

CitruScan: Un Enfoque Basado en Redes Neuronales para Proteger la Producción de Limones



Colaboración

Juan A. Guerrero-Ibáñez; Juan Contreras-Castillo,
Universidad de Colima

Fecha de recepción: 22 de agosto de 2025

Fecha de aceptación: 13 de septiembre de 2025

RESUMEN: Una de las principales amenazas para la producción de limón en México son las enfermedades causadas por virus, bacterias y hongos. La detección temprana de enfermedades es esencial para reducir las pérdidas económicas, sin embargo, los métodos tradicionales son costosos y poco eficientes. En este trabajo se propone CitruScan, un modelo híbrido que combina redes neuronales convolucionales (CNN) para la extracción de características con Random Forest (RF) para la clasificación de imágenes de hojas de limón. Se empleó un dataset de 1354 imágenes (8 clases de enfermedades más hojas sanas), obtenidas de Kaggle, dividido en conjuntos de entrenamiento, validación y prueba. Los resultados muestran que CitruScan alcanzó una exactitud superior al 99%, superando a métodos convencionales como ResNet, MobileNet y VGG16 usados en trabajos publicados en la literatura. Los resultados muestran la eficacia de la propuesta como una herramienta para el diagnóstico temprano de enfermedades en plantaciones de limón.

PALABRAS CLAVE: Aprendizaje profundo, clasificación de enfermedades, hojas de limón, redes generativas adversariales, redes neuronales convolucionales.

ABSTRACT: One of the main threats to lemon production in Mexico is diseases caused by viruses, bacteria, and fungi. Early disease detection is essential to reduce economic losses; however, traditional methods are costly and inefficient. In this work, we propose CitruScan, a hybrid model that combines Convolutional Neural Networks (CNNs) for feature extraction with Random Forest (RF) for the classification of lemon leaf images. A dataset of 1,354 images (8 disease classes plus healthy leaves), obtained from Kaggle, was used and divided into training, validation, and testing sets. The results show that CitruScan achieved an accuracy above 99%, outperforming conventional methods such as ResNet, MobileNet, and VGG16 reported in the literature. The results demonstrate the effectiveness of the proposed approach as a tool for the early diagnosis of diseases in lemon plantations.

KEYWORDS: Convolutional neural networks, Deep learning, disease classification, generative adversarial networks, lemon leaves.

INTRODUCCIÓN

El limón es uno de los frutos de gran importancia para el consumo nacional y es uno de los productos de mayor exportación en México. De acuerdo con datos publicados por el gobierno de México, en 2024, nuestro país se consolidó como el segundo productor y exportador de limón a nivel mundial, solamente por debajo de la India [1]. Colima ocupa el tercer lugar a nivel nacional en la producción de limón, solo por detrás de Michoacán, y Veracruz, produciendo en el 2023 un total de 312,047 toneladas [1].

El gobierno federal expresa que existe una serie de patógenos que afectan a los cítricos y que pueden ocasionar efectos

adversos como reducción de la producción, calidad del fruto o incluso la muerte de los árboles [2]. Las enfermedades que afectan a la producción de limón en México incluyen Antracnosis, el virus de la tristeza, araña roja, entre otras.

Para la detección de enfermedades en los cultivos comúnmente se aplican tres métodos: i) identificación visual por personas expertas, ii) análisis de características morfológicas para identificación de patógenos y iii) técnicas de diagnóstico molecular y microbiológico [3]. Sin embargo, estos métodos son costosos, es susceptible a errores subjetivos por parte de los expertos y se invierte mucho tiempo, lo que puede impulsar la proliferación de la enfermedad. De acuerdo con la Organización de las Naciones Unidas para la alimentación y la agricultura (FAO) las enfermedades son responsables de pérdidas que oscilan entre 20% y 40% de la producción total [4].

En este contexto, es necesario implementar técnicas de diagnóstico y monitoreo temprano con el fin de identificar enfermedades en los cultivos y evitar su pronta diseminación.

Con respecto a la identificación de enfermedades en el limón, las redes neuronales convolucionales (CNN – por sus siglas en inglés Convolutional Neural Networks) se han convertido en uno de los métodos preferidos para la clasificación y detección. La mayoría de los trabajos hacen uso de redes neuronales pre-entrenadas. Por ejemplo, en [5] se utilizaron ResNet, MobileNet y VGG16 para identificar enfermedades en las hojas de limón dulce. Los resultados mostraron que MobileNet alcanzó la mejor exactitud (89%), sin embargo, no detallaron las características del dataset. La limitación del enfoque es la falta de análisis profundo de los datos y la dependencia de modelos pre-entrenados.

En [6] utilizaron una versión modificada de AlexNet para la detección de enfermedades en cultivos, mostrando resultados competitivos en clasificación. Las limitantes del artículo son el uso de un dataset reducido lo que impacta la generalización en campo.

En [7] se presenta un modelo de diagnóstico de enfermedades en hojas basado en segmentación y AlexNet. La propuesta mejora la extracción de características al reducir ruido en las imágenes. La principal limitante es su procesamiento complejo y sensible a variaciones de iluminación, lo que puede afectar su aplicabilidad en escenarios agrícolas reales.

Por otro lado, en [8] se propuso un modelo híbrido basado en CNN y Support Vector Machine (SVM) para detectar niveles de severidad de cancros en limón, con un dataset generado en los campos de

Punjab, India y estaba formado por 6 categorías de severidad de la enfermedad. El modelo alcanzó una exactitud media de 94.03%. Sin embargo, el conjunto de datos es cerrado y los autores no proporciona información sobre la estructura del mismo, limitando la reproducibilidad del estudio.

Los autores en [9] aplicaron modelos de aprendizaje profundo con aprendizaje de transferencia (DenseNet-201, ResNet-50, ResNet-152V2, y Xception) y utilizaron un conjunto de datos propio. Los resultados mostraron que el mejor modelo fue Xception con una exactitud alrededor de 94.34%. Si bien el modelo obtuvo un buen desempeño, la principal limitación es su alto costo computacional, lo que restringe su aplicación práctica en entornos agrícolas con pocos recursos.

Finalmente, los autores en [10] propusieron un modelo híbrido CNN-SVM para clasificar 4 enfermedades en hojas de limón, utilizando un conjunto de datos de 4821 imágenes y obteniendo una exactitud de 89.6%. Sin embargo, a pesar de usar un conjunto de datos amplio, su exactitud fue inferior a 90%.

Se puede observar que los trabajos previos muestran resultados que rara vez superan el 94% de exactitud y tienen limitaciones con respecto a conjunto de datos reducidos y costos computacionales altos. En este trabajo presentamos CitruScan, un modelo híbrido que combina CNN para el proceso de extracción de características con Random Forest(RF) para el proceso de clasificación de imágenes. Además del aporte metodológico, CitruScan tiene un impacto práctico directo en el ámbito agrícola. La herramienta puede ser implementada en aplicaciones móviles permitiendo a los productores de cítricos identificar de forma temprana enfermedades en las hojas, incluso sin experiencia previa en diagnóstico visual.

MATERIAL Y MÉTODOS

CitruScan es un modelo híbrido CNN-RF para la clasificación e identificación de enfermedades en las hojas de limón. Como entradas de la arquitectura se tienen un conjunto de datos de imágenes de hojas de limón. La salida de la arquitectura muestra tres elementos claves: i) una etiqueta indicando la enfermedad que se detectó o confirmado que la hoja está sana de acuerdo con la imagen, ii) una etiqueta representando el resultado de clasificación que se obtuvo como resultado con CitruScan y iii) el porcentaje de confidencia asociado con la predicción realizada por el modelo.

Modelo de CitruScan

Este estudio propone un modelo para detectar enfermedades en las hojas del limón. La Figura 1 muestra los componentes de CitruScan, compuesta por dos bloques: i) un bloque de aprendizaje de características y ii) un bloque de clasificación.

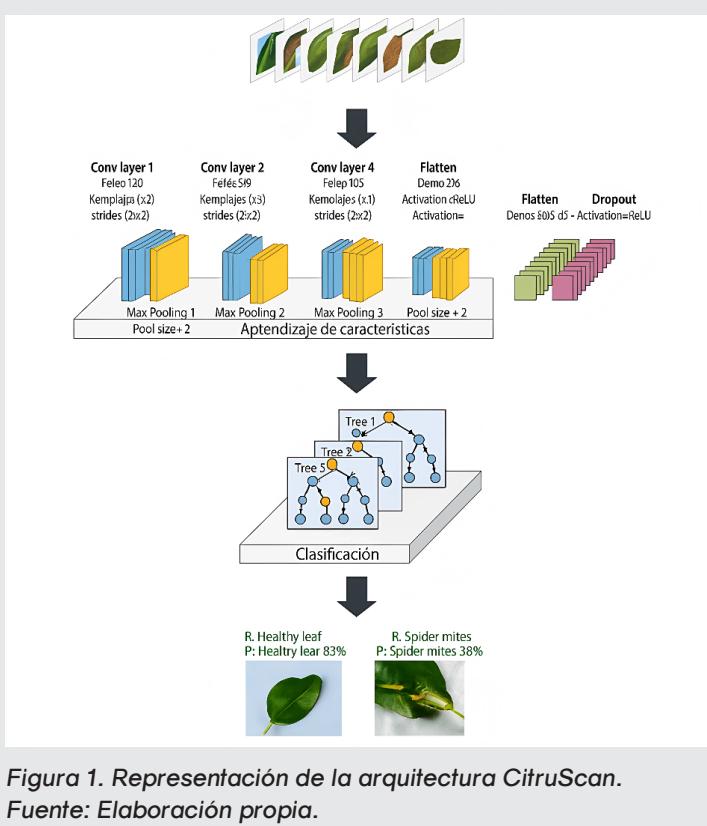


Figura 1. Representación de la arquitectura CitruScan.

Fuente: Elaboración propia.

El bloque de aprendizaje de características recibe las imágenes de entrada en un tamaño definido de 128x128 píxeles con tres canales de color, a las cuales se les aplicó normalización de los valores de los píxeles. La red neuronal que se implementó consta de 4 capas convolucionales con tamaños de filtros de 16, 32, 64 y 128, cada una con un kernel de 3x3 y la función de activación ReLU (Rectified Lineal Unit). Posteriormente, se aplica una capa Flatten para transformar el tensor multidimensional en un vector, a continuación, se aplica una capa densa para reducir la dimensionalidad y facilitar el aprendizaje de las características extraídas.

Para el entrenamiento se utilizó el optimizador Adam, se definió la tasa de aprendizaje inicial de 0.001, el batch size de 32. Se aplicaron 100 épocas y se utilizó el ambiente MGPU (Multiple-Graphics Processing Unit) de colab.

El bloque de clasificación recibe las características extraídas por la red neuronal y aplica el proceso de clasificación mediante el algoritmo Random Forest (RF). Para tener un equilibrio entre rendimiento y costo computacional se aplicó una configuración de 500 árboles, una profundidad máxima de 20, y como criterio de división Gini.

Conjunto de datos (dataset)

Para el entrenamiento y validación de CitruScan se utilizó el dataset público Lemon leaf disease dataset (LLDD), un conjunto de datos con imágenes de alta calidad diseñado para entrenar y validar modelos de aprendizaje máquina para la clasificación de enfermedades del limón [11]. El dataset consiste en un total de 1354 imágenes en formato jpg

con tamaños de 1000x1000 a 300dpi. Las imágenes están distribuidas en 9 categorías que representan 8 enfermedades (Anthracnose, Bacterial Blight, Citrus Canker, Curl Virus, Deficiency Leaf, Dry Leaf, Sooty Mould, y Spider Mites) y la categoría de hoja saludable (healthy leaf). La distribución del conjunto de datos se muestra en la tabla 1.

Para garantizar una evaluación justa del modelo, el dataset se dividió en tres subconjuntos: 70% para entrenamiento, 15% para validación y 15% para prueba, utilizando un muestreo estratificado para mantener el balance de clases.

Tabla 1. Distribución del conjunto de datos por categoría.

| Categoría | Número de imágenes |
|------------------|--------------------|
| Anthracnose | 100 |
| Bacterial Blight | 105 |
| Citrus Canker | 178 |
| Curl Virus | 115 |
| Deficiency Leaf | 193 |
| Dry Leaf | 186 |
| Sooty Mould | 153 |
| Spider Mites | 114 |
| Healthy leaf | 210 |
| Total | 1354 |

Fuente: Elaboración propia.

La Figura 2 muestra un ejemplo de las imágenes utilizadas para el presente trabajo.

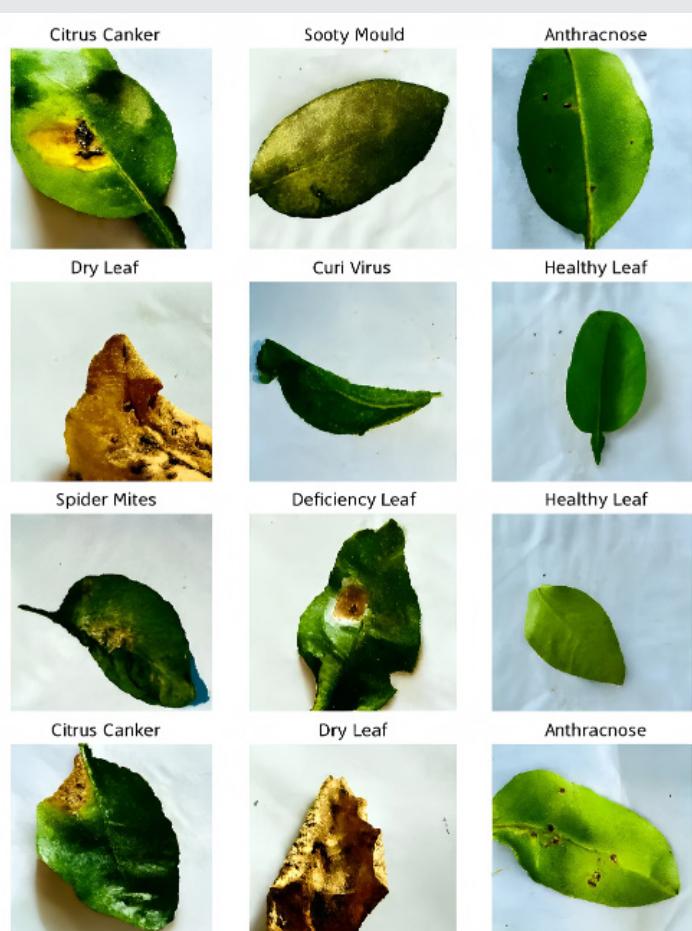


Figura 2. Representación de imágenes usadas en CitruScan.

Fuente: Elaboración propia.

Uno de los retos más comunes para los conjuntos de datos es el overfitting, es decir, un problema que hace que el modelo "memorice" los datos de entrenamiento más que tener un aprendizaje generalizado. Para reducir este efecto, se utilizan técnicas de aumento de datos, que incrementan el tamaño del conjunto de entrenamiento y mejoran la capacidad de generalización [12].

Los métodos tradicionales expanden el conjunto de datos mediante la creación de nuevas imágenes usando las imágenes existentes aplicando transformaciones como traslación, rotación, ajuste de brillo, acercamiento, alejamiento, transformaciones, entre otras técnicas. Sin embargo, estas técnicas pueden generar imágenes redundantes y pérdida de calidad de las imágenes.

Las técnicas recientes son conocidas como Redes generativas adversariales (GAN - Generative Adversarial Networks) [13], las cuales aplican métodos de aprendizaje profundo incluyendo Vanilla GAN [14], Deep Convolutional GAN [15] o Conditional GAN [16] para la generación de muestras sintéticas que mantienen características estadísticas del conjunto de datos original.

En este trabajo se implementó una Deep Convolutional GAN (DCGAN) para incrementar y balancear el conjunto de datos de hojas de limón. El generador incluyó 4 capas convolucionales transpuestas, activación ReLU en las capas ocultas y Tanh en la salida. El discriminador incluyó 4 capas convolucionales, activación LeakyReLU en las capas ocultas y Sigmoid en la salida. El optimizado fue Adam con tasa de aprendizaje de 0.0002 y $\beta_1=0.5$, además se alicaron 200 épocas, completando un total de 3600 imágenes, balanceando las imágenes en cada categoría a 400. La validación de las imágenes solamente se realizó mediante inspección visual por un experto agrónomo.

CitruScan combina CNN, Random Forest y DCGAN, lo cual se considera su principal innovación, ofreciendo ventajas frente a redes convolucionales profundas con regularización, incluyendo una mejora de la capacidad de generalización al separar la extracción de características de la clasificación (Random Forest), reduciendo el riesgo de sobreajuste en escenarios con conjuntos de datos limitados, y DCGAN permite un balance de las clases.

RESULTADOS

En esta sección, describimos la configuración del escenario de pruebas y los resultados obtenidos en la evaluación de desempeño del modelo propuesto.

Configuración

El desarrollo del modelo se realizó en Google Colaboratory [17], un ambiente de desarrollo gratuito

que usa el lenguaje Python y se ejecuta en la nube. El modelo fue entrenado con 100 épocas y se aplicó early stopping para controlar el rendimiento del modelo durante todas las épocas para reducir el overfitting y mejorar de esta manera la generalización de la red neuronal. Se definió un tamaño de batch de 32, se aplicó el optimizador Adam, la tasa de aprendizaje inicial se configuró a 0.001, como es un problema multi-clase, se utilizó la función de pérdida cross-entropy categorial.

Métricas de evaluación

Para analizar el rendimiento del modelo propuesto, se utilizaron las cuatro métricas más utilizadas: accuracy, precisión, recall y F1 score.

Accuracy representa el comportamiento del modelo a través de todas las clases. El parámetro accuracy se calcula como el radio entre el número correcto de predicciones entre el total de predicciones.

La precisión representa la exactitud del modelo en clasificar una muestra como positiva.

Recall mide la habilidad del modelo para detectar muestras positivas y F1 score combina las métricas de precisión y recall para obtener un único valor.

Otro de las métricas que se utilizó fue la matriz de confusión, un método que permite representar visualmente los resultados del algoritmo clasificador. En esta representación visual se desglosa el número de instancias reales de una clase específica frente al número de instancias previstas para esa clase. La matriz de confusión se puede utilizar para calcular otras métricas de rendimiento del modelo, como la precisión y la coincidencia, entre otras.

Discusión

De acuerdo con los resultados obtenidos por el modelo y que se muestran en la Figura 3, el modelo alcanzó un 99.1% de exactitud en la fase de entrenamiento y 99.3 en la fase de validación

La Figura 4 muestra la matriz de confusión que el modelo propuesto obtuvo durante su evaluación. La figura muestra que el modelo propuesto alcanza un desempeño muy alto en la clasificación de las enfermedades de las hojas de limón, reflejando predicciones correctas en la mayoría de las clases y errores de clasificación mínimos, lo cual indica una adecuada capacidad de generalización del modelo. El resultado muestra la robustez del esquema CNN + RF, validando su eficacia en escenarios multiclas.

La Tabla 2 muestra los resultados obtenidos con las diferentes métricas analizadas para cada una de las clases. Se observa que el modelo muestra un comportamiento equilibrado en su desempeño en

todas las métricas para cada categoría, lo que evita el sesgo hacia una categoría en particular. El sistema no solo obtiene una exactitud global alta, sino un rendimiento similar de forma individual.

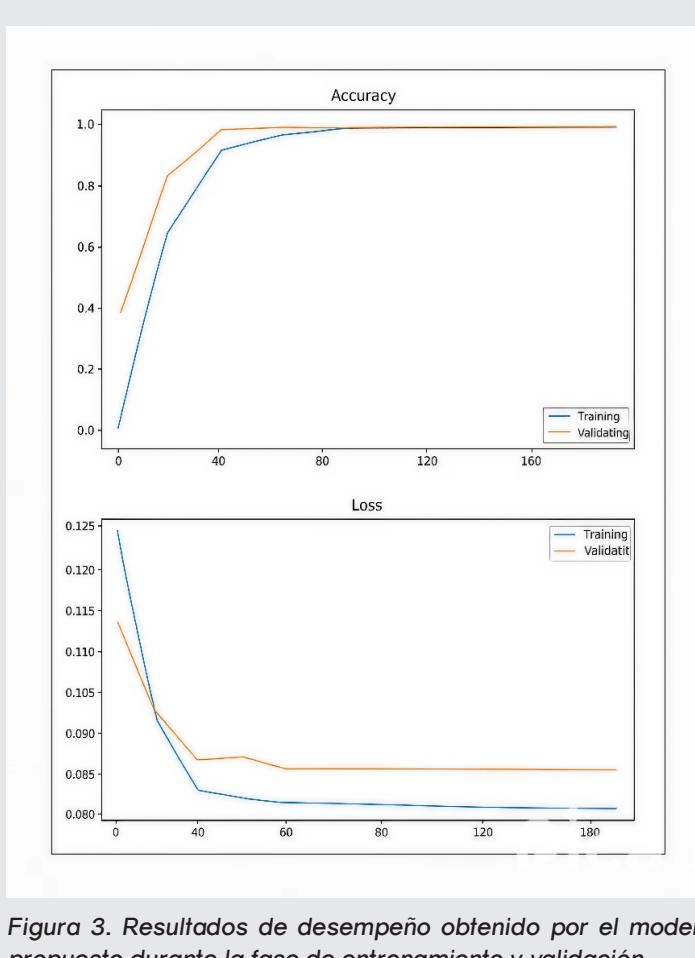


Figura 3. Resultados de desempeño obtenido por el modelo propuesto durante la fase de entrenamiento y validación.

Fuente: Elaboración propia generada por el modelo propuesto.

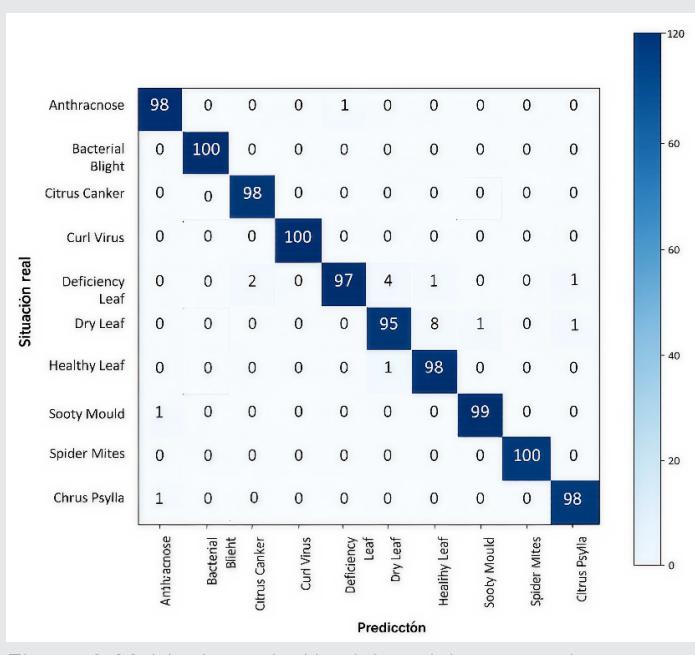


Figura 4. Matriz de confusión del modelo propuesto.

Fuente: Elaboración propia.

Desde nuestra opinión, el desempeño alcanzado se puede atribuir al uso como clasificador de Random Forest en lugar de un modelo puro basado en CNN. RF contribuye a una mejor generalización, lo que reduce la probabilidad de sobreajuste. Además, el incremento del conjunto de datos para lograr su balance logró reducir la probabilidad de sobreajuste y una mayor diversidad de las categorías, que como lo muestran los resultados se refleja en una mejora del desempeño del modelo.

Tabla 2. Tabla de desempeño por clase.

| Categoría | Precisión (%) | Recall (%) | F1-score (%) |
|------------------|---------------|------------|--------------|
| Anthracnose | 98.99 | 98.99 | 98.49 |
| Bacterial Blight | 100.00 | 100.00 | 100.00 |
| Citrus Canker | 98.00 | 98.00 | 98.00 |
| Curl Virus | 100.00 | 100.00 | 100.00 |
| Deficiency Leaf | 95.10 | 97.00 | 96.04 |
| Dry Leaf | 95.00 | 95.00 | 95.00 |
| Sooty Mould | 99.00 | 99.00 | 99.0 |
| Spider Mites | 100.00 | 100.00 | 100.00 |
| Healthy leaf | 97.03 | 98.00 | 97.51 |

Fuente: Elaboración propia.

El tiempo invertido en el proceso de entrenamiento fue de 5987 segundos en el ambiente MGPU (Multiple-Graphics Processing Unit) de colab. Se puede observar claramente que el modelo obtuvo mejores resultados que los obtenidos en los trabajos analizados en la sección 2 y que se resumen en la Tabla 3.

Tabla 3. Distribución del conjunto de datos por categoría.

| Trabajo | Modelos usados | Resultado obtenido |
|------------------------------|---|--------------------|
| Reddy & Thirunavukkarasu [5] | Modelos pre-entrenados ResNet, MobileNet y VGG16 | 89% |
| Kukreja [8] | CNN y SVM | 94.03% |
| Pramanik [9] | DenseNet-201, ResNet-50, ResNet-152V2, and Xception | 94.34% |
| Gupta [10] | CNN y SVM | 89.6% |
| Modelo propuesto | CNN y RF | 99.4% |

Fuente: Elaboración propia.

Por otro lado, la Figura 5 presenta la matriz de confusión, la cual refleja que el modelo propuesto fue capaz de alcanzar valores de precisión del 100% para casi la tercera parte de las categorías evaluadas, una quinta parte obtuvo valores de 99%, otra quinta parte alcanzó un 98% de exactitud y solamente una categoría es la que obtuvo un rendimiento del 95%, el cual es mejor que los valores obtenidos por los modelos con los que se compararon.

Después del entrenamiento y validación del modelo, el modelo y los pesos obtenidos se guardaron como un archivo de datos jerárquico para realizar predicciones

con el modelo. Se utilizaron 100 imágenes aleatorias para realizar predicciones del modelo, los resultados se muestran en la Figura 6. Para cada predicción, se muestra la imagen, el resultado real, y el resultado de predicción junto con el porcentaje de exactitud obtenido con el modelo propuesto.

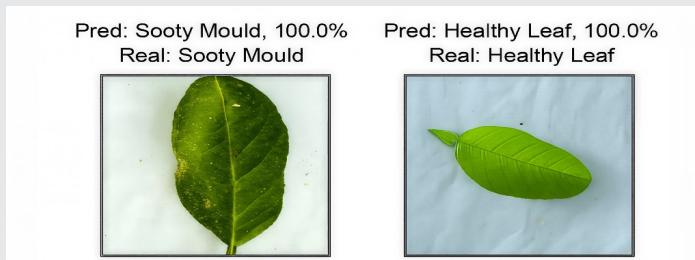


Figura 5. Representación de resultados de predicción obtenidos con el modelo propuesto.

Fuente: Elaboración propia generada por el modelo propuesto.

Aunque los resultados fueron buenos, existen limitaciones como el tamaño del conjunto de datos, que, aunque se aumentó, sigue siendo un conjunto de datos pequeño. Además, este estudio uso imágenes sintéticas, lo que podría generar sesgos al aplicarse en entornos reales. Finalmente, cabe aclarar que debido a que algunos trabajos usaron conjuntos de datos privados o cerrados, es difícil establecer un comparativo justo. Si bien nuestro trabajo utilizó un conjunto de datos público, la heterogeneidad de conjuntos de datos en la literatura puede sesgar la comparación de los modelos.

CONCLUSIONES

En este trabajo se propone un modelo híbrido basado en redes neuronales convolucionales y el algoritmo Random Forest para la identificación y clasificación de enfermedades en las hojas del limón. La red neuronal propuesta demostró tener alto desempeño en la extracción de características, permitiendo al clasificador alcanzar una exactitud de 99%. Además de la exactitud, se evaluaron métricas complementarias como precisión, recall y F1-score, que confirmaron la capacidad del modelo para discriminar de forma confiable entre diferentes enfermedades.

La combinación CNN+RF contribuyó a la generalización a diferencia de modelos basados únicamente en transfer learning. El uso de DCGAN contribuyó en el balanceo de las categorías, reduciendo el riesgo de sobreajuste.

Sin embargo, el modelo tiene algunas limitaciones, incluyendo que el conjunto de datos sigue siendo relativamente pequeño (3600 imágenes) y se generó parcialmente con datos sintéticos. Además, el modelo solo se evaluó en condiciones de laboratorio, por lo que es necesario evaluar el rendimiento en escenarios de campo real.

BIBLIOGRAFÍA

- [1] Secretaría de Agricultura y Desarrollo Rural, “Ratifica México su posición como segundo productor y exportador mundial de limones y limas,” gob.mx. Accessed: Feb. 28, 2025. [Online]. Available: <http://www.gob.mx/agricultura/prensa/ratifica-mexico-su-posicion-como-segundo-productor-y-exportador-mundial-de-limones-y-limas>.
- [2] Secretaría de Agricultura y Desarrollo Rural, “Plagas y enfermedades comunes del limón,” gob.mx. Accessed: Feb. 26, 2025. [Online]. Available: <http://www.gob.mx/agricultura/articulos/plagas-y-enfermedades-comunes-del-limon>.
- [3] C. H. Bock, G. H. Poole, P. E. Parker, and T. R. Gottwald, “Plant Disease Severity Estimated Visually, by Digital Photography and Image Analysis, and by Hyperspectral Imaging,” *Critical Reviews in Plant Sciences*, vol. 29, no. 2, pp. 59–107, 2010, doi: 10.1080/07352681003617285.
- [4] Food and Agriculture Organization of the United Nations, “FAO - News Article: Climate change fans spread of pests and threatens plants and crops, new FAO study.” Accessed: Nov. 10, 2022. [Online]. Available: <https://www.fao.org/news/story/en/item/1402920/icode/>.
- [5] K. R. G. Reddy and K. S. Thirunavukkarasu, “Deep Learning Approach for Disease Detection in Sweet Lemon Leaves to Enhance Citrus Fruit Yield,” in 2024 8th International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2024, pp. 786–792. doi: 10.1109/ICECA63461.2024.10800763.
- [6] J.-F. Yeh, S.-Y. Wang, and Y.-P. Chen, “Crop Disease Detection by Image Processing Using Modified Alexnet,” in 2021 IEEE 3rd Eurasia Conference on Biomedical Engineering, Healthcare and Sustainability (ECBIOS), 2021, pp. 159–160. doi: 10.1109/ECBIOS51820.2021.9510426.
- [7] K. Jayaprakash and S. P. Balamurugan, “Design of Optimal Multilevel Thresholding based Segmentation with AlexNet Model for Plant Leaf Disease Diagnosis,” in 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT), 2022, pp. 1473–1479. doi: 10.1109/ICSSIT53264.2022.9716233.
- [8] V. Kukreja, R. Sharma, and R. Yadav, “Improving Citrus Farming Through Efficient and Accurate Diagnosis of Lemon Citrus Canker Disease with Deep Learning,” in 2023 4th International Conference for Emerging Technology (INCET), 2023, pp. 1–5. doi: 10.1109/INCET57972.2023.10170174.

[9] A. Pramanik, A. Zabed Khan, A. A. Biswas, and M. Rahman, "Lemon Leaf Disease Classification Using CNN-based Architectures with Transfer Learning," in 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2021, pp. 1–6. doi: 10.1109/ICCCNT51525.2021.9579586.

[10] R. Gupta, M. Kaur, N. Garg, H. Shankar, and S. Ahmed, "Lemon Diseases Detection and Classification using Hybrid CNN-SVM Model," in 2023 Third International Conference on Secure Cyber Computing and Communication (ICSCCC), 2023, pp. 326–331. doi: 10.1109/ICSCCC58608.2023.10176828.

[11] M. Shaheen, "Lemon Leaf Disease Dataset (LLDD)." Accessed: Feb. 26, 2025. [Online]. Available: <https://www.kaggle.com/datasets/mahmoudshaheen1134/lemon-leaf-disease-dataset-lldd>.

[12] F. Konidaris, T. Tagaris, M. Sdraka, and A. Stafylopatis, "Generative Adversarial Networks as an Advanced Data Augmentation Technique for MRI Data," in VISIGRAPP, 2019.

[13] I. Goodfellow et al., "Generative Adversarial Networks," Advances in Neural Information Processing Systems, vol. 3, Jun. 2014, doi: 10.1145/3422622.

[14] A. Bisht, K. Rawat, J. P. Bhati, S. Vats, V. Sharma, and S. Singh, "Generating Images Using Vanilla Generative Adversarial Networks," in 2024 4th International Conference on Technological Advancements in Computational Sciences (ICTACS), 2024, pp. 463–468. doi: 10.1109/ICTACS62700.2024.10840720.

[15] R. Gupta and V. Gupta, "Performance Analysis of Different GAN Models: DC-GAN and LS-GAN," in 2021 7th International Conference on Signal Processing and Communication (ICSC), 2021, pp. 222–227. doi: 10.1109/ICSC53193.2021.9673478.

[16] Mohebbanaaz, M. Jyothirmai, K. Mounika, E. Sravani, and B. Mounika, "Detection and Identification of Fake Images using Conditional Generative Adversarial Networks (CGANs)," in 2024 IEEE 16th International Conference on Computational Intelligence and Communication Networks (CICN), 2024, pp. 606–610. doi: 10.1109/CICN63059.2024.10847379.

[17] Google, "Google Colab," Google collaborative. Accessed: Mar. 01, 2025. [Online]. Available: <https://colab.research.google.com/>.

