

Universidad Autónoma del Estado de México

Centro Universitario UAEM Texcoco



T E S I S

**Uso de Técnicas de Visión Artificial para la
Identificación de Daños Foliareos Causados por Plagas y
Enfermedades en Plantas de Jitomate**

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

Presenta:

Ernesto García Amaro

Director de tesis: Dr. Jair Cervantes Canales

Tutor: Dr. José Sergio Ruiz Castilla

Tutor: Dr. Farid García Lamont

Índice general

1. Introducción	8
1.1. Planteamiento del problema	9
1.2. Justificación	10
1.3. Objetivo general	10
1.4. Objetivos específicos	11
1.5. Hipótesis	11
1.6. Pregunta de investigación	11
2. Estado del arte	12
2.1. Trabajos de revisión	12
2.2. Trabajos con técnicas de aprendizaje automático y procesamiento digital de imágenes	14
2.3. Trabajos con técnicas de aprendizaje profundo	18
2.3.1. Desarrollo de arquitecturas CNN	30
2.4. Trabajos con implementaciones de robótica	34
2.5. Otros	35
3. Materiales y métodos	38
3.1. Definiciones	38
3.2. Espacios de color	39
3.2.1. Espacio de color RGB	39
3.2.2. Espacio de color HSV	40
3.2.3. Espacio de color YCbCr	40
3.2.4. Espacio de color L*a*b*	41
3.3. Técnicas de segmentación	41
3.3.1. Otsu	41
3.3.2. PCA	42
3.4. Técnicas de extracción de características	42
3.4.1. Características de textura	43
3.4.2. Características cromáticas	44
3.5. Técnicas de equilibrio de clases	47
3.5.1. Submuestreo aleatorio	47
3.5.2. Sobremuestreo con réplicas	48
3.5.3. Sobremuestreo con el algoritmo SMOTE	49
3.6. Técnicas de clasificación	49

3.6.1. Máquinas de vectores soporte (SVM)	50
3.6.2. Redes neuronales artificiales (RNA)	50
3.6.3. Naive bayes (NB)	51
3.6.4. K-vecinos más cercanos (KNN)	52
3.6.5. Bosques aleatorios (BA)	52
3.6.6. Regresión logística (RL)	53
3.7. Métricas de desempeño	53
4. Metodología	55
4.1. Conjunto de datos	55
4.2. Módulos del método propuesto inicial	57
4.2.1. Segmentación	58
4.2.2. Extracción de características	59
4.2.3. Clasificación	62
4.3. Módulo de preprocesamiento	63
4.4. Módulo de equilibrio de clases	65
4.5. Influencia de técnicas de extracción de características	69
5. Resultados experimentales	71
5.1. Resultados del método propuesto inicial	71
5.1.1. Resultados con la técnica de segmentación Otsu	72
5.1.2. Resultados con la técnica de segmentación PCA	72
5.2. Resultados con el módulo de preprocesamiento	74
5.3. Resultados con el módulo de equilibrio de clases	78
5.4. Resultados de la influencia de técnicas de extracción de características	80
6. Discusiones	83
6.1. Métodos de segmentación	83
6.2. Técnicas de extracción de características	84
6.3. Técnicas de equilibrio de clases	85
6.4. Algoritmos de aprendizaje automático	86
7. Conclusiones y trabajo futuro	87
7.1. Conclusiones	87
7.2. Trabajo futuro	88
8. Artículos publicados y en revisión	89
8.1. Artículo publicado	89
8.2. Artículo en revisión	90

Índice de figuras

3.1. Línea del tiempo de la inteligencia artificial, Figura tomada de [1].	39
3.2. PCA, algoritmo para reducción de dimensiones.	42
3.3. Submuestreo aleatorio.	48
3.4. Sobremuestreo con réplicas.	48
3.5. Sobremuestreo con el algoritmo SMOTE.	49
4.1. Conjunto de datos Plantvillage.	57
4.2. Método propuesto inicial.	57
4.3. PCA y Otsu: Dos métodos usados en la segmentación de imágenes.	58
4.4. Extracción de características texturales.	60
4.5. Extracción de características cromáticas.	61
4.6. Extracción de características híbridas.	62
4.7. Clasificación del método propuesto inicial.	62
4.8. Método propuesto con preprocesamiento.	63
4.9. Preprocesamiento de un espacio de color RGB a otros.	64
4.10. Segmentación con preprocesamiento.	65
4.11. Método propuesto con equilibrio de clases	66
4.12. Submuestreo aleatorio.	67
4.13. Sobremuestreo con réplicas y sobremuestreo con el algoritmo SMOTE.	69
4.14. Extracción de características de las técnicas usadas	70
5.1. Gráfica de resultados con dos métodos de segmentación.	74
5.2. Desempeño de exactitud de técnicas de extracción de características.	75
5.3. Gráfica de diagramas de caja con métodos de aprendizaje automático.	76
5.4. Desempeño de exactitud de técnicas de extracción de características por clasificador.	77
5.5. Exactitud de técnicas de equilibrio de clases.	79
5.6. Desempeño de precisión por método de aprendizaje automático.	80
5.7. Desempeño de técnicas de extracción de características.	81
5.8. Desempeño de exactitud de algoritmos de aprendizaje automático.	82
6.1. Gráfica de resultados sin preprocesamiento y con preprocesamiento previo.	84
6.2. Exactitud de técnicas de equilibrio de clases.	86
8.1. Capítulo de libro publicado en la editorial Springer.	89
8.2. Artículo para revista JCR (Computación y Sistemas).	90
8.3. Artículo para la revista IJCOPI.	91

Índice de tablas

2.1. Desempeño de trabajos relacionados con técnicas de aprendizaje automático y procesamiento digital de imágenes.	18
2.2. Desempeño de trabajos relacionados con técnicas de aprendizaje profundo.	36
2.3. Desarrollo de arquitecturas CNN.	37
2.4. Trabajos con implementaciones de robótica.	37
4.1. Distribución de imágenes del conjunto de datos.	56
4.2. Distribución de imágenes del conjunto de datos submuestreado.	67
4.3. Partición del conjunto de datos.	68
5.1. Resultados de exactitud con la técnica de segmentación Otsu en la etapa 1.	72
5.2. Resultados de precisión con la técnica de segmentación Otsu aplicada en la etapa 1.	72
5.3. Resultados de exactitud con la técnica de segmentación PCA en la etapa 1.	73
5.4. Resultados de precisión con la técnica de segmentación PCA en la etapa 1.	73
5.5. Resultados de exactitud con el módulo de preprocesamiento.	75
5.6. Resultados de precisión con el módulo de preprocesamiento.	76
5.7. Matriz de confusión del algoritmo SVM.	78
5.8. Desempeño de algoritmos de clasificación frente a técnicas de equilibrio de clases.	78
5.9. Desempeño de precisión de los clasificadores.	79
5.10. Exactitud de algoritmos de aprendizaje automático frente a técnicas de extracción de características.	81
5.11. Resultados de exactitud de algoritmos de aprendizaje automático frente al vector de características concatenado.	82

Resumen

Actualmente, el área agrícola se ha diversificado en la producción de diferentes cultivos, logrando incrementar la producción por planta y calidad del fruto, realizando esta actividad a cielo abierto como en ambientes protegidos (invernaderos). Bajo esta premisa, la distribución de cultivos se ha extendido hacia el extranjero, obteniendo ingresos financieros considerables; sin embargo, una de las principales preocupaciones de los productores son los descensos económicos por la llegada no deseada de enfermedades y plagas que afectan a los cultivos. Por ende, en esta investigación, se ha implementado un sistema para el reconocimiento de daños foliares causados por enfermedades y plagas en plantas de jitomate, que funja como herramienta de apoyo para la correcta identificación con base en técnicas de visión artificial. Para resolver esta problemática, se ha desarrollado una metodología con una estructura modular, representada por las siguientes etapas: preprocesamiento, segmentación, extracción de características, equilibrio de clases, y clasificación. Así mismo, se ha usado el conjunto de datos Plantvillage, el cual contiene diez clases distintas, representadas por ocho enfermedades, una plaga, y una clase completamente sana. Derivado de las fases experimentales desarrolladas a lo largo de esta investigación, el sistema propuesto, alcanza un rendimiento del 94.65 % de exactitud.

Palabras clave: procesamiento digital de imágenes, reconocimiento de patrones, visión artificial, daños foliares, enfermedades y plagas del jitomate.

Abstract

Currently, the agricultural area has diversified into the production of different crops, managing to increase production per plant and quality of the fruit, carrying out this activity in the open air as well as in protected environments (greenhouses). Under this premise, the distribution of crops has spread abroad, obtaining considerable financial income; however, one of the main concerns of producers is the economic decline due to the unwanted arrival of diseases and pests that affect crops. Therefore, in this research, a system has been implemented for the recognition of foliar damage caused by diseases and pests in tomato plants, which serves as a support tool for the correct identification based on artificial vision techniques. To solve this problem, a methodology with a modular structure has been developed, represented by the following stages: preprocessing, segmentation, feature extraction, balancing of classes, and classification. Likewise, the Plantvillage dataset has been used, which contains ten different classes, represented by eight diseases, one pest, and one completely healthy class. Derived from the experimental phases developed throughout this research, the proposed system reaches a performance of 94.65% accuracy.

Keywords: digital image processing, pattern recognition, artificial vision, foliar damage, tomato pests and diseases.

Capítulo 1

Introducción

Actualmente, México juega un papel importante en la exportación de una gran diversificación de cultivos bajo agricultura a cielo abierto, con un porcentaje de producción del 97.7%, y agricultura protegida (invernaderos) del 2.3% [2]; asimismo, de la producción total de jitomate, 60.8% se obtienen de ambientes protegidos; Chiapas es el principal productor de café en ambientes protegidos, Guanajuato de brócoli, Cd. de México de nochebuena y Sinaloa de jitomate; por ende, en el país se han incrementado un gran número de invernaderos; logrando un aumento en la producción por planta y calidad del fruto; se ha llegado a estos resultados con la implementación de nuevos métodos automatizados para el cuidado de invernaderos, controlando temperatura, humedad e iluminación; impactando en el cuidado de la siembra, nutrición, crecimiento y cosecha de la misma. Además, en México, la superficie sembrada de jitomate es de 42 383.3 hectáreas, con una producción anual de 2 860 305.19 toneladas [3].

En esta investigación, se ha revisado el estado del arte, en áreas como: procesamiento digital de imágenes, robótica, aprendizaje automático, aprendizaje profundo, entre otras; identificando enfermedades en plantas, detección de estrés en plantas, medición del nivel de madurez de frutos, reconocimiento de plantas a través de la hoja, usando diferentes metodologías, como: redes neuronales artificiales (RNA) tradicionales, redes neuronales convolucionales (CNN), algoritmo gradiente descendiente, K-vecinos más cercanos (KNN), algoritmos probabilísticos, K-means, máquinas de vectores soporte (SVM), etc. Obteniendo un punto de referencia de trabajos realizados, analizando el aporte de cada uno de ellos.

Además, se ha implementado una metodología computacional basada en herramientas y

técnicas de visión artificial, logrando reconocer nueve daños foliares, causados por enfermedades y plagas que afectan a hojas de plantas de jitomate y una clase completamente sana, las cuales son mencionadas a continuación: virus del mosaico del tomate, moho de la hoja, tizón temprano, mancha foliar, araña roja de dos puntos, mancha foliar por septoria, tizón tardío, mancha bacteriana, virus del enrollamiento de la hoja amarilla del tomate y hojas sanas, siendo esta una herramienta de apoyo. Bajo la identificación precisa en este sistema implementado, se lograrán disminuir pérdidas financieras y la aplicación excesiva de productos químicos en la producción de jitomate.

A continuación se da una breve descripción de cada uno de los capítulos contenidos en este documento. En el Capítulo 1, se describe la problemática, justificación, objetivos, hipótesis, y pregunta de investigación; en el Capítulo 2, se plasma el fundamento teórico del método implementado; en el Capítulo 3, se hace una revisión profunda de trabajos relacionados con esta investigación; en el Capítulo 4, se describe la metodología propuesta, con base en un sistema modular secuencial; en el Capítulo 5, se despliegan los resultados experimentales obtenidos; en el Capítulo 6, se discuten los resultados obtenidos; en el Capítulo 7, se exponen las conclusiones y trabajo futuro; y finalmente, en el Capítulo 8, se muestran las publicaciones y trabajos sometidos que se encuentran en revisión.

1.1. Planteamiento del problema

En la actualidad, la identificación precisa de enfermedades en hojas de plantas de jitomate, se ha convertido en un reto global. Expertos en el área, regularmente realizan la identificación de forma manual u ocular, emitiendo un diagnóstico no preciso o erróneo, sugiriendo aplicar algunos productos químicos como fungicidas o pesticidas, perjudicando a las plantas, seres vivos y al medio ambiente.

Con el paso del tiempo, la producción de diferentes frutos ha crecido de manera exponencial, teniendo ingresos financieros considerables en algunos estados del país; sin embargo, existen algunos riesgos en el proceso de crecimiento de los cultivos. Productores han reportado descensos económicos, debido a enfermedades que han atacado a plantas de jitomate en regiones del estado

de Puebla, Hidalgo, Guanajuato, Sinaloa, entre otros; o inclusive, plantíos contaminados en su totalidad, reflejando pérdidas en los balances financieros.

Algunas de las enfermedades más comunes en plantas de jitomate son: pudrición radicular, cáncer bacteriano del tomate, peca y mancha bacteriana, moho de las hojas, moho gris, tizón temprano, tizón tardío, y cenicilla polvorienta [4], presentadas por variaciones de humedad, sequía, temperatura, residuos de cosechas anteriores, viento, insectos, nublados, y negligencia de los operadores de plantíos, diagnosticando la enfermedad a través del tallo, hoja, o fruto. Posteriormente a la identificación de alguna anomalía en la planta, el productor acude a expertos para diagnosticar el daño foliar, considerada una identificación tardía y con cierto grado de avance; así mismo, se le aplica la dosis recomendada para controlar y/o eliminar dicha enfermedad, generando gastos adicionales; en el peor de los escenarios, las plantas enfermas son desechadas corriendo el riesgo que los cultivos vecinos sean contaminados; por ende, las plantas son extraídas completamente para evitar la propagación del patógeno. Una de las principales causas de pérdida en la producción de jitomate es la identificación no precisa de daños foliares; por esta razón, se pretende desarrollar un sistema que identifique, virus del mosaico del tomate, moho de la hoja, tizón temprano, mancha foliar, araña roja de dos puntos, mancha foliar por septoria, tizón tardío, mancha bacteriana, virus del enrollamiento de la hoja amarilla del tomate, y hojas sanas, en hojas de plantas de jitomate.

1.2. Justificación

La identificación precisa de daños foliares causados por plagas y enfermedades en plantas de jitomate, ayudará a disminuir pérdidas financieras y el uso excesivo o erróneo de productos químicos, reduciendo la afección a plantas, seres vivos, y al medio ambiente.

1.3. Objetivo general

Implementar algoritmos de visión artificial para la identificación de daños foliares causados por plagas y enfermedades en plantas de jitomate.

1.4. Objetivos específicos

1. Definir un conjunto de datos.
2. Preprocesar imágenes de entrada.
3. Implementar algoritmos para segmentar imágenes.
4. Implementar algoritmos para extraer características.
5. Implementar métodos para el equilibrio de clases.
6. Experimentar con algoritmos de aprendizaje automático.

1.5. Hipótesis

Bajo la aplicación de algoritmos de visión artificial, se lograrán identificar daños foliares causados por plagas y enfermedades en plantas de jitomate de forma precisa; obteniendo un sistema que sirva como herramienta de apoyo para toma de decisiones.

1.6. Pregunta de investigación

¿ Pueden algoritmos de visión artificial, identificar daños foliares causados por plagas y enfermedades en plantas de jitomate de forma precisa ?

Capítulo 2

Estado del arte

En este capítulo, se plasman los trabajos relacionados del estado del arte con esta investigación. Destacando investigaciones enfocadas en el reconocimiento de plantas a través de la hoja, detección de daños foliares, identificación de plagas y enfermedades en diferentes plantas, detección de estrés en plantas, entre otras; resolviendo estas problemáticas con técnicas y métodos de procesamiento digital de imágenes, inteligencia artificial, algoritmos de aprendizaje automático, y aprendizaje profundo. Además, haciendo una comparación en forma de tabla de cada trabajo revisado, considerando: objetivo del trabajo, conjunto de datos usado, método aplicado, y mejor desempeño obtenido. Los apartados son encabezados por las siguientes secciones: trabajos de revisión, trabajos con técnicas de aprendizaje automático y procesamiento digital de imágenes, trabajos con técnicas de aprendizaje profundo, desarrollo de arquitecturas CNN, y trabajos con implementaciones de robótica.

2.1. Trabajos de revisión

Anil A. Bharate y M. S. Shirdhonkar en su trabajo, y derivado del decremento financiero, causado por la afectación de enfermedades en diferentes plantas, impactando a la producción agrícola; los agricultores, requieren expertos para el monitoreo en tiempo real de los cultivos, evitando pérdidas monetarias; por ende, en esta investigación los autores han realizado una revisión de métodos desarrollados por distintos autores para la detección de enfermedades en plantas, usando procesamiento digital de imágenes; en el primer apartado se describen algunas

enfermedades y la forma en la que se propagan; en manzana, los autores describen enfermedades como *Apple Scab*, *Apple Rot*, *Marssonina Leaf Blotch*, *Black Rot Canker* y *Apple Mosaic*; en uva, *Black Rot*, *Powdery Mildew*, *Downy Mildew*, *Anthracnose*, *Bacterial Leaf Spot* y *Rust*; y en granada *Bacterial Blight*, *Aspergillus Fruit Rot*, *Anthracnose*, *Cercospora Fruit Spot* y *Cercospora leaf Spot*, diagnosticadas a través de la hoja y en algunos casos en el fruto; posteriormente se describen trabajos realizados en el ámbito de detección y clasificación de enfermedades en plantas y el nivel de madurez de frutos, considerando, manzana, pimiento, granada, jitomate, fresa, hoja de algodón, entre otros; resolviendo la propuesta con SVM, RNA, redes neuronales convolucionales (CNN), algoritmos de agrupación *K-Means* y lógica difusa en conjunción con transformadas de Laplace (LPT); además, se visualiza y explica una arquitectura modular de un sistema para la detección de enfermedades en plantas usando procesamiento digital de imágenes. Los autores de este trabajo concluyen que la detección temprana de enfermedades ayuda a los agricultores a mejorar el rendimiento de los cultivos, impactando en el producto interno bruto (PBI) de la India; asimismo, color, textura y características morfológicas en las plantas, son variables que permiten el diagnóstico acertado de enfermedades [5].

Gittaly Dhingra et al. en su trabajo, la principal problemática, como en otros casos es la disminución en las ganancias por producción, impactando directamente en la pérdida financiera; esto, derivado de las enfermedades que atacan a los cultivos tanto en invernaderos como a cielo abierto; por ende, en esta investigación se desarrolló un estudio acerca de la detección y clasificación de enfermedades en plantas de uva, café, soya, maíz, tabaco, sandía, arroz, algodón, trigo, pepino, caña de azúcar, girasol, calabaza, chile, olivo, arboles de sombra, orquídea, té, remolacha, jitomate, entre otras; resolviéndolos con diferentes métodos y técnicas de procesamiento digital de imágenes y visión artificial, a través, de parámetros de agrupamiento, RNA, algoritmo NB, lógica difusa, PSO, OPDPA, dimensiones fractales, SVM, análisis discriminante, AdaBoost, entre otros; concluyendo que los trabajos analizados deben implementar algoritmos más novedosos y herramientas para lograr mejores resultados, considerando parámetros de precisión y calidad, muy necesarios para la industria altamente competitiva y cambiante [6].

Kaiyan Lin et al. en su investigación, la problemática perseguida, se deriva de la pérdida de producción agrícola crónica, impactando y minimizando las ganancias financieras; esto, resul-

tado del estrés que se presenta en plantas en invernaderos, disminuyendo la cantidad y calidad del fruto; en este artículo se revisaron avances recientes de técnicas de visión artificial para la detección de estrés hídrico, estrés nutricional, y estrés por enfermedad y plagas en plantas en invernaderos, considerando plantas de jitomate, manzana, lechuga, maíz, algodón y pepino; y la plaga mosca blanca. Para la solución de los trabajos citados, se implementaron técnicas de visión computacional, como: redes neuronales artificiales, algoritmos de agrupamiento, algoritmos bioinspirados, clasificación estadística, entre otras. Los autores concluyen que la segmentación en la actualidad sigue siendo un reto cuando se trabaja con imágenes con luz no controlada; además, los algoritmos desarrollados son para una planta específica y toma desventaja al considerar usarlos para otros cultivos distintos; asimismo, color, textura y morfología son características de suma importancia para detectar y clasificar estrés y, enfermedades en plantas [7].

2.2. Trabajos con técnicas de aprendizaje automático y procesamiento digital de imágenes

Jair Cervantes et al. en su trabajo, ha identificado plantas a través de la hoja, usando el conjunto de datos ICL considerando diferentes familias, la metodología aplicada para solucionar el reconocimiento de hojas se basa en un sistema de reconocimiento de patrones; analizando la influencia de la combinación de diferentes técnicas de extracción de características en imágenes RGB, considerando características texturales (Haralick y LBP), cromáticas (características de color elementales y características de color usando momentos de Hu) y geométricas (características geométricas elementales, momentos de Hu, momentos de flusser, momentos R, y descriptores de Fourier), evaluándolas con algoritmos de aprendizaje automático (SVM, redes neuronales, naive bayes, y regresión logística) [8].

Jair Cervantes et al. en su investigación, ha desarrollado un sistema para el reconocimiento de plantas a partir de la hoja, basado en métodos de extracción de características, extrayendo patrones de textura (LBP y Haralick), color (características de color elementales y características de color usando momentos invariantes de Hu) y forma (características geométricas elementales, características invariantes de Hu, momentos invariantes de flusser, momentos R, y descriptores

de Fourier), posteriormente, se han implementado tres técnicas de selección de características (selección de características basada en la correlación, selección de características basada en la ganancia de información y un algoritmo genético), obteniendo una reducción en las dimensiones de la información, finalmente, utilizaron algoritmos de aprendizaje automático para la clasificación [9].

Ismail El Massi et al. en su trabajo, ha aplicado métodos de extracción de características y reconocimiento de patrones, bajo una propuesta combinatoria de clasificadores con estructura serie y paralelo basado en el algoritmo SVM, con la finalidad de detectar y reconocer daños causados por enfermedades en las hojas de las plantas. El mejor modelo implementado, logra un rendimiento del 91.11 % [10].

Jagadeesh Basavaiah et al. en su investigación, logra identificar cuatro enfermedades y una clase de hojas sanas de plantas de jitomate, bajo la implementación de múltiples métodos de extracción de características, considerando histogramas de color, momentos Hu, Haralick y LBP, clasificando con bosques aleatorios y árboles de decisión, donde, el mejor rendimiento se obtiene con bosques aleatorios con un 94.00 % [11].

Gayatri Pattnaik y K. Parvathi en su trabajo, detectan y clasifican plagas en plantas de jitomate, el conjunto de datos fue extraído de un repositorio de Internet, el cual contiene diez clases pertenecientes a diez plagas, las técnicas implementadas, se basan en un sistema de reconocimiento de patrones, como extractores de características se usó el algoritmo HOG y LBP, para la clasificación se utilizó el algoritmo SVM, el desempeño del sistema propuesto obtuvo un rendimiento del 97.00 % [12].

Benjamín Luna-Benoso et al. en su investigación, logra detectar y reconocer tres enfermedades a través de lesiones foliares previamente segmentadas, así mismo, las técnicas de extracción de características usadas fueron los momentos de color y la matriz de co-ocurrencia de nivel de gris (GLCM), posteriormente, para una predicción más robusta, se implementaron tres algoritmos en una arquitectura paralela, considerando SVM, KNN y perceptrón multicapa (MLP), basados en una regla de decisión final de voto mayoritario, el rendimiento del modelo se logró con un 97.39 % respectivamente [13].

Peter J. Burt y Edward H. Adelson en su trabajo, describió una técnica para la codificación de imágenes, la imagen filtrada es representada con una densidad de muestra reducida, además, el proceso de codificación, es equivalente a muestrear la imagen como operadores laplacianos de muchas escalas, por lo tanto, el código tiende a mejorar características destacadas de la misma, así como la compresión y otros factores; derivado de este método para procesamiento de imágenes, los autores en esta investigación hacen la descripción puntual del algoritmo para codificar y decodificar una imagen en distintas escalas [14].

Shitala Prasad et al. en su investigación, abordan el problema de la segmentación de una imagen, siendo un desafío principal en la separación del fondo de una imagen y la extracción de la región de interés, tomando en cuenta la variación de color, textura y forma; en el tema de segmentación de imágenes se han realizado trabajos que han aportan resultados significativos hasta el día de hoy; por ende, los autores de esta investigación desarrollaron un sistema para la detección y reconocimiento de enfermedades en imágenes de hoja de plantas en ambiente natural, usando segmentación no supervisada basado en bloques, incluyendo enfermedades como Tikka en el maní, *Powdery Mildew* y *Downy Mildew* atacando a diferentes especies; plantas de manzana y mango enfermas de *Erysiphe Necator* y *Pseudoperonospora Humul*; tizón tardío y tizón temprano en las plantas de jitomate y papa, roya en plantas de maíz y manzana; asimismo, se usó el software Matlab 7.7 (R2008b), para ejecutar una metodología propia, donde, en primera instancia se digitalizó una imagen de entrada, posteriormente se realizó una conversión a color HSI, después se dividió la imagen en 25 bloques, después, se aplica la segmentación no supervisada sobre los bloques, y finalmente se encuentra la región enferma integrando la imagen original. En este trabajo se concluyó que los resultados de este método son mejores que los métodos ya existentes, como la segmentación basada en histogramas y la segmentación basada en bordes, especialmente para el conjunto de datos de hojas enfermas [15].

Youwen Tian et al. en su investigación, la principal problemática que plasman los autores en este trabajo, es la disminución masiva de la producción de cultivos; por ende, se ha desarrollado un sistema en el cual se adoptó una tecnología avanzada de procesamiento digital de imágenes para evaluar con precisión el grado de daño de las enfermedades en los cultivos, enfermedades como: *Downy Mildew*, *Powdery Mildew* y *Scab* en pepino; *Big Spot*, *Leaf Blight* y *Gray Leaf*

Spot en maíz; y *Downy Mildew*, *Powdery Mildew* y *Anthracnose* en uva; el método adoptado se implementó apoyado de un sistema modular en tres etapas, procesamiento de imágenes, segmentación y clasificación, midiendo el grado de daño; se usó MATLAB 7.01; concluyendo que, la precisión en la clasificación mejoró en gran medida, clasificando para pepino 50.1 %, maíz 70.1 %, y uva 50.1 %, utilizando tecnología de procesamiento digital de imágenes, lo que redujo el tiempo y los costos de evaluación, proporcionando datos precisos para el estudio de otros aspectos del cultivo [16].

Shitala Prasad et al. en su trabajo, manifiestan que la India, es un país agrícola, y la mayoría de la población depende de ella; por ende, para aumentar la tasa de producción se requiere mejoras tecnológicas; por ende, en esta investigación se ha implementado un sistema biométrico automatizado para la detección temprana de enfermedades en hojas de plantas, incluyendo enfermedades como *Tikka* en el maní; *Powdery Milde* y *Downy Mildew* en plantas de manzana y mango; *Late Blight* y *Early Blight* en plantas de papa y jitomate; y *Rust* en manzana; usando una metodología propia en tres pasos, detección de la lesión, extracción de características aplicando GWT (*Gabor Wavelet Transform*) y clasificación de vectores de características usando SVM, procesando la información con Matlab 7.7 (2008b) logrando un porcentaje de precisión del 89 %; los autores concluyen que el sistema biométrico implementado es robusto e independiente, logrando la identificación de enfermedades en las hojas de plantas [17].

Diogo Nunes et al. en su trabajo, la principal problemática que plasman los autores en esta investigación, es el uso de productos químicos en los cultivos de jitomate, trayendo serios problemas a la salud humana y al medio ambiente; por ende, se ha implementado un sistema para la detección de enfermedad en follaje en imágenes de plantas de jitomate, promoviendo que los agricultores utilicen estos sistemas para reducir la gestión de los productos químicos en los cultivos. En este artículo, se logró detectar de forma temprana la enfermedad tizón tardío en jitomate, implementando herramientas móviles basadas en Android, plataforma web, técnicas de filtrado de imágenes y reconocimiento de patrones basado en redes neuronales perceptrón multicapa, logrando un porcentaje de precisión del 94.12 %. Los autores garantizan la inocuidad de los alimentos, la protección del consumidor y oportunidades a los pequeños agricultores para llevar a cabo sus operaciones de manera más productiva; asimismo, se diseñaron herramientas

de software para utilizar aparatos de bajo costo y garantizar la facilidad de uso [18].

David G. Lowe en su investigación, presenta un método para extraer características invariantes de las imágenes, utilizándose para realizar una correspondencia confiable entre diferentes vistas de un objeto o escena; asimismo, las características son invariables para la escala y rotación de la imagen, además de incluir técnicas de reconocimiento de objetos [19].

Daniel Ayala Niño et al. en su trabajo, ha realizado una comparación de un sistema de reconocimiento de patrones para la clasificación de hojas complejas, las imágenes contienen hojas de la misma familia, pero de diferente especie, la metodología aplicada se divide en las siguientes etapas: segmentación, extracción y selección de características, y clasificación, frente a técnicas de aprendizaje profundo, específicamente dos arquitecturas de redes CNN propuestas por los autores. El conjunto de datos utilizado en esta investigación, fue proporcionado por el departamento de frutas del Colegio de Postgraduados [20].

Tabla 2.1: Desempeño de trabajos relacionados con técnicas de aprendizaje automático y procesamiento digital de imágenes.

Referencia	Objetivo del trabajo	Conjunto de datos	Método aplicado	Mejor Desempeño
[8]	Reconocimiento de plantas	ICL	Haralick, LBP, Hu, Flusser, momentos R, Fourier, momentos R, SVM, RNA, NB, y RL	100 %
[9]	Identificación de plantas	ICL	Haralick, LBP, Hu, Flusser, momentos R, Fourier, momentos R, algoritmo genético, SVM, RNA, NB, y RL	100 %
[10]	Reconocimiento de enfermedades en plantas	6 clases	Haralick y SVM	91.11 %
[11]	Identificación de enfermedades en plantas	5 clases	Histogramas de color, Hu, Haralick, LBP, BA y árboles de decisión	94.00 %
[12]	Detección y clasificación de plagas en plantas	10 clases	HOG, LBP, y SVM	97.00 %
[13]	Reconocimiento de lesiones foliares	3 clases	Momentos de color, GLCM, SVM, KNN, y MLP	97.39 %
[15]	Detección de enfermedades en plantas en ambiente natural	8 clases	HSI y métodos de procesamiento digital de imágenes para segmentación autónoma	N/A
[16]	Evaluación precisa del grado del daño de las enfermedades en los cultivos	9 clases	Tecnología de procesamiento digital de imágenes	70.1 %
[17]	Reconocimiento temprano de enfermedades en plantas	6 clases	GWT y SVM	89.00 %
[18]	Detección de enfermedad en follaje	2 clases	Filtrado de imágenes y MLP	94.12 %
[20]	Clasificación de hojas complejas	Departamento de frutas del Colegio de Postgraduados (8 clases)	Hu, Fourier, Flusser, momentos R, LBP, Haralick, Gabor, NB, SVM, LR, árboles de decisión, y CNN	95.29 %

2.3. Trabajos con técnicas de aprendizaje profundo

Mads Dyrmann et al. en su trabajo, el problema que se aborda, se centra en la maleza, presentándose en los campos agrícolas, lo cual se considera una amenaza, y es de suma importancia

el manejo y tratamiento de la misma; por consiguiente, en esta investigación se ha implementado un sistema de reconocimiento de especies de plantas (maleza), a través, del desarrollo de una de red neuronal convolucional profunda CNN; se construyó la red desde cero, entrenada y probada con 10,413 imágenes de seis dataset distintos para 22 especies, con imágenes capturadas desde ambientes cambiantes, considerando cultivo en sus etapas iniciales de crecimiento; los resultados experimentales, arrojan una precisión del 86.2% para el método propuesto [21].

Halimatu Sadiyah Abdullahi et al. en su investigación, el problema se centra en el aumento exponencial de la población, misma, que exige un rápido crecimiento de los alimentos con la llegada de la tecnología, reduciendo el trabajo laborioso, maximizando la producción y reduciendo el impacto de la presión sobre el medio ambiente; por ende, se ha propuesto un modelo que reconoce y clasifica imágenes para optimizar la producción en plantación de maíz, usando algoritmos de inteligencia artificial; el sistema se prueba para tres clases distintas (plantas saludables, plantas parcialmente saludables e insalubres), obteniendo 1,918 imágenes de una granja, se uso la *CNN VGG16* como extractor de características y el algoritmo SVM como clasificador; los resultados arrojaron una tasa de precisión del 99.58% [22].

Xuanxin Liu et al. en su trabajo, la principal problemática es la identificación de plantas con métodos tradicionales, siendo un reconocimiento deficiente, debido a la iluminación desigual y ciclos de crecimiento; por consiguiente, se ha propuesto un modelo de redes neuronales recurrentes convolucionales (C-RNN) para la identificación de plantas, centrada en la observación para imitar los comportamientos humanos; la C-RNN se compone de dos partes, la CNN que se usa como extractor de características, y la RNN usada para sintetizar características de cada imagen para la predicción final, para las pruebas se usó el dataset Flavia y BJFU100; logrando una tasa de clasificación del 99.65% para la C-RNN [23].

Mostafa Mehdipour Ghazi et al. en su investigación, la principal problemática es la evaluación de diferentes arquitecturas pre-entrenadas CNN, midiendo el rendimiento de estas; por ende, se ha aplicado aprendizaje por transferencia, usando CNN, para la identificación de especies de plantas; los modelos implementados y probados en este trabajo, son GoogLeNet, AlexNet y VGGNet, mismos, que medimos su desempeño con el dataset LifeCLEF-2015, el cual contiene 91,758 imágenes para 1,000 clases distintas, entre árboles, hierbas y helechos; después de la

experimentación realizada, se combinaron dos de los mejores modelos, GoogLeNet y VGGNet usando la técnica de fusión, logrando una precisión general del 80.18 % [24].

Sue Han Lee et al. en su investigación, se centra en la detección de nuevas y extrañas especies de plantas, con el fin de mejorar la industria farmacéutica, equilibrar el ecosistema, así como la productividad agrícola y la sostenibilidad; por ende, se ha desarrollado un modelo de identificación de plantas, usando CNN; la CNN propuesta se basa en la red pre-entrenada AlexNet, probada con el dataset ImageNet de la competencia ILSVRC-2012, identificando 44 especies distintas, además, se ha implementado el algoritmo backpropagation para un aprendizaje y optimización de filtros; a partir de los resultados experimentales, se logra una precisión del 99.5 % superando a la soluciones convencionales [25].

Danzi Wu et al. en su trabajo, propuso una función de pérdida para la identificación de imágenes de plantas de grano fino, que podría relacionar jerarquías del árbol taxonómico en la función objetivo de aprendizaje profundo; se entrenaron ocho redes convolucionales (GoogLeNet, ResNet-50, Inception-v3, Inception-ResNet-v2, MobileNet-v2, ShuffleNet-v2, DenSeNet-169 y SENet-154) por dos funciones de pérdida, asimismo, se experimentó con el dataset PlantCLEF-2015 y PlantCLEF-2017; posterior a las pruebas realizadas, se obtuvo una precisión del 84.07 % para PlantCLEF-2017 [26].

Jonas Krause et al. en su trabajo, identifican plantas, partiendo de una imagen natural o de ambiente no controlado, hasta el momento ha sido el principal reto y un problema desafiante; así mismo, los autores de esta investigación, han propuesto el sistema WTPlant para la identificación de plantas en imágenes de ambientes naturales, con un enfoque de aprendizaje profundo; la CNN se ha entrenado para clasificar 100 especies de plantas; la primera etapa se implementó para el pre-procesamiento para el análisis de múltiples escalas, asimismo, se desarrolló una CNN para extraer las características más discriminatorias; los experimentos iniciales mostraron que el sistema fue capaz de detectar con una precisión del 99.3 % [27].

Chao Chen et al. en su investigación, y con la llegada de dispositivos móviles inteligentes, en la actualidad, es posible capturar imágenes de flores de forma no profesional, lo cual es un reto, ya que al momento de procesar dichas imágenes, se tienen con un ligero desenfoque, no estando

en condiciones para realizar dicha tarea; por consiguiente, en este trabajo, se ha realizado un método de clasificación automática de dos pasos para flores silvestres, basado en redes CNN; en primera instancia, procesamos las imágenes de flores clasificándolas como imágenes borrosas o claras, posterior a ello, extraemos características usando la CNN, clasificando de acuerdo a las características aprendidas; la CNN propuesta contiene cuatro capas convolucionales, tres capas totalmente conectadas y una Softmax, clasificando para 17 especies, el entrenamiento fue ejecutado bajo la plataforma de hardware Intel i5-8400, 8Gb de memoria ram, GPU Nvidia GTX1050, junto con el framework TensorFlow; después del entrenamiento y la experimentación realizada, el modelo arroja una precisión del 90.20 % [28].

Srdjan Sladojevic et al. en su investigación, el problema de las enfermedades en las plantas, se relaciona estrechamente con los problemas de la agricultura sostenible y el cambio climático; por consiguiente, en este trabajo se ha desarrollado un modelo de reconocimiento de enfermedades en hojas de plantas, a través, del uso de redes CNN; la arquitectura propuesta, clasifica 13 enfermedades en plantas de pera, melocotón, manzana y uva, distinguiendo las hojas de su entorno, asimismo, se tomo como base la red CaffeNet que contiene ocho capas de aprendizaje, cinco capas convolucionales y tres completamente conectadas, además, se ha probado el modelo con un dataset que contiene 33,469 imágenes; el hardware usado esta compuesto de 16Gb de memoria ram, procesador intel core i7-4790, tarjeta gráfica GeForce GTX TITAN X 12 Gb con un OS Linux Ubuntu 14.04; posterior a la experimentación, se ha logrado una precisión promedio del 96.3 % [29].

Yang Lu et al. en su trabajo, la problemática como en otras investigaciones, son las enfermedades que afectan a las plantas, impactando en la economía y la salud alimentaría; por ende, en este trabajo proponen un nuevo método de identificación de enfermedades del arroz, basado en técnicas de redes CNN; la arquitectura de la CNN implementada fue basada en redes pre-entrenadas, se usó un dataset de 500 imágenes de hojas y tallos de arroz, sanos y enfermos, capturados de un campo experimental, se redimensionaron las imágenes a 224x224 píxeles; para las pruebas, se consideraron 10 tipos enfermedades distintas; derivado de la experimentación, el modelo logra una precisión del 95.48 % [30].

Xihai Zhang et al. en su investigación, la problemática principal, es derivada del número de

enfermedades del maíz y el daño que causan han aumentado; por ende, se proponen modelos mejorados de CNN para la identificación de enfermedades en hojas de plantas de maíz; en este trabajo se mejoraron dos modelos CNN, considerando GoogLeNet y Cifar10, probando su desempeño para nueve clases distintas (southern leaf blight, brown spot, Curvularia leaf spot, rust, dwarf mosaic, gray leaf spot, round spot, northern leaf blight y healthy leaf), disminuyendo el número de parámetros en comparación con las redes VGG y AlexNet; los resultados muestran un porcentaje de precisión para GoogLeNet del 98.9 % y 98.8 % para Cifar10 [31].

Ramar Ahila Priyadharshini et al. en su investigación, abordan el problema de la seguridad alimentaria, ya que en algunos casos se hace identificación de alguna enfermedad de forma visual, siendo una detección manual e inexacta, conduciendo al uso incorrecto de fungicidas y pesticidas, afectando a los seres humanos con enfermedades crónicas dañinas y a su vez reduciendo la calidad y cantidad de producción; por ende, se probó un modelo de aprendizaje profundo para la clasificación de enfermedades en hojas de plantas de maíz, CNN; la metodología propuesta esta basada en la arquitectura CNN LeNet modificada, formada por dos capas convolucionales y tres capas totalmente conectadas, asimismo, fue validado el desempeño con el dataset Plantvillage, considerando cuatro clases, una clase para hojas completamente sanas y tres clases para enfermedades (northern leaf blight, gray leaf spot y common rust), con un total de 4,365 imágenes para entrenamiento y validación, como parte del pre-procesamiento, se le aplicó a las imágenes de entrada el algoritmo PCA whitening, posterior a ello, la primera capa convolucional extrae entidades de bajo nivel como: bordes, líneas, curvas y esquinas, y la segunda capa aprende características globales; producto de la experimentación, se obtuvo una precisión del 97.89%; los autores concluyen, que derivado de este estudio, el tamaño de núcleo 3x3 es el más adecuado para la clasificación de esta tarea [32].

Jing Hu et al. en su trabajo, la problemática principal es la desafiante tarea de reconocimiento de imágenes de hojas de plantas, ya que tiene variaciones morfológicas, de textura, coloración y venación; por esta causa, en este trabajo se propuso una red neuronal convolucional de fusión de múltiples escalas (MSF-CNN), para el reconocimiento de hojas de plantas; las redes MSF-CNN realizan operaciones de concatenación de mapas de características aprendidas, sobre imágenes de distintas escalas desde una sola vista, el modelo propuesto tiene cuatro capas

convolucionales, probando el desempeño con dos dataset, MalayaKew MK con 46288 imágenes y LeafSnap con 7719 imágenes, realizando una evaluación cruzada entre ellas; después de la experimentación realizada, se obtuvo una precisión del 98.74 %; concluyendo que usando los dataset mencionados anteriormente, se demuestra que el método propuesto es superior a los métodos de reconocimiento de hojas de plantas de última generación [33].

Anandhakrishnan MG Joel Hanson et al. en su trabajo, la problemática como en otros trabajos, es la identificación manual y visual por un experto, lo cual se considera una detección tardía e inexacta; por ende, se ha desarrollado un modelo de reconocimiento de enfermedades en hojas de plantas, a través, del uso de redes CNN; en la parte metodológica, en primera instancia, se ha usado un dataset de un repositorio localizado en Internet, contemplando 30880 imágenes para entrenamiento y 2589 para validación, posterior a ello se han pre-procesado las imágenes que entran a la CNN, normalizando cada una de ellas; finalmente, se ha implementado una CNN, red neuronal artificial invariante de cambio o invariante de espacio SIANN, nombrada en función de su arquitectura de pesos compartidos y características de invariancia de traducción; después de la experimentación y el ajuste a la red, el porcentaje de precisión, arroja un 95 % del resultado final global [34].

Uday Pratap Singh et al. en su investigación, la principal problemática en este trabajo, está basada en la pérdida por producción, afectando los incrementos financieros causados por enfermedades en plantas de hojas de mango; por ende, se ha propuesto una MCNN (Multi-layer Convolutional Neural Network) para la clasificación de hojas de mango, infectadas por la enfermedad fúngica Anthracnose; la metodología propuesta, se validó con dos dataset, uno de creación propia en tiempo real con 1070 imágenes y en el segundo con 1130 imágenes del conjunto de datos PlantVillage, para cuatro clases distintas; las dimensiones de las imágenes fueron de 128x128, asimismo, se pre-procesaron con la técnica de ecualización de histogramas y central square crop; la CNN propuesta, esta basada en la arquitectura AlexNet, con seis capas convolutivas, la etapa de entrenamiento fue ejecutada en 100 épocas, usando el 80 % para el aprendizaje y 20 % para clasificación; después de la experimentación realizada, el modelo obtuvo una precisión del 97.13 %, superando a las técnicas con las que fue comparada [35].

Peng Jiang et al. en su trabajo, es preocupante que plagas y enfermedades afectan a los

cultivos en diferentes ámbitos, hoy en día, resultan gravemente afectadas hojas de plantas de manzana, el humano, tiene métodos de identificación de enfermedades rudimentarios, siendo técnicas poco confiables; por ende, en esta investigación se propone un modelo para la detección en tiempo real de enfermedades (*Alternaria leaf spot*, *Brown spot*, *Mosaic*, *Grey spot* y *Rust*) en hojas de plantas de manzana, usando redes CNN; la arquitectura propuesta INAR-SSD, fue basada en la CNN GoogLeNet (VGG-INCEP) y la concatenación de Rainbow, el modelo fue validado con el dataset ALDD con 26377 imágenes, conteniendo imágenes de laboratorio y en condiciones reales, asimismo, la CNN fue entrenada para cinco clases, el modelo SSD utiliza un módulo Inception para extraer características de varios tamaños y una concatenación Rainbow para mejorar las conexiones entre capas; resultado de la experimentación, el modelo INAR-SSD obtiene un 78.80 % de precisión [36].

Konstantinos P. Ferentinos en su investigación, la principal problemática es la identificación óptica de enfermedades en plantas de forma no precisa, ya que una detección visual por agrónomos experimentados o fitopatólogos no garantiza un diagnóstico acertado, en consecuencia, esto da lugar a conclusiones y tratamientos erróneo; por lo consiguiente, en este trabajo se han desarrollado modelos de redes neuronales convolucionales para la detección y diagnóstico de enfermedades en hojas de varias plantas; la metodología propuesta fue basada en aprendizaje profundo con imágenes de entrada de dimensiones 256×256 , los modelos (AlexNet, AlexNet-OWTBn, GoogLeNet, Overfeat y VGG) fueron entrenados con datasets de ambiente abierto, sumando un total de 87,848 imágenes con 25 plantas distintas para 58 clases; derivado de la experimentación, la tasa de precisión más alta, alcanza un porcentaje del 99.53 % para la CNN VGG [37].

Bin Liu et al. en su trabajo, manifiestan que, en la actualidad, los trabajos realizados, usan métodos de pre-procesamiento de imágenes complejos y no garantizan altas tasas de identificación para las enfermedades en plantas; por ende, en esta investigación, proponen un modelo de redes CNN profundas, para la identificación de enfermedades en hojas de plantas de manzana; así mismo, la CNN implementada fue basada en AlexNet, siendo probada con un dataset de 13,689 imágenes, considerando cuatro enfermedades (*Mosaic*, *Rust*, *Brown spot*, y *Alternaria leaf spot*); posterior a la experimentación, se ha logrado una precisión general del 97.62 % [38].

Alvaro Fuentes et al. en su investigación, se enfocan en la problemática basada en pérdidas económicas, generada por las enfermedades y plagas que afectan a las plantas de tomate; por ende, se han probado diferentes arquitecturas, basadas en aprendizaje profundo para la detección de enfermedades y plagas, considerando (leaf mold, gray mold, canker, plague, miner, low temperature, powdery mildew, whitefly y nutritional excess) en hojas de plantas de tomate, capturando las imágenes en un ambiente no controlado; las arquitecturas probadas fueron VGG y ResNet, estas, recaen en tres familiar distintas, Faster Region-based Convolutional Neural Network (Faster R-CNN), Region-based Fully Convolutional Network (R-FCN) y Single Shot Multibox Detector (SSD), mostrando su desempeño con 5,000 imágenes de un dataset de creación propia, obtenidas de granjas en la península de Corea, las pruebas se desarrollaron bajo un hardware con las siguientes características: Intel Core I7 3.5 GHz y dos GPUs Nvidia GeForce TitanX; después de la experimentación realizada, el sistema muestra una precisión del 83.06 % [39].

Melike Sardogan et al. en su investigación, aborda el problema de enfermedades en plantas, afectando el crecimiento y el rendimiento de los cultivos, impactando en el ámbito social, ecológico y económico en la agricultura; por ende, en este trabajo, se desarrolló un modelo CNN en conjunción con el algoritmo Learning Vector Quantization (LVQ) para la detección y clasificación de enfermedades (Bacterial spot, Late blight, Septoria spot y Yellow curved) en imágenes RGB de hojas de plantas de tomate; como aprendizaje de características se usó una arquitectura CNN, y el algoritmo LVQ para la clasificación, el modelo propuesto se entrenó y validó con valores obtenidos del conjunto de datos Plantvillage, usando 500 imágenes, las cuales, 400 fueron para el aprendizaje en la etapa de entrenamiento y 100 para validación; asimismo, el porcentaje de precisión para este modelo probado fue del 86 %; los autores concluyen que el método propuesto reconoce de manera efectiva cuatro tipos de enfermedades, asimismo, se podría mejorar el porcentaje de precisión, usando diferentes filtros o distintos tamaños de capas convolucionales [40].

Shradha Verma et al. en su investigación, y derivado de las enfermedades que afectan a las plantas de uva, ha repercutido en la pérdida de granos alimenticios en todo el mundo, poniendo en riesgo a la producción agrícola y el rendimiento de los cultivos; asimismo, los fungicidas y

otros productos químicos tienen un efecto adverso en el ecosistema; por consiguiente, en este trabajo se ha implementado y evaluado el rendimiento de las arquitecturas de redes neuronales convolucionales ResNet50 y VGG16; para el entrenamiento y la clasificación se usó el banco de datos Plantvillage con ponderaciones 70-30, utilizando 4062 imágenes, contemplando tres enfermedades (black rot, leaf blight y black measles) y una clase de plantas sanas, las imágenes procesadas fueron de dimensiones de 256×256 píxeles; así mismo, los resultados indican que la CNN ResNet50 se desempeñó de forma efectiva, obteniendo un porcentaje del 98.76 %, en comparación con VGG16 que obtuvo 95.48 % [41].

Endang Suryawati et al. en su trabajo, el principal problema, es la detección tardía de enfermedades, amenazando la seguridad alimentaria mundial y generando desastres a los pequeños agricultores que carecen de recursos financieros para contrarrestarlas y superarlas, teniendo pérdidas significativas en los cultivos. Con este trabajo, buscan evaluar el desempeño de tres arquitecturas y una propuesta base de redes neuronales convolucionales para la detección de enfermedades, como: tomato bacterial spot, tomato early blight, tomato late blight, tomato leaf Mold, tomato septoria leaf spot, tomato spider mites, tomato target spot, tomato mosaic virus y tomato yellow leaf curl virus en hojas de plantas de jitomate; se probó el desempeño de la arquitectura de aprendizaje profundo AlexNet con cinco capas convolucionales, VGGNet con trece capas convolucionales, GoogleNet con nueve capas convolucionales y una propuesta base con dos capas convolucionales; el entrenamiento y validación se realizó con banco de datos Plantvillage, usando 18,160 imágenes de diez clases distintas, contemplando nueve enfermedades y una clase saludable, tomando el 80 % para entrenamiento, 10 % para validación y el resto para pruebas, la dimensión de las imágenes evaluadas para todas las arquitecturas fue de 64×64 píxeles; posteriormente, derivado de la experimentación se obtienen resultados significativos logrando que la arquitectura VGGNet supere al resto de los algoritmos, desempeñándose con un porcentaje de precisión de 95.24 % y una pérdida de 0.17; concluyendo, que una arquitectura más profunda puede ser más benéfica para esta tarea [42].

Halil Durmus et al. en su trabajo, la preocupación es eminente, ya que el jitomate es uno de los cultivos más producidos en todo el mundo, según la Organización de las Naciones Unidas; así mismo, se ven afectadas las plantas por plagas y enfermedades, combatiéndolas con

métodos costosos, perjudicando la salud humana y de las plantas; por ende, se desarrolló una investigación, acerca, de detección de diferentes enfermedades (bacterial spot, early blight, late blight, leaf mold, septoria leaf spot, spider mites, target spot, mosaic virus, y yellow curl virus) a través de las hojas de plantas de jitomate, usando aprendizaje profundo ejecutado bajo la plataforma de cómputo móvil Nvidia Jetson TX1 en un robot en tiempo real, probando dos arquitecturas de redes neuronales artificiales, AlexNet y SqueezeNet, mostrando el desempeño de entrenamiento y validación. Para el entrenamiento, se tomó el conjunto de datos de Plantvillage, conteniendo 54.309 imágenes etiquetadas para 14 cultivos diferentes, de los cuales, en este trabajo se utilizaron diez clases distintas para jitomate, incluyendo las hojas sanas; resultado de la experimentación, se concluye que el algoritmo SqueezeNet resultó ser un buen candidato para la clasificación de aprendizaje profundo móvil, debido a que es más ligero y contribuye al ahorro computacional con respecto a AlexNet; sin embargo, AlexNet demostró un poco más de precisión con 95.65 % que SqueezeNet con 94.30 % [43].

Jia Shijie et al. en su trabajo, la principal problemática se deriva de la identificación manual de alguna plaga o enfermedad, detectándola manualmente basada en la experiencia, siendo laboriosa, con mucho tiempo invertido y propensa a errores; esto, impactando en el rendimiento de la producción. En esta investigación, se construyó una arquitectura de red neuronal convolucional, para detectar plagas y enfermedades del tomate (hojas sanas, Tomato bacterial spot, Tomato early blight, Tomato late blight, Tomato leaf mold, Tomato septoria leaf spot, Tomato target spot, Tomato mosaic virus, Yellow leaf curl virus, Tomato gray spot y Tomato two spotted spider mite), basada en la CNN VGG16; asimismo, se creó un conjunto de datos de hojas de plantas de tomate de enfermedades y plagas, conteniendo 11 clases, cada clase con 640 imágenes, teniendo un total de 7040 imágenes; para la primera prueba se tomó como base la arquitectura VGG16 para extracción de características, tomando la salida de la penúltima capa totalmente conectada y extrayendo los descriptores de características de entrenamiento para que sea entrenado el clasificador SVM, posteriormente se realizó detección de una plaga o enfermedad; en la segunda prueba se validó el algoritmo VGG16 de extremo a extremo, basado en el modelo original de la misma; después de los resultados de experimentación, se obtuvo un porcentaje de precisión del 89 %, concluyendo que el alto porcentaje de precisión se debe a imágenes de prueba de alta calidad relativa, es decir, fondo simple, objeto centrado y luz controlada [44].

Keke Zhang et al. en su trabajo, la principal problemática, se deriva de las enfermedades que afectan a los cultivos de plantas de jitomate; logrando perjudicar directamente la producción y el comercio agrícola; en esta investigación los autores evaluaron arquitecturas de redes neuronales convolucionales profundas, para identificar enfermedades como: *Corynespora leaf spot disease*, *Early blight*, *Late blight*, *Leaf mold disease*, *Septoria leaf spot*, *Two-spotted spider mite*, *Virus disease*, *Yellow leaf curl disease* y *Health*, en hojas de plantas de jitomate, a través, de aprendizaje por transferencia; como primera instancia se probaron dos métodos de optimización, *Stochastic Gradient Descent (SGD)* y *Adaptive Moment Estimation (Adam)* a tres redes CNN pre-entrenadas (*AlexNet*, *GoogleNet* y *ResNet*); *ResNet* obtuvo el mejor porcentaje de precisión en comparación con *AlexNet*, *GoogleNet*, asimismo, resultado de la prueba anterior, *ResNet* fue puesta a prueba modificando parámetros de configuración, como: tamaño de lote, número de iteraciones de entrenamiento y número de capas totalmente conectadas (ajuste fino); el conjunto de datos proviene de un repositorio de datos de acceso abierto; la implementación se desarrolló en Matlab, bajo ambiente Windows10, usando una GPU NVIDIA con 4GB de memoria o NVIDIA GTX1080Ti con 11GB de memoria; asimismo, resultado de las pruebas realizadas a la CNN *ResNet*, se ha obtenido un porcentaje de precisión del 97.28%; concluyendo que un gran tamaño de lote y un gran número de iteraciones, asegura mejorar la precisión del modelo objetivo, la configuración del tamaño del lote y la cantidad de iteraciones depende del conjunto de datos y la red utilizada [45].

Guan Wang et al. en su investigación, la principal problemática se centra en tres puntos: seguridad alimentaria, manejo de la enfermedad y pérdida del rendimiento por producción; en este trabajo, los autores proponen modelos de aprendizaje profundo para el diagnóstico automático de la gravedad de enfermedades en plantas, basado en imágenes; por ende, se experimentó aplicando pruebas a diferentes arquitecturas de redes neuronales convolucionales profundas, la primera con distintas capas convolucionales, entrenada desde cero; asimismo, *Inception-V3*, *ResNet50*, *VGG19* y *VGG16*, con aprendizaje por transferencia; logrando que *VGG16* superara a las demás con un porcentaje de precisión del 90.4%, identificando la enfermedad *Black Rot* causada por el hongo *Botryosphaeria Obtusa* en distintas fases, como: etapa sana, etapa temprana, etapa intermedia y etapa final, en hojas de plantas de manzana; además, se usó el conjunto de datos *Plantvillage*; las pruebas se realizaron en un equipo de cómputo workstations con OS

Ubuntu, con un hardware i5 6500 16GB de memoria, acelerada por una GPU Geforce GTX TITAN X 12GB de memoria. Derivado de la comparación de las arquitecturas CNN anteriormente mencionadas, se concluye que, el ajuste fino en estos modelos profundos pre-entrenados, puede mejorar significativamente el rendimiento con pocos datos [46].

Yu Sun et al. en su trabajo, como principal preocupación, logran cerrar la brecha taxonómica botánica; desarrollando un modelo de aprendizaje profundo para la identificación de imágenes de plantas en un entorno natural no controlado. En esta investigación, se probaron distintas versiones de la red neuronal convolucional ResNet, usando el framework de Keras; el modelo propuesto ResNet26 alcanza una tasa de reconocimiento del 91.78 %, comparándolo y superando al modelo original, ResNet18 con 89.27 %, ResNet34 con 88.28 % y ResNet50 con 86.15 %; el conjunto de datos usado fue el BJFU100, que contiene 10,000 imágenes de 100 especies de plantas ornamentales. Por otro lado, la arquitectura propuesta ResNet26 y los modelos originales, se sometieron a otros experimentos con el conjunto de datos de hojas Flavia, calculando el desempeño de la misma, logrando obtener un porcentaje de precisión del 99.65 % para la ResNet26, superando a los modelos originales de ResNet18, 34 y 50; esta implementación se logró, utilizando un equipo de cómputo servidor con OS Linux versión Ubuntu 16.04, con un procesador i7-3770 a 3.40 Ghz con 16Gb de memoria, con una GPU GTX1070 con 8Gb de memoria; los autores concluyen que el aprendizaje profundo es una tecnología prometedora para la clasificación de plantas a gran escala en entornos naturales [47].

Zhihao Jiao et al. en su investigación, se enfocan en la identificación de plantas partiendo de la estructura fisiológica de las hojas, el núcleo de esta investigación se basa en técnicas de aprendizaje profundo, implementando redes CNN, específicamente realizando una propuesta de Redes Adversarias Generativas Condicionales, probadas con el conjunto de datos de plantas ICL, alcanzando un desempeño para Top1 del 93.41 % y para Top4 un 96.63 % respectivamente, así mismo, comparando el rendimiento con arquitecturas CNN existentes [48].

Aitor Gutierrez et al. en su trabajo, la principal problemática, se centra en la detección de plagas por observación ocular, siendo un método insuficiente para cultivos grandes y no preciso; por lo consiguiente, en esta investigación se presenta un desarrollo y comparación para la detección e identificación de plagas (*Bemisia tabaci* y *Trialeurodes vaporariorum*) más comunes

en plantas de jitomate y pimiento de invernadero; los autores han desarrollado dos enfoques diferentes, visión computacional y estrategias de aprendizaje automático, mismas, que han sido comparadas con una solución de aprendizaje profundo, además, se creó un dataset para evaluar los modelos desarrollados; después de la experimentación, los resultados muestran un porcentaje de precisión del 82.34 % para el modelo de aprendizaje automático [49].

2.3.1. Desarrollo de arquitecturas CNN

Rupesh Kumar Srivastava et al. en su investigación, la problemática principal, nace del nivel de profundidad de una CNN, ya que involucra el aprendizaje de características en la etapa de entrenamiento, siendo un problema actual del desarrollo de la misma; los autores de este trabajo, presentan una arquitectura de redes neuronales convolucionales CNN, diseñada para facilitar el entrenamiento basado en el gradiente descendente estocástico SGD para redes muy profundas; se diseñó e implementó la red Highway Networks, permitiendo el flujo de información sin obstáculos, a través de varias capas convolutivas, caracterizándose por el uso de unidades de compuertas, que aprenden a enrutar el flujo de información en toda la red, asimismo, se ha medido el desempeño de las redes con el dataset CIFAR-10 y MNIST; posterior a la experimentación con diferentes profundidades, se obtiene una precisión del 92.24 %, concluyendo que un modelo extremadamente profundo puede tener un desempeño eficiente [50].

Rupesh Kumar Srivastava et al. en su trabajo, el problema que abordan los autores, parte, de lo largo y profundo de una CNN, ya que el proceso de entrenamiento se ha convertido en una tarea compleja; por ende, se ha desarrollado una nueva arquitectura CNN para superar esto; la propuesta de esta investigación, se centra en una inspiración de una red recurrente de memoria a corto plazo y utilizan unidades de compuerta adaptativas para regular el flujo de información, se probaron los diferentes modelos desarrollados con el dataset MNIST, CIFAR-10 y CIFAR-100; derivado de la experimentación realizada, se logró un porcentaje de precisión del 92.46 % para CIFAR-10 [51].

Adriana Romero et al. en su investigación, el objetivo es abordar el problema de compresión de la red aprovechando la profundidad; por lo consiguiente, proponemos un enfoque novedoso

para entrenar redes delgadas y menos profundas (pero aún profundas), llamadas FitNets; se validó el método propuesto en los dataset de referencia MNIST, CIFAR-10, CIFAR-100, SVHN y AFLW; posterior a la experimentación, se ha logrado una precisión del 91.61 % para CIFAR-10 [52].

Alex Krizhevsky et al, en su investigación, la principal problemática, se deriva de las limitantes que ofrecen las redes neuronales artificiales tradicionales, ya que intentan resolver problemas complejos con resultados por debajo de lo esperado; por ende, en este trabajo, los autores contribuyen en el entrenamiento de una red neuronal convolucional profunda (CNN), clasificando con el conjunto de datos ImageNet; el modelo, se probó en la competencia de desafío de reconocimiento visual ILSVRC-2010 e ILSVRC-2012; la CNN desarrollada, está compuesta de 60 millones de parámetros, 650 mil neuronas, 5 capas convolucionales y 3 capas totalmente conectadas; por otro lado, ImageNet, contiene 15 millones de imágenes y 22 mil clases; de las cuales, para este desarrollo se usaron 1.2 millones de imágenes de alta resolución para 1000 clases distintas; en el concurso ILSVRC-2010, se logró obtener una tasa de error del 37.5 % y 17.0 % para top-1 y top-5; asimismo, en la competencia ILSVRC-2012, resultó con un porcentaje del 15.3 % en cuanto a la tasa de error; los autores concluyen, que, partiendo de los resultados obtenidos, demuestran que una CNN grande y profunda, es capaz de lograr valores récord en un conjunto de datos altamente complejo [53].

Gao Huang et al. en su investigación, el problema fundamental de este trabajo, se centra en la disminución de la tasa de error de la CNN propuesta, comparándola con los métodos de vanguardia que existen hasta el momento; por ende, se han desarrollado modelos de redes neuronales convolucionales DenseNet, probando el desempeño de la misma, con cuatro dataset, CIFAR-10, CIFAR-100, SVHN e ImageNet; la metodología se desarrolló interconectando cada una de las capas de la DenseNet a todas las capas posteriores, mejorando el flujo de información entre ellas, asimismo, con este método, se redujeron sustancialmente los parámetros y el uso de cómputo; los modelos propuestos, demostraron un porcentaje de tasa de error para la variable top-5 de 3.46 % para el dataset CIFAR-10, 17.18 % para CIFAR-100, 1.59 % para SVHN y 6.15 % para ImageNet [54].

Forrest N. Iandola et al. en su trabajo, la principal problemática a abordar, es la reducción de

una CNN, siendo escalable a otras plataformas y su aplicación se diversifique; por ende, en este trabajo, los autores contribuyen en el desarrollo de una red CNN, clasificando con el conjunto de datos ImageNet; la CNN SqueezeNet, es un arquitectura que tiene 50 veces menos parámetros que AlexNet y se comprime en 0.5 MB menos, manteniendo la precisión de sus antecesores; posterior a la experimentación, se logro una precisión del 82.5%; los autores concluyen, que, partiendo de los resultados obtenidos, SqueezeNet es un buen candidato para una gran variedad de aplicaciones, especialmente aquellas que el tamaño del modelo es importante, como en el entrenamiento en servidores distribuidos, exportar un modelo desde la nube y aplicaciones de sistemas embebidos en un FPGA [55].

Christian Szegedy et al. en su trabajo, la problemática se deriva del progreso pasible de las CNN en años atrás, misma, que se había detenido, no sólo por falta de hardware más potente, conjunto de datos y modelos más grandes, sino también por consecuencia de nuevas ideas, algoritmos y arquitecturas de redes mejoradas; por ende, los autores proponen una arquitectura de red neuronal convolucional profunda (GoogLeNet), para la detección y clasificación en el concurso de desafío de reconocimiento visual a gran escala ILSVRC-14; la CNN propuesta está compuesta por 22 capa convolucionales y 5 capas totalmente conectadas, puesta a prueba con el conjunto de datos ImageNet, en el concurso anteriormente mencionado, considerando, 1000 clases; además, GoogLeNet usa 12 veces menos parámetros que la arquitectura AlexNet, mientras que es significativamente más precisa; los resultados demuestran una tasa de error de top-5, del 6.67%, superando a otras arquitecturas con las que fue comparada; se concluye que, este método, aumento en la calidad con un incremento modesto en los requisitos computacionales, en comparación con arquitecturas menos profundas y estrechas [56].

Karen Simonyan y Andrew Zisserman, en su investigación, el principal desafío en este trabajo, fue la profundidad de las capas convolucionales de los modelos propuestos; la configuración de filtros, en cada una de las capas son pequeñas, con dimensiones de 3x3, y en algunos casos 1x1; con esta investigación los autores buscan, evaluar la profundidad de una arquitectura de redes convolucionales ConvNets profundas, reconociendo imágenes a gran escala y validando el modelo con el conjunto de datos ImageNet en el concurso ILSVRC-2014; el equipo de VGG aseguro un segundo lugar en el concurso, obteniendo una tasa de error del 7.3%, posterior a ello,

se redujo al 6.8 %; concluyendo que la profundidad de las ConvNets, es benéfica en la precisión de la clasificación, asimismo, el rendimiento mostrado en el conjunto de datos ImageNet, se puede lograr con las arquitecturas aquí propuestas [57].

Christian Szegedy et al. en su investigación, la principal problemática en este trabajo, es agilizar el proceso de entrenamiento a un bajo costo computacional en las CNN; por ende, se han implementado dos arquitecturas de red CNN residuales Inception-ResNet-v1, Inception-ResNet-v2 y una CNN Inception-v4 pura; mismas, que fueron probadas, mostraron su desempeño en el concurso ILSVRC-2015 con el conjunto de datos ImageNet; los modelos propuestos, demostraron un porcentaje de error de 3.08 % para top-5 [58].

Christian Szegedy et al. en su trabajo, y derivado del éxito obtenido en la competencia ILSVRC-2012, se han impulsado nuevas líneas de investigación, enfocadas a redes CNN de mayor rendimiento; mejorando las arquitecturas de red, impactando en la amplitud y profundidad de las mismas; por ende, en esta investigación se describen y prueban arquitecturas CNN (Inception-v3), el modelo contiene 42 capas de profundidad, identificando 1000 clases del conjunto de datos ImageNet, iterando 100 épocas de entrenamiento; el modelo propuesto, se probó en el concurso ILSVRC-2012, logrando obtener un porcentaje de error del 3.5 % para top-5; demostrando que se pueden alcanzar resultados de alta calidad con una resolución de campo tan baja como 79x79, además, la factorización y la reducción de dimensiones dentro de la red neuronal, puede tener un costo computacionalmente bajo, manteniendo la calidad de la misma [59].

Kaiming He et al. en su trabajo, la principal problemática de esta investigación, es la degradación generada en la etapa de entrenamiento, afectando directamente a la precisión; por ende, se desarrolló un modelo de aprendizaje residual para facilitar el entrenamiento en una CNN, sustancialmente más profundas que las usadas en la actualidad; la propuesta metodológica, parte del diseño tradicional de una CNN, sólo que a esta, le es agregado más capas convolutivas y funciones de aprendizaje residuales, las cuales conectan de manera directa un par de capas convolucionales, obteniendo pequeños aprendizajes por cada conexión, asimismo, se hicieron pruebas con CNN simples y CNN residuales comparando tasas de error, con diferentes capas de profundidad, parámetros y tamaño de las capas convolutivas; además, participaron en el concurso ILSVRC-2015 y COCO 2015 ganando el primer lugar, obteniendo una tasa de error

del 3.57 % para la variable top-5; los autores concluyen que las redes residuales pueden disfrutar de ganancias de precisión a partir de una profundidad mucho mayor, con mejores resultados que sus predecesoras [60].

2.4. Trabajos con implementaciones de robótica

Halil Durmus et al. en su trabajo, la preocupación es eminente, ya que el jitomate es uno de los cultivos más producidos en todo el mundo, según la Organización de las Naciones Unidas; así mismo, se ven afectadas las plantas por plagas y enfermedades, combatiéndolas con métodos costosos, perjudicando la salud humana y de las plantas; por ende, se desarrolló una investigación, acerca, de detección de diferentes enfermedades (bacterial spot, early blight, late blight, leaf mold, septoria leaf spot, spider mites, target spot, mosaic virus, y yellow curl virus) a través de las hojas de plantas de jitomate, usando aprendizaje profundo ejecutado bajo la plataforma de cómputo móvil Nvidia Jetson TX1 en un robot en tiempo real, probando dos arquitecturas de redes neuronales artificiales, AlexNet y SqueezeNet, mostrando el desempeño de entrenamiento y validación. Para el entrenamiento, se tomó el conjunto de datos de Plantvillage, conteniendo 54.309 imágenes etiquetadas para 14 cultivos diferentes, de los cuales, en este trabajo se utilizaron diez clases distintas para jitomate, incluyendo las hojas sanas; resultado de la experimentación, se concluye que el algoritmo SqueezeNet resulto ser un buen candidato para la clasificación de aprendizaje profundo móvil, debido a que es más ligero y contribuye al ahorro computacional con respecto a AlexNet; sin embargo, AlexNet demostró un poco más de precisión con 95.65 % que SqueezeNet con 94.30 % [43].

Noa Schor et al. en su trabajo, las enfermedades en las plantas, son la principal causa de descensos del rendimiento por producción y aumento de uso de pesticidas, impactando en los balances financieros; por ende, se ha desarrollado y evaluado un sistema de detección robótica para la identificación de enfermedades (Powdery mildew y Tomato spotted wilt virus) en plantas; el sistema se basa, en un manipulador que facilita alcanzar múltiples poses de detección; asimismo, se desarrollaron algoritmos basados en PCA y en el coeficiente de variación CV; los resultados experimentales demuestran que, el porcentaje de precisión más alto fue de 95.2 %

para PCA, mientras los otros métodos fueron inferiores a este [61].

2.5. Otros

Pim Lindhout et al. en su investigación, la principal problemática se deriva de las nuevas razas de *Cladosporium Fulvum* (*Fulvia Fulva*), agente causal de la enfermedad moho de las hojas, perjudicando a la producción de tomate (*Lycopersicon esculentum* Mill); estas razas evolucionan rápidamente, superando a los genes de resistencia introducidos en cultivos comerciales; por ende, en este trabajo se desarrolló experimentación para comparar la resistencia de genes, frente a diferentes razas de *Cladosporium Fulvum*; resultado de la experimentación, el gen Cf6 probó ser resistente frente a todas las razas disponibles a diferencia de los demás; asimismo, se concluye que, en la producción de tomate, se ve en la necesidad de introducir nuevos genes de resistencia, siempre y cuando aparezcan nuevas razas con genes sumamente complejos [62].

Miaoying Tian et al. con su investigación, describen y caracterizan funcionalmente el inhibidor EPI10 de la proteasa (enzima que rompe enlaces proteínicos) extracelular; el gen EPI10 fue regulado a la alza durante la infección de *P. Infestans*, agente causal de la enfermedad tizón tardío en la planta de tomate, lo que permite tener un papel importante durante la patogénesis; asimismo, el EPI10 inhibió e interactuó con la proteína P69B del tomate. Resultado de la experimentación, se encontró que *P. Infestans*, desarrolló dos inhibidores de proteasa diferentes y estructuralmente divergentes, identificados en base al algoritmo de Laskowski [63].

Tabla 2.2: Desempeño de trabajos relacionados con técnicas de aprendizaje profundo.

Referencia	Objetivo del trabajo	Conjunto de datos	Método aplicado	Desempeño
[21]	Reconocimiento de especies de plantas (maleza)	6 conjuntos de datos distintos para 22 especies	CNN propia	86.2 %
[22]	Reconocimiento de imágenes para optimizar la producción en plantación de maíz	3 clases (propio)	VGG16+SVM	99.58 %
[23]	Identificación de plantas	Flavia y BJFU100	C-RNN (CNN+RNN)	99.65 %
[24]	Identificación de especies de plantas	LifeCLEF-2015	GoogLeNet, AlexNet y VGGNet	80.18 %
[25]	Detección de nuevas y extrañas especies de plantas	44 especies distintas	CNN propuesta basada en AlexNet	99.5 %
[26]	Identificación de plantas de grano fino	PlantCLEF-2015 y PlantCLEF-2017	GoogLeNet, ResNet-50, Inception-v3, Inception-ResNet-v2, MobileNet-v2, ShuffleNet-v2, DenSeNet-169 y SENet-154	84.07 %
[27]	Identificación de plantas en ambientes no controlados	100 especies	CNN propia	99.3 %
[28]	Clasificación automática de flores silvestres	17 especies	CNN propia	90.20 %
[29]	Reconocimiento de enfermedades en plantas	13 clases	Se tomo como base la red CaffeNet	96.3 %
[30]	Identificación de enfermedades	10 clases	Arquitectura CNN pre-entrenada	95.48 %
[31]	Identificación de enfermedades en hojas de plantas de maíz	9 clases	Modelos CNN mejorados de GoogLeNet y Cifar10	98.9 %
[32]	Clasificación de enfermedades en hojas de plantas de maíz	Plantvillage (4 clases)	CNN LeNet modificada	97.89 %
[33]	Reconocimiento de hojas en plantas	MalayaKew MK y LeafSnap	MSF-CNN	98.74 %
[34]	Reconocimiento de enfermedades en hojas de plantas	Conjunto de datos extraído de un repositorio en Internet	CNN SIANN	95 %
[35]	Clasificación de enfermedades en plantas de mango	Conjunto de datos propio y Plantvillage	MCNN propuesta basada en AlexNet	97.13 %
[36]	Detección en tiempo real de enfermedades en plantas de manzana	ALDD (5 clases)	INAR-SSD propuesta basada en GoogLeNet (VGG-INCEP) y Rainbow	78.80 %
[37]	Detección y diagnóstico de enfermedades en hojas de varias plantas	Conjunto de datos de ambiente abierto (58 clases)	AlexNet, AlexNetOWTBn, GoogLeNet, Overfeat y VGG	99.53 %
[38]	Identificación de enfermedades en hojas de plantas de manzana	4 clases	CNN basada en AlexNet	97.62 %
[39]	Detección de enfermedades y plagas en hojas de plantas de tomate	Conjunto de datos propio	Faster R-CNN, R-FCN, y SSD	83.06 %
[40]	Detección y clasificación de enfermedades en hojas de plantas de tomate	Plantvillage	CNN+LVQ	86 %
[41]	Reconocimiento de enfermedades en plantas de uva	Plantvillage (3 clases)	ResNet50 y VGG16	98.76 %
[42]	Detección de enfermedades en hojas de plantas de jitomate	Plantvillage (10 clases)	AlexNet, VGGNet, GoogleNet, y CNN propia	95.24 %
[43]	Detección de diferentes enfermedades en plantas de jitomate	Plantvillage (10 clases)	AlexNet y SqueezeNet	95.65 %
[44]	Detección de plagas y enfermedades en plantas de tomate	Conjunto de datos propio (11 clases)	VGG16+SVM y VGG16	89 %
[45]	Identificación de enfermedades en hojas de plantas de jitomate	Plantvillage	AlexNet, GoogleNet y ResNet	97.28 %
[46]	Diagnóstico automático de la gravedad de enfermedades en plantas de manzana	Plantvillage	CNN propia, Inception-V3, ResNet50, VGG19, y VGG16	90.4 %
[47]	Identificación de plantas en un entorno natural no controlado	BJFU100 y Flavia	Distintas versiones de ResNet	99.65 %
[48]	Identificación de plantas	ICL	Redes Adversarias Generativas Condicionales	96.63 %
[49]	Detección e identificación de plagas en plantas de jitomate y pimiento	Conjunto de datos propio (2 clases)	Aprendizaje automático y aprendizaje profundo	82.34 %

Tabla 2.3: Desarrollo de arquitecturas CNN.

Referencia	Objetivo del trabajo	Conjunto de datos	Arquitectura desarrollada	Desempeño
[50]	Entrenamiento óptimo de la CNN	CIFAR-10 y MNIST	CNN Highway Networks	92.24 %
[51]	Optimizar el proceso de entrenamiento de la CNN	MNIST, CIFAR-10 y CIFAR-100	CNN recurrente de memoria a corto plazo	92.46 %
[52]	Proponen CNN delgadas y menos profundas (pero aún profunda)	MNIST, CIFAR-10, CIFAR-100, SVHN, y AFLW	FitNets	91.61 %
[53]	Optimización de aprendizaje de la CNN	ImageNet	AlexNet	Tasa de error del 37.5 % y 17.0 % para top-1 y top-5
[54]	Disminución de la tasa de error de la CNN	CIFAR-10, CIFAR-100, SVHN, e ImageNet	DenseNet	Tasa de error top-5 1.59 %
[55]	Reducción de parámetros de la CNN	ImageNet	SqueezeNet	82.5 %
[56]	Proponen una nueva CNN con menos parámetros que otras arquitecturas	ImageNet	GoogLeNet	Tasa de error de top-5 del 6.67 %
[57]	Optimización de la profundidad de las capas convolucionales	ImageNet	ConvNets	Tasa de error de 6.8 %
[58]	Agilizar el proceso de entrenamiento a un bajo costo computacional	ImageNet	Inception-ResNet-v1, Inception-ResNet-v2, e Inception-v4	Tasa de error 3.08 % para top-5
[59]	Mejoras en la arquitectura de red impactando en la amplitud y profundidad	ImageNet	Inception-v3	Tasa de error 3.5 % para top-5
[60]	Desarrollo de un modelo de aprendizaje residual para optimizar el entrenamiento	ImageNet	CNN residual	Tasa de error 3.57 % para top-5

Tabla 2.4: Trabajos con implementaciones de robótica.

Referencia	Objetivo del trabajo	Conjunto de datos	Método aplicado	Desempeño
[43]	Detección de diferentes enfermedades en plantas de jitomate	Plantvillage (10 clases)	AlexNet, SqueezeNet, y un robot en tiempo real	95.65 %
[61]	Identificación de enfermedades en plantas	2 clases	PCA, coeficiente de variación (CV), y robot	95.2 %

Capítulo 3

Materiales y métodos

3.1. Definiciones

Inteligencia artificial (IA): La IA es una rama de las ciencias computacionales, este tipo de sistemas pasan por un proceso de entrenamiento o aprendizaje, percibiendo su entorno, tomando decisiones, y tomando acciones. Los sistemas de IA dependen de algoritmos que aprenden, basados en técnicas de aprendizaje automático y aprendizaje profundo, junto con grandes conjuntos de datos con representaciones bien definidas de la verdad objetiva.

Aprendizaje automático: El aprendizaje automático, básicamente se puede entender como la selección manual de funciones en el flujo de trabajo, posteriormente se desarrolla la etapa de entrenamiento del modelo. El aprendizaje automático y el aprendizaje profundo en algunos casos se podría pensar en una similitud en la terminología, sin embargo, son métodos que resuelven un mismo problema, pero de forma distinta. Algunas de las técnicas consideradas en el aprendizaje automático, se incluyen las siguientes: árboles de decisión, redes neuronales artificiales tradicionales, máquinas de vectores soporte, algoritmos probabilísticos, entre otros.

Aprendizaje profundo: El aprendizaje profundo, se desprende como un subconjunto del aprendizaje automático, el cual se basa en las vías neuronales del cerebro humano. Se centra en múltiples capas entre la capa de entrada y la capa de salida, además de la profundidad de cada una de ellas. Así mismo, otra característica particular acerca del aprendizaje profundo, es la forma de aprender automáticamente las características más relevantes. Las técnicas más

comunes de aprendizaje profundo incluyen redes neuronales convolucionales, redes neuronales recurrentes, redes Q profundas, entre otras.

Modelo: Es un conjunto de algoritmos entrenados que predicen, dado un conjunto de entradas bien definidas [1].

En la Figura 3.1, se visualiza la línea del tiempo de la inteligencia artificial, la cual es considerada una de las ramas de la computación, además, se plasma la relación que tiene con las subramas aprendizaje automático y aprendizaje profundo.

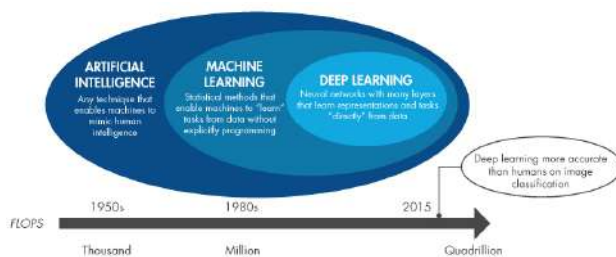


Figura 3.1: Línea del tiempo de la inteligencia artificial, Figura tomada de [1].

3.2. Espacios de color

Hay una diversificación de espacios de color, donde, se separa brillo, saturación, tonalidad, matiz, intensidad, etc, según sea el caso, dimensionándolos a distintas componentes; así mismo, los espacios de color orientados al hardware se consideran los siguientes: red, green, y blue (RGB), implementado en monitores y videocámaras; CMY, implementado en impresoras; YIQ, implementado en la televisión; por otro lado, los espacios de color usados para el procesado de imágenes en color destacan los siguientes: RGB; hue, saturation, y value (HSV); hue, saturation, e intensity (HSI); $L^*a^*b^*$; $L^*u^*v^*$; YUV; y YCbCr. A continuación, se da referencia de los espacios de color usados en esta investigación.

3.2.1. Espacio de color RGB

RGB, es una representación de los colores red, green, y blue respectivamente, además, podría decirse que es una estructura para componentes orientados al hardware, proyectando los canales

que representa en sistemas de proyección y adquisición de imágenes como monitores y videocámaras, además, usado en la tecnología de la televisión; así mismo, siendo una representación fundamental para equipos de cómputo, escáneres, y para el almacenamiento de imágenes. Cada color es representado por 1 byte, sumando 24 bits por cada píxel de la imagen, la combinación de los 3 bytes, será el color mostrado en el equipo electrónico para cada píxel. La intensidad de los tres canales, determinan la tonalidad y brillo del píxel resultante [64]. Aunque este espacio de color no es el óptimo para el preprocesado de imágenes, se usa para tal fin, la deficiencia que presenta el espacio de color RGB en esta área, es por la alta correlación que hay entre sus tres componentes.

3.2.2. Espacio de color HSV

El espacio de color HSV, por sus siglas en inglés (hue, saturation, value), representa los componentes separados del matiz, saturación y valor de un píxel. El HSV se ve más fácilmente como un cono, siendo el matiz la posición alrededor del borde, representando el color, la saturación es la distancia desde el centro hasta el borde, diluyendo el matiz en la luz blanca, finalmente, el valor representa la intensidad o brillo del color, tomando una posición entre la parte superior del cono y la parte baja. Además, los componentes matiz y saturación imitan la forma en que los humanos visualizan el color. Como ya se ha comentado en el párrafo anterior, el matiz representa el color, donde, girando alrededor del borde del cono, encontramos el color rojo, posteriormente el color verde y finalmente el color azul. La saturación representa la intensidad del color, cuanto más cerca del centro del cono se encuentre el valor de saturación, menos se expresará el valor del matiz. Finalmente, la posición arriba y abajo del cono representa el valor o brillo, hacia la parte inferior habrá píxeles más oscuros y píxeles más claros en la parte superior [65] [66].

3.2.3. Espacio de color YCbCr

El espacio de color YCbCr es una variante que proviene del espacio de color CIE, donde la estructura de colores se emplea para estandarizar la televisión digital. Los componentes de este espacio de color, son representados de la siguiente manera: Y representa la luminancia, y CbCr son los componentes de crominancia; este espacio es usado frecuentemente para el procesamiento

digital de imágenes, teniendo un campo específico en la compresión de imágenes. El valor que toma la componente Y, se escala entre 0 y 1, el cual representa la luminancia de un píxel; además, YCbCr utiliza el formato de archivo JPEG para comprimir altamente la información de color [65].

3.2.4. Espacio de color L*a*b*

Para realizar una transformación de un espacio de color a otro, se usan técnicas de procesamiento digital de imágenes como estas, dimensionando las componentes actuales a otro espacio distinto, logrando un preprocesado óptimo para un fin específico. Al igual que el espacio de color de la sección anterior, el espacio de color L*a*b* es una variante que proviene del espacio de color CIE. L*a*b*, está representado por tres componentes, en los cuales se logra separar la intensidad y tonalidad, donde, la componente L* representa la intensidad, a* y b* representan los componentes de tonalidad [67].

3.3. Técnicas de segmentación

En esta sección, se plasman dos técnicas para segmentar imágenes, así mismo, se explica el fundamento teórico de cada tópico. Estos métodos logran separar objetos o regiones de interés dentro de imágenes. A continuación se hace una descripción de estos algoritmos.

3.3.1. Otsu

El algoritmo de segmentación borde adaptativo, Otsu, es un método basado en umbrales, el cual busca el umbral óptimo k^* en imágenes, maximizando la varianza entre clases, refiera la ecuación (3.1). Así mismo, los píxeles se dividen en dos clases por un umbral [68] [69] [70].

$$\sigma_B^2(k^*) = \max_{1 \leq k < L} \sigma_B^2(k) \quad (3.1)$$

3.3.2. PCA

El algoritmo análisis de componentes principales (por sus siglas en inglés PCA), el objetivo principal es la reducción de dimensiones. Así mismo, el algoritmo PCA toma varios vectores o componentes, los cuales son procesados reduciendo el tamaño de las dimensiones a una componente principal [71] [72] [61]. Este algoritmo funge como el núcleo de la etapa de segmentación, aplicado a los canales de una imagen. En la Figura 3.2, las columnas 1,2, y 3, representan las componentes de una imagen en un espacio de color RGB respectivamente, posteriormente, bajo la ejecución del algoritmo PCA, se reduce a una componente principal.

1	2	3		1
0.9437	0.0457	0.9500	→ PCA →	0.6110
0.8153	0.0578	0.9289		0.5571
0.8242	0.0551	0.9400		0.6130
0.8215	0.0544	0.9314		0.6163
0.8936	0.0440	0.9729		0.6222
0.5335	0.0963	0.6794		0.6170
0.4103	0.1344	0.6251		0.5776
0.9110	0.0532	0.9609		0.6295
0.9329	0.0574	0.8981		0.5532
0.9388	0.0517	0.9299		0.5324
0.9483	0.0510	0.9269		0.4786
0.9175	0.0543	0.9246		0.4697
0.8835	0.0523	0.9484		0.5140
0.8569	0.0463	0.9173		0.5137
0.9387	0.0504	0.9386		0.5399
0.9436	0.0495	0.9493		0.5349
0.9433	0.0494	0.9497		0.5412
0.9768	0.0374	0.9587		0.5599
0.3371	0.1601	0.6239		0.6035
0.5914	0.8475	0.1040		0.6359
0.8764	0.9375	0.0531		

Figura 3.2: PCA, algoritmo para reducción de dimensiones.

3.4. Técnicas de extracción de características

La extracción de características, es un proceso delicado y es considerada piedra angular para los algoritmos de aprendizaje automático, la correcta implementación de las técnicas de extracción, definen los patrones reunidos para el proceso de reconocimiento de alguna área específica. En este apartado, la imagen es representada a través de un conjunto de valores numéricos con un alto potencial discriminativo, eliminando características redundantes, disminuyendo la dimensión de las mismas. Anteriormente, en el campo del procesamiento de imágenes, las características extraídas eran representadas por un vector que contenía todos los píxeles de la imagen integral, impactando directamente en el costo computacional por el alto procesamiento de la información. Hoy en día, los vectores de características extraídos, se reducen con base en la téc-

nica implementada, optimizando el costo computacional. En este documento, las características extraídas se dividen en dos vertientes, características de textura y características de color. A continuación se da detalle de los métodos y técnicas usadas para la extracción de características.

3.4.1. Características de textura

Las características de textura, específicamente extraen información a través de la superficie del objeto que se desea analizar, donde, las técnicas implementadas buscan patrones muy peculiares como: superficies ásperas, rugosas, delicadas, uniformes, granuladas, entre otras, hallando descriptores particulares sobre una disposición espacial en el área previamente segmentada. La información extraída, se puede presentar a lo largo del área segmentada, por lo tanto, estos algoritmos funcionan aún cuando la textura ha sido desplazada. Para este fin se ha puesto en marcha el algoritmo de Haralick [73], que básicamente se implementa de la siguiente manera, en primera instancia se obtiene la matriz de coocurrencia, la cual mide como coocurren pares o vectores de píxeles, posteriormente, se calculan los 14 descriptores de Haralick, dependiendo de la información que queramos extraer, se especifican que ecuaciones se van a implementar. Los descriptores de Haralick se describen a continuación:

$$f_1 = \sum_i \sum_j [p(i, j)^2] \quad (3.2)$$

$$f_2 = \sum_{n=0}^{N_g-1} n^2 \left\{ \begin{array}{l} \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \\ |i - j| = n \end{array} \right\} \quad (3.3)$$

$$f_3 = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} [ijp(i, j) - \mu_x \mu_y]}{\sigma_x \sigma_y} \quad (3.4)$$

$$f_4 = \sum_i \sum_j (i - \mu_x)^2 p(i, j) \quad (3.5)$$

$$f_5 = \sum_i \sum_j \frac{1}{1 + (i - j)^2} p(i, j) \quad (3.6)$$

$$f_6 = \sum_{i=2}^{2N_g} iP_{x+y}(i) \quad (3.7)$$

$$f_7 = \sum_{i=2}^{2N_g} (i - f_8)^2 P_{x+y}(i) \quad (3.8)$$

$$f_8 = - \sum_{i=2}^{2N_g} P_{x+y}(i) \log\{P_{x+y}(i)\} \quad (3.9)$$

$$f_9 = - \sum_i \sum_j p(i, j) \log\{p(i, j)\} \quad (3.10)$$

$$f_{10} = \sum_{i=0}^{N_g-1} (i - f_8)^2 P_{x-y}(i) \quad (3.11)$$

$$f_{11} = - \sum_{i=0}^{N_g-1} P_{x-y}(i) \log\{P_{x-y}(i)\} \quad (3.12)$$

$$f_{12} = \frac{HXY - HXY1}{\text{máx}\{HX, HY\}} \quad (3.13)$$

$$f_{13} = (1 - e^{[-2(HXY2-HXY)]})^{\frac{1}{2}} \quad (3.14)$$

$$f_{14} = (\text{Segundo valor propio mas grande de } Q)^{\frac{1}{2}} \quad (3.15)$$

3.4.2. Características cromáticas

Las características de color proveen mucha información, la cual puede ser extraída partiendo de un espacio de color específico, considerando escala de grises; rojo, verde, y azul (RGB por sus siglas en inglés), y hue, saturation, y value (HSV), por mencionar algunos, extrayendo descriptores a través de algoritmos como: descriptores de Contraste, momentos de Hu, descriptores de Fourier, transformada discreta del coseno (TDC) y características de Gabor. Cabe mencionar que el espacio de color usado para recuperar información con las distintas técnicas de extrac-

ción de características ya mencionadas en este párrafo, es el formato RGB, el cual consta de 3 bytes o 24 bits de profundidad por cada píxel de la imagen analizada, representando 1 byte de información para el canal R, 1 byte para el canal G, y finalmente, 1 byte para el canal B.

Los descriptores de Contraste [74], se obtienen con base en la diferencia entre el área previamente segmentada y su entorno; así mismo, entre menor sea la diferencia, menor será el contraste. Este cálculo se define de la siguiente manera:

$$K_1 = \frac{G - G_e}{G_e}, \quad K_2 = \frac{G - G_e}{G + G_e} \quad \text{and} \quad K_3 = \ln(G/G_e). \quad (3.16)$$

dónde, G y G_e denotan el valor promedio de la región segmentada y su vecindario.

Los momentos de Hu se basan en los momentos invariantes algebraicos de Boole, integrando información de la variable de color del área segmentada [75]. Esta técnica es considerada un método estadístico, el cual describe el centroide de un objeto en la imagen. En total son 7 momentos que se han definido, por cada uno se obtiene 1 descriptor, tomando en cuenta que es aplicado a una imagen a escala de grises o 1 byte de profundidad o a su equivalente 8 bits. Además, los momentos de Hu son invariantes a escala, traslación, y rotación, calculándose de la siguiente manera:

$$\phi_1 = \eta_{20} + \eta_{02} \quad (3.17)$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad (3.18)$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (3.19)$$

$$\phi_4 = (\eta_{30} - 3\eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad (3.20)$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \quad (3.21)$$

$$(3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + \quad (3.22)$$

$$4(\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}))$$

$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - \quad (3.23)$$

$$(\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

Por otro lado, se han extraído características de Fourier [76] [66], para su cálculo, utilizan funciones senos y cosenos bajo la aplicación de la transformada discreta de Fourier (TDF), obteniendo resultados con números complejos, así mismo, derivado de la implementación de esta técnica en imágenes de un sólo canal, se generan 8 características, las cuales son reunidas en un vector para su futuro procesado. Para una función bidimensional, los descriptores de Fourier se obtienen de la siguiente manera:

$$F(u, v) = \frac{1}{\sqrt{MN}} \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} I(x, y) \cdot e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})}. \quad (3.24)$$

La transformada discreta del coseno (TDC) [77], es usada en el área de procesamiento digital de imágenes, específicamente en la compresión de imágenes; además, usa transformaciones como base y funciones coseno de diferente longitud de onda. Una peculiaridad muy particular de la TDC en relación con la transformada discreta de Fourier (TDF), es que se limita al uso de coeficientes reales. La TDC en dos dimensiones se deriva de forma directa de la definición del caso unidimensional, en total, se obtienen 4 descriptores, tomando en cuenta que es aplicada a una imagen a escala de grises o 1 byte de profundidad o a su equivalente 8 bits, y es calculada de la siguiente manera:

$$F(u, v) = \frac{2}{\sqrt{MN}} \sum_{x=1}^N \sum_{y=1}^M I(x, y) \cdot c_u \cdot \cos\left(\frac{\pi(2x-1)u}{2N}\right) \cdot c_v \cdot \cos\left(\frac{\pi(2y-1)v}{2M}\right) \quad (3.25)$$

$$= \frac{2c_u c_v}{\sqrt{MN}} \sum_{x=1}^N \sum_{y=1}^M I(x, y) \cdot E_u^N(x) \cdot E_v^M(y)$$

Los descriptores de Gabor, es una técnica híbrida de extracción de características de textura. Siendo una función compuesta por el núcleo de la transformada de Fourier en una función Gaussiana. El cálculo de esta técnica se basa en la familia de funciones de Gabor [78] [17]. Derivado de la implementación de este método en imágenes de un sólo canal, se generan 67 características, las cuales son reunidas en un vector para su futuro procesado. Los descriptores de Gabor es un filtro 2D, representado por la siguiente ecuación:

$$g(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right] \exp(2\pi j u_0 x). \quad (3.26)$$

3.5. Técnicas de equilibrio de clases

Un conjunto de datos está en desequilibrio, cuando la cantidad de instancias de una clase es mayor (clase mayoritaria) que el resto de las clases (clases minoritarias), un desbalance en el conjunto de datos, puede afectar el desempeño de los algoritmos de aprendizaje automático, sesgando los resultados hacia la clase mayoritaria. En este trabajo se incluyen tres técnicas para el equilibrio de clases, considerando técnicas heurísticas y no heurísticas.

3.5.1. Submuestreo aleatorio

El submuestreo aleatorio, es considerado un método de equilibrio de clases no heurístico. Bajo su desarrollo, primero se localiza la clase con el menor número de instancias, posteriormente se identifican las clases restantes con número de instancias superior a la clase minoritaria, finalmente, en cada clase mayoritaria detectada, se selecciona el número de instancias de la clase minoritaria de forma aleatoria, logrando conjunto de datos simétrico [79]. En la Figura 3.3, las barras de color azul representan el número de instancias de las clases de un conjunto de datos asimétrico determinado, así mismo, como se puede observar, la clase 1 funge como clase minoritaria, y las clases restantes como clases mayoritarias, posterior a la aplicación de la técnica submuestreo aleatorio, como resultado, se obtiene un conjunto de datos simétrico respecto a la clase minoritaria, representado con las barras de color verde.

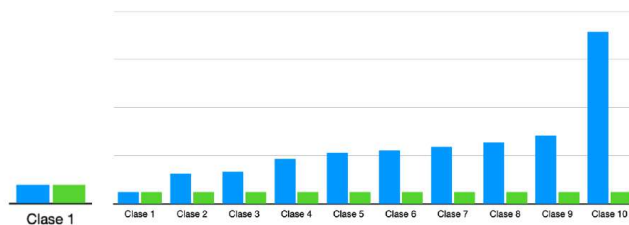


Figura 3.3: Submuestreo aleatorio.

3.5.2. Sobremuestreo con réplicas

Así como el submuestreo aleatorio, el sobremuestreo con réplicas es considerado un método de equilibrio de clases no heurístico. Esta técnica, antes de muestrear las clases, se aplica una segmentación al conjunto de datos, regularmente se divide de la siguiente manera, 70 % para conjunto de entrenamiento y 30 % como conjunto de prueba u 80 % como conjunto de entrenamiento y 20 % para pruebas. Posterior a la segmentación del conjunto de datos, el muestreo con réplicas o copias de instancias, realiza un sobremuestreo exclusivamente de las clases minoritarias del conjunto de entrenamiento, logrando una simetría con la clase mayoritaria [80]. En la Figura 3.4, las barras de color azul representan el número de instancias de las clases de un conjunto de datos asimétrico determinado, así mismo, como se puede observar, la clase 10 funge como clase mayoritaria, y las clases restantes como clases minoritarias, posterior a la aplicación de la técnica sobremuestreo con réplicas, como resultado, se obtiene un conjunto de datos simétrico respecto a la clase mayoritaria, representado con las barras de color verde.

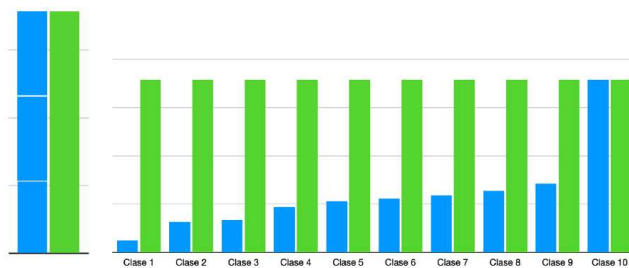


Figura 3.4: Sobremuestreo con réplicas.

3.5.3. Sobremuestreo con el algoritmo SMOTE

La técnica de sobremuestreo de minorías sintéticas (por sus siglas en inglés SMOTE), como en el sobremuestreo con réplicas, antes del muestreo, es aplicada una segmentación al conjunto de datos, posteriormente se sobremuestra sólo el conjunto de entrenamiento. El algoritmo SMOTE es considerado un método heurístico, el objetivo principal se deriva de la distribución asimétrica entre las clases de un conjunto de datos determinado, logrando realizar un balanceo entre las instancias de las diferentes clases, basado en la creación de muestras sintéticas a partir de una distancia y vecindad de las clases minoritarias. Los datos sintéticos se crean a través de las características o atributos de las instancias, así mismo, son introducidos a lo largo de los segmentos de línea que unen a cualquiera o todos los vecinos más cercanos de las clases minoritarias, además, dependiendo de la cantidad de sobremuestreo requerido, los k vecinos se eligen al azar [81]. En la Figura 3.5, las barras de color azul representan el número de instancias de las clases de un conjunto de datos asimétrico determinado, así mismo, como se puede observar, la clase 10 funge como clase mayoritaria, y las clases restantes como clases minoritarias, posterior a la aplicación de la técnica SMOTE, como resultado, se obtiene un conjunto de datos simétrico respecto a la clase mayoritaria, representado con las barras de color verde.

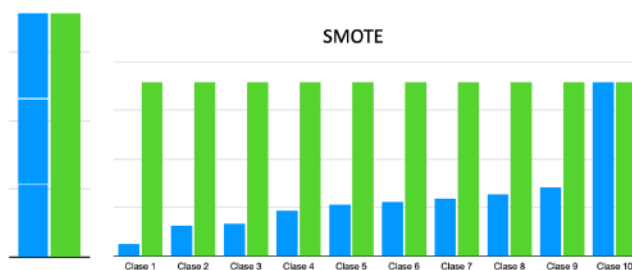


Figura 3.5: Sobremuestreo con el algoritmo SMOTE.

3.6. Técnicas de clasificación

En esta sección, se definen los algoritmos usados para la clasificación, así mismo, se da una breve descripción de cada uno de ellos. Los métodos de clasificación utilizados se consideraron los siguientes: máquinas de vectores soporte (SVM), redes neuronales artificiales (RNA), naive bayes (NB), K-vecinos más cercanos (KNN), bosques aleatorios (BA), y regresión logística (RL).

3.6.1. Máquinas de vectores soporte (SVM)

SVM, es un algoritmo para clasificación, que bajo su implementación, realiza una separación entre clases a través de la búsqueda de un hiperplano óptimo siempre que las clases sean linealmente separables, aplicado sobre los datos de entrenamiento. Las instancias que se encuentran al borde de cada clase, fungen como vectores soporte. Cuando las clases no son linealmente separables, se utilizan funciones llamadas kernels, que transfieren los datos a otro espacio altamente dimensional, logrando la separabilidad entre ellas. El entrenamiento de una SVM, se denota en la ecuación (3.27), lo cual resuelve un problema de programación cuadrática, así mismo, el hiperplano óptimo se define en la ecuación (3.28), donde, S representa los vectores soporte, y finalmente, la clasificación de un nuevo objeto x viene dada por la ecuación (3.29). Para una revisión más detallada acerca del método, vea [82].

$$\max_{\alpha_i} -\frac{1}{2} \sum_{i,j=1}^l \alpha_i y_i \alpha_j y_j \mathbf{K}(x_i \cdot x_j) + \sum_{i=1}^l \alpha_i \quad (3.27)$$

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

$$\sum_{i \in S} (\alpha_i y_i) K(\mathbf{x}_i, \mathbf{x}_j) + b = 0. \quad (3.28)$$

$$f(x) = \text{sign} \left(\sum_{i \in S} (\alpha_i y_i) K(x_i, x_j) + b \right). \quad (3.29)$$

3.6.2. Redes neuronales artificiales (RNA)

Las redes neuronales artificiales trata de imitar el proceso de aprendizaje y solución del cerebro humano, esto se consigue, con la ayuda de métodos computacionales que ayuden a contribuir a un mejor desempeño del mismo. Los humanos, para resolver problemas de la vida cotidiana, toman conocimiento previo, adquirido de la experiencia de alguna área en específico, así mismo, las redes neuronales artificiales, recaban información de ejemplos o situaciones para construir modelos o sistemas que puedan tomar decisiones de manera automática. Uno de los métodos de aprendizaje más usados y el que fue utilizado en este trabajo es el algoritmo backpropagation

(BP); el cual de forma general, realiza el proceso de aprendizaje y clasificación en cuatro apartados, inicialización de pesos, propagación hacia adelante, propagación hacia atrás y actualización de pesos. Primero, se realiza una inicialización aleatoria de pesos, en consecuencia, las ecuaciones (3.30) y (3.31) definen la propagación directa, donde las unidades en cada capa determinan sus estados por las entradas que reciben de las capas inferiores, posteriormente, la complejidad de este algoritmo se refleja en la propagación hacia atrás, donde desciende de las unidades de las capas superiores a las inferiores, comenzando su cálculo con la ecuación (3.32), y finalmente, se actualiza cada unidad de salida. Para una revisión más profunda de este algoritmo, vea [83].

$$x_j = \sum_i y_i w_{ji}. \quad (3.30)$$

$$y_j = \frac{1}{1 + e^{-x_j}}. \quad (3.31)$$

$$\partial E / \partial y_j = y_j - d_j. \quad (3.32)$$

3.6.3. Naive bayes (NB)

Los algoritmos basados en el teorema de Bayes, en particular NB, los valores de precisión del estimador numérico, se eligen en función del análisis de los datos de entrenamiento; asimismo, este método es fundamental para resolver problemas de clasificación de patrones cuando la distribución de probabilidad de los patrones es conocida. Un clasificador Bayesiano usa un enfoque probabilístico para asignar la clase a un ejemplo. 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 un objeto con k características $X_k = [x_1, x_2, \dots, x_k]$, para este caso, el conjunto de características define a un objeto específico. Posterior a ello, el algoritmo calcula la probabilidad condicional posterior $p(C_i|X_k)$ usando la regla de Bayes:

$$p(C_i|X_k) = \frac{p(X_k|C_i)p(C_i)}{p(X_k)}, i = 1, 2, \dots, m. \quad (3.33)$$

En la ecuación (3.33) $p(X_k|C_i)$, $p(C_i)$ y $p(X_k)$ son calculadas a partir de los datos de entre-

namiento. Asimismo, partiendo del teorema de Bayes, para una observación dada (X_k), la clase a la que pertenece está dada por la probabilidad posterior máxima:

$$f(X_k) = \arg \max_i p(C_i|X_k) \quad (3.34)$$

Para realizar un análisis a profundidad del algoritmo NB, referirse a [84].

3.6.4. K-vecinos más cercanos (KNN)

El algoritmo KNN, clasifica un nuevo punto del conjunto de datos, basándose en la distancia euclidiana, encontrando las k distancias más cercanas al objeto a clasificar. En primera instancia, se calculan las distancias del nuevo punto a cada objeto del conjunto de datos, la distancia euclídea de un punto a a un punto b se calcula de la siguiente manera:

$$d(P_0, P_1) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (3.35)$$

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 número votos. Para realizar un análisis a profundidad del algoritmo KNN, referirse a [85].

3.6.5. Bosques aleatorios (BA)

Bosques aleatorios, es un algoritmo compuesto de clasificadores de árboles de decisión, cada árbol depende de los valores de un vector aleatorio, muestreando de forma independiente y con la misma distribución para todos los árboles del bosque. El error de generalización para los bosques converge hasta un límite a medida que aumenta el número de árboles en el bosque. Cuando un modelo se generaliza y se produce un error, depende de la fuerza de los arboles individuales en el bosque y la correlación entre ellos. Al seleccionar aleatoriamente las características para dividir cada nodo, se producen tasas de error que se comparan favorablemente con el algoritmo Adaboost pero son más robustas con respecto al ruido. En [86], se plasma el proceso de clasificación con el algoritmo BA, asimismo, se especifica la caracterización de la precisión, el uso de características aleatorias, la selección de entradas aleatorias, las entradas

de combinaciones lineales, la operación del algoritmo Adaboost, los efectos del ruido de salida, las entradas débiles de datos, los bosques aleatorios para regresión, teoremas y ecuaciones que conllevan a la ejecución del clasificador BA.

3.6.6. Regresión logística (RL)

El algoritmo regresión logística, es considerado un método de aprendizaje máquina de análisis predictivo, basado en conceptos de probabilidad. El núcleo de este algoritmo, es la función sigmoide, proporcionando una salida discreta, la cual, tiende a limitar la función de coste entre 0 y 1. El modelo de regresión logística relaciona π_i con las características mediante la ecuación (3.36), donde, $\beta = (\beta_0, \beta_1, \dots, \beta_p)$ es el vector de coeficientes de regresión [87, 88].

$$\text{logit}(\pi_i) = \log\left(\frac{\pi_i}{1 - \pi_i}\right) = X_i\beta. \quad (3.36)$$

3.7. Métricas de desempeño

El rendimiento de los algoritmos de aprendizaje automático se evalúan normalmente a través de una matriz de confusión, compuesta por columnas y filas, donde, las columnas son las clases predichas y las filas pertenecen a las clases actuales. Así mismo, a continuación se definen las métricas de desempeño usadas para medir el rendimiento del método propuesto, donde, (FN *por sus siglas en inglés*) representa los falsos negativos, (FP *por sus siglas en inglés*) representa los falsos positivos, (TN *por sus siglas en inglés*) representa los verdaderos negativos, y finalmente (TP *por sus siglas en inglés*) representa los verdaderos positivos.

$$\text{Acc} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (3.37)$$

$$\text{Precision} = \frac{TP}{(TP + FP)} \quad (3.38)$$

$$\text{Recall} = \frac{TP}{(TP + FN)} \quad (3.39)$$

$$F - Measure = \frac{(2 * precision * recall)}{(precision + recall)} \quad (3.40)$$

$$FP - Rate = \frac{FP}{(FP + TN)} \quad (3.41)$$

$$MCC = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (3.42)$$

Capítulo 4

Metodología

En este capítulo, se detalla el conjunto de datos usado en esta investigación, descrito en la Sección 4.1, además, las etapas del método propuesto y la forma en la que fueron implementados. El núcleo de la metodología aplicada, está basada en un sistema de reconocimiento de patrones, considerando segmentación, extracción de características y clasificación. A lo largo de los experimentos realizados, la metodología sufrió cambios en la estructura definida al inicio de la investigación, por ende, este capítulo se ha dividido en cuatro partes, haciendo hincapié en la distribución de los algoritmos usados en cada etapa del esquema del sistema propuesto. En cada actualización de la metodología, se ha nombrado de la siguiente manera: módulos del método propuesto inicial, Sección 4.2; módulo de preprocesamiento, Sección 4.3; módulo de equilibrio de clases, Sección 4.4; y finalmente, influencia de técnicas de extracción de características, Sección 4.5.

4.1. Conjunto de datos

En la experimentación desarrollada en este trabajo, se ha usado un conjunto de datos de ambiente libre, el cual fue obtenido de un repositorio de Internet, nombrado Plantvillage [89]. Este conjunto de datos, está compuesto de diez clases diferentes, ocho enfermedades, considerando virus del mosaico del tomate con 373 imágenes, moho de la hoja con 952 imágenes, tizón temprano con 1000 imágenes, mancha foliar con 1404 imágenes, mancha foliar por septoria con 1771 imágenes, tizón tardío con 1908 imágenes, mancha bacteriana con 2127 imágenes, y virus

Tabla 4.1: Distribución de imágenes del conjunto de datos.

Clase	Nombre común	Nombre científico	Número de imágenes
a	Virus del mosaico del tomate	Tomato mosaic virus	373
b	Moho de la hoja	Fulvia fulva	952
c	Tizón temprano	Alternaria solani	1000
d	Mancha foliar	Corynespora cassiicola	1404
e	Hojas sanas	Hojas completamente sanas	1591
f	Araña roja de dos puntos	Tetranychus urticae	1676
g	Mancha foliar por septoria	Septoria lycopersici	1771
h	Tizón tardío	Phytophthora infestans	1908
i	Mancha bacteriana	Xanthomonas campestris pv. vesicatoria	2127
j	Virus del enrollamiento de la hoja amarilla del tomate	Tomato yellow leaf curl virus	5357

del enrollamiento de la hoja amarilla del tomate con 5357 imágenes; y una plaga, araña roja de dos puntos con 1676 imágenes, y finalmente, de la clase completamente sana se tienen 1591 imágenes; sumando un total de 18159 imágenes procesadas, las imágenes están en un espacio de color RGB con dimensiones de 256×256 píxeles, véase la Tabla 4.1, y relaciónela visualmente con la Figura 4.1.

Las características visuales que comparten los distintos daños foliares de las diez clases del conjunto de datos Plantvillage, son muy notorias para el ojo humano en algunos casos, enfáticamente en los colores manifestados en las hojas, como el color café, marrón oscuro, y el color amarillo, además de algunas manchas y/o puntos muy peculiares; para discriminar entre las diez clases consideradas, se contempló la aplicación de métodos y técnicas computacionales, ya que de forma manual, ocular, o tradicional se podría emitir un diagnóstico erróneo.

En el Capítulo 3, estado del arte, se plasman algunos trabajos relacionados con esta investigación, donde, se visualiza el uso del conjunto de datos Plantvillage, procesando las imágenes con distintos métodos computacionales. Además, las imágenes del conjunto de datos Plantvillage, fueron obtenidas en un ambiente controlado de iluminación con un fondo homogéneo.

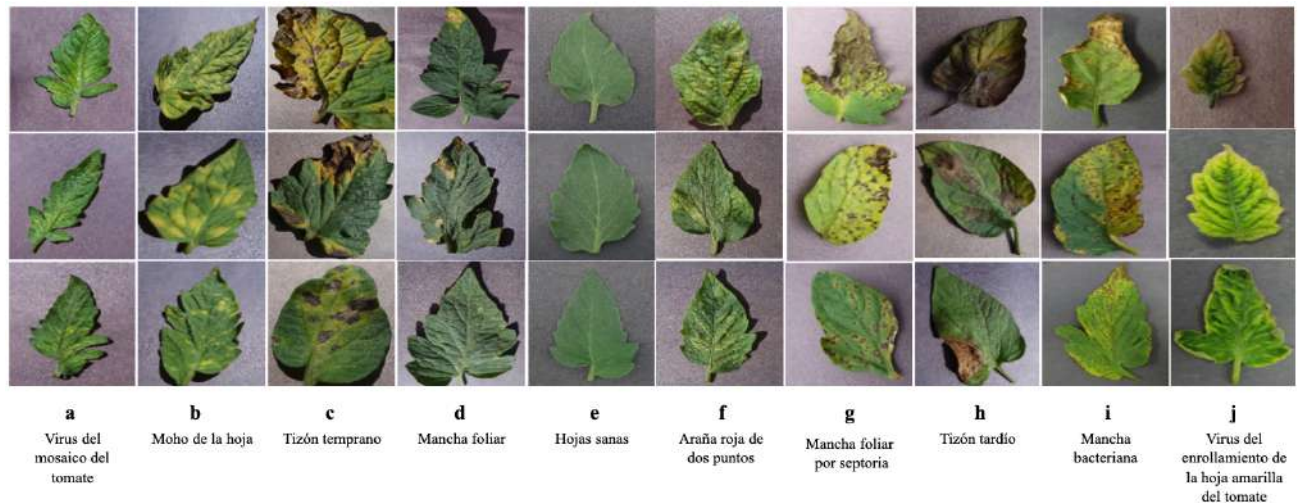


Figura 4.1: Conjunto de datos Plantvillage.

4.2. Módulos del método propuesto inicial

En este primer despliegue del método propuesto, se han planteado tres etapas, considerando el módulo de 1) segmentación, 2) extracción de características, y 3) clasificación, estructurados de manera secuencial. Cabe mencionar que, en la disposición modular del diagrama metodológico presentado en esta sección, no hay una implementación de preprocesado de imágenes al inicio de la ejecución del sistema. En los siguientes párrafos se explica cada una de las etapas desarrolladas de este proceso. En la Figura 4.2 se visualiza el diagrama del método propuesto.

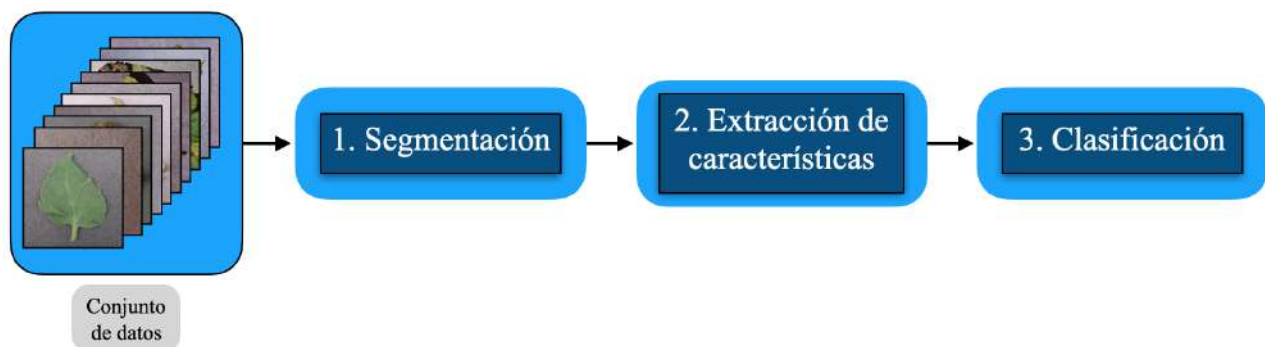


Figura 4.2: Método propuesto inicial.

4.2.1. Segmentación

En primera instancia, las imágenes pertenecientes al conjunto de datos son ingresadas al sistema, posterior a ello, son procesadas por esta etapa. El propósito de esta sección, es separar la hoja de su entorno, resultando el área de interés la cual será analizada en módulos posteriores. Las pruebas experimentales en este apartado, se han aplicado de manera autónoma con base en el algoritmo de segmentación frontera adaptativa Otsu y el algoritmo fase de análisis de componentes principales (PCA por sus siglas en inglés). Al realizar una segmentación prominente, el sistema utilizará la región o área segmentada para determinar los bordes y calcular las propiedades mediante las técnicas de extracción de características implementadas en la siguiente etapa.

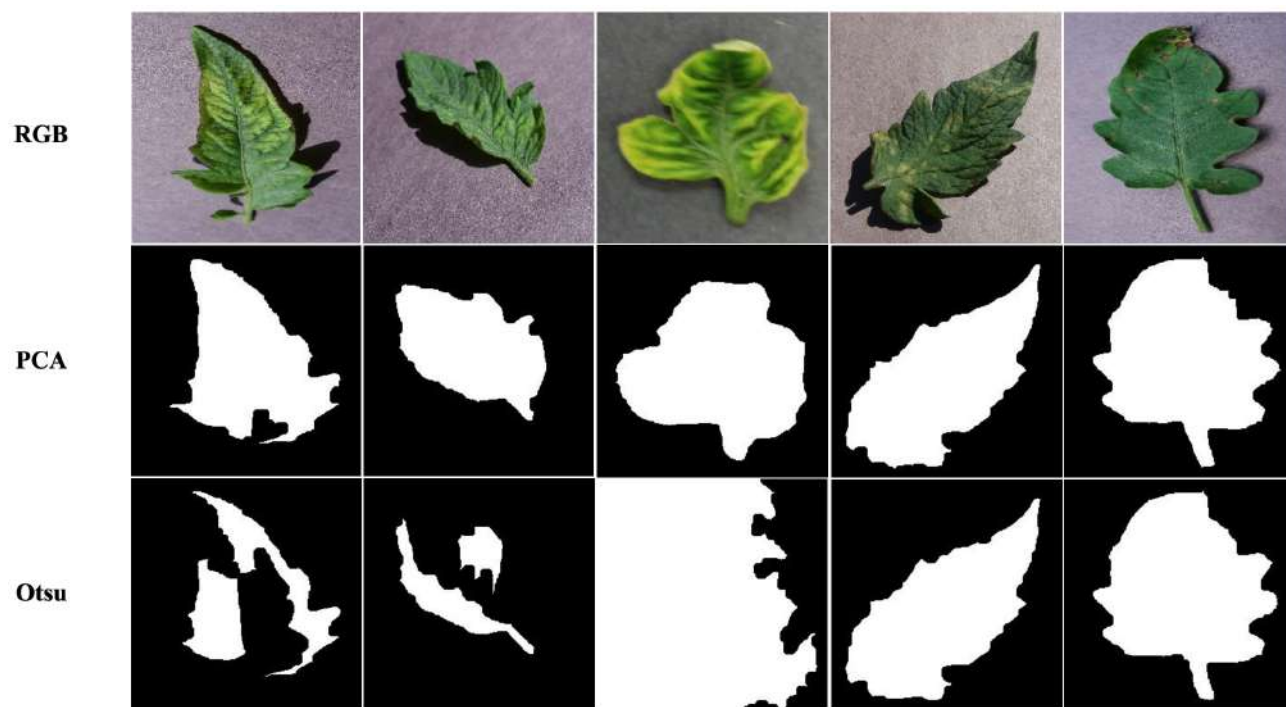


Figura 4.3: PCA y Otsu: Dos métodos usados en la segmentación de imágenes.

En esta primera etapa del método propuesto inicial, como se mencionó en el párrafo anterior, se implementaron dos técnicas computacionales para separar la hoja de su entorno, aunque el conjunto de datos usado tiene un fondo homogéneo, fue un reto la implementación de una segmentación automática, ya que se procesaron imágenes en un espacio de color RGB, sabiendo que hay mucha correlación entre sus componentes, además, de las sombras que existen en las

imágenes. En la Figura 4.3, se visualiza una comparación de los métodos de segmentación aplicados, de los cuales, se han considerado Otsu y PCA, fungiendo como el núcleo de esta etapa, así mismo, se reflejan 5 ejemplos, donde, se puede ver el desempeño cualitativo de cada técnica usada.

4.2.2. Extracción de características

La extracción de características en imágenes, es un proceso crítico que se debe tratar con mucho cuidado, ya que los algoritmos que operan sobre este módulo podrían extraer patrones que describan a una imagen de forma competente o en su defecto, se obtendrían patrones redundantes que no generalicen las clases de manera eficiente, impactando directamente en el rendimiento de los algoritmos de aprendizaje automático. Anteriormente las características extraídas de una imagen eran representadas por los píxeles de la misma, esto repercutía en la longitud del vector generado, siendo este muy grande y con un gasto de recursos computacionales alto. Las características extraídas en esta etapa se catalogan en dos tipos, características de textura y características cromáticas, sin embargo, las pruebas experimentales se hicieron de tres formas, con características de 1) textura, 2) cromáticas e, 3) híbridas (texturales/cromáticas).

Características de textura

Las características de textura, específicamente extraen información a través de la superficie de la hoja, donde, las técnicas implementadas buscan patrones muy peculiares como: superficies ásperas, rugosas, delicadas, uniformes, granuladas, entre otras, hallando descriptores particulares sobre una disposición espacial en el área previamente segmentada. La información extraída, se puede presentar a lo largo de la región de interés, por lo tanto, estos algoritmos funcionan aún cuando la textura ha sido desplazada. Para este fin se ha puesto en marcha el algoritmo de Haralick, que básicamente se implementa de la siguiente manera, en primera instancia se obtiene la matriz de coocurrencia, la cual mide como coocurren pares o vectores de píxeles, posteriormente, se calculan los 14 descriptores de Haralick. Los descriptores de Haralick se mencionan a continuación: segundo momento angular, contraste, correlación, suma de cuadrados, momento de diferencia inversa, suma promedio, suma de entropía, suma de varianzas, entropía, diferencia de varianzas, diferencia de entropías, medida de correlación I, medida de correlación II, y má-

ximo coeficiente de correlación.

De cada descriptor de Haralick aplicado, se obtiene media y rango del canal R, G, y B de la imagen integral, obteniendo un vector de 84 características de longitud. El vector de características texturales X_t puede ser representado de la siguiente manera:

$$X_t = [x_1, x_2, \dots, x_{84}] \quad (4.1)$$

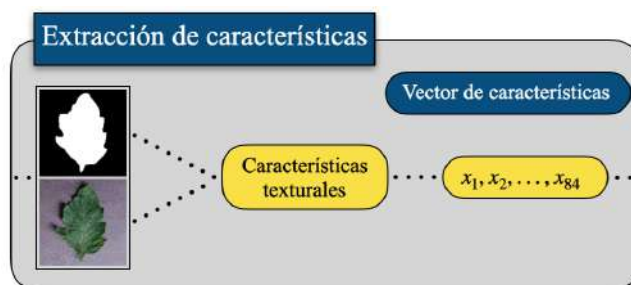


Figura 4.4: Extracción de características texturales.

En la Figura 4.4, se visualiza gráficamente como se extraen características texturales de una imagen en un espacio de color RGB. Cabe mencionar que, antes de la ejecución de esta etapa, las imágenes son segmentadas, así mismo, posterior a ello, el vector de características generado, entra al módulo de clasificación.

Características cromáticas

Las características cromáticas proveen mucha información partiendo de imágenes en un espacio de color específico, en nuestro caso, obtenemos descriptores a través de tres canales primarios del espacio de color RGB, buscando cambios de intensidad o tonalidad. Los algoritmos usados en esta sección se nombran a continuación, así como el número de características obtenidas de cada uno para las tres componentes R, G y B: los descriptores de Contraste en su cálculo en el área segmentada, han obtenido 15 descriptores, 5 para cada componente; las características de Gabor obtienen 201 descriptores, 67 para cada componente; los momentos de Hu obtienen 21 descriptores, 7 para cada componente; la transformada discreta del coseno (TDC) ha obtenido 12 características, 4 para cada componente; y finalmente los descriptores de Fourier han obtenido 24 descriptores, 8 para cada componente. El vector generado con estas cinco técnicas

de extracción de características es de 273 descriptores, representado por el vector X_c , como se muestra a continuación:

$$X_c = [x_1, x_2, \dots, x_{273}] \quad (4.2)$$

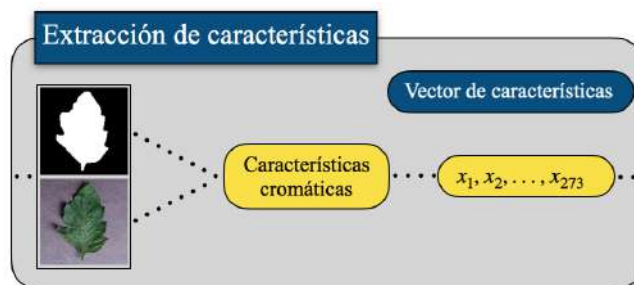


Figura 4.5: Extracción de características cromáticas.

En la Figura 4.5, se visualiza gráficamente como se extraen características cromáticas de una imagen en un espacio de color RGB. Cabe mencionar que, antes de la ejecución de esta etapa, las imágenes son segmentadas, así mismo, posterior a ello, el vector de características generado, entra al módulo de clasificación.

Características híbridas

El objetivo de unir las características texturales y cromáticas, fue para probar el rendimiento y aporte de los dos tipos de descriptores considerados en esta investigación. En los experimentos desarrollados con las características híbridas, se hizo una concatenación del vector X_t representando a las características de textura, y el vector X_c representando a las características cromáticas, obteniendo un nuevo vector con una longitud de 357 descriptores, representado por $X_h = X_t + X_c$. El vector resultante se describe a continuación.

$$X_h = [x_1, x_2, \dots, x_{357}] \quad (4.3)$$

En la Figura 4.6, se visualiza gráficamente como se extraen características híbridas de una imagen en un espacio de color RGB. Cabe mencionar que, antes de la ejecución de esta etapa, las imágenes son segmentadas, así mismo, posterior a ello, el vector de características generado, entra al módulo de clasificación.

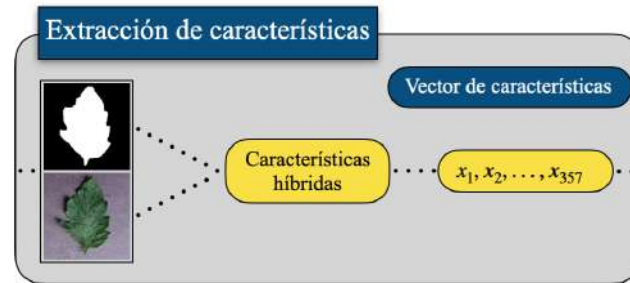


Figura 4.6: Extracción de características híbridas.

4.2.3. Clasificación

Posterior a la generación de los descriptores, producto de la implementación de los algoritmos de extracción de características usados en esta investigación, los vectores generados en la etapa anterior representan las entradas a este módulo, probando el rendimiento frente a los algoritmos de aprendizaje automático, de los cuales figuran los siguientes: máquinas de vectores soporte (SVM), algoritmo backpropagation para redes neuronales artificiales (RNA), naive bayes (NB), K-vecinos más cercanos (KNN), y bosques aleatorios (BA), discriminando entre las diez clases del conjunto de datos Plantvillage.

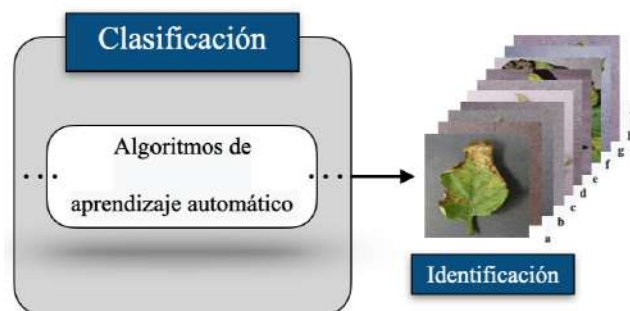


Figura 4.7: Clasificación del método propuesto inicial.

En la Figura 4.7, se visualiza gráficamente el proceso de clasificación con el método propuesto inicial. Cabe mencionar que, antes de la ejecución de esta etapa, se extraen las características de textura, cromáticas, e híbridas, así mismo, posterior a ello, son identificadas las clases, midiendo el rendimiento del modelo construido con las métricas de desempeño definidas en la Sección 3.7.

4.3. Módulo de preprocesamiento

Hasta este punto de la investigación, se han desarrollado algunas pruebas experimentales, las cuales, prueban el desempeño y nivel de efectividad del método propuesto inicial, considerando tres etapas. En esta sección, se dan a conocer los detalles del módulo de preprocesamiento aplicado a todas las imágenes del conjunto de datos, este módulo se ha implementado antes de la etapa de segmentación de la Figura 4.2, posteriormente, la metodología sufre un cambio, actualizando la estructura modular, como se visualiza en la Figura 4.8. Producto de este ajuste, las cuatro etapas resultantes se nombran y se ejecutan de la siguiente manera: 1) preprocesamiento, 2) segmentación, 3) extracción de características, y 4) clasificación. El núcleo de este apartado, es una transformación de un espacio de color a otro, con el objetivo de tener una segmentación más prominente. En las siguientes subsecciones, se dan a conocer detalles del preprocesamiento aplicado. En la Figura 4.8, se muestra la metodología actualizada, así mismo, se realizan pruebas experimentales con esta estructura.

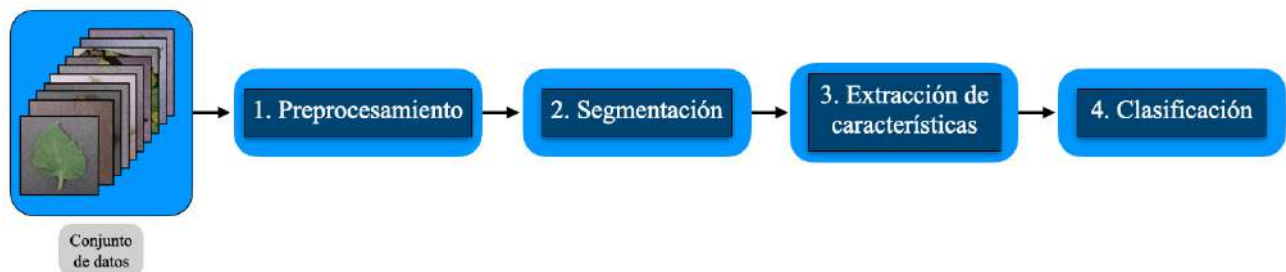


Figura 4.8: Método propuesto con preprocesamiento.

Antes de decidir el espacio de color con el que se iba a trabajar, se realizaron algunas pruebas con distintas técnicas, visualizando el desempeño cualitativo de cada una de ellas. En la Figura 4.9, se visualiza el preprocesado aplicado a imágenes en formato RGB, transformándolas a diferentes espacios de color, como: HSV, YCbCr, y YIQ, las dimensiones de las imágenes se mantuvieron en el formato original, el cual es de 256×256 píxeles. En la etapa 1 de la Figura 4.8, se preprocesan las imágenes del conjunto de datos, que consiste en una transformación del espacio de color RGB al espacio de color $L^*a^*b^*$. La intensidad de los diferentes componentes del espacio de color RGB, determina tanto el tono como el brillo, este formato es ideal para desplegar colores en sistemas electrónicos como, televisión y equipos de adquisición de imágenes,

sin embargo, no es el más adecuado para el procesamiento de imágenes en color, debido a la alta correlación entre sus canales. Por esta razón, en este módulo, se ha usado el espacio de color $L^*a^*b^*$, el cual está definido por tres variables que lo componen, L^* es la intensidad, a^* y b^* los componentes de tonalidad. La ubicación de este espacio de color es similar al espacio RGB, pero la posición de las variables es distinta. Además, la intensidad se desacopla de la cromaticidad, imitando la forma en que los humanos perciben los colores.

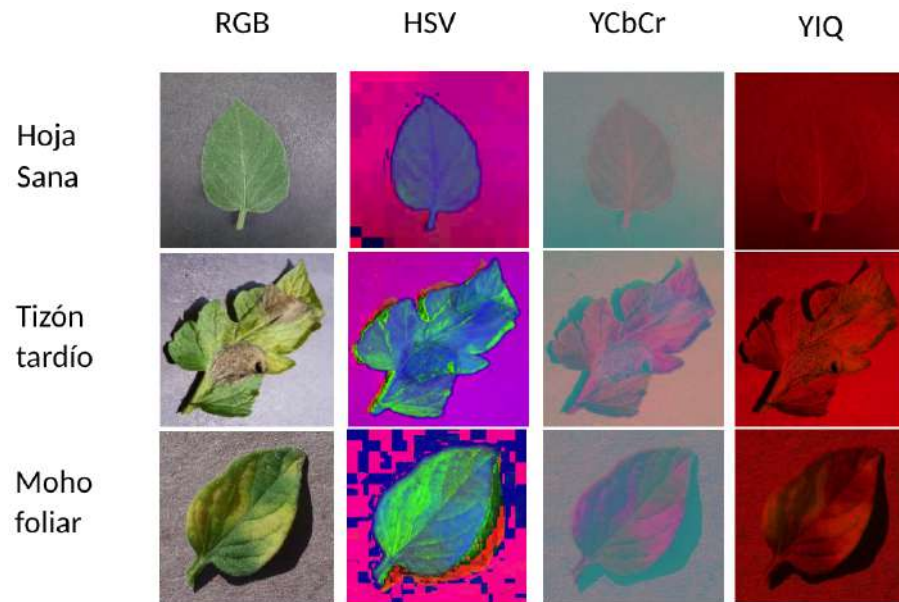


Figura 4.9: Preprocesamiento de un espacio de color RGB a otros.

Tras el preprocesado de la etapa 1 de la Figura 4.8, se ha extraído el área de interés de las imágenes del conjunto de datos, bajo la implementación del algoritmo de análisis de componentes principales (PCA), separando la hoja de su entorno. Posteriormente, se extraen características de textura, cromáticas, e híbridas (texturales/cromáticas) a partir de la región segmentada, generando los vectores de características. Finalmente, en el módulo de clasificación, el algoritmo de aprendizaje automático máquinas de vectores soporte (SVM), el algoritmo backpropagation para redes neuronales artificiales (RNA), K-vecinos más cercanos (KNN), bosques aleatorios (BA), y regresión logística (RL), toman como entrada los vectores generados de la etapa anterior, discriminando entre las diez clases del conjunto de datos Plantvillage, así mismo, se ejecutan las pruebas experimentales pertinentes. En la Figura 4.10, se visualizan dos ejemplos para dos hojas distintas, haciendo una comparación cualitativa en la forma que segmentaba

antes del preprocesamiento implementado y posterior a ello; en la subimagen a), se aplica una segmentación con el algoritmo PCA sin preprocesamiento previo, por otro lado, en la subimagen b) se segmentó nuevamente con el algoritmo PCA, pero con un preprocesamiento previo, a través de una transformación de un espacio de color a otro, obteniendo una segmentación más prominente, así mismo, se concluye que, el método de segmentación implementado tiene un mejor desempeño al aplicar una etapa de preprocesamiento previa.

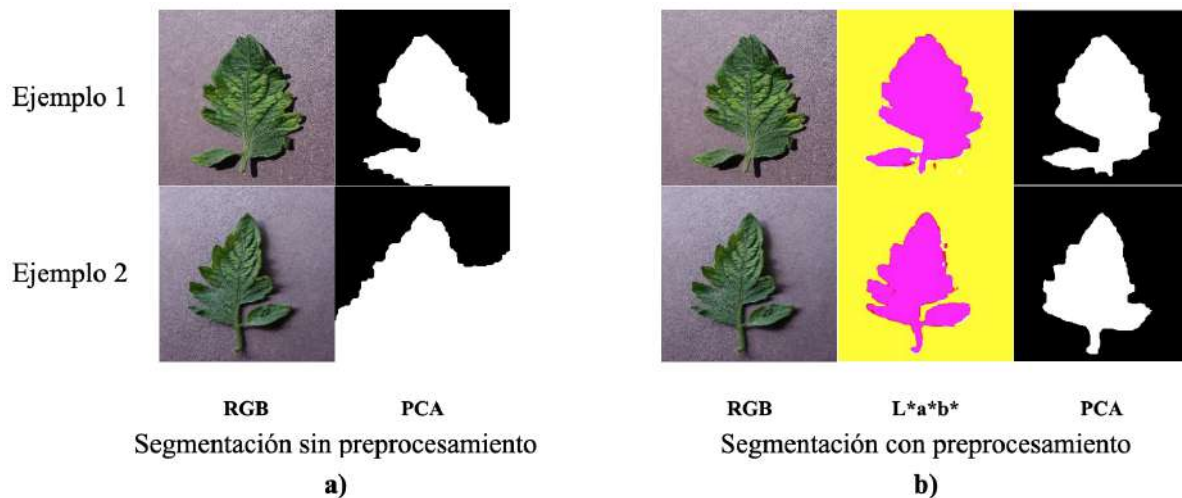


Figura 4.10: Segmentación con preprocesamiento.

4.4. Módulo de equilibrio de clases

En la Sección 4.1, se define el conjunto de datos usado en esta investigación, en la cual se consideran diez clases distintas, así mismo, analizando esta información, se puede visualizar una proporción asimétrica entre clases, donde la clase con menor número de imágenes pertenece a la enfermedad virus del mosaico del tomate con 373 imágenes (clase minoritaria) y la clase con mayor número de imágenes pertenece a la enfermedad virus del enrollamiento de la hoja amarilla del tomate con 5357 imágenes (clase mayoritaria). Por ende, se ha implementado un módulo de equilibrio de clases partiendo del método propuesto de la Figura 4.8, desarrollando esta etapa entre los módulos extracción de características y clasificación. Resultado de esta adición modular, se ha conseguido una metodología con cinco etapas, nombradas de forma secuencial con base en su ejecución, 1) preprocesamiento, 2) segmentación, 3) extracción de características,

4) equilibrio de clases, y 5) clasificación, ver Figura 4.11.

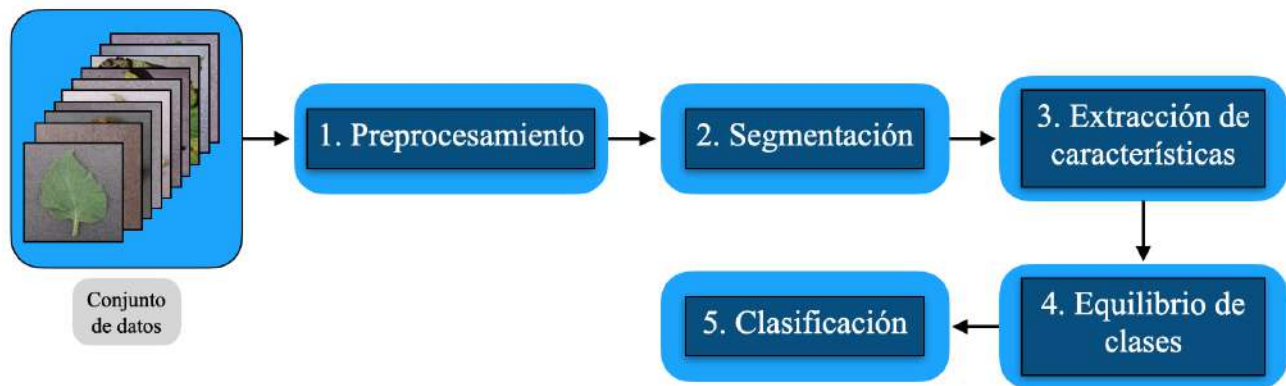


Figura 4.11: Método propuesto con equilibrio de clases

Los métodos de equilibrio de clases usados en esta sección, recaen en dos apartados, técnicas no heurísticas, considerando submuestreo aleatorio, Sección 3.5.1 y sobremuestreo con réplicas, Sección 3.5.2, y técnicas heurísticas, implementando el algoritmo SMOTE, Sección 3.5.3. Posterior al equilibrio de clases descrito en esta sección, la metodología resultante se puede observar en la Figura 4.11, con esta actualización del método propuesto, nuevamente se realizaron las pruebas experimentales pertinentes, donde, las imágenes se preprocesan transformándolas de un espacio de color a otro, después las imágenes son segmentadas con el algoritmo PCA, en la etapa de extracción de características se consideraron descriptores cromáticos solamente; como técnicas de equilibrio de clases se implementaron las mencionadas en esta sección, y finalmente, se probó el desempeño con los algoritmos nombrados a continuación: SVM, RNA, KNN, y BA. Para equilibrar las clases con la técnica de submuestreo aleatorio, inicialmente, se localiza la clase con menor número de instancias (clase minoritaria), que en nuestro caso está representada con la enfermedad virus del mosaico del tomate con 373 instancias, vinculada a la clase (a), posteriormente, se localiza el resto de las clases, de las cuales se consideran (b, c, d, e, f, g, h, i, j). Después de haber identificado la clase minoritaria de sus contrapartes; las clases (b, c, d, e, f, g, h, i, j) se submuestran eligiendo 373 instancias de cada una de forma aleatoria de la distribución del número de instancias original (Tabla 4.1), resultando un conjunto de datos simétrico con un total 3730 instancias, ver Tabla 4.2.

Tabla 4.2: Distribución de imágenes del conjunto de datos submuestreado.

Clase	Nombre común	Número de instancias
a	Virus del mosaico del tomate	373
b	Moho de la hoja	373
c	Tizón temprano	373
d	Mancha foliar	373
e	Hojas sanas	373
f	Araña roja de dos puntos	373
g	Mancha foliar por septoria	373
h	Tizón tardío	373
i	Mancha bacteriana	373
j	Virus del enrollamiento de la hoja amarilla del tomate	373

La secuencia de pasos para lograr el equilibrio de clases con la técnica de submuestreo aleatorio se muestra a continuación.

1. Identificar la clase minoritaria.
2. Identificar las clases mayoritarias.
3. Seleccionar de forma aleatoria la cantidad de instancias de la clase minoritaria en cada clase mayoritaria.
4. Conjunto de datos simétrico, ver Tabla 4.2 o ver Figura 4.12.

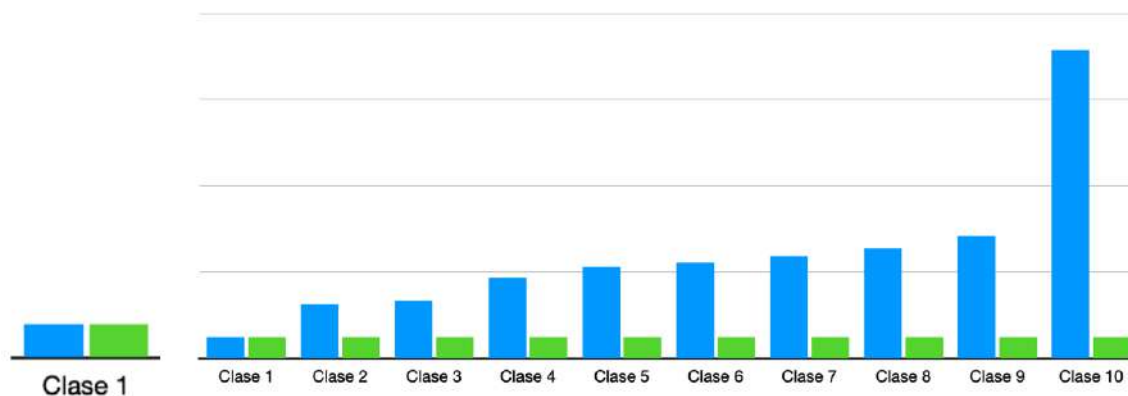


Figura 4.12: Submuestreo aleatorio.

Las técnicas de sobremuestreo que se usaron en esta investigación fue el sobremuestreo con réplicas, Sección 3.5.2, y el sobremuestreo con el algoritmo SMOTE, Sección 3.5.3. Así mismo, antes de implementarlos, el conjunto de datos se ha segmentado en dos secciones a partir del conjunto de datos original, la primera partición como conjunto de entrenamiento, representando el 70 % de cada una de las clases, y la segunda partición como conjunto de pruebas, representando el 30 % de cada una de las clases, ver Tabla 4.3.

Tabla 4.3: Partición del conjunto de datos.

Clase	Nombre común de la enfermedad o plaga	Partición de entrenamiento (70 %)	Partición de pruebas (30 %)
a	Virus del mosaico del tomate	261	112
b	Moho de la hoja	666	286
c	Tizón temprano	700	300
d	Mancha foliar	983	421
e	Hojas sanas	1114	477
f	Araña roja de dos puntos	1173	503
g	Mancha foliar por septoria	1240	531
h	Tizón tardío	1336	572
i	Mancha bacteriana	1489	638
j	Virus del enrollamiento de la hoja amarilla del tomate	3750	1607

Posterior al particionado del conjunto de datos, se sobremuestra sólo la partición que pertenece al conjunto de entrenamiento de las clases minoritarias (a, b, c, d, e, f, g, h, i), alcanzando una simetría con la clase mayoritaria (j), que pertenece al daño foliar virus del enrollamiento de la hoja amarilla del tomate, muestreando con el sobremuestreo con réplicas y con el algoritmo SMOTE.

La secuencia de pasos para lograr el equilibrio de clases con el método de sobremuestreo con réplicas y SMOTE se muestra a continuación.

1. Particionar el conjunto de datos en dos secciones, subconjunto de entrenamiento y subconjunto de prueba.
2. Identificar la clase mayoritaria en el subconjunto de entrenamiento.
3. Identificar las clases minoritarias en el subconjunto de entrenamiento.
4. Muestrear sólo el subconjunto de entrenamiento.

5. Realizar copias de instancias de las clases minoritarias hasta alcanzar una simetría con la clase mayoritaria en el caso de sobremuestreo con réplicas / creación de instancias sintéticas o artificiales hasta alcanzar una simetría con la clase mayoritaria en el caso del algoritmo SMOTE.
6. Conjunto de datos simétrico, ver Figura 4.13.

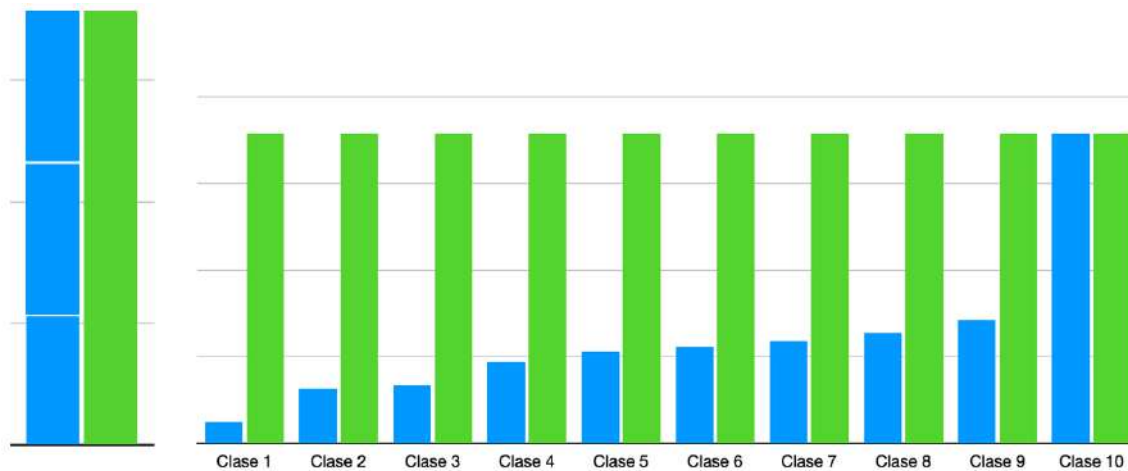


Figura 4.13: Sobremuestreo con réplicas y sobremuestreo con el algoritmo SMOTE.

4.5. Influencia de técnicas de extracción de características

Como ya se ha comentado en el Capítulo 2, la importancia de los algoritmos de extracción de características tanto texturales como cromáticos, en este apartado, se realiza un análisis de la influencia de los distintos métodos de extracción de características implementados en esta investigación, impactando en el desempeño de los algoritmos de aprendizaje automático utilizados.

Tomando como base el último despliegue metodológico representado por la Figura 4.11, se han realizado nuevamente pruebas experimentales, haciendo énfasis en la etapa tres, donde, se extrajeron características de manera individual con los distintos algoritmos definidos en la Sección 3.4, considerando: Haralick, descriptores de Contraste, momentos de Hu, descriptores

de Fourier, TDC y características de Gabor, analizando la influencia que tiene cada uno de ellos en la clasificación. Para realizar los experimentos, hemos considerado los siguientes algoritmos para cada una de las etapas, quedando de la siguiente manera: en la etapa de preprocesamiento, se considero una transformación de RGB a $L^*a^*b^*$; para segmentar imágenes, se ha considerado el algoritmo PCA; en la etapa de extracción de características se implementó el proceso descrito en esta sección; posteriormente, en la etapa de equilibrio de clases se contemplo el algoritmo SMOTE; y finalmente, en la etapa de clasificación se han considerado los siguientes algoritmos de aprendizaje automático: SVM, RNA, KNN, BA.

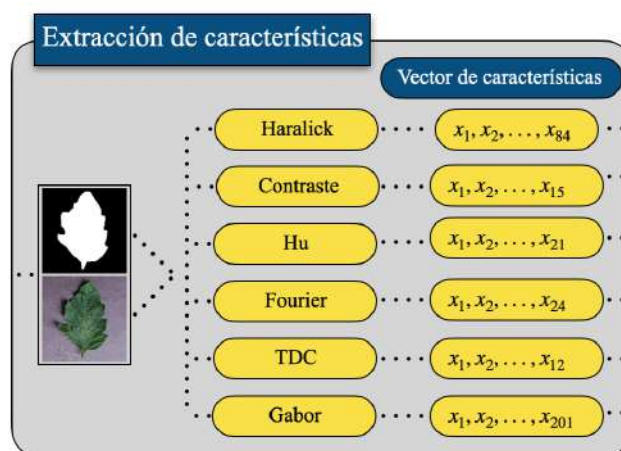


Figura 4.14: Extracción de características de las técnicas usadas

En la Figura 4.14, se visualiza gráficamente el proceso de extracción de características de forma independiente de cada una de las técnicas usadas en esta investigación. Cabe mencionar que, antes de la ejecución de esta etapa, las imágenes del conjunto de datos son segmentadas funcionando como el núcleo del módulo el algoritmo PCA, así mismo, posterior a ello, se implementa el algoritmo SMOTE para el equilibrio del conjunto de datos.

Capítulo 5

Resultados experimentales

En este capítulo, se visualizan los resultados de las pruebas experimentales de esta investigación, así mismo, los resultados se expresan en forma de tabla, matrices de confusión, y en gráficas con diagramas de caja, en el caso de las gráficas con diagramas de caja, se ha asignando un color específico según el tipo de prueba realizada. Para las pruebas de la Sección 5.2, se ha asignado el color verde en los diagramas de caja, para los experimentos de la Sección 5.3, se ha definido el color azul, y finalmente para las pruebas de la Sección 5.4, se ha asignado el color rojo.

5.1. Resultados del método propuesto inicial

Los resultados que se plasman en esta sección, son producto de experimentos con el método propuesto inicial, expresado en la Figura 4.2. Los resultados de este apartado, difieren de la implementación del algoritmo aplicado en la etapa 1 de la metodología, considerando el algoritmo Otsu y PCA; en los módulos posteriores, se han considerado los siguientes métodos, en la etapa extracción de características, se incluyeron descriptores de textura, cromáticos, e híbridos (texturales/cromáticos); finalmente, en la etapa de clasificación, se probó el rendimiento del sistema con NB, KNN, BA, RNA, y SVM. A continuación se describen los resultados.

Tabla 5.1: Resultados de exactitud con la técnica de segmentación Otsu en la etapa 1.

	Texturales	Cromáticas	Texturales/Cromáticas
NB	37.86 %	38.26 %	40.37 %
RNA	81.44 %	87.69 %	82.21 %
KNN	72.39 %	74.59 %	79.66 %
BA	76.17 %	79.67 %	81.95 %
SVM	88.87 %	91.73 %	93.46 %

5.1.1. Resultados con la técnica de segmentación Otsu

En la Tabla 5.1, se visualizan los resultados con la implementación del algoritmo Otsu en la etapa de segmentación, los métodos usados en la etapa de extracción de características y en la etapa de clasificación se han descrito en el párrafo anterior. Así mismo, los resultados reportados, fueron obtenidos con base en la métrica de exactitud, definida en la Sección 3.7, ecuación (3.37). El algoritmo con mejor rendimiento para esta prueba experimental, fue el clasificador SVM, obteniendo una exactitud para características de textura del 88.87 %, para características cromáticas un 91.73 % y para características híbridas (texturales/cromáticas) un 93.46 %. Además, en la Tabla 5.2, se despliegan resultados del desempeño de precisión de cada clase del conjunto de datos, definido en la Tabla 4.1, frente a los algoritmos de aprendizaje automático.

Tabla 5.2: Resultados de precisión con la técnica de segmentación Otsu aplicada en la etapa 1.

Clase	Texturales					Cromáticas					Texturales/Cromáticas				
	NB	RNA	KNN	BA	SVM	NB	RNA	KNN	BA	SVM	NB	RNA	KNN	BA	SVM
a	0.111	0.616	0.573	0.716	0.823	0.115	0.769	0.510	0.856	0.855	0.117	0.709	0.639	0.851	0.901
b	0.120	0.749	0.598	0.761	0.834	0.238	0.840	0.718	0.779	0.852	0.194	0.702	0.759	0.835	0.884
c	0.159	0.549	0.485	0.575	0.726	0.131	0.706	0.524	0.671	0.793	0.141	0.626	0.577	0.692	0.831
d	0.287	0.752	0.561	0.637	0.815	0.327	0.788	0.572	0.663	0.830	0.345	0.734	0.636	0.702	0.881
e	0.778	0.932	0.884	0.903	0.956	0.711	0.946	0.889	0.911	0.972	0.732	0.902	0.927	0.927	0.981
f	0.162	0.762	0.591	0.685	0.867	0.235	0.830	0.602	0.725	0.906	0.223	0.757	0.675	0.738	0.929
g	0.380	0.766	0.687	0.722	0.863	0.605	0.838	0.703	0.734	0.895	0.594	0.838	0.764	0.760	0.903
h	0.444	0.726	0.635	0.701	0.831	0.300	0.803	0.708	0.770	0.883	0.339	0.792	0.758	0.790	0.893
i	0.391	0.851	0.752	0.795	0.902	0.446	0.919	0.761	0.808	0.942	0.449	0.809	0.834	0.840	0.954
j	0.791	0.912	0.852	0.813	0.961	0.929	0.959	0.870	0.861	0.978	0.926	0.905	0.893	0.872	0.982

5.1.2. Resultados con la técnica de segmentación PCA

En la Tabla 5.3, se visualizan los resultados con la implementación del algoritmo PCA en la etapa de segmentación, los métodos usados en la etapa de extracción de características y

Tabla 5.3: Resultados de exactitud con la técnica de segmentación PCA en la etapa 1.

	Texturales	Cromáticas	Texturales/Cromáticas
NB	39.12 %	44.76 %	46.13 %
RNA	81.81 %	88.27 %	81.46 %
KNN	73.57 %	76.1 %	80.23 %
BA	77.27 %	80.77 %	82.77 %
SVM	89.81 %	92.71 %	93.86 %

en la etapa de clasificación se han descrito en párrafos anteriores. Así mismo, los resultados reportados, fueron obtenidos con base en la métrica de exactitud, definida en la Sección 3.7, ecuación (3.37). El algoritmo que mejor se desempeñó fue el clasificador SVM, obteniendo una exactitud para características de textura de 89.81 %, para características cromáticas un 92.71 %, y para características híbridas (texturales/cromáticas) un 93.86 %. Además, en la Tabla 5.4, se despliegan resultados del desempeño de precisión de cada clase del conjunto de datos, definido en la Tabla 4.1, frente a los algoritmos de aprendizaje automático.

Tabla 5.4: Resultados de precisión con la técnica de segmentación PCA en la etapa 1.

Clase	Texturales					Cromáticas					Texturales/Cromáticas				
	NB	RNA	KNN	BA	SVM	NB	RNA	KNN	BA	SVM	NB	RNA	KNN	BA	SVM
a	0.111	0.682	0.579	0.733	0.819	0.091	0.794	0.570	0.846	0.852	0.113	0.540	0.656	0.865	0.877
b	0.095	0.739	0.625	0.759	0.873	0.567	0.825	0.721	0.832	0.903	0.456	0.746	0.776	0.853	0.914
c	0.180	0.644	0.486	0.608	0.738	0.104	0.705	0.535	0.687	0.795	0.133	0.651	0.612	0.701	0.805
d	0.297	0.729	0.556	0.653	0.837	0.368	0.797	0.590	0.673	0.836	0.364	0.732	0.634	0.704	0.890
e	0.837	0.937	0.911	0.915	0.969	0.712	0.947	0.922	0.932	0.979	0.781	0.901	0.935	0.933	0.984
f	0.199	0.761	0.616	0.695	0.869	0.340	0.831	0.620	0.714	0.909	0.268	0.739	0.660	0.752	0.916
g	0.382	0.739	0.718	0.738	0.875	0.590	0.887	0.733	0.777	0.931	0.551	0.718	0.776	0.797	0.933
h	0.440	0.716	0.656	0.692	0.842	0.460	0.806	0.739	0.764	0.883	0.487	0.736	0.761	0.772	0.909
i	0.370	0.842	0.753	0.800	0.899	0.427	0.925	0.768	0.819	0.952	0.432	0.826	0.828	0.851	0.955
j	0.806	0.915	0.859	0.829	0.967	0.918	0.959	0.875	0.867	0.982	0.920	0.929	0.904	0.879	0.984

En la gráfica de la Figura 5.1, se observan los mejores resultados de esta sección, comparando dos métodos de segmentación, visualizando en el eje x los algoritmos de aprendizaje automático probados; y en el eje y se muestra el porcentaje de exactitud que alcanza cada uno de los clasificadores utilizados, esta gráfica es producto de la experimentación con técnicas de extracción de características híbridas (texturales/cromáticas). La exactitud más alta, fue obtenida por el método de segmentación PCA, excepto con el clasificador RNA. La diferencia del porcentaje de exactitud obtenida, a través de las técnicas de segmentación Otsu y PCA, son mínimas, concluyendo que, los dos métodos de segmentación son efectivos para la metodología propuesta.

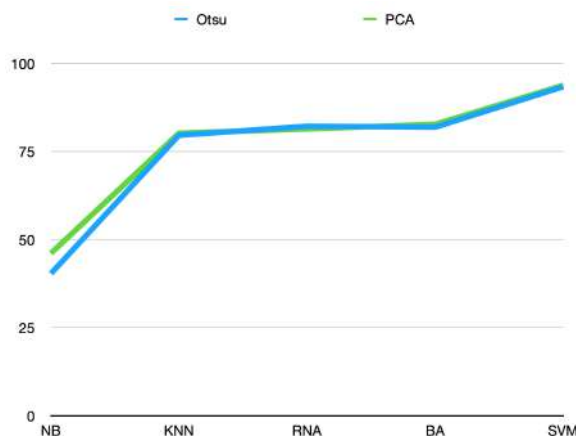


Figura 5.1: Gráfica de resultados con dos métodos de segmentación.

5.2. Resultados con el módulo de preprocesamiento

Los resultados que se plasman en esta sección, son producto de experimentos con el método propuesto junto al módulo de preprocesamiento, expresado en la Figura 4.8. En la etapa 1 preprocesamiento, es aplicado una transformación a las imágenes de entrada, cambiándolas de un espacio de color RGB a un espacio de color $L^*a^*b^*$; posteriormente, en la etapa de segmentación, es ejecutado el algoritmo PCA; como técnicas de extracción de características se implementaron descriptores de textura, cromáticos, e híbridos (texturales/cromáticos); finalmente, en la etapa de clasificación, se probó el rendimiento del sistema con RL, KNN, BA, RNA, y SVM. A continuación se describen los resultados. En la Tabla 5.5, se muestran resultados de los algoritmos de aprendizaje automático, evaluando los vectores extraídos con las técnicas de extracción de características mencionadas anteriormente. Para esta prueba, el algoritmo con menor rendimiento fue KNN, con 82.67% para las características cromáticas y para la características híbridas (texturales/cromáticas) un 84.13%; además, para la prueba con características texturales, superó solamente al clasificador RL con 74.95%. El clasificador KNN y RL tuvieron un comportamiento muy similar en su desempeño. El tercer mejor algoritmo para esta prueba fue BA, obteniendo un 77.91% para características texturales, para características cromáticas y características híbridas (texturales/cromáticas), se ha superado el 85% de exactitud. Uno de los clasificadores con mejor desempeño para esta prueba fueron las RNA usando el algoritmo de aprendizaje backpropagation, con un porcentaje de exactitud del 81.83%

Tabla 5.5: Resultados de exactitud con el módulo de preprocesamiento.

	Texturales	Cromáticas	Texturales/Cromáticas
KNN	74.95 %	82.67 %	84.13 %
RL	73.95 %	83.95 %	86.05 %
BA	77.91 %	85.12 %	86.63 %
RNA	81.83 %	90.65 %	83.76 %
SVM	89.40 %	93.69 %	94.46 %

para las características texturales; para las características cromáticas un 90.65 %; y finalmente para las características híbridas (texturales/cromáticas) un rendimiento del 83.76 %. El mejor desempeño lo obtuvo el algoritmo SVM, con una exactitud de 89.40 % para características texturales; para características cromáticas un 93.69 %, y finalmente, para características híbridas (texturales/cromáticas), se demostró un 94.46 %. En la Tabla 5.5, los mejores resultados se han obtenido con base en los descriptores híbridos (texturales/cromáticas), excepto con la RNA, ya que el mejor desempeño se ha obtenido con características cromáticas. Además, en los diagramas de caja de la Figura 5.2, es notable, que el rendimiento de las características híbridas (texturales/cromáticas) supera considerablemente las características de textura, y ligeramente las características cromáticas.

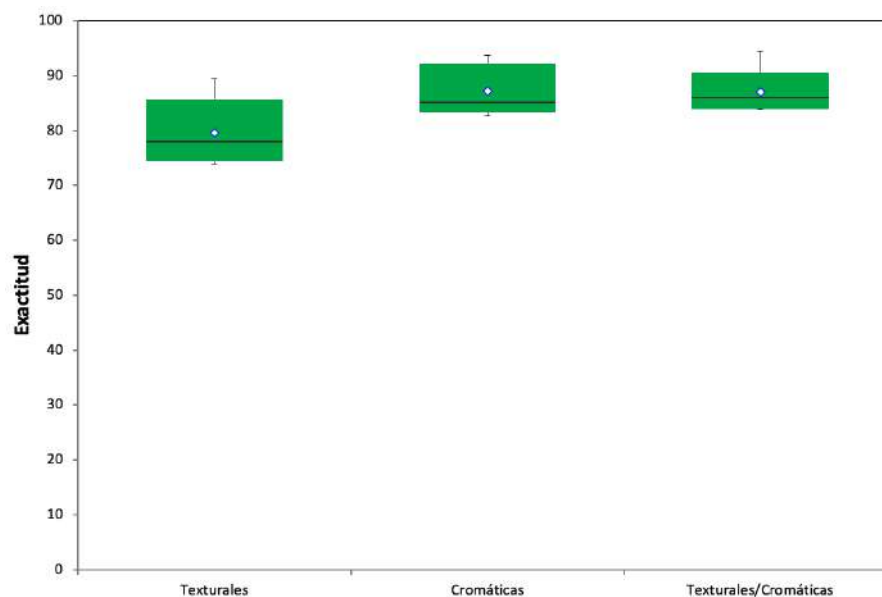


Figura 5.2: Desempeño de exactitud de técnicas de extracción de características.

En la Tabla 5.6, se visualizan las precisiones de cada una de las clases del conjunto de datos

Tabla 5.6: Resultados de precisión con el módulo de preprocesamiento.

Clase	KNN	RL	BA	RNA	SVM
a	0.863	0.798	0.942	0.903	0.920
b	0.794	0.808	0.875	0.884	0.903
c	0.674	0.652	0.721	0.721	0.805
d	0.700	0.772	0.787	0.848	0.907
e	0.949	0.955	0.946	0.963	0.988
f	0.949	0.821	0.803	0.871	0.932
g	0.792	0.794	0.836	0.869	0.935
h	0.813	0.759	0.786	0.838	0.905
i	0.839	0.895	0.887	0.944	0.967
j	0.928	0.952	0.927	0.976	0.989

usado, donde, la clase con el resultado más alto pertenece a la clase j , con un porcentaje de precisión de 0.989%, así mismo, esta tabla se expresa gráficamente en la Figura 5.3. En la Figura 5.3, la gráfica de caja del algoritmo KNN y RL, tienen los rangos de datos más grandes, donde los valores están más dispersos o separados que sus contrapartes; asimismo, los rangos de datos de precisión del algoritmo BA y RNA están moderadamente más concentrados que el algoritmo KNN y RL; finalmente, el algoritmo con mejor desempeño fue el clasificador SVM, ya que los datos están más concentrados en comparación con los clasificadores restantes.

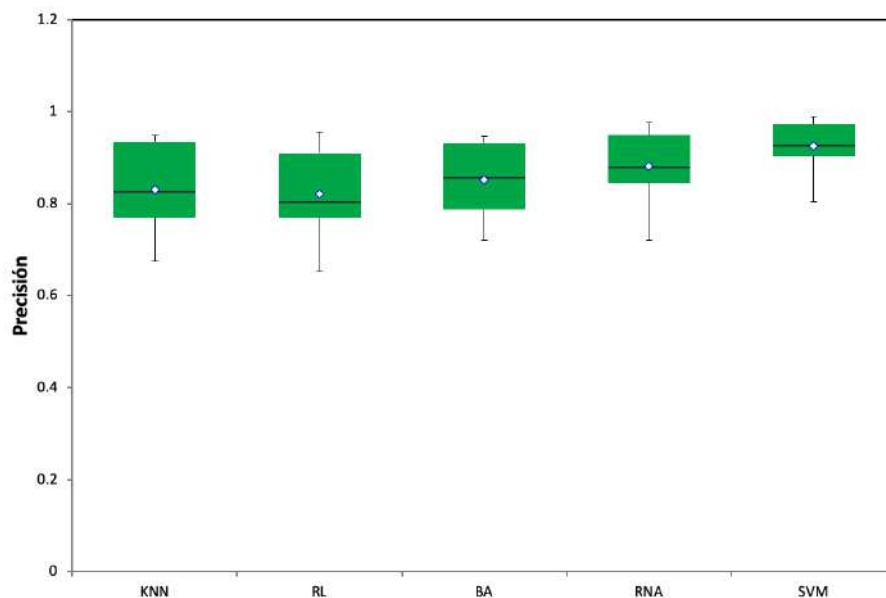


Figura 5.3: Gráfica de diagramas de caja con métodos de aprendizaje automático.

En la Figura 5.4, se grafican las exactitudes de los clasificadores considerados en estas prue-

bas, midiendo las técnicas de extracción de características utilizadas. El algoritmo más deficiente fue KNN, donde las exactitudes obtenidas son más dispersas en comparación con el resto de algoritmos. El clasificador con mejor desempeño fue SVM, ya que los datos están más concentrados en comparación con sus contrapartes.

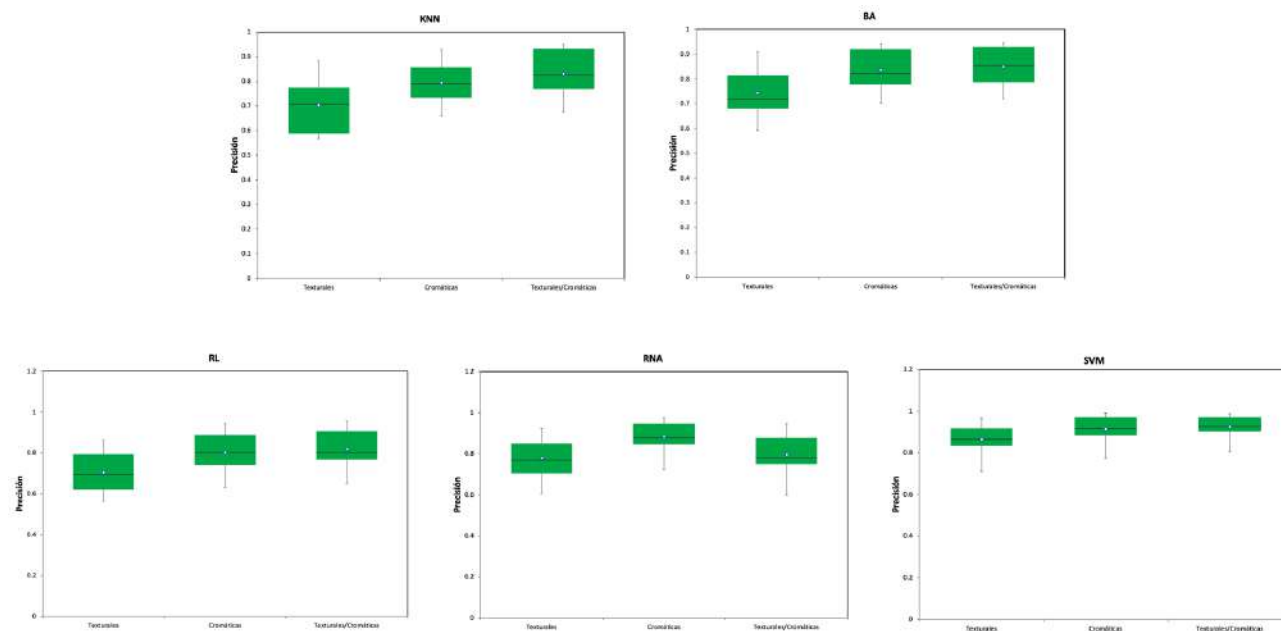


Figura 5.4: Desempeño de exactitud de técnicas de extracción de características por clasificador.

En la Tabla 5.7, se visualiza una matriz de confusión, considerando el mejor desempeño del método propuesto junto al módulo de preprocesamiento, destacando el algoritmo SVM, además, se realiza un análisis de confusión entre clases. La nomenclatura de las diez clases consideradas se define en la Sección 4.1. El análisis por clase se expresa textualmente de la siguiente manera: para la clase a, el algoritmo se ha confundido más con la clase (g); para la clase b, la mayor confusión fue con la clase (h); para la clase c, la mayor confusión fue con la clase (h); para la clase d, la mayor confusión fue con la clase (f); para la clase e, la mayor confusión fue con la clase (d); para la clase f, la mayor confusión fue con la clase (d); para la clase g, la mayor confusión fue con la clase (b); para la clase h, la mayor confusión fue con la clase (c); para la clase i, la mayor confusión fue con la clase (c); y para la clase j, la mayor confusión fue con la clase (i).

Tabla 5.7: Matriz de confusión del algoritmo SVM.

	a	b	c	d	e	f	g	h	i	j
a	323	2	0	1	0	3	7	1	0	1
b	3	823	15	8	1	3	17	19	2	2
c	4	13	791	15	1	11	12	83	8	11
d	2	7	14	1226	5	59	3	7	7	2
e	0	1	3	13	1372	2	1	5	1	0
f	6	5	10	53	1	1503	3	8	1	7
g	8	25	22	9	3	6	1456	18	14	8
h	5	27	91	12	4	11	36	1566	15	9
i	0	3	19	6	1	1	14	14	2005	18
j	0	5	17	9	0	14	8	10	20	5179

5.3. Resultados con el módulo de equilibrio de clases

Los resultados que se plasman en esta sección, son producto de experimentos con el método propuesto junto al módulo de equilibrio de clases, expresado en la Figura 4.11. Así mismo, en este apartado se ofrece una comparativa de tres métodos de equilibrio de clases. Las pruebas fueron ejecutadas de la siguiente manera: en la etapa 1 preprocesamiento, es aplicado una transformación a las imágenes de entrada, cambiándolas de un espacio de color RGB a un espacio de color $L^*a^*b^*$; posteriormente, en la etapa de segmentación, es ejecutado el algoritmo PCA; como extractores de características se implementaron descriptores cromáticos, vistos en la Sección 3.4.2; así mismo, en la etapa de equilibrio de clases, se aplicaron tres técnicas vistas en la Sección 3.5; finalmente, en la etapa de clasificación, se probó el rendimiento del sistema con el algoritmo KNN, BA, RNA, y SVM. A continuación se describen los resultados. En la Tabla 5.8, se visualiza el rendimiento de exactitud de métodos de aprendizaje automático frente a las técnicas de equilibrio de clases. El clasificador más competente fue el algoritmo SVM. Por otro lado, la técnica de equilibrio de clases más deficiente fue el submuestreo aleatorio.

Tabla 5.8: Desempeño de algoritmos de clasificación frente a técnicas de equilibrio de clases.

	Submuestreo	Sobremuestreo	SMOTE
KNN	71.44 %	78.90 %	78.97 %
BA	76.89 %	84.32 %	84.57 %
RNA	84.58 %	91.11 %	90.34 %
SVM	85.87 %	92.69 %	92.61 %

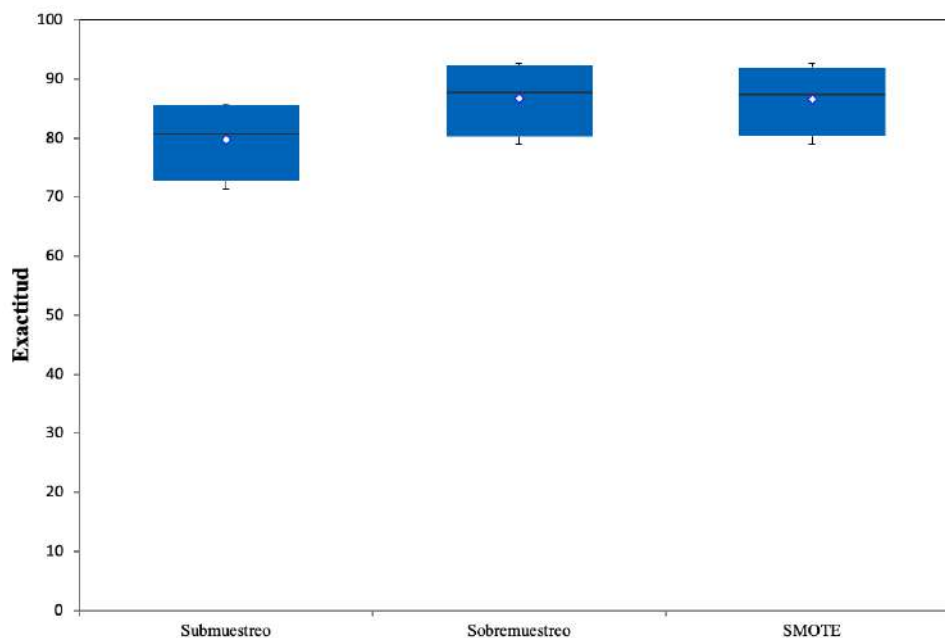


Figura 5.5: Exactitud de técnicas de equilibrio de clases.

La técnica de equilibrio de clases con mejor rendimiento fue obtenida con el submuestreo con réplicas, sin embargo, cabe mencionar que la técnica sobremuestreo con réplicas y SMOTE tienen un desempeño muy similar. En la Figura 5.5, se grafican los resultados de la Tabla 5.8. En la Tabla 5.9, se plasman resultados con la métrica de precisión, midiendo el rendimiento de cada una de las clases del conjunto de datos, frente a los clasificadores considerados en este apartado. Así mismo, las clases más discriminantes, están representadas por la clase *e* y la clase *f*. En la Figura 5.6, se grafican los resultados de la Tabla 5.9.

Tabla 5.9: Desempeño de precisión de los clasificadores.

Clase	KNN	BA	RNA	SVM
a	0.755	0.848	0.911	0.921
b	0.688	0.806	0.924	0.876
c	0.555	0.647	0.733	0.726
d	0.620	0.746	0.834	0.864
e	0.894	0.949	0.967	0.985
f	0.677	0.767	0.875	0.923
g	0.710	0.811	0.892	0.927
h	0.741	0.766	0.821	0.878
i	0.816	0.855	0.954	0.966
j	0.946	0.950	0.981	0.981

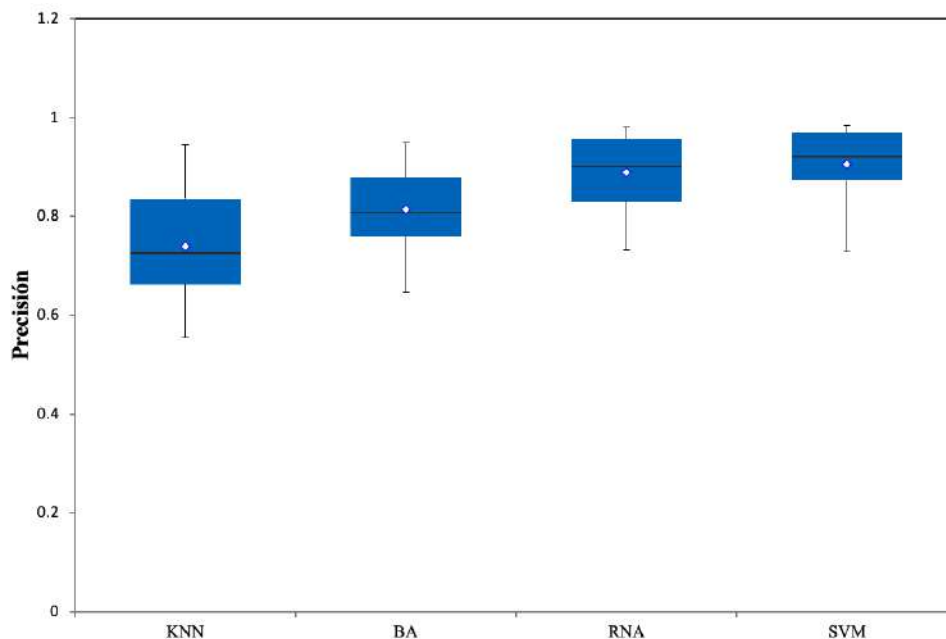


Figura 5.6: Desempeño de precisión por método de aprendizaje automático.

5.4. Resultados de la influencia de técnicas de extracción de características

Posterior a las pruebas experimentales ejecutadas en esta sección del trabajo de investigación, los resultados visualizados en la Tabla 5.10; fueron obtenidos bajo la implementación de la última actualización del método propuesto de la Figura 4.11, donde, se usaron algoritmos de aprendizaje automático frente a los vectores recuperados de las diferentes técnicas de extracción de características. Así mismo, cada clasificador usado, se han probado frente a los patrones recuperados de manera individual, considerando los descriptores de Contraste, características de Gabor, momentos de Hu, TDC, Haralick, y descriptores de Fourier, plasmando la exactitud para cada prueba realizada. Las pruebas fueron ejecutadas de la siguiente manera: en la etapa 1 preprocesamiento, es aplicado una transformación a las imágenes de entrada, cambiándolas de un espacio de color RGB a un espacio de color $L^*a^*b^*$; posteriormente, en la etapa de segmentación, es ejecutado el algoritmo PCA; como extractores de características se consideraron los vistos en la Sección 3.4 de manera individual; finalmente, en la etapa de clasificación, se probó el rendimiento con los clasificadores KNN, RNA, BA, y SVM. El mejor porcentaje obtenido, se ha

manifestado en el experimento llevado a cabo con el algoritmo SVM frente a las características de Gabor, con una exactitud de 90.87 %.

Tabla 5.10: Exactitud de algoritmos de aprendizaje automático frente a técnicas de extracción de características.

	Contraste	Gabor	Hu	TDC	Fourier	Haralick
KNN	55.29 %	74.06 %	36.62 %	57.29 %	41.94 %	72.41 %
RNA	54.36 %	73.16 %	33.11 %	60.20 %	63.45 %	72.90 %
BA	64.47 %	80.14 %	45.11 %	63.28 %	63.95 %	77.79 %
SVM	61.59 %	90.87 %	31.77 %	63.25 %	64.41 %	84.85 %

En la gráfica de la Figura 5.7, se puede visualizar la influencia de cada técnica de extracción de características implementada en esta investigación, donde, el método más influyente es representado por las características de Gabor; así mismo, la técnica más deficiente, corresponde a los siete momentos de Hu.

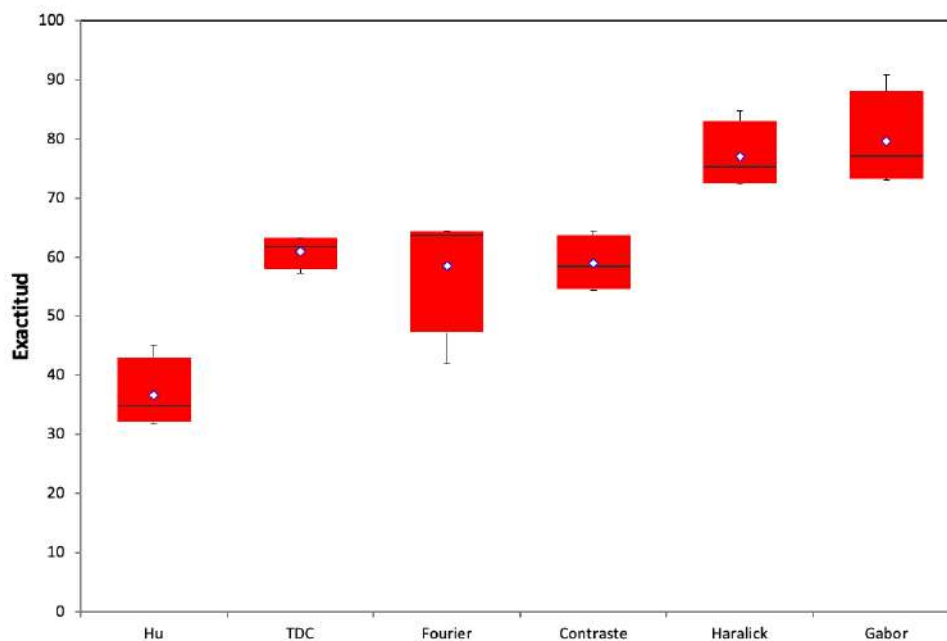


Figura 5.7: Desempeño de técnicas de extracción de características.

Por otro lado, en la gráfica de la Figura 5.8, se ilustra el desempeño de las técnicas de aprendizaje automático, esquematizando la exactitud a través de diagramas de caja, donde, el mejor rendimiento lo ha obtenido el diagrama de caja perteneciente al algoritmo SVM, así mismo, el clasificador más deficiente fue para el algoritmo KNN.

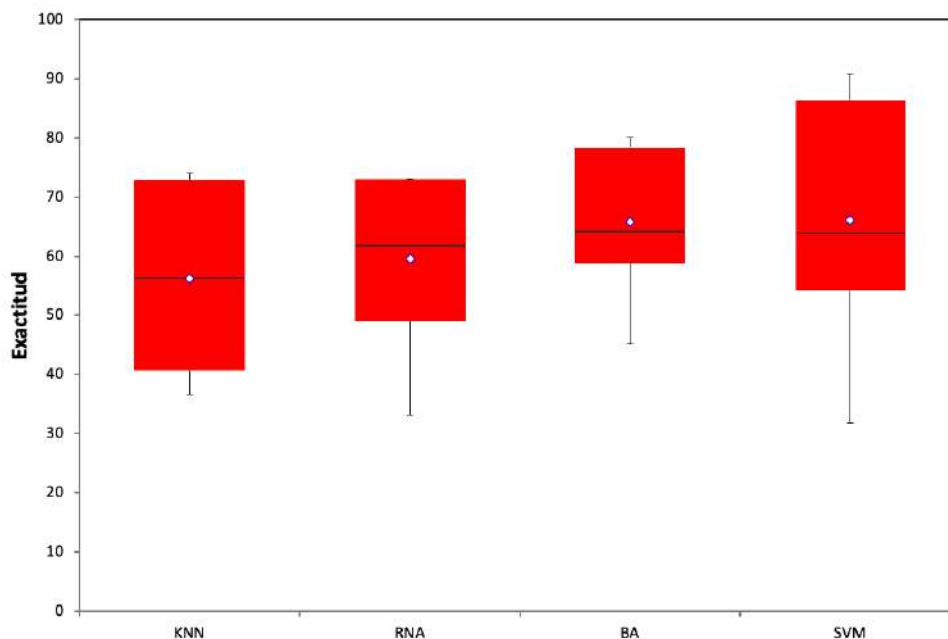


Figura 5.8: Desempeño de exactitud de algoritmos de aprendizaje automático.

Tabla 5.11: Resultados de exactitud de algoritmos de aprendizaje automático frente al vector de características concatenado.

	Exactitud
KNN	82.15 %
RNA	86.04 %
BA	86.51 %
SVM	94.65 %

Después de los experimentos realizados en esta sección de la investigación, se han desarrollado pruebas experimentales adicionales, donde, cada vector recuperado de la etapa de extracción de características ha sido concatenado, teniendo como resultado un vector de 273 descriptores, probado frente a los algoritmos de aprendizaje automático. Derivado de estas pruebas, se han conseguido los resultados de la Tabla 5.11, donde, el desempeño más alto, fue con el algoritmo SVM con un 94.65 % de exactitud, y el rendimiento más deficiente, lo ha conseguido el clasificador KNN con un 82.15 %.

Capítulo 6

Discusiones

Las diferentes etapas de la metodología aplicada para el reconocimiento de daños foliares causados por enfermedades y plagas que afectan a hojas de plantas de jitomate, pasaron por una fase de pruebas experimentales, donde, algoritmos para el preprocesado de imágenes, segmentación, extracción de características, equilibrio de clases, y clasificación, fueron medidos con base en el rendimiento de la clasificación. En este capítulo, se discuten los resultados obtenidos, derivado de los ensayos científicos realizados al método propuesto.

6.1. Métodos de segmentación

La tarea de separar un objeto o área de interés de su entorno en imágenes, se denomina segmentación, entre más optimizada sea esta labor, mejor desempeño tendrá en etapas posteriores. El módulo de segmentación para cualquier sistema de reconocimiento de patrones, es fundamental, ya que sin un buen desempeño de esta actividad, podría afectar negativamente a todo el sistema implementado.

En la primera fase experimental, visualizada en la Figura 4.2, en la etapa 1, se implementaron dos métodos de segmentación, basados en Otsu y PCA como núcleo de la misma. Las pruebas realizadas con estos dos algoritmos, han demostrado que, el mejor rendimiento lo ha obtenido el método PCA con base en las cualidades de los resultados. Posteriormente, para optimizar aún más el proceso de segmentación, se implementó una etapa de preprocesamiento aplicada a cada imagen del conjunto de datos previo al módulo de segmentación, ver Figura 4.8, cambiando

el estado actual de un espacio de color RGB a un espacio de color $L^*a^*b^*$. Derivado de las pruebas desarrolladas, se concluye que, el sistema propuesto tiene mejor rendimiento con una etapa de preprocesamiento al inicio de la estructura modular que sin ella. El sistema arroja un 93.86 % de exactitud para las pruebas sin preprocesamiento previo, y un 94.46 % de exactitud con la implementación del módulo de preprocesamiento al inicio de la metodología, ver gráfica de la Figura 6.1. Cabe mencionar que, para ambas pruebas, en la etapa de segmentación se uso el algoritmo PCA, como extractores de características se usaron características híbridas, y finalmente como método de clasificación el algoritmo SVM.

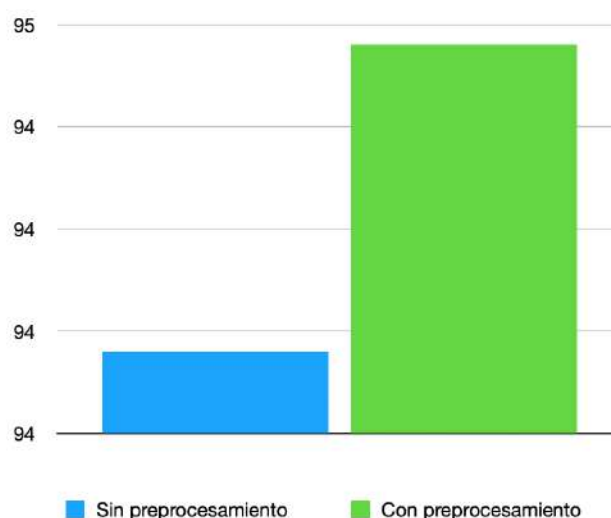


Figura 6.1: Gráfica de resultados sin preprocesamiento y con preprocesamiento previo.

6.2. Técnicas de extracción de características

Para poder extraer información relevante que describa a un objeto en una imagen, se necesita la correcta elección de algoritmos de extracción de características que ayuden a recuperar patrones de manera óptima. Así mismo, como ya se ha comentado en capítulos anteriores, en esta investigación se implementaron extractores de características de textura, cromáticas, e híbridas, mostrando su desempeño en la clasificación con distintos métodos de aprendizaje automático. Se demostró que, bajo las pruebas experimentales desarrolladas, la técnica de extracción de características más influyente fue la híbrida (texturales/cromáticas). Se puede inferir que, al combinar las características de textura y cromáticas, los algoritmos considerados para cada tipo

de patrones, aportan ciertos descriptores que ayudan al rendimiento en la clasificación. Así mismo, los algoritmos usados en esta investigación, como: descriptores de Contraste, características de Gabor, momentos de Hu, TDC, Haralick, y descriptores de Fourier, fueron considerados en la implementación del sistema propuesto como patrones de textura y cromáticos, definidos en la Sección 3.4, y derivado de las pruebas experimentales desarrolladas en la Sección 5.4, se puede inferir que los daños foliares en las hojas de plantas de jitomate consideradas en este trabajo son mas sensibles a las características de Gabor y Haralick.

6.3. Técnicas de equilibrio de clases

Un conjunto de datos, regularmente tiene una proporción asimétrica entre clases, esto se debe a que en un entorno real no hay un equilibrio igualitario entre el número de muestras de las clases para un objetivo específico. En esta investigación se usó un conjunto de datos de un repositorio de Internet de ambiente libre, el cual contiene 10 clases distintas, así mismo, la distribución del número de instancias entre clases está en desequilibrio, por ende, en este trabajo, se han implementado tres técnicas de equilibrio de clases, considerando, submuestreo aleatorio, sobremuestreo con réplicas, y sobremuestreo con el algoritmo SMOTE.

Posterior a los experimentos realizados con las tres técnicas de muestreo de clases (Sección 5.3), se ha demostrado lo siguiente, la técnica sobremuestreo con réplicas y sobremuestreo con el algoritmo SMOTE, en su desempeño obtuvieron resultados muy similares, sin embargo, la técnica de sobremuestreo con réplicas supero a el algoritmo SMOTE ligeramente con un 92.69% de exactitud, siendo este el resultado más alto para estas pruebas experimentales. Además, el resultado más bajo lo ha obtenido la técnica de submuestreo aleatorio, inferimos que, la causa de este rendimiento se debe a la pérdida de información.

En la gráfica de la Figura 6.2, se visualiza el desempeño de exactitud en la clasificación, bajo la implementación de tres técnicas de muestreo de clases consideradas en esta investigación.

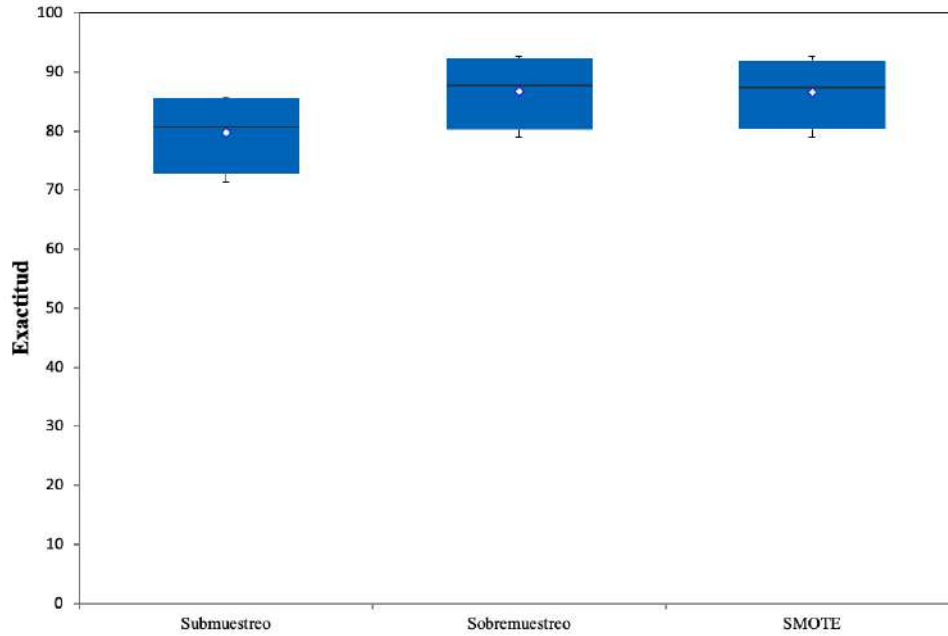


Figura 6.2: Exactitud de técnicas de equilibrio de clases.

6.4. Algoritmos de aprendizaje automático

Los algoritmos de aprendizaje automático, regularmente están al final de la estructura modular de un sistema de reconocimiento de patrones, midiendo el rendimiento en la clasificación con métricas de desempeño con base en una matriz de confusión. A lo largo de toda la etapa experimental realizada en esta investigación, como técnica de clasificación se han usado los algoritmos NB, KNN, RL, BA, RNA, y SVM, descritos en la Sección 3.6. Así mismo, se ha demostrado que, el algoritmo con mejor rendimiento para todas las pruebas realizadas mencionadas en este manuscrito, ha sido el algoritmo SVM.

Capítulo 7

Conclusiones y trabajo futuro

7.1. Conclusiones

Derivado del método propuesto en esta investigación, considerando las etapas de preprocesamiento, segmentación, extracción de características, equilibrio de clases y clasificación, para la identificación de daños foliares causados por enfermedades y plagas en hojas de plantas de jitomate, se concluye lo siguiente:

- La técnica de preprocesado de imágenes aplicada en este trabajo, bajo la transformación de un espacio de color RGB a un espacio de color $L^*a^*b^*$, trajo consigo un mejor rendimiento en la clasificación, comparado con la ausencia de este proceso.
- El método que separa la hoja de su entorno de manera prominente, e implementado en la mayor parte de la experimentación de esta investigación en la etapa de segmentación, fue el algoritmo PCA, frente al algoritmo Otsu.
- En la etapa de extracción de características, se midió el rendimiento en la clasificación probando con características texturales, cromáticas, e híbridas (texturales/cromáticas), donde, la técnica más influyentes y con mejor rendimiento fueron las características híbridas.
- De las tres técnicas usadas para el equilibrio de clases, considerando submuestreo aleatorio, sobremuestreo con réplicas y sobremuestreo con el algoritmo SMOTE, el método más

sobresaliente fue el sobremuestreo con réplicas, sin embargo, el algoritmo SMOTE quedo ligeramente abajo respecto al sobremuestreo con réplicas; además, podemos inferir que en la técnica submuestreo aleatorio existe un rendimiento bajo debido a la pérdida de información.

- De los algoritmos de aprendizaje automático usados en este trabajo de investigación, considerando: NB, KNN, RL, BA, RNA, y SVM, el que obtuvo mejor desempeño fueron las máquinas de vectores soporte (SVM).
- Derivado de las pruebas experimentales realizadas con distintos algoritmos de extracción de características de forma independiente, inferimos que los daños foliares en las hojas de jitomate son más sensibles a las técnicas de extracción de características de textura.

7.2. Trabajo futuro

El análisis, la implementación, y desarrollo experimental de los métodos aplicados en este trabajo de investigación, ayudará a tener una visión más clara de las bondades de los algoritmos usados, así mismo, se pueden aplicar las técnicas y métodos utilizados en este documento en áreas similares o alternas, o en su defecto realizar propuestas adicionales. Algunas sugerencias de trabajo futuro se describen a continuación:

- Implementar algoritmos de selección de características.
- Con base en este estudio, se podría pensar en el reconocimiento no solo de daños foliares abordados en este trabajo, sino en daños foliares adicionales.
- Un sistema de reconocimiento de enfermedades y plagas no solamente aplicado a plantas de jitomate, sino a diferentes plantas.
- Nutrir el conjunto de datos con daños foliares con deficiencias de nutrientes.
- Experimentar con algoritmos adicionales para la extracción de características.
- Migrar el prototipo a un ambiente real, sitio web o móvil.

Capítulo 8

Artículos publicados y en revisión

8.1. Artículo publicado

En la Figura 8.1, se visualiza la primera página del primer artículo desarrollado, así mismo, se participó en el congreso ICIC 2020, generando un capítulo de libro publicado en la editorial Springer. Se anexa el link: https://doi.org/10.1007/978-3-030-60796-8_9



Figura 8.1: Capítulo de libro publicado en la editorial Springer.

8.2. Artículo en revisión

En la Figura 8.2, se visualiza la primera página del segundo artículo desarrollado, así mismo, se **sometió** al JCR Computación y Sistemas. Cabe mencionar que desde la fecha de envío (03 de abril de 2021) se encuentra en estatus de **en revisión**.



Figura 8.2: Artículo para revista JCR (Computación y Sistemas).

En la Figura 8.3, se visualiza la primera página del tercer artículo desarrollado, así mismo, se **sometió** a la revista IJCOPI (International Journal of Combinatorial Optimization Problems and Informatics). Cabe mencionar que desde la fecha de envío (06 de agosto de 2022) se encuentra en estatus **enviado**.

Influence of feature extraction algorithms in the classification for the identification of foliar damage in tomato plants

Ernesto García Amaro¹, Jair Cervantes Canales¹, Farid García Lamont¹, José Sergio Ruiz Castilla¹.

¹Universidad Autónoma del Estado de México, Fraccionamiento El Tejocote, Texcoco, Estado de México, México.

ernestogarciaamaro@gmail.com, jcervantes@uaemex.mx, fgarcial@uaemex.mx, jsergioruizc@gmail.com

Abstract:

This manuscript presents a comparative analysis of the influence of feature extraction techniques on foliar damage classification caused by pests and diseases that affect tomato crops through the Plantvillage image dataset. In the implemented system, five stages of the proposed method have been considered, contemplating, preprocessing, segmentation, feature extraction, classes balancing, and classification. The experiments showed that the most discriminative feature extraction method, which best describes foliar damage, is texture patterns, getting 84.85% for the Haralick features and 90.87% for the Gabor features. In addition, the experiments developed with the concatenated feature vectors, classified with the SVM algorithm, have shown a performance of 94.65%.

Keywords: feature extraction, pattern recognition, artificial vision, tomato diseases and pests.

1. Introduction

Computational sciences have increased their development in recent years, applying methods for solving problems of everyday life in our environment, achieving promising results; however, the cusp has not been reached, and the contribution to science continues to advance. In the last years, computational techniques' use, implementation, and development have increased exponentially in multiple areas. Some research areas such as the medicine [1,2], education [3,4], home [5], aeronautics [6,7], automotive industry [8,9], energetic area [10,11], telecommunications [12,13], agriculture [14,15], among other, innovating processes that years ago were done manually. In agriculture, computational techniques have been used to solve multiple problems [16-20]; one of the applications we will focus on in this work is the automatic identification of pests and diseases from images.

The foliar damage is caused by diseases and pests that affect tomato plants produced by variations in temperature, humidity, wind, drought, residues from previous harvests, or poor crop management, directly impacting losses due to low production, reducing financial balances. Traditionally, the identification of these anomalies is carried out visually by experts in the area. However, it is considered a manual and inaccurate method since the nature of foliar damage could be, in some cases, very similar, incurring in an excessive or erroneous application of chemical products in the crops, directly affecting the plants and the consumer.

The identification of foliar damage uses traditional automatic computational techniques by a series of steps: preprocessing, segmentation, feature extraction, and classification. Image preprocessing focuses mainly on applying filters to digital images to improve their quality and appearance, the filters permit highlighting specific details that need to be pointed out. On the other hand, segmentation techniques allow obtaining only the region of the desired object; feature extraction methods look for patterns in images to describe objects, currently, three types of features stand out, textural, chromatic, and geometric patterns; finally, machine learning algorithms develop a training stage from the received descriptors, build models, and make a prediction on the test data.

One of the phases that most affects the performance of foliar damage classification is the type of features used. Different feature extraction techniques have been developed with the growing development of computational methods for image analysis and processing.

Moreover, feature extraction techniques play an essential role in a pattern recognition system, allowing essential descriptors to be obtained from the processed images, making a difference between different classes of a given dataset, even when they share similarities.

Figura 8.3: Artículo para la revista IJCOPI.

Bibliografía

- [1] MathWorks, “Machine learning vs deep learning.” ebook, <https://explore.mathworks.com/machine-learning-vs-deep-learning>, 2019. 1994-2019 The MathWorks.
- [2] INEGI, “Encuesta nacional agropecuaria 2017,” Report ENA. 2017, Instituto Nacional de Estadística y Geografía, México, <http://www.beta.inegi.org.mx/proyectos/encagro/ena/2017/>, 2017.
- [3] INEGI, “Encuesta nacional agropecuaria 2019,” Report ENA. 2019, Instituto Nacional de Estadística y Geografía, <https://www.inegi.org.mx/temas/agricultura/>, 2019.
- [4] CESAVERG, *Campaña manejo fitosanitario del jitomate*. Comité Estatal de Sanidad Vegetal de Guanajuato, A.C., Irapuato Guanajuato, 2016 ed., 2016.
- [5] A. A. Bharate and M. Shirdhonkar, “A review on plant disease detection using image processing,” in *2017 International Conference on Intelligent Sustainable Systems (ICISS)*, pp. 103–109, IEEE, 2017.
- [6] G. Dhingra, V. Kumar, and H. D. Joshi, “Study of digital image processing techniques for leaf disease detection and classification,” *Multimedia Tools and Applications*, vol. 77, no. 15, pp. 19951–20000, 2018.
- [7] K. Lin, J. Chen, H. Si, and J. Wu, “A review on computer vision technologies applied in greenhouse plant stress detection,” in *Chinese Conference on Image and Graphics Technologies*, pp. 192–200, Springer, 2013.

- [8] J. Cervantes, J. Taltempa, F. G. Lamont, J. S. R. Castilla, A. Y. Rendon, and L. D. Jalili, “Análisis comparativo de las técnicas utilizadas en un sistema de reconocimiento de hojas de planta,” *Revista Iberoamericana de Automática e Informática industrial*, vol. 14, no. 1, pp. 104–114, 2017.
- [9] J. Cervantes, F. G. Lamont, L. R. Mazahua, A. Z. Hidalgo, and J. S. R. Castilla, “Complex identification of plants from leaves,” in *International Conference on Intelligent Computing*, pp. 376–387, Springer, 2018.
- [10] I. E. Massi, Y. Es-saady, M. E. Yassa, and D. Mammass, “Combination of multiple classifiers for automatic recognition of diseases and damages on plant leaves,” *Signal, Image and Video Processing*, vol. 15, pp. 789–796, oct 2020.
- [11] J. Basavaiah and A. A. Anthony, “Tomato leaf disease classification using multiple feature extraction techniques,” *Wireless Personal Communications*, vol. 115, pp. 633–651, jun 2020.
- [12] G. Pattnaik and K. Parvathi, “Automatic detection and classification of tomato pests using support vector machine based on HOG and LBP feature extraction technique,” in *Advances in Intelligent Systems and Computing*, pp. 49–55, Springer Singapore, nov 2020.
- [13] B. Luna-Benoso, J. C. Martínez-Perales, J. Cortés-Galicia, R. Flores-Carapia, and V. M. Silva-García, “Detection of diseases in tomato leaves by color analysis,” *Electronics*, vol. 10, p. 1055, apr 2021.
- [14] P. Burt and E. Adelson, “The laplacian pyramid as a compact image code,” *IEEE Transactions on communications*, vol. 31, no. 4, pp. 532–540, 1983.
- [15] S. Prasad, P. Kumar, and A. Jain, “Detection of disease using block-based unsupervised natural plant leaf color image segmentation,” in *International Conference on Swarm, Evolutionary, and Memetic Computing*, pp. 399–406, Springer, 2011.
- [16] Y. Tian, L. Wang, and Q. Zhou, “Grading method of crop disease based on image processing,” in *International Conference on Computer and Computing Technologies in Agriculture*, pp. 427–433, Springer, 2011.

- [17] S. Prasad, P. Kumar, R. Hazra, and A. Kumar, “Plant leaf disease detection using gabor wavelet transform,” in *International Conference on Swarm, Evolutionary, and Memetic Computing*, pp. 372–379, Springer, 2012.
- [18] D. Nunes, C. Werly, G. K. Vianna, and S. M. S. da Cruz, “Early discovery of tomato foliage diseases based on data provenance and pattern recognition,” in *International Provenance and Annotation Workshop*, pp. 229–231, Springer, 2014.
- [19] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [20] D. A. Niño, J. S. R. Castilla, M. D. A. Zenteno, and L. D. Jalili, “Complex leaves classification with features extractor,” in *International Conference on Intelligent Computing*, pp. 758–769, Springer, 2019.
- [21] M. Dyrmann, H. Karstoft, and H. S. Midtiby, “Plant species classification using deep convolutional neural network,” *Biosystems Engineering*, vol. 151, pp. 72–80, 2016.
- [22] H. S. Abdullahi, R. Sheriff, and F. Mahieddine, “Convolution neural network in precision agriculture for plant image recognition and classification,” in *2017 Seventh International Conference on Innovative Computing Technology (Intech), Ieee, Londrés*, pp. 1–3, 2017.
- [23] X. Liu, F. Xu, Y. Sun, H. Zhang, and Z. Chen, “Convolutional recurrent neural networks for observation-centered plant identification,” *Journal of Electrical and Computer Engineering*, vol. 2018, 2018.
- [24] M. M. Ghazi, B. Yanikoglu, and E. Aptoula, “Plant identification using deep neural networks via optimization of transfer learning parameters,” *Neurocomputing*, vol. 235, pp. 228–235, 2017.
- [25] S. H. Lee, C. S. Chan, P. Wilkin, and P. Remagnino, “Deep-plant: Plant identification with convolutional neural networks,” in *2015 IEEE international conference on image processing (ICIP)*, pp. 452–456, IEEE, 2015.
- [26] D. Wu, X. Han, G. Wang, Y. Sun, H. Zhang, and H. Fu, “Deep learning with taxonomic loss for plant identification,” *Computational Intelligence and Neuroscience*, vol. 2019, 2019.

- [27] J. Krause, G. Sugita, K. Baek, and L. Lim, “Wtplant (what’s that plant?) a deep learning system for identifying plants in natural images,” in *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pp. 517–520, 2018.
- [28] C. Chen, Q. Yan, M. Li, and J. Tong, “Classification of blurred flowers using convolutional neural networks,” in *Proceedings of the 2019 3rd International Conference on Deep Learning Technologies*, pp. 71–74, 2019.
- [29] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, “Deep neural networks based recognition of plant diseases by leaf image classification,” *Computational intelligence and neuroscience*, vol. 2016, 2016.
- [30] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, “Identification of rice diseases using deep convolutional neural networks,” *Neurocomputing*, vol. 267, pp. 378–384, 2017.
- [31] X. Zhang, Y. Qiao, F. Meng, C. Fan, and M. Zhang, “Identification of maize leaf diseases using improved deep convolutional neural networks,” *IEEE Access*, vol. 6, pp. 30370–30377, 2018.
- [32] R. A. Priyadharshini, S. Arivazhagan, M. Arun, and A. Mirnalini, “Maize leaf disease classification using deep convolutional neural networks,” *Neural Computing and Applications*, vol. 31, no. 12, pp. 8887–8895, 2019.
- [33] J. Hu, Z. Chen, M. Yang, R. Zhang, and Y. Cui, “A multiscale fusion convolutional neural network for plant leaf recognition,” *IEEE Signal Processing Letters*, vol. 25, no. 6, pp. 853–857, 2018.
- [34] A. Hanson, M. Joel, A. Joy, and J. Francis, “Plant leaf disease detection using deep learning and convolutional neural network,” *International Journal of Engineering Science*, vol. 5324, 2017.
- [35] U. P. Singh, S. S. Chouhan, S. Jain, and S. Jain, “Multilayer convolution neural network for the classification of mango leaves infected by anthracnose disease,” *IEEE Access*, vol. 7, pp. 43721–43729, 2019.

- [36] P. Jiang, Y. Chen, B. Liu, D. He, and C. Liang, “Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks,” *IEEE Access*, vol. 7, pp. 59069–59080, 2019.
- [37] K. P. Ferentinos, “Deep learning models for plant disease detection and diagnosis,” *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018.
- [38] B. Liu, Y. Zhang, D. He, and Y. Li, “Identification of apple leaf diseases based on deep convolutional neural networks,” *Symmetry*, vol. 10, no. 1, p. 11, 2018.
- [39] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, “A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition,” *Sensors*, vol. 17, no. 9, p. 2022, 2017.
- [40] M. Sardogan, A. Tuncer, and Y. Ozen, “Plant leaf disease detection and classification based on cnn with lvq algorithm,” in *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, pp. 382–385, IEEE, 2018.
- [41] S. Verma, A. Chug, A. P. Singh, P. Rajvanshi, and S. Sharma, “Deep learning based plant disease diagnosis for grape plant,”
- [42] E. Suryawati, R. Sustika, R. S. Yuwana, A. Subekti, and H. F. Pardede, “Deep structured convolutional neural network for tomato diseases detection,” in *2018 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pp. 385–390, IEEE, 2018.
- [43] H. Durmuş, E. O. Güneş, and M. Kırıcı, “Disease detection on the leaves of the tomato plants by using deep learning,” in *2017 6th International Conference on Agro-Geoinformatics*, pp. 1–5, IEEE, 2017.
- [44] J. Shijie, J. Peiyi, H. Siping, *et al.*, “Automatic detection of tomato diseases and pests based on leaf images,” in *2017 Chinese Automation Congress (CAC)*, pp. 2537–2510, IEEE, 2017.
- [45] K. Zhang, Q. Wu, A. Liu, and X. Meng, “Can deep learning identify tomato leaf disease?,” *Advances in Multimedia*, vol. 2018, 2018.

- [46] G. Wang, Y. Sun, and J. Wang, “Automatic image-based plant disease severity estimation using deep learning,” *Computational intelligence and neuroscience*, vol. 2017, 2017.
- [47] Y. Sun, Y. Liu, G. Wang, and H. Zhang, “Deep learning for plant identification in natural environment,” *Computational intelligence and neuroscience*, vol. 2017, 2017.
- [48] Z. Jiao, L. Zhang, C.-A. Yuan, X. Qin, and L. Shang, “Plant leaf recognition based on conditional generative adversarial nets,” in *International Conference on Intelligent Computing*, pp. 312–319, Springer, 2019.
- [49] A. Gutierrez, A. Ansuategi, L. Susperregi, C. Tubío, I. Rankić, and L. Lenža, “A benchmarking of learning strategies for pest detection and identification on tomato plants for autonomous scouting robots using internal databases,” *Journal of Sensors*, vol. 2019, 2019.
- [50] R. K. Srivastava, K. Greff, and J. Schmidhuber, “Highway networks,” *arXiv preprint arXiv:1505.00387*, 2015.
- [51] R. K. Srivastava, K. Greff, and J. Schmidhuber, “Training very deep networks,” in *Advances in neural information processing systems*, pp. 2377–2385, 2015.
- [52] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, “Fitnets: Hints for thin deep nets,” *arXiv preprint arXiv:1412.6550*, 2014.
- [53] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, pp. 1097–1105, 2012.
- [54] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.
- [55] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, “Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5 mb model size,” *arXiv preprint arXiv:1602.07360*, 2016.

- [56] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.
- [57] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [58] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in *Thirty-first AAAI conference on artificial intelligence*, 2017.
- [59] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.
- [60] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [61] N. Schor, A. Bechar, T. Ignat, A. Dombrovsky, Y. Elad, and S. Berman, “Robotic disease detection in greenhouses: combined detection of powdery mildew and tomato spotted wilt virus,” *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 354–360, 2016.
- [62] P. Lindhout, W. Korta, M. Cislik, I. Vos, and T. Gerlagh, “Further identification of races of *cladosporium fulvum* (*fulvia fulva*) on tomato originating from the netherlands france and poland,” *Netherlands Journal of Plant Pathology*, vol. 95, no. 3, pp. 143–148, 1989.
- [63] M. Tian, B. Benedetti, and S. Kamoun, “A second kazal-like protease inhibitor from *phytophthora infestans* inhibits and interacts with the apoplastic pathogenesis-related protease p69b of tomato,” *Plant physiology*, vol. 138, no. 3, pp. 1785–1793, 2005.
- [64] R. C. Gonzalez and R. E. Woods, *Digital image processing*. Pearson education india, 2009.
- [65] S. D. Gupta and Y. Ibaraki, *Plant image analysis, fundamentals and applications*. 6000 Broken Sound Parkway NW, Suite 300 Boca Raton, FL 33487-2742: Boca Raton: CRC Press, 2015.

- [66] E. Cuevas, D. Zaldívar, and M. Pérez, *Procesamiento digital de imágenes con MATLAB & Simulink*. Ra-Ma, 2016.
- [67] F. Garcia-Lamont, J. Cervantes, A. López, and L. Rodriguez, “Segmentation of images by color features: A survey,” *Neurocomputing*, vol. 292, pp. 1–27, 2018.
- [68] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, pp. 62–66, jan 1979.
- [69] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital image processing using MATLAB*. Pearson Education India, 2004.
- [70] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis and Machine Vision*. Springer US, 1993.
- [71] Q. xia Hu, J. Tian, and D. jian He, “Wheat leaf lesion color image segmentation with improved multichannel selection based on the chan–vese model,” *Computers and Electronics in Agriculture*, vol. 135, pp. 260–268, apr 2017.
- [72] L. Cao, K. Chua, W. Chong, H. Lee, and Q. Gu, “A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine,” *Neurocomputing*, vol. 55, pp. 321–336, sep 2003.
- [73] R. M. Haralick, K. Shanmugam, and I. Dinstein, “Textural features for image classification,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, pp. 610–621, nov 1973.
- [74] I. R. Donis-González, D. E. Guyer, and A. Pease, “Postharvest noninvasive classification of tough-fibrous asparagus using computed tomography images,” *Postharvest Biology and Technology*, vol. 121, pp. 27–35, nov 2016.
- [75] M.-K. Hu, “Visual pattern recognition by moment invariants,” *IEEE Transactions on Information Theory*, vol. 8, pp. 179–187, feb 1962.
- [76] Y. Mingqiang, K. Kidiyo, R. Joseph, *et al.*, “A survey of shape feature extraction techniques,” *Pattern recognition*, vol. 15, no. 7, pp. 43–90, 2008.

- [77] S. Dabbaghchian, M. P. Ghaemmaghani, and A. Aghagolzadeh, "Feature extraction using discrete cosine transform and discrimination power analysis with a face recognition technology," *Pattern Recognition*, vol. 43, pp. 1431–1440, apr 2010.
- [78] D. Gabor, "Theory of communication: Journal of the institute of electrical engineers, 93, 429–457," 1946.
- [79] S. Kotsiantis, D. Kanellopoulos, P. Pintelas, *et al.*, "Handling imbalanced datasets: A review," *GESTS International Transactions on Computer Science and Engineering*, vol. 30, no. 1, pp. 25–36, 2006.
- [80] A. Amin, S. Anwar, A. Adnan, M. Nawaz, N. Howard, J. Qadir, A. Hawalah, and A. Husain, "Comparing oversampling techniques to handle the class imbalance problem: A customer churn prediction case study," *IEEE Access*, vol. 4, pp. 7940–7957, 2016.
- [81] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, jun 2002.
- [82] V. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999.
- [83] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, oct 1986.
- [84] G. H. John and P. Langley, "Estimating continuous distributions in bayesian classifiers,"
- [85] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine Learning*, vol. 6, pp. 37–66, jan 1991.
- [86] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [87] J. S. Borges, J. M. Bioucas-Dias, and A. R. S. Marcal, "Bayesian hyperspectral image segmentation with discriminative class learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, pp. 2151–2164, jun 2011.

- [88] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM Algorithm,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, pp. 1–22, sep 1977.
- [89] D. P. Hughes and M. Salathe, “An open access repository of images on plant health to enable the development of mobile disease diagnostics,” 2015.