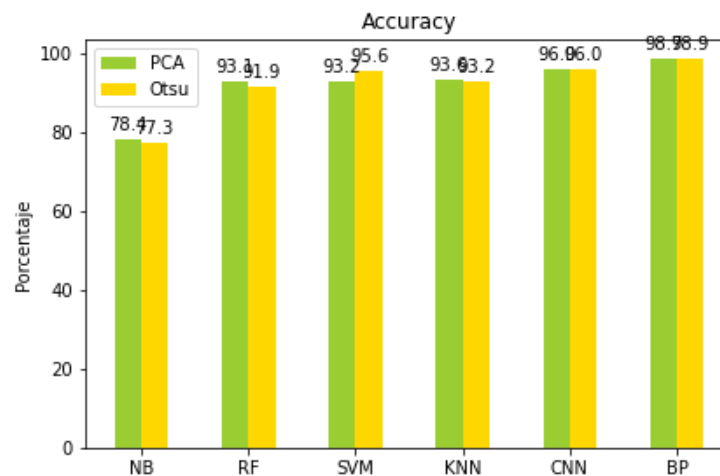


Figura 5.1: Gráfica mostrando los métodos para segmentar y sus resultados. Elaboracion propia en Python

En la Tabla 5.3 se pueden distinguir los resultados obtenidos al implementar el método PCA y comparando los algoritmos de clasificación utilizados; Backpropagation con 97.8 % es el porcentaje de mejor precisión al extraer características cromáticas.

Algoritmo	Acc	P	E	S	R	F-m	MCC
Naive Bayes	77 %	0.77 %	0.62 %	0.91 %	0.77 %	0.765 %	0.563 %
Random Forests	93.9 %	0.939 %	0.94 %	0.90 %	0.939 %	0.939 %	0.88 %
SVM	94.8 %	0.949 %	0.92 %	0.97 %	0.948 %	0.948 %	0.897 %
KNN	94.8 %	0.949 %	0.926 %	0.97 %	0.948 %	0.948 %	0.897 %
Backpropagation	97.8 %	0.978 %	0.97 %	0.986 %	0.978 %	0.978 %	0.956 %

Tabla 5.3: Resultados con método PCA y características cromáticas.

En la Tabla 5.4 se pueden distinguir los resultados obtenidos al implementar el método de segmentación Otsu y comparando los algoritmos de clasificación utilizados; Backpropagation con 97.9 % es el porcentaje de mejor precisión al extraer características cromáticas.

Algoritmo	Acc	P	E	S	R	F-m	MCC
Naive Bayes	79 %	0.79 %	0.62 %	0.89 %	0.79 %	0.787 %	0.6 %
Random Forests	93.2 %	0.933 %	0.90 %	0.958 %	0.932 %	0.932 %	0.865 %
SVM	96.4 %	0.965 %	0.942 %	0.986 %	0.964 %	0.964 %	0.929 %
KNN	94.2 %	0.944 %	0.912 %	0.972 %	0.942 %	0.942 %	0.886 %
Backpropagation	97.9 %	0.979 %	0.972 %	0.986 %	0.979 %	0.979 %	0.958 %

Tabla 5.4: Resultados con segmentación Otsu y características cromáticas.

En la Tabla 5.5 se pueden distinguir los resultados obtenidos al implementar el método de segmentación Otsu y comparando los algoritmos de clasificación utilizados; Backpropagation con 98.89 % es el porcentaje de mejor precisión al extraer una combinación de características texturales y cromáticas.

Algoritmo	Acc	P	E	S	R	F-m	MCC
Naive Bayes	78.68 %	0.80 %	0.66 %	0.89 %	0.68 %	0.784 %	0.592 %
Random Forests	93.63 %	0.937 %	0.908 %	0.946 %	0.936 %	0.936 %	0.874 %
SVM	98.38 %	0.984 %	0.966 %	0.982 %	0.984 %	0.984 %	0.968 %
KNN	94.64 %	0.947 %	0.918 %	0.956 %	0.946 %	0.946 %	0.894 %
Backpropagation	98.89 %	0.989 %	0.98 %	0.978 %	0.989 %	0.989 %	0.978 %

Tabla 5.5: Resultados con segmentación Otsu y con la combinación de características texturales y cromáticas.

En la Tabla 5.6 se pueden distinguir los resultados obtenidos al implementar el método PCA y comparando los algoritmos de clasificación utilizados; Backpropagation con 98.6 % es el porcentaje de mejor precisión al extraer una combinación de características texturales y cromáticas.

Algoritmo	Acc	P	E	S	R	F-m	MCC
Naive Bayes	78.03 %	0.804 %	0.63 %	0.912 %	0.78 %	0.776 %	0.583 %
Random Forests	94.73 %	0.949 %	0.906 %	0.966 %	0.947 %	0.947 %	0.896 %
SVM	97.57 %	0.976 %	0.95 %	0.978 %	0.976 %	0.976 %	0.952 %
KNN	94.53 %	0.947 %	0.896 %	0.976 %	0.945 %	0.945 %	0.893 %
Backpropagation	98.58 %	0.986 %	0.968 %	0.98 %	0.986 %	0.986 %	0.972 %

Tabla 5.6: Resultados con método PCA y con la combinación de características texturales y cromáticas.

Para finalizar con los resultados obtenidos en la primera etapa, en la Tabla 5.7 se muestra el porcentaje de accuracy obtenido al hacer la prueba en una red neuronal convolucional; se muestra de una manera separada de los demás conjuntos de datos mostrados en las tablas anteriores ya que no se realiza segmentación alguna con ningún método previo a la corrida, debido que, al hacer pruebas de segmentación previas a las corridas, el porcentaje de accuracy disminuye en lugar de mejorar; esta es la razón de hacer las corridas con las imágenes originales sin ningún tipo de tratamiento de segmentación ni extracción de características.

Algoritmo	Acc	P	E	S	R	F-m	MCC
CNN	97 %	96 %	0.92 %	1.0 %	0.96 %	0.96 %	0.954 %

Tabla 5.7: Resultados con la red neuronal convolucional (CNN)

Aplicamos validación cruzada con 5 k-Folds y usando 1000 imágenes. El 80 % de las imágenes se utilizaron para entrenamiento y el 20 % de las imágenes se utilizaron para la validación. Después de realizar la validación cruzada se obtuvieron los resultados de la Tabla 5.8 . El promedio de los resultados obtenidos muestra que se logró una precisión del 93 %.

Sección	VP	FP	FN	VN	Acc	P	E	S	F1
1	93 %	7 %	1 %	99 %	0.96 %	0.93 %	0.93 %	0.99 %	0.96 %
2	97 %	3 %	1 %	99 %	0.98 %	0.97 %	0.97 %	0.99 %	0.98 %
3	84 %	16 %	5 %	95 %	0.89 %	0.85 %	0.84 %	0.95 %	0.89 %
4	97 %	3 %	0 %	100 %	0.98 %	0.97 %	0.97 %	1.0 %	0.98 %
5	95 %	5 %	3 %	97 %	0.96 %	0.95 %	0.95 %	0.97 %	0.96 %
Promedio					0.95 %	0.93 %	0.93 %	0.98 %	0.95 %

Tabla 5.8: Resultados Validación Cruzada.

En la siguiente figura podemos observar una representación gráfica donde se presentan los mejores resultados de la segunda etapa de este trabajo, mostrando que el mejor método de segmentación utilizado fue Otsu, utilizando la extracción de características (texturales, cromáticas e híbridas) y mostrando el mejor algoritmo probado que arrojó los porcentajes a la vista. La barra roja representa los resultados en las pruebas implementadas con el método de segmentación Otsu y la extracción de características texturales, mostrando la precisión más alta obtenida con el algoritmo Backpropagation 98.98 %, la barra amarilla nos presenta los resultados en las pruebas realizadas con el método de segmentación Otsu y la extracción de características híbridas (texturales y cromáticas), mostrando la precisión más alta obtenida también con el algoritmo Backpropagation 98.89 %; la barra verde muestra los resultados en las pruebas implementadas con el método de segmentación Otsu y la obtención de características cromáticas, de igual manera que en los casos anteriores, muestra la precisión más alta obtenida con el algoritmo Backpropagation 97.9 % ; en la barra azul se muestra el porcentaje de efectividad usando las redes neuronales convolucionales que obtuvieron un 97 % de precisión. Finalmente, la barra gris muestra una precisión del 93 % con validación cruzada.

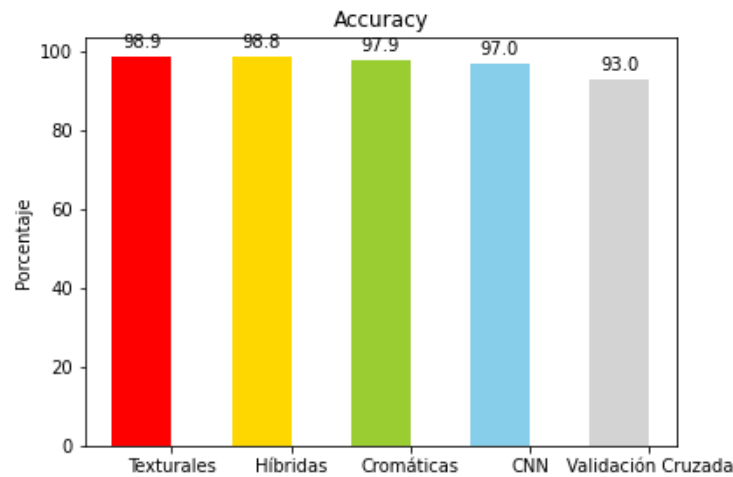


Figura 5.2: Gráfica de resultados con la mejor precisión. Elaboracion propia en Python

5.2. Redes Neuronales Convolucionales

La siguiente tabla corresponde a la tercera etapa, donde se pueden ver los resultados comparados con los resultados obtenidos con diferentes métodos encontrados en el estado del arte, el porcentaje mencionado corresponde a la métrica de Accuracy.

Autor	Tipo de imágenes	CNN	SVM	CNN+SVM
<i>Silva et. al. 2019</i>	Marcas de ganado	-	-	93.28
<i>Silva et. al. 2020</i>	Marcas de ganado	-	-	95.34
<i>Niu et. al. 2011</i>	Textos	-	-	94.4
<i>Abien et. al. 2019</i>	Dataset MNIST (números)	-	-	91.86
<i>Campos et. al. 2017</i>	Cultivo, suelo y maleza	-	-	94.3
<i>Jiang H. et. al. 2020</i>	Maíz y maleza	-	-	97.51
<i>Yeshwanth Sai et. al. 2020</i>	Perros	-	-	93.57
<i>Miao Ma et. al. 2016</i>	Gestos manuales	-	-	96.1
<i>Yang Li et. al. 2020</i>	Maíz, manzana y uva	-	-	94
<i>Zhicheng Wang et. al. 2017</i>	Detección de fuego	95.79	-	-
<i>Dechant et. al. 2017</i>	Tizón en la hoja de maíz	96.7	-	-
<i>Sibiya 2019</i>	Enfermedades de maíz	87 - 99	-	-
<i>Xihai Shang et. al. 2017</i>	Enfermedades de maíz	98.8 y 98.9	-	-
<i>Sumita Mishra et. al. 2020</i>	Enfermedades de maíz	88.46	-	-
<i>Zhanquan Sun et. al. 2017</i>	Imágenes falsas en la red	95.2	89.45	97.2
Método propuesto	Maíz y Maleza	-	-	98

Figura 5.3: Tabla comparativa de Resultados.

El método propuesto arrojó resultados con un porcentaje de precisión excelente, ya que ha sido comprobado y se puede observar en la tabla anterior que solo dos métodos de los 15 revisados en el estado del arte lo superan; es un método muy parecido al propuesto, ya que usan CNN para detectar enfermedades de maíz. (*Sibiya & Sumbwanyambe, 2019*) detecta la enfermedad de tizón en la hoja de maíz con accuracy del 99 %, pero las detecciones de las otras enfermedades no rebasan el 87 %. De igual forma (*Zhang et al., 2018*) detecta enfermedades en la hoja del maíz con un porcentaje de accuracy del 98.9 % con apoyo de modelos avanzados de redes pre-entrenadas como Google Net y conjuntos de datos con miles de imágenes como CIFAR-10 que ayudaron a obtener un 98.8 % respectivamente. Por lo que podemos mencionar que el porcentaje obtenido por nuestro

método es altamente efectivo con una precisión del **98 %**

En las siguientes imágenes podemos observar el valor del área bajo la curva (AUC por sus siglas en inglés) y su gráfica.

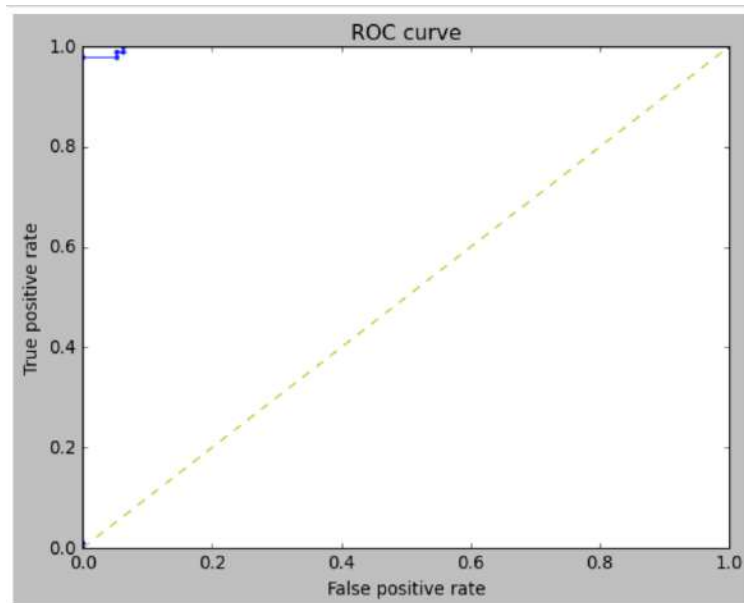


Figura 5.4: Curva ROC. Elaboracion propia en Python

AUC = 0.9988995598239296

Figura 5.5: Valor del area bajo la curva.

Para la cuarta etapa, nuestro conjunto de datos sobre las malas hierbas del maíz fué enriquecido y se tomó del entorno natural de cultivos de maíz. Se obtuvieron imágenes del estado de Hidalgo en el 2020; un año despues se obtuvieron del estado de Puebla y San Luis Potosí. Se utilizó la cámara de un iPhone 8 plus de 12 megapíxeles en posición vertical hacia el suelo para adquirir imágenes aéreas de gran calidad y que pudieran reducir la influencia del reflejo de la luz solar, todo esto a campo abierto y lo más parecido a las características y cambios ambientales para hacerlo lo más real posible. Se recolectaron imágenes de maíz y malas hierbas en días diferentes, iluminación y fondo del suelo, por ejemplo, humedad, nublado, soleado y residuos de paja en algunas situaciones. Se recogieron un total de 4500 imágenes de maleza y maíz, pero se realizó una depuración quedándonos solamente con 4167 imágenes funcionales para realizar las pruebas en nuestra CNN, una muestra de estas imágenes se pueden ver en las figuras 4.4 y 4.5

Como han podido observar, las imágenes de maíz son muy precisas o comunes en los cultivos, independientemente de que sean de diferentes estados, esto sucede al menos en México, no obstante, en la maleza si se puede identificar mayor variedad de hierbas malas por cada estado. En la experimentación de la fase 4, las imágenes originales que se utilizaron fueron: *2711 de maíz* y *1456 de maleza*, las cuales se redimensionaron a 150×150 para realizar la corrida en el lenguaje de programación Python ya que al usarlas con su tamaño original hubiera causado retrasos en

las corridas.

(Kamilaris & Prenafeta-Boldú, 2018) en su trabajo “Deep Learning in agriculture: A survey” nos muestra que la combinación de algunos algoritmos mejora la exactitud (Accuracy) y la precisión, si se comparan con los utilizados de forma individual o separada, lo podemos ver en los trabajos de (Lottes et al., 2017) y (Ramos et al., 2021)

Para mostrar los resultados obtenidos durante las pruebas realizadas, se usan las abreviaturas siguientes: Accuracy=Acc, Precisión=P, F1 score=F1, Coeficiente de Correlación de Matthew=Mcc, Sensibilidad=S, Especificidad=E y AUC= Area Bajo la curva que son las *medidas de desempeño* generadas en los resultados y time = tiempo de ejecución de cada prueba.

Algoritmo	Épocas	Acc	P	F1	Mcc	S	E	AUC	Time
CNN (Rmsprop)	300	<u>98.43</u>	99.23	98.76	96.65	98.68	98.29	99.2	300 min.
CNN (Adam)	200	<u>98.31</u>	99.04	98.66	96.39	98.36	98.29	99.32	180 min.
CNN ± SVM (Linear)	100	<u>99.72</u>	99.88	99.76	99.42	99.82	99.65	99.96	165 min.

Figura 5.6: Resultados en las pruebas usando CNN y CNN+SVM.

(Kamilaris & Prenafeta-Boldú, 2018) , menciona que para la detección de maleza y maíz en general se utilizan: cámaras ópticas y fotodetectores, imágenes hiperespectrales y multiespectrales; además de que las técnicas de análisis de datos más utilizadas para ello son: clasificación de píxeles basada en el clustering de k-means y el clasificador de Bayes, técnicas de extracción de características con FFT y GLCM, clasificación basada en wavelets y filtrado de Gabor, algoritmos genéticos, técnicas difusas, redes neuronales artificiales, segmentación por erosión y dilatación, regresión logística, detección de bordes, detección de color, análisis de componentes principales.

En la siguiente tabla se puede ver el resultado obtenido comparado contra los resultados de los diferentes métodos encontrados y mencionados en el apartado del estado del arte, la mayoría son experimentos de detección de maleza en algún tipo de cultivo y la métrica de desempeño que se muestra es el Accuracy.

Nombre	Imágenes	Maleza	Cultivo	Algoritmo
<i>Burgos-Artizzu et. al. 2011</i>	Maleza y cultivo	95%	80%	FIP + RCRD
<i>Dos Santos Ferreira et. al. 2017</i>	Maleza en cultivo de Soya	98.8%	N/A	CNN
<i>Dyrmann et. al. 2016</i>	Maleza	86.2%	N/A	VGG16
<i>Dyrmann et. al. 2017</i>	Maleza en cultivo de cereal	93%	94%	CNN
<i>Fann et. al. 2021</i>	Enfermedades en maíz	N/A	97.65 %	Red profunda DBN
<i>Farooq et. al. 2019</i>	Maleza	94.72	N/A	CNN
<i>Inkyu Sa. et. al. 2018</i>	Maleza	80%	N/A	CNN
<i>Kounalakis et. al. 2017</i>	Maleza en pastizales	89.09%	N/A	CNN
<i>Lameski et. al. 2017</i>	Maleza en cultivo de zanahoria	71.3%	64.1%	CNN
<i>Liu et. al. 2019</i>	Maleza en cultivo de zanahoria	91%	N/A	SVM
<i>Lottes et. al. 2017</i>	Maleza en cultivo de remolacha azucarera	97%	90%	RGB+NIR
<i>Mortensen et. al. 2016</i>	Maleza	79%	N/A	VGG16 CNN
<i>Rebetez et. al. 2016</i>	Maleza	90%	N/A	CNN + HistNN
<i>Xinshao and Cheng 2015</i>	Maleza	90.96%	N/A	PCANet + LMC
<i>McCool et. al. 2017</i>	Maleza	93.9%	N/A	Adapted Version of Inception V3
<i>Milioto et. al. 2017</i>	Maleza	97%	N/A	CNN
<i>Kussul et. al. 2017</i>	Maíz	N/A	94.60%	CNN
<i>Gómez et. al. 2020</i>	Maleza y Maíz	98.98%	98.98%	Backpropagation
<i>Gómez et. al. 2021</i>	Maleza y Maíz	97%	97%	CNN
<i>Ramos et. al. 2021</i>	Maleza y Maíz	98%	98%	CNN+SVM
<u>Método propuesto</u>	Maleza y Maíz	99.72%	99.72%	CNN+SVM

Figura 5.7: Comparación de resultados obtenidos vs el método propuesto.

El método propuesto arrojó resultados con un porcentaje de precisión excelente, ya que ha sido comprobado y se puede observar en la tabla anterior que ninguno de los 20 métodos encontrados en el estado del arte lo superan. La mayoría de los autores tratan de clasificar maleza y maíz o algún otro tipo de cultivo, también la mayoría de los métodos de clasificación utilizan redes neuronales convolucionales y algunas variantes de éstas como los modelos avanzados de redes pre-entrenadas

VGG16 e Inception V3; son pocos los autores que aún utilizan métodos tradicionales de machine learning para sus trabajos. Por lo que podemos mencionar que el porcentaje obtenido por nuestro método propuesto es altamente efectivo con un Accuracy de **99.72%** y una precisión del **99.88%**

```

[[575  1]
 [  3 864]]

Accuracy o Exactitud 0.9972279972279973
Precisión 0.9988439306358381
F1_score 0.997690531177829
MCC 0.994228483956718

Sensitivity : 0.9982638888888888
Specificity : 0.9965397923875432

```

Figura 5.8: Matriz de confusión y demás métricas de evaluación.

En la matriz de confusión se puede observar que solo 1 elemento fue considerado como falso positivo para la maleza, mientras que para el maíz solo 3 elementos fueron considerados falsos negativos del total de las imágenes utilizadas para las pruebas, también se muestran los resultados de las otras métricas de evaluación utilizadas.

En las siguientes imágenes podemos observar las gráficas de precisión, pérdida y area bajo la curva.

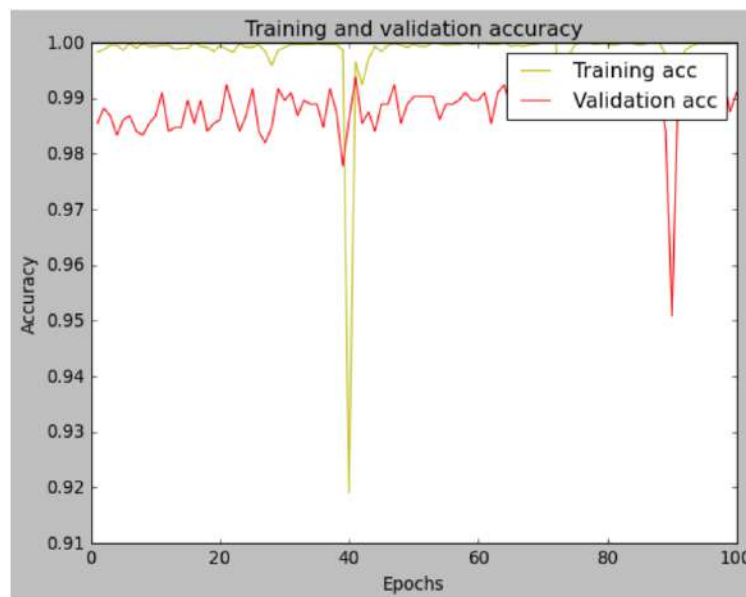


Figura 5.9: Gráfica de precisión en el entrenamiento.

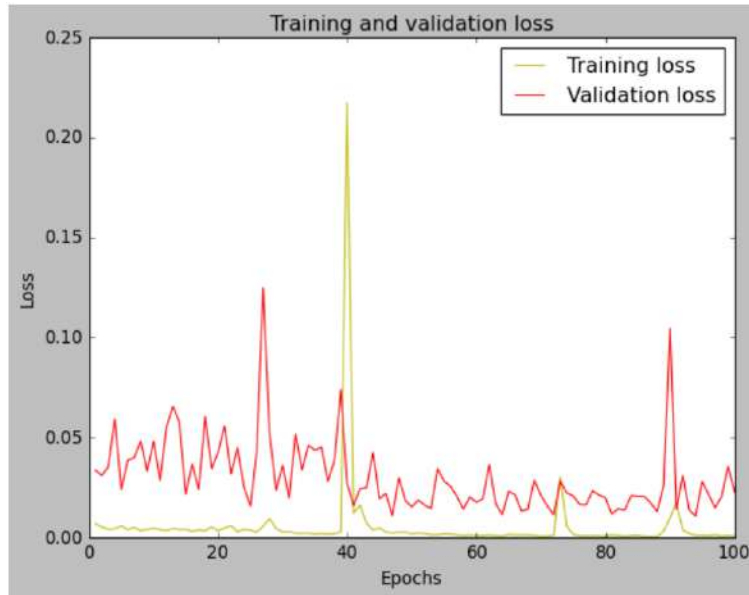


Figura 5.10: Gráfica de pérdida en el entrenamiento.

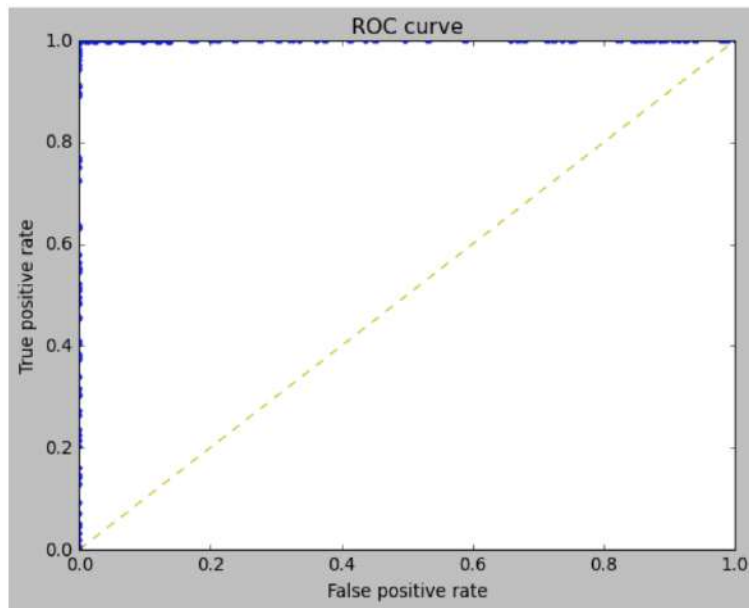


Figura 5.11: Gráfica del area bajo la curva.

Capítulo 6

Discusiones

Hoy en día los sensores, drones y la Inteligencia Artificial a través del aprendizaje máquina y aprendizaje profundo, comienzan a desempeñar un papel esencial en las prácticas de la agricultura inteligente o agricultura de precisión. La espectroscopia y los sensores terrestres han despertado un gran interés en el campo de la detección de malas hierbas. La mayoría de los estudios se centran en la detección en condiciones ideales, como en interiores o bajo iluminación artificial y se necesitan más estudios en el entorno real del campo para probar la aplicabilidad de esta tecnología.

6.1. Técnicas clásicas vs CNN

En las técnicas clásicas de clasificación encontramos que utilizando el método de segmentación Otsu y extrayendo características texturales, arroja la precisión más alta obtenida con el algoritmo Backpropagation 98.98%. Las desventajas que podemos encontrar son: que en ocasiones nuestras imágenes no se ven bien segmentadas, se necesitan dos programas computacionales diferentes para la obtención de resultados esperados; para la generación del vector de características utilizando Matlab se lleva demasiado tiempo computacional, además de sumarle el tiempo de clasificación en el software Weka genera que estos resultados se desestimen frente a los encontrados por la CNN aún siendo un muy buen porcentaje de precisión.

Con la técnica de redes neuronales convolucionales se alcanzó el porcentaje de precisión de 96% y 97% en la primera y segunda etapa respectivamente. Con el algoritmo híbrido, donde utilizamos la combinación de las CNN como extractor de características y las SVM como clasificador, se obtuvo un 98% en la tercera etapa y un 99.88% en la etapa final. En la Fig. 5.6 se ven los tiempos computacionales generados durante las pruebas en CNN, los cuales son una tercera parte de los invertidos en los algoritmos usados en las técnicas clásicas, debido a que se necesita el funcionamiento de un solo software para la obtención de los resultados esperados.

Un inconveniente o una barrera que se debe considerar en el método propuesto, es que en el uso del aprendizaje profundo se necesitan grandes conjuntos de datos que sirven de entrada durante el procedimiento de entrenamiento. Por lo que, el uso de un Dataset propio en las condiciones más reales posibles, además del desbalanceo de datos y un Dataset cercano a las 5000 imágenes funciona para cultivo de maíz y maleza; pero no garantiza que podría funcionar con la misma precisión para algún otro cultivo debido a la complejidad del problema, es decir, cantidad de clases, precisión requerida, etc. Otra limitación es el hecho de que los modelos de aprendizaje profundo

pueden aprender algunos problemas particularmente bien, como en nuestro método propuesto e incluso generalizar en algunos aspectos, pero no pueden generalizar más allá del conjunto de datos utilizado.

6.2. Comparación con otros trabajos relacionados

El método propuesto en la tercera etapa arrojó resultados con un porcentaje de precisión altamente efectivo del 98% , ya que ha sido comprobado y se puede observar en la Fig. 5.3 que solo dos métodos de los revisados en el estado del arte lo superan; ya que usan CNN para detectar enfermedades de maíz. Esto sucede con apoyo de modelos avanzados de redes pre-entrenadas como Google Net y conjuntos de datos con miles de imágenes como CIFAR-10 que ayudaron a obtener precisiones superiores a la encontrada.

El método propuesto en la cuarta etapa arrojó resultados con un porcentaje de precisión excelente, con un Accuracy de 99.72% y una precisión del 99.88% las cuales se pueden observar en la Fig. 5.7 que ninguno de los 20 métodos encontrados en el estado del arte lo superan. La mayoría de los autores tratan de clasificar maleza y maíz, también la mayoría de los métodos de clasificación utilizan CNN y algunas variantes de éstas como los modelos avanzados de redes pre-entrenadas VGG16 e Inception V3.

El aprendizaje profundo se debe aplicar para resolver diversos problemas agrícolas de clasificación o predicción, relacionados con la visión por computadora y el análisis de imágenes. Los beneficios son alentadores para alcanzar una agricultura inteligente y sostenible, además de una producción de alimentos más segura y sin contaminantes.

Capítulo 7

Conclusiones

Esta investigación tuvo como objetivo identificar plantas de maíz para diferenciarlas de la maleza usando Visión Artificial. La limitación principal que se encontró fue que no existía un Dataset que sirviera para poder realizar los experimentos deseados para diferenciar el maíz de la maleza; por lo que se realizó la recolección de imágenes en 3 estados diferentes de la República Mexicana creando así un Dataset propio. Con base en las fases de experimentación con el Dataset utilizado, se puede concluir que el uso de algoritmos tradicionales como Naive Bayes, Random Forest, K-nn, SVM y Backpropagation arrojaron buenos porcentajes de precisión en los resultados de los experimentos, pero se necesita de pre-procesamiento en las imágenes de maíz y maleza para obtenerlos, esto requiere más recursos y tiempo computacional.

Por otro lado, las últimas dos fases de experimentación confirman que el algoritmo híbrido utilizado donde la extracción de características hecha por la CNN y la clasificación hecha por el algoritmo SVM ha obtenido excelentes resultados en menor tiempo y sin hacer uso de pre-procesamiento de las imágenes; el porcentaje de efectividad logrado en la detección de maíz y maleza ha sido de un 99.88 % de precisión.

Las recomendaciones para ampliar o continuar con el tema de investigación sería aumentar la cantidad de imágenes del Dataset tanto de maíz como de maleza con Data Augmentation o poder obtener imágenes de más estados de la República Mexicana para realizar nuevos experimentos con algoritmos como Adaboost, Gboost y XGBoost que son usados en la actualidad para mejorar el rendimiento de los algoritmos de aprendizaje. Se tiene pensado publicar el Dataset con el cual se trabajó en las diferentes etapas de experimentación para facilitar a mas estudiantes e investigadores los medios para replicar las pruebas aquí realizadas o mejorarlas.

Capítulo 8

Publicaciones

Primer artículo COMIA 2020 - Publicado en la Revista Research in Computer Science - “Clasificación de plantas de maíz y maleza: Hacia la mejora de la fertilización en México” https://www.rcs.cic.ipn.mx/2020_149_8/

ISSN 1870-4069

**Clasificación de plantas de maíz y maleza:
Hacia la mejora de la fertilización en México**

Marcos Yamir Gómez-Ramos, J. Sergio Ruiz-Castilla,
Farid Garefa-Lamont

Universidad Autónoma del Estado de México,
Centro Universitario UAEM Texcoco,
Estado de México,
México

{marcos.yamir, jsorgiorulzc}@gmail.com,
fglamont@yahoo.com.mx

Resumen. El cultivo del maíz es muy importante en México y aún podemos encontrar métodos tradicionales de labranza desde la siembra hasta la cosecha; sin embargo, se llegan a observar algunos ejemplos de uso de maquinaria sofisticada. El maíz se siembra en hileras llamadas surcos, separados por unos 0.75 m entre sí. Se siembran de 4 a 5 semillas por mata a una distancia de 0.50 m entre cada una. Por diversos factores, no siempre nace el 100% de semillas, incluso en ocasiones no germina ninguna semilla. En las primeras semanas se lleva a cabo la fertilización, que consiste en depositar una cantidad de fertilizante químico al pie de la planta. Dicho proceso se hace de manera manual, en el que algún trabajador coloca un puñado de la sustancia por mata, buscando hacerlo de la mejor manera posible, aunque no exacta. Por ende, la actividad resulta lenta, tardada y costosa. Actualmente se ha implementado el uso de maquinaria que deposita fertilizante a lo largo de todo el surco. Con este método se hace más rápido y barato. Sin embargo, se esparce fertilizante en donde hay ausencia de mata de maíz o incluso a la maleza. El fertilizante que no es absorbido por la planta se va al subsuelo, generando contaminación de los mantos acuíferos. El presente trabajo propone la clasificación de plantas de maíz y de maleza, pues al identificar la planta de maíz es posible depositar fertilizante a éstas evitando la maleza e incluso donde no hay planta. Para lograr el objetivo, se obtuvieron 1000 imágenes de maíz y maleza. Se aplicaron métodos de procesamiento de imágenes, como segmentación con PCA y Otsu; después se aplicaron algoritmos de clasificación: Naive Bayes, Random Forest, SVM, KNN, CNN y redes neuronales Backpropagation logrando resultados con hasta el 98.98% de precisión. Con la finalidad de que la solución mejore el proceso de fertilización.

Palabras clave: Visión artificial, extracción de características, clasificación, fertilización.

Figura 8.1: Primer página del primer artículo publicado.

Segundo artículo Publicado en International Journal of combinatorial optimization problems and informatics. (IJCOPI, 2021) “Classification of corn plants and weed based on characteristics of color and texture using methods of segmentation Otsu and PCA” <https://www.ijcopi.org/ojs/article/view/218>



www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 12(3), Sep-Dec 2021, 98-108. ISSN: 2007-1558.

Corn plants and weeds classification using the Otsu segmentation method and PCA

Marcos Yamir Gómez Ramos, José Sergio Ruiz Castilla, Farid García Lamont

Universidad Autónoma del Estado de México, Doctorado en Ciencias de la Computación, Centro Universitario UAEM Texcoco, Texcoco, Estado de México, México.

marcos.yamir@gmail.com, jserroruiz@gmail.com, fglamont@yahoo.com.mx

Abstract. The corn crop is very important in Mexico. Corn is fertilized manually or with machinery. When fertilization is manual, it consists of depositing fertilizer to each corn plant. Whereas machine fertilization, involve of dropping fertilizer along the furrow continuously. Manual fertilization is effective, but it is expensive and time-consuming. Machine fertilization can be inefficient, because fertilizer is deposited in the weeds or where there is no corn plant. When the fertilizer is not absorbed by the plant, it can damage the aquifers. This project presents algorithms to classify corn plants and weeds, hoping to contribute to automated fertilization or identified weeds to apply herbicide or eliminate. We took hundreds of pictures of corn plants and weeds in corn crops. The images were segmented using the Otsu method. As well as, the images were processed with the PCA algorithm. We apply classification algorithms such as Naive Bayes, Random Forest, SVM, KNN and Backpropagation. We also apply a convolutional neural network (CNN). We finally got 99.97% as the best result with the Backpropagation classifier.

Keywords: Classification, Backpropagation, Segmentation, Corn plants, Weeds, Otsu, PCA.

Article info

Received Jan 17, 2021

Accepted May 11, 2021

1 Introduction

Corn is very important to Mexico. Corn is representative in the social, cultural and economic aspects. There are two classes of corn in Mexico: white and yellow. The original varieties are 59 in Mexico. Corn production during 2017 was 27.8 million tons in Mexico. The main producers are: Sinaloa 22%, Jalisco 14%, Estado de Mexico 8%, Michoacán 7%, Guanajuato 6%, Guerrero 5%, Veracruz 5%, Chiapas 5%, Chihuahua 4%, Puebla 4% and other 20% are divided among other states. Mexico is the eighth producer in the world [1].

The corn crop must grow without weeds; however, this is not always possible. Weeds must be eliminated by applying herbicides such as Acetochlor or Fluroxipyr 30 days after sowing. Nitrogen fertilizer should be applied 50 to 65 days after sowing. [2]. When the weeds are not removed, the fertilizer is absorbed by the corn plant, but at the same time by the weeds. Wild plants can affect corn growth and production. Fertilizers not absorbed by plants can affect aquifers.

Figura 8.2: Primer página del segundo artículo publicado.

Tercer artículo Publicado en la revista Communications in Computer and Information Science de Springer. (2021) “Hybrid algorithm of convolutional neural networks and vector support machines in classification” https://link.springer.com/chapter/10.1007/978-3-030-90241-4_21



Hybrid Algorithm of Convolutional Neural Networks and Vector Support Machines in Classification

Marcos Yamir Gómez Ramos^(✉), José Sergio Ruíz Castilla,
and Farid García Lamont

Universidad Autónoma del Estado de México (UAEMEX), Jardín Zumpango s/n,
Fraccionamiento El Tejocote, Texcoco, Estado de México, Mexico
{fgarcial, jsruizc}@uaemex.mx, mgomezr008@alumno.uaemex.mx
<https://www.uaemex.mx>

Abstract. Looking for the improvement of the classification, we propose a hybrid algorithm to identify the corn plant and the weed. With the aim of improving the fertilization and herbicide application processes. An efficient process can avoid wasted fertilizers and decrease subsoil contamination. The purpose is to identify the corn plant to specify the fertilizer application in an automated and precise way. Whereas, the identification of the weed allows to apply herbicides directly. In this work we propose a hybrid method with Convolutional Neural Networks (CNN) to extract characteristics from images and Vector Support Machines (SVM) for classification. We obtained effectiveness results, a percentage of 98%, being higher than those compared to the state of the art.

Keywords: Classification · Convolutional neural network · Support vector machine · Hybrid algorithm

1 Introduction

According to the national agricultural survey in Mexico (NAS, 2019), corn is the second crop in the country in terms of annual production, surpassed only by sugar cane, but above crops such as wheat, sorghum, tomato, Chile and beans [1]. Of the 64 breeds reported in Mexico, 59 can be considered native. Mexico is the cradle of corn. Also, it is the origin of an enormous diversity of varieties. The corn has been the sustenance of their peoples. The population uses corn in countless ways. Finally, corn is the source of cultural and social wealth for Mexicans [2].

Chemical fertilizers unquestionably have effects on the environment. Fertilizers generate a high risk of environmental damage. Fertilizers can contaminate subsoil groundwater in the area of application. A fertilizer is a substance that is used to provide nutrients to the soil. The objective of fertilizers is to increase the concentration of nutrients to favor and promote plant growth [3].

Figura 8.3: Primer página del tercer artículo publicado.

Referencias

- Aakif, A. & Khan, M. F. (2015). Automatic classification of plants based on their leaves. *Biosystems Engineering*, 139, 66–75.
- Agarap, A. F. (2017). An architecture combining convolutional neural network (cnn) and support vector machine (svm) for image classification.
- Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, 6(1), 37–66.
- ASERCA, A. d. S. a. l. C. y. D. d. M. A. (2019). Maíz grano cultivo representativo de México.
- Baez, I. (2018). Los robots: Un nuevo paradigma en la agricultura.
- Balasubramaniam, P. & Ananthi, V. (2016). Segmentation of nutrient deficiency in incomplete crop images using intuitionistic fuzzy c-means clustering algorithm. *Nonlinear Dynamics*, 83.
- Baruffaldi, J. M. (2016). Redes neuronales adversarias para el reconocimiento de malezas.
- Basly, H., Ouarda, W., Sayadi, F. E., Ouni, B., & Alimi, A. M. (2020). Cnn-svm learning approach based human activity recognition. In El Moataz, A., Mammass, D., Mansouri, A., & Nouboud, F. (Eds.), *Image and Signal Processing*, (pp. 271–281)., Cham. Springer International Publishing.
- Breiman, L. (2001). Random forest. *Machine Learning*, 45(1), 5–32.
- Burkov, A. (2019). volume 1. Andriy Burkov.
- Campos, Y., Sossa, H., & Pajares, G. (2017). Comparative analysis of texture descriptors in maize fields with plants, soil and object discrimination. *Precision Agriculture*, 18, 1–19.
- Cervantes, J., Lamont, F. G., Mazahua, L. R., Hidalgo, A. Z., & Castilla, J. S. R. (2018). Complex identification of plants from leaves. In *Intelligent Computing Methodologies* (pp. 376–387). Springer International Publishing.
- Cervantes, J., Taltempa, J., García-Lamont, F., Castilla, J. S. R., Rendon, A. Y., & Jalili, L. D. (2017). Análisis comparativo de las técnicas utilizadas en un sistema de reconocimiento de hojas de planta. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 14(1), 104–114.
- Chau, A. L., Hernandez, R. R., Mora, V. T., Canales, J. C., Mazahua, L. R., & Lamont, F. G. (2017). Detection of compound leaves for plant identification. *IEEE Latin America Transactions*, 15(11), 2185–2190.
- Chen, Y.-B., Wang, D., Ge, X.-L., Zhao, B.-G., Wang, X.-C., & Wang, B.-C. (2016). Comparative proteomics of leaves found at different stem positions of maize seedlings. *Journal of Plant Physiology*, 198, 116–128.

- chen He, D. & Wang, L. (1990). Texture unit, texture spectrum, and texture analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 28(4), 509–512.
- Cheng, B. & Matson, E. T. (2015). A feature-based machine learning agent for automatic rice and weed discrimination. In *ICAISC*.
- Choi, K. H., Han, S. K., Han, S. H., Park, K.-H., Kim, K.-S., & Kim, S. (2015). Morphology-based guidance line extraction for an autonomous weeding robot in paddy fields. *Comput. Electron. Agric.*, 113(C), 266–274.
- Chollet, F. (2018). volume 1. Manning Publications Co.
- Chua, L. S. (2016). Untargeted MS-based small metabolite identification from the plant leaves and stems of *impatiens balsamina*. *Plant Physiology and Biochemistry*, 106, 16–22.
- Cuevas, E., Zaldivar, D., & Pérez, M. (2010). volume 1. Alfaomega-Rama.
- Dong, M., Mu, S., Su, T., & Sun, W. (2019). Image recognition of peanut leaf diseases based on capsule networks. In *Communications in Computer and Information Science* (pp. 43–52). Springer Singapore.
- Durmus, H., Güneş, E. O., & Kırcı, M. (2017). Disease detection on the leaves of the tomato plants by using deep learning. In *2017 6th International Conference on Agro-Geoinformatics*, (pp. 1–5).
- E.G., A., J.C., C., J.E., C., J.S.R, C., & F.G., L. (2020). Identification of diseases and pests in tomato plants through artificial vision. In *Intelligent Computing Methodologies*, volume 12465 of *Lecture Notes in Computer Science*, (pp. 98–109). Springer, Cham.
- feng Wang, X., Wang, Z., & wen Zhang, S. (2019). Segmenting crop disease leaf image by modified fully-convolutional networks. In *Intelligent Computing Theories and Application* (pp. 646–652). Springer International Publishing.
- Fuentes, A., Yoon, S., Kim, S. C., & Park, D. S. (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors*, 17(9).
- Gabor, D. (1946). Theory of communication. *Journal of the Institution of Electrical Engineers*, (93), 429–441.
- Garcia, F., Cervantes, J., Lopez, A., & Alvarado, M. (2016). Fruit classification by extracting color chromaticity, shape and texture features: Towards an application for supermarkets. *IEEE Latin America Transactions*, 14(7), 3434–3443.
- Garcia-Lamont, F., Cervantes, J., López, A., & Rodriguez, L. (2018). Segmentation of images by color features: A survey. *Neurocomputing*, 292, 1–27.
- Gonzalez, R., Woods, R., & Eddins, S. (2004). volume 1. Pearson Education India.
- Guijarro, M., Riomoros, I., Pajares, G., & Zitinski, P. (2015). Discrete wavelets transform for improving greenness image segmentation in agricultural images. *Comput. Electron. Agric.*, 118, 396–407.

- Gómez, M. Y., Castilla, J. S. R., & Lamont, F. G. (2020). Clasificación de plantas de maíz y maleza: Hacia la mejora de la fertilización en México. *Research in Computing Science*, 149(8), 683–697.
- Gómez, M. Y., Castilla, J. S. R., & Lamont, F. G. (2021). Classification of corn plants and weed based on characteristics of color and texture using methods of segmentation otsu and pca. *International Journal of Combinatorial Optimization Problems and Informatics*, 12(3), 98–108.
- Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6), 610–621.
- Harrison, M. (2019). volume 1. O'Reilly.
- Haug, S., Michaels, A., Biber, P., & Ostermann, J. (2014). Plant classification system for crop /weed discrimination without segmentation. *IEEE Winter Conference on Applications of Computer Vision*, 1142–1149.
- Hlaing, S. H. & Khaing, A. S. (2014a). Weed and crop segmentation and classification using area thresholding. *International Journal of Research in Engineering and Technology*, 03, 375–382.
- Hlaing, S. H. & Khaing, A. S. (2014b). Weed and crop segmentation and classification using area thresholding. *International Journal of Research in Engineering and Technology*, 03, 375–382.
- Hu, M.-K. (1962). Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2), 179–187.
- Iglesias, R. B. (2013). volume 1. Editorial UOC, S.L.
- Imanov, E. & Alzouhbi, A. K. (2018). Machine learning comparative analysis for plant classification. In *13th International Conference on Theory and Application of Fuzzy Systems and Soft Computing — ICAFS-2018* (pp. 586–593). Springer International Publishing.
- Jalili, L. D., Morales, A., Cervantes, J., & Ruiz-Castilla, J. S. (2016). Improving the performance of leaves identification by features selection with genetic algorithms. In *Communications in Computer and Information Science* (pp. 103–114). Springer International Publishing.
- Jiang, G., Wang, Z., & Liu, H. (2015). Automatic detection of crop rows based on multi-roi. *Expert Systems with Applications*, 42(5), 2429–2441.
- Jiang, H., Zhang, C., Qiao, Y., Zhang, Z., Zhang, W., & Song, C. (2020). Cnn feature based graph convolutional network for weed and crop recognition in smart farming. *Computers and Electronics in Agriculture*, 174, 105450.
- Jiao, Z., Zhang, L., Yuan, C.-A., Qin, X., & Shang, L. (2019). Plant leaf recognition based on conditional generative adversarial nets. In *Intelligent Computing Theories and Application* (pp. 312–319). Springer International Publishing.
- John, G. H. & Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, (pp. 338–345), San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Kamilaris, A. & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147, 70–90.

- Kazmi, W., Garcia-Ruiz, F. J., Nielsen, J., Rasmussen, J., & Jørgen Andersen, H. (2015). Detecting creeping thistle in sugar beet fields using vegetation indices. *Computers and Electronics in Agriculture*, *112*, 10–19. Precision Agriculture.
- Lameski, P., Zdravevski, E., Trajkovik, V., & Kulakov, A. (2017). Weed detection dataset with rgb images taken under variable light conditions. In *ICT Innovations*.
- Lanlan, W. & You-xian, W. (2009). Weed/corn seedling recognition by support vector machine using texture features. *African Journal of Agricultural Research*, *4*, 840–846.
- Li, Y., Nie, J., & Chao, X. (2020). Do we really need deep cnn for plant diseases identification? *Computers and Electronics in Agriculture*, *178*, 105803.
- Liang, C., Xiong, J., Zheng, Z., Zhong, Z., Li, Z., Chen, S., & Yang, Z. (2020). A visual detection method for nighttime litchi fruits and fruiting stems. *Computers and Electronics in Agriculture*, *169*, 105192.
- Lottes, P., Khanna, R., Pfeifer, J., Siegwart, R., & Stachniss, C. (2017). Uav-based crop and weed classification for smart farming. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, (pp. 3024–3031).
- Martínez, A. C. (2017). Reconocimiento de imágenes con redes convolucionales en c.
- Meng, Q., Qiu, R., He, J., Zhang, M., Ma, X., & Liu, G. (2015). Development of agricultural implement system based on machine vision and fuzzy control. *Computers and Electronics in Agriculture*, *112*, 128–138. Precision Agriculture.
- Montalvo Martínez, M., Guijarro, M., Guerrero, J., & Ribeiro, A. (2016). Identification of plant textures in agricultural images by principal component analysis. (pp. 391–401).
- MSC, G. (2019). Principales cultivos de México.
- Muñoz, M. J. (2011). Reconocimiento de malezas por medio de matrices de coocurrencia, dimensión fractal y redes neuronales artificiales.
- Niño, D. A., Castilla, J. S. R., Zenteno, M. D. A., & Jalili, L. D. (2019). Complex leaves classification with features extractor. In *Intelligent Computing Theories and Application* (pp. 758–769). Springer International Publishing.
- Niu, X.-X. & Suen, C. Y. (2012). A novel hybrid cnn–svm classifier for recognizing handwritten digits. *Pattern Recognition*, *45*(4), 1318–1325.
- Nowakowski, K., Boniecki, P., Tomczak, R., & Raba, B. (2011). Identification process of corn and barley kernels damages using neural image analysis. *Journal of Research and Applications in Agricultural Engineering*, *56*, 103–105.
- Oleary, M. (2016). Maíz: de México para el mundo.
- P. Constante, A. Gordón, O. C. E. P. I. E. F. A. (2016). Artificial vision techniques for strawberry's industrial classification. *IEEE Latin America Transactions*, *14*(6).
- Padilla, J. L. B. (2016). Robot móvil terrestre para la aplicación de agroquímicos en invernaderos controlados.

- Parvathi, S. K. M., Boigelot, B., & Mercatoris, B. (2015). Effective segmentation of green vegetation for resource-constrained real-time applications. *Precision Agriculture*, 257–266.
- Pattanayak, S. (2017). volume 1. Springer Science.
- Prasad, S., Kumar, P., Hazra, R., & Kumar, A. (2012). Plant leaf disease detection using gabor wavelet transform. In Panigrahi, B. K., Das, S., Suganthan, P. N., & Nanda, P. K. (Eds.), *Swarm, Evolutionary, and Memetic Computing*, (pp. 372–379)., Berlin, Heidelberg. Springer Berlin Heidelberg.
- Pueschel, P., Newnham, G., Rock, G., Udelhoven, T., Werner, W., & Hill, J. (2013). The influence of scan mode and circle fitting on tree stem detection, stem diameter and volume extraction from terrestrial laser scans. *ISPRS Journal of Photogrammetry and Remote Sensing*, 77, 44–56.
- Ramos, M. Y. G., Castilla, J. S. R., & Lamont, F. G. (2021). Hybrid algorithm of convolutional neural networks and vector support machines in classification. In *Advanced Research in Technologies, Information, Innovation and Sustainability. ARTIIS 2021*, volume 1485 of *Communications in Computer and Information Science*, (pp. 263–274). Springer, Cham.
- Ramírez, D. (2012). Desarrollo de un método de procesamiento de imágenes para la discriminación de maleza en cultivos de maíz.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533–536.
- Schor, N., Bechar, A., Ignat, T., Dombrovsky, A., Elad, Y., & Berman, S. (2016). Robotic disease detection in greenhouses: Combined detection of powdery mildew and tomato spotted wilt virus. *IEEE Robotics and Automation Letters*, 1(1), 354–360.
- Serrano, A. G. (2016). volume 1. Alfaomega Grupo Editor, S.A. de C.V.
- Shijie, J., Peiyi, J., Siping, H., & sLiu Haibo (2017). Automatic detection of tomato diseases and pests based on leaf images. *2017 Chinese Automation Congress (CAC)*, 2537–2510.
- Sibiya, M. & Sumbwanyambe, M. (2019). A computational procedure for the recognition and classification of maize leaf diseases out of healthy leaves using convolutional neural networks. *AgriEngineering*, 1(1), 119–131.
- Silva, A. R. E., Álvarez Cobelas, M., & González, E. M. (2017). Impactos del nitrógeno agrícola en los ecosistemas acuáticos. *Ecosistemas*, 26(1), 37–44.
- Silva, C. & Welfer, D. (2019). A novel hybrid svm-cnn method for extracting characteristics and classifying cattle branding. *Latin American Journal of Computing*, 6(1), 9 – 16.
- Sonka, M., Hlavac, V., & Boyle, R. (2010). volume 1. Springer, Boston, MA.
- Suzuki, R., Fukuta, S., Matsumoto, Y., Hasegawa, T., Kojima, H., Hotta, M., & Miyake, N. (2016). Development of reverse transcription loop-mediated isothermal amplification assay as a simple detection method of chrysanthemum stem necrosis virus in chrysanthemum and tomato. *Journal of Virological Methods*, 236, 29–34.

- Tewari, V., Ashok Kumar, A., Nare, B., Prakash, S., & Tyagi, A. (2014). Microcontroller based roller contact type herbicide applicator for weed control under row crops. *Computers and Electronics in Agriculture*, *104*, 40–45.
- Torres-Sánchez, J., López-Granados, F., & Peña, J. (2015). An automatic object-based method for optimal thresholding in uav images: Application for vegetation detection in herbaceous crops. *Computers and Electronics in Agriculture*, *114*, 43–52.
- Vapnik, V. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, *10*(5), 988–999.
- Wang, L. & He, D.-C. (1990). Texture classification using texture spectrum. *Pattern Recognition*, *23*(8), 905–910.
- Wei, X., Jia, K., Lan, J., Li, Y., Zeng, Y., & Wang, C. (2014). Automatic method of fruit object extraction under complex agricultural background for vision system of fruit picking robot. *Optik - International Journal for Light and Electron Optics*, *125*(19), 5684–5689.
- Yang, C.-C. (2000). Application of artificial neural networks in image recognition and classification of crop and weeds. *Canadian Biosystems Engineering / Le Genie des biosystems au Canada*, *42*, 147–152.
- Ye, M., Cao, Z.-G., Yu, Z., & Bai, X. (2015). Crop feature extraction from images with probabilistic superpixel markov random field. *Computers and Electronics in Agriculture*, *114*.
- Zaforas, M. (2017). Inteligencia artificial como servicio: reconocimiento de imágenes.
- Zhang, K., Wu, Q., Liu, A., & Meng, X. (2018). Can deep learning identify tomato leaf disease? *Advances in Multimedia*, *2018*, 1–10.
- Zhang, S., Wang, X., & Wang, Z. (2019). Weed recognition in wheat field based on sparse representation classification. In *Intelligent Computing Theories and Application* (pp. 511–519). Springer International Publishing.
- Zhang, X., Qiao, Y., Meng, F., Fan, C., & Zhang, M. (2018). Identification of maize leaf diseases using improved deep convolutional neural networks. *IEEE Access*, *6*, 30370–30377.
- Zhao, C., Chan, S. S., Cham, W.-K., & Chu, L. (2015). Plant identification using leaf shapes—a pattern counting approach. *Pattern Recognition*, *48*(10), 3203–3215.
- Zheng, Y., Yuan, C.-A., Shang, L., & Huang, Z.-K. (2019). Leaf recognition based on capsule network. In *Intelligent Computing Theories and Application* (pp. 320–325). Springer International Publishing.