

Detección de enfermedades en hojas de tomate usando aprendizaje profundo

Edgar R. Arredondo-Basurto¹, J. Félix Serrano-Talamantes^{1,*}, Mauricio Olguín-Carbajal¹,
 Jacobo Sandoval-Gutiérrez², Juan C. Herrera-Lozada¹,
 Israel Rivera-Zárate¹, Miguel Hernández-Bolaños¹

¹ Instituto Politécnico Nacional,
 Centro de Innovación y Desarrollo Tecnológico en Cómputo,
 México

² Universidad Autónoma Metropolitana, Unidad Lerma,
 México

{jfserrano, molguinc, jlozada, irivera, mbolanos}@ipn.mx, earredondob2500@alumno.ipn.mx,
 j.sandoval@correo.ler.uam.mx

Resumen. En 2023, el sector agrícola de México representó el 2.2% del Producto Interno Bruto (PIB). Sin embargo, este sector se encuentra continuamente amenazado por bacterias, virus o enfermedades que podrían tener un impacto financiero significativo en los agricultores. Por esta razón, la identificación y detección temprana de enfermedades adquiere una gran relevancia. Para lograrlo, el agricultor debe contar con una formación integral que abarque diversas disciplinas, experiencia en el reconocimiento de síntomas y un conocimiento profundo de la amplia gama de causas de estas enfermedades. En la presente investigación, centrada en las plantas de tomate, se desarrolló un sistema de reconocimiento de imágenes basado en técnicas de aprendizaje profundo (deep learning) para identificar enfermedades en los cultivos de tomate y ponerlo a disposición de los usuarios finales a través de un sistema web. Las pruebas realizadas muestran hasta un 98% de precisión en la clasificación, lo que subraya su potencial en aplicaciones agrícolas.

Palabras clave. Red neuronal artificial, clasificador, aprendizaje profundo (deep learning), aprendizaje automático (machine learning).

Tomato Leaf Disease Detection Using Deep Learning

Abstract. In 2023, Mexico agricultural sector represented 2.2% of GDP, however, this sector is continuously threatened by bacteria, virus or diseases that could have a significant financial impact on farmers. This is why early

identification and detection of diseases become relevant. To achieve this, the farmer must have comprehensive training encompassing various disciplines, experience in symptoms recognition, and a deep knowledge of the wide range of causes of these diseases. In the present research, focused on tomatoes plants, an image recognition system based on deep learning techniques was developed to identify diseases in cultivated tomatoes and to make it available to end users through a web system. The tests performed show up to 98% accuracy in classification, which underlines its potential in agricultural applications.

Keywords. Artificial neural network, classifier, deep learning, machine learning.

1. Introducción

En el 2023 el sector agrícola en México representó el 2.2% del PIB. En el último trimestre de ese año se situó en torno a los 627 mil millones de pesos mexicanos. En México los cultivos con mayor crecimiento fueron: agave (69.6%), cebolla (23.2%), zarzamora (11.1%), plátano (5.7%) y tomate (1.4%) [1].

El principal estado productor de tomate del país es Sinaloa con 684,333 toneladas en 2020, seguido por San Luis Potosí (380,175 T), Michoacán (248,499 T), Zacatecas (234,866 T) y Jalisco (196,897 T), estados que en conjunto producen el 51.8% del total nacional; mientras que

Culiacán es el principal municipio productor con 230,826 toneladas. El 99% del volumen total se envía a Estos Unidos, seguido de Canadá y Japón [2].

De acuerdo con la Secretaría de Agricultura y Desarrollo Rural (SADER) en el mundo surgen nuevas amenazas a los vegetales y cada vez están más cerca de las fronteras de la República Mexicana, por lo que el país debe estar en alerta permanente. De no tener el cuidado sanitario adecuado, se reduciría en un 10% el sistema de producción agrícola, según estima la dependencia gubernamental. Aunado a esto, el arribo de nuevos fitopatógenos ha logrado confundirse con aquellos conocidos, impidiendo que el técnico tenga certidumbre de que patógeno se trata y por consiguiente no tenga éxito al aplicar un control fitosanitario adecuado en el cultivo [3].

Por su parte, las enfermedades se consideran uno de los principales factores limitantes en el cultivo de tomate. Por ejemplo, el tizón temprano (*Alternaria solani*) es una de las enfermedades más comunes y significativas a nivel mundial, capaz de causar una reducción considerable en la producción de tomate, además de provocar serios daños en los frutos. Del mismo modo, el tizón tardío (*Phytophthora infestans*) es la enfermedad más devastadora que afecta al tomate. Puede destruir plantaciones completas en menos de una semana y convertirse en una fuente de infección para otras plantas [2].

Por lo tanto, resulta fundamental facilitar la identificación y detección temprana de las enfermedades tan pronto como surjan, ya que esto permite mitigar su impacto y prevenir su propagación. Esta acción puede ser decisiva para seleccionar el tratamiento adecuado, limitando la propagación y evitando pérdidas económicas significativas. Para lograrlo, el agricultor debe contar con una formación integral que abarque diversas disciplinas, experiencia en el reconocimiento de los síntomas, y un conocimiento profundo sobre la gran diversidad de causas que provocan estas enfermedades. Esta propuesta retoma el estado del arte en la detección de enfermedades del tomate, en particular a través de algoritmos de aprendizaje profundo, con la finalidad de crear un sistema Web clasificador que asista a los agricultores en el diagnóstico de posibles enfermedades en la planta

del tomate, utilizando el reconocimiento de imágenes de los síntomas visibles que las mismas presentan. A través de esta herramienta, se busca identificar las afecciones de manera rápida y precisa, mejorando la eficiencia en el manejo de enfermedades y reduciendo las pérdidas en los cultivos.

2. Estado del arte

R. Backes et al. [4] propusieron el procesamiento de imágenes de hojas de plantas con el objetivo de caracterizarlas en función del color de la textura. Este enfoque consiste en tratar la imagen de la hoja de árbol de manera individual para cada canal de color, modelándola como una superficie. Los resultados obtenidos demostraron el potencial de este desarrollo, superando modelos de análisis como los Momentos de Cromaticidad y los descriptores EEE de Gabor.

Debido a la necesidad de identificar especies diferentes de árboles, Mzoughi et al. desarrollaron un sistema automatizado de identificación de plantas usando técnicas de visión por computadora [5]. Los autores identificaron dos aproximaciones diferentes para el análisis de las características de las hojas: a) el conocimiento botánico específico y b) la división en partes específicas de la hoja a analizar. Su técnica de dividir la hoja en partes, como lo hacen los expertos en botánica, les proporcionó mejores resultados que con otras técnicas vigentes en ese momento. Hicieron uso de 3,070 imágenes para un total de 70 especies diferentes.

Yanikoglu et al. desarrollaron un sistema automático de identificación de plantas [6], en el que se presentaron varias de las dificultades más comunes para dicha identificación, tales como la iluminación, la pose, las variaciones de orientación, la diferencia de formas dependientes de la madurez o composición de la planta. Su sistema presentó una eficiencia entre el 61 y 81 % clasificando 126 especies de plantas.

Basado en investigaciones anteriores, sobre identificación de plantas, Prasad presentó una investigación para identificar plantas enfermas [7], en un sistema automatizado usando visión móvil. Su desarrollo consistió en un sistema cliente servidor para detección, y diagnóstico de

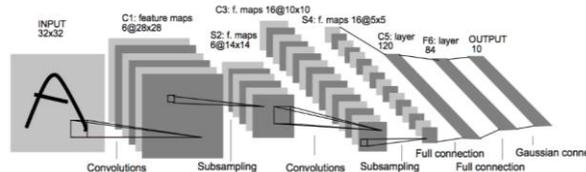


Fig. 1. Arquitectura de una Red Neuronal Convolutional

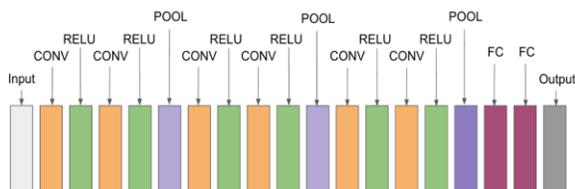


Fig. 2. Ejemplo de arquitectura CNN

enfermedades en las hojas usando una transformada Wavelet de Gabor en conjunto con una matriz de co-ocurrencia como sistema de reconocimiento de patrones, logrando un reconocimiento de hasta un 93% de eficiencia en condiciones óptimas.

En 2017, Brahimi et al. Propusieron un sistema clasificador con aprendizaje automático [8], el cual analiza imágenes de las hojas mediante una Red Neuronal Convolutional o CNN como algoritmo de aprendizaje. Dicha propuesta, utilizó un conjunto de 14,828 imágenes de hojas de tomate infectadas con nueve diferentes enfermedades. Una de las ventajas de usar la CNN fue la extracción automática de características con la finalidad de utilizar directamente las imágenes sin pre-procesamiento. Otra de sus aportaciones es que uso un método de visualización con la finalidad de comprender y poder localizar regiones infectadas en las hojas de la planta de tomate a analizar. Sus resultados alcanzaron una eficiencia de entre 98% y 99%. También se han desarrollado otros trabajos con resultados destacables, los cuales utilizan redes neuronales convolucionales [9, 10, 11].

Foysal et al. propusieron un sistema automatizado de clasificación de enfermedades en el tomate [12], usando Deep Learning en una Red Neuronal Convolutional, su propuesta uso una Red Neuronal Convolutional Profunda (DCNN, Deep Convolutional Neural Network,) de 15 capas. El sistema fue capaz de identificar hasta cinco diferentes tipos de enfermedades con una alta tasa de precisión y una baja tasa de entropía.

En 2020, Muchtar et al. [13] desarrollaron un sistema para la detección de la enfermedad fúngica Septoria en hojas de tomate, utilizando como plataforma de procesamiento una tarjeta Intel Neural Compute Stick. También usaron una CNN, pero usando un sistema integrado por una Raspberry Pi en conjunto con una NCS (Neural Computing Stick). El rango de precisión fue desde un 82 hasta un 100% con una tasa de precisión del 95.89%.

Un desarrollo más reciente, de 2021, se llevó a cabo por Brdar et al. para detectar la vulnerabilidad de ciertos tipos de tomates a infecciones fúngicas [14], En este estudio se revisaron imágenes hiperespectrales cercanas al infrarrojo (NIR) en el rango de 1,000 a 1,700 nanómetros y técnicas de aprendizaje automático con la finalidad de construir modelos que permitan predecir de forma automática la susceptibilidad de ciertas cepas de tomates a infecciones fúngicas en el futuro. Dicho desarrollo utilizó modelos de regresión por validación K-fold cruzada, XG_Boost y Random Forest. Los resultados indicaron que ciertas cepas de tomate tenían una predisposición más alta a los contagios fúngicos con una correlación de hasta 0.92.

En 2022 los investigadores Jie Di y Qing Li usaron una Red Neuronal Convolutional Mejorada (ICNN, improved Convolutional Neural Network) como método de detección de enfermedades en hojas de manzana [15]. Proponiendo un nuevo modelo de detección de objetivo de enfermedades de hoja de manzana basado en Deep Learning. Usaron 1,404 imágenes con cuatro diferentes tipos de enfermedades para el modelado y como método de evaluación. Los investigadores propusieron un ajuste de tamaño y reorganización, la red neuronal usó núcleos de convolución de 1x1, 1x1, y 3x3, con el objetivo de llevar a cabo la reducción de la dimensionalidad de características y lograr un incremento en la profundidad de la red sin aumentar la complejidad computacional. Los resultados obtenidos mostraron una precisión media promedio del 99.9%.

En 2024 Ahmed et al. desarrollaron un sistema de visión por computadora para detectar enfermedades en guayabas [16]. El sistema desarrollado para detección y categorización se basó en una arquitectura CNN. El conjunto de datos usado fue de 306 imágenes para frutas y

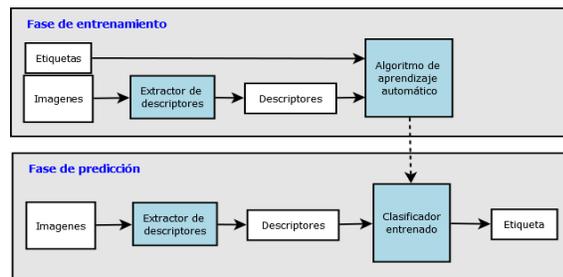


Fig. 3. Clasificación con algoritmos de aprendizaje automático

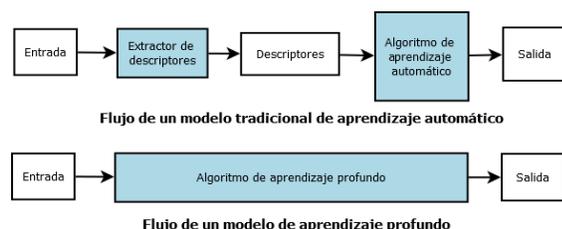


Fig. 4. Diferencias entre la clasificación con modelos de aprendizaje automático y aprendizaje profundo

hojas de guayaba, sanas y enfermas. Se clasificaron cuatro enfermedades y se obtuvo una precisión promedio de 74% para entrenamiento y de 34% para pruebas del conjunto de datos usados.

3. Marco teórico

3.1 Algoritmos de aprendizaje profundo

Las CNN son redes neuronales con el propósito de simular el comportamiento de la corteza visual, por lo que asumen que la entrada de la red son imágenes. La arquitectura de una CNN está compuesta por tres capas: Capa convolucional (CONV), capa de agrupamiento (*pooling*) y capa conectada completamente (*fully-connected*), ver figura 1.

Generalmente, a cada capa convolucional le sigue una capa con funciones de activación, como ReLU (figura 2). Finalmente, la parte final de la red está formada por capas completamente conectadas (*fully connected*). Debe mencionarse que los pesos de las capas son aprendidos en la etapa de entrenamiento, pero sólo en las capas completamente conectadas y las convolucionales.

Existen distintas arquitecturas con un nombre asignado. Las más comunes son las siguientes:

- LeNet. Es una red CNN que se desarrolló en los años noventa del siglo veinte por Y. LeCun.
- AlexNet. Desarrollado por A. Krizhevsky, I. Sutskever y G. Hinton.
- GoogLeNet. La ganadora del ILSVRC 2014.

Asimismo, el aprendizaje por transferencia (*Transfer Learning*) se basa en reutilizar elementos desarrollados en procesos previos de aprendizaje automático y aplicarlos a nuevos modelos con usos y aplicaciones similares. Este enfoque permite compartir el conocimiento adquirido entre diferentes modelos. El concepto clave de este tipo de aprendizaje es la generalización, es decir, transferir el conocimiento previamente adquirido para que pueda ser utilizado por otro modelo en distintos escenarios o condiciones. En este caso, no se necesita reconstruir y volver a entrenar un modelo de aprendizaje desde cero.

3.2 Comparación de clasificación de imágenes usando modelos de aprendizaje automático y aprendizaje profundo

La clasificación de imágenes usando algoritmos de aprendizaje automático está compuesta de dos fases:

1. Fase de entrenamiento. En esta fase se entrena el algoritmo usando un conjunto de datos pre-clasificados (muestras etiquetadas).
2. Fase de predicción. En esta fase se utiliza el algoritmo entrenado para predecir la etiqueta de imágenes fuera del conjunto de entrenamiento.

La fase de entrenamiento, para el problema de clasificación de imágenes, tiene dos etapas principales:

1. Extracción de descriptores. En esta etapa se hace uso del conocimiento general en el área para seleccionar y extraer descriptores que serán usados por el algoritmo de aprendizaje automático, de acuerdo con el tipo de imágenes a clasificar. HOG y SIFT (Histograma de Gradientes Orientados y

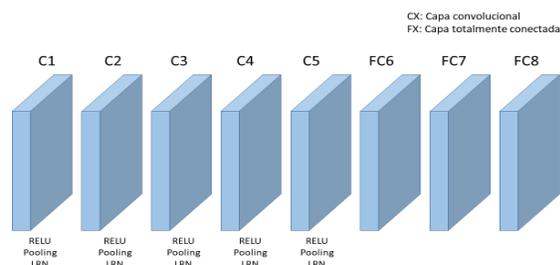


Fig. 5. Arquitectura CNN propuesta para reconocimiento de enfermedades en hojas de tomate

Tabla 1. Conjunto de datos (enfermedades de las hojas del tomate)

| Índice | Clase | Número de imágenes | Porcentaje |
|--------------|--|--------------------|----------------|
| 0 | Virus rizado del tomate amarillo | 4,032 | 24.56% |
| 1 | Virus mosaico del tomate | 325 | 1.98% |
| 2 | Corynespora cassiicola. Mancha blanca | 1,356 | 8.26% |
| 3 | Araña Roja | 1,628 | 9.92% |
| 4 | Septoria | 1,723 | 10.49% |
| 5 | Passalora fulva. Moho en la hoja | 904 | 5.51% |
| 6 | Tizón tardío del tomate (phytophthora infestans) | 1,781 | 10.85% |
| 7 | Tizón temprano (Alternaria Solani) | 952 | 5.80% |
| 8 | Mancha bacteriana | 2,127 | 12.94% |
| 9 | Hoja sana | 1,591 | 9.69% |
| TOTAL | | 16,419 | 100.00% |

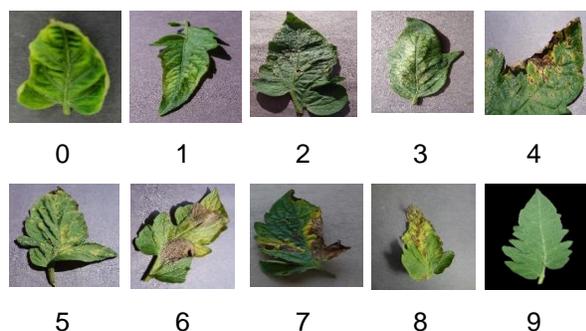


Fig. 6. Conjunto de imágenes identificando las enfermedades de la hoja del tomate

Transformación de Características Invariante a la Escala, respectivamente) son ejemplos

de descriptores usados en la clasificación de imágenes.

- Entrenamiento del modelo. En esta etapa se hace uso de un conjunto de entrenamiento compuesto por descriptores de imágenes y sus etiquetas correspondientes para entrenar el modelo de aprendizaje automático.

Ahora, en la fase de predicción, se aplica el mismo proceso de extracción de descriptores a imágenes nuevas y los descriptores obtenidos se pasan al algoritmo de aprendizaje automático para predecir la etiqueta o clase a la que pertenece, ver figura 3.

En los algoritmos tradicionales de aprendizaje automático, este proceso es elaborado manualmente, es decir, la selección y extracción debe ser diseñada e implementada por el programador, con base en el conocimiento y recomendaciones de expertos. En cambio, en los algoritmos de aprendizaje profundo, la ingeniería de descriptores es realizada automáticamente por el algoritmo. La ingeniería de descriptores es costosa, consume tiempo importante y requiere de cierto nivel de experiencia.

Un descriptor mal seleccionado condena al fracaso al resto del clasificador. Es así como los algoritmos de aprendizaje profundo prometen resultados más precisos comparados con los algoritmos tradicionales de aprendizaje automático, con menos o incluso sin ingeniería de descriptores, ver figura 4.

4. Metodología

La metodología propuesta consta de las siguientes etapas:

- Implementación del clasificador,
 - Calibración fina,
 - Pruebas y resultados experimentales,
- Implementación del sistema Web.

5. Implementación del clasificador

5.1. Calibración final

En la fase de entrenamiento de este desarrollo, se llevó a cabo un proceso de calibración fina de una red pre-entrenada en ImageNet, con el fin de

Tabla 2. Conjunto de pruebas de validación para comprobar que nuestra metodología es la adecuada

| ID | Descripción de la prueba | Resultado |
|----|---|---|
| 1 | Seleccionar imágenes que son previamente conocidas, y etiquetadas | Se clasifica la imagen en alguna de las 9 enfermedades o en una hoja sana |
| 2 | Seleccionar imágenes que no corresponden con ninguna de las 10 clases validadas | La respuesta indica que la imagen procesada no corresponde con ninguna de las 10 clases validadas |
| 3 | Comprobar que la eficiencia de clasificación sea mayor al 98% | El resultado de clasificación fue de 98.82% |
| 4 | Comprobar que el tiempo de respuesta sea menor a 10 ms. | 5.82 mseg |

| CLASE/PREDICCIÓN | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|------------------|-----|-----|-----|------|-----|----|-----|-----|-----|-----|
| 0 | 336 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 350 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 |
| 2 | 0 | 0 | 425 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 1 | 0 | 0 | 1071 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 1 | 0 | 0 | 0 | 314 | 0 | 1 | 0 | 3 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 75 | 0 | 0 | 0 | 0 |
| 6 | 18 | 1 | 0 | 0 | 0 | 1 | 257 | 3 | 0 | 1 |
| 7 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 196 | 2 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 379 | 3 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 191 |

Fig. 7. Matriz de confusión de las 10 clases validadas

transferir este aprendizaje previo al modelo de detección de enfermedades del tomate. Consiste en sustituir la capa de salida de la red por una nueva capa que contiene las distintas clases que queremos clasificar.

Los modelos pre-entrenados utilizados en esta propuesta fueron la Alex Net y la GoogLeNet; la primera de ellas fue ganadora en la competencia ILSVRC de 2012 y la segunda en 2014. En particular, el framework de aprendizaje profundo Caffe tiene su versión de la red Alex Net llamada CaffeNet [18]. Esta red contiene cinco capas convolucionales y tres capas completamente conectadas. La función de activación ReLu se aplica después de cada capa convolucional y completamente conectada. De acuerdo con lo anterior, la red pre-entrenada de AlexNet que se utilizó para este desarrollo se configuró con cinco redes convolucionales de C1 a C5, y con las tres redes totalmente conectadas representadas como FC6, FC7 y FC8 en la figura 5.

Por otra parte, el conjunto de datos utilizado para el proceso de calibración fina consistió en 16,419 imágenes distribuidas en diez clases, las cuales corresponden a 9 enfermedades del tomate por identificar y 1 de plantas sanas. como se muestra en la tabla 1.

Estas imágenes forman parte del repositorio de acceso abierto PlantVillage.org y fueron compartidas por Brahim M [8]. Asimismo, en el caso de la propuesta de Brahim et al, se utilizaron solo clases correspondientes a enfermedades del tomate, sin considerar plantas sanas, por lo que en nuestra propuesta añadimos el conjunto de las imágenes sanas.

Las imágenes originales son a color y muestran únicamente una hoja de tomate sobre un fondo homogéneo, lo que simplifica las actividades de preprocesamiento, en virtud de que únicamente se requiere convertir la imagen a escala de grises, ver figura 6.

Continuando con el proceso de calibración fina, los modelos de la CaffeNet y GoogLeNet están disponibles en el repositorio oficial de Caffe en <https://github.com/BVLC/caffe/tree/master/models/>. Los archivos incluidos con cada modelo y su finalidad son los siguientes.

- *.caffemodel: Contiene los pesos de la red.
- train_val.prototxt: Definición del modelo en el que se especifican los datos de entrada, el número de salidas y la especificación de cada capa de la red convolucional.
- solver.prototxt: Definición del solucionador, el cual es responsable de la optimización del modelo. En este archivo se definen los parámetros de ejecución.

El proceso de entrenamiento de la red se encuentra acompañado de etapas periódicas de prueba que sirven para evaluar el desempeño de la red. Es así como el conjunto de imágenes de entrada debe separarse en dos subconjuntos, uno exclusivamente para entrenamiento y otro para pruebas de la red.

Del conjunto de datos disponible, mostrado en la tabla 1, se eligió un 80% del total de imágenes para usarse en el entrenamiento y el restante para las pruebas. La separación se llevó a cabo de forma aleatoria manteniendo la proporción por cada clase.

Tabla 3. Métricas de desempeño del clasificador

| Métrica | Fórmula |
|---------------------|---|
| Accuracy | $\frac{\sum_{i=1}^N 1_{\hat{Y}_i=Y_i}}{N}$ |
| Precision | $\frac{\sum_{j=1}^N 1_{\hat{Y}_j=Y_j} * 1_{Y_j=i}}{\sum_{j=1}^N 1_{\hat{Y}_j=i}}$ |
| Precision total | $\frac{\sum_{i=1}^C Precision_i}{C}$ |
| Recall _i | $\frac{\sum_{j=1}^N 1_{\hat{Y}_j=Y_j} * 1_{Y_j=i}}{\sum_{j=1}^N 1_{Y_j=i}}$ |
| Recall total | $\frac{\sum_{i=1}^C Recall_i}{C}$ |
| F _i | $2 \cdot \frac{Precision_i \cdot Recall_i}{Precision_i + Recall_i}$ |
| F total | $\frac{\sum_{i=1}^C F_i}{C}$ |

donde:

$Y_i, (\hat{Y}_i)$ representan el índice real (predicho) de la clase de la imagen.

N representa el número de imágenes, en este caso son 3,637.

C representa el número de clases, en este caso son 10.

$1_{a=b}$ representa la función indicatriz, la cual es uno si la condición $a=b$ se cumple, en caso contrario es cero.

Los resultados de la calibración fina de ambas redes son los siguientes:

- CaffeNet. Precisión mayor al 99% en 8 mil iteraciones (aproximadamente en 120 horas en un equipo con CPU i3, 4GB RAM).
- GoogLeNet. Precisión mayor al 97% en 2 mil iteraciones, en casi el triple de tiempo que la CaffeNet; en virtud de que requeriría de tiempo adicional considerable para alcanzar o superar a la CaffeNet, se seleccionó esta última como clasificador en las actividades posteriores.

5.2. Pruebas y resultados experimentales

Se definieron 4 pruebas generales como método de validación, cuyos resultados se muestran en la tabla 2.

En particular, para medir la efectividad de clasificación, se registraron las predicciones del conjunto de pruebas en la matriz de confusión presentada en la figura 7 y se calcularon las métricas descritas en la tabla 3 [8].

Con base en dichos cálculos, los valores obtenidos de cada métrica son los siguientes:

- Accuracy: 98.82%,
- Precision total: 98.45%,
- Recall total: 98.53%,
- F total: 98.46%.

Adicionalmente, en las figuras 8, 9 y 10 se presentan algunas de las pruebas de clasificación, correspondientes a una hoja con mancha bacteriana, sana y con virus del rizado amarillo del tomate, respectivamente.

Por otra parte, se hizo también un estudio comparativo con respecto a las referencias mencionadas en el estado del arte, como se anotó en la tabla 4.

Cabe mencionar que normalmente el método usado para medir y extraer rasgos de textura es la matriz de co-ocurrencia usando niveles de gris en la imagen (GLCM o Grey Level Co-occurrence Matrix).

Esto produce histograma en niveles de grises de 2 dimensiones para una dupla de píxeles (el vecino y el de referencia). La relación espacial entre el píxel de referencia y su vecino puede ser en cualquiera de las 8 direcciones. Cuando se



Fig. 8. Clasificación de la imagen de una hoja con mancha bacteriana

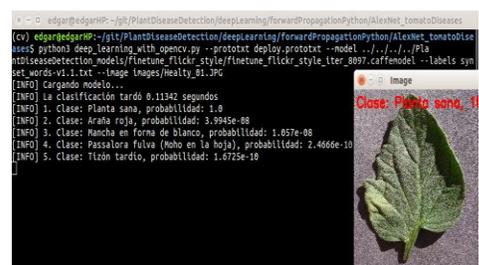


Fig. 9. Clasificación de la imagen de una hoja sana

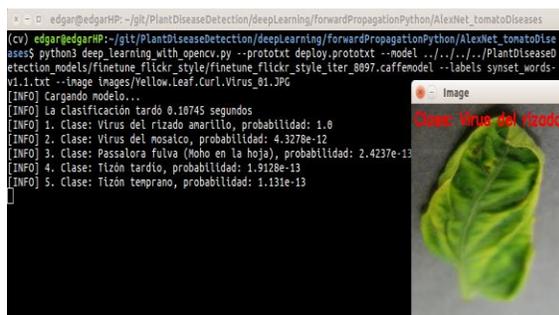


Fig. 10. Clasificación de la imagen de una hoja con el virus del rizado Amarillo del tomate

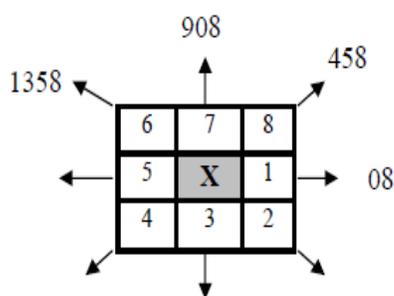


Fig. 12. Los ocho vecinos del píxel X de acuerdo con el ángulo deseado para calcular la matriz de coocurrencia

Tabla 4. Análisis comparativo con respecto a las referencias.

| Estudio | Descriptor | Clasificador | Eficiencia(%) |
|-----------------------|---|--------------|---------------|
| Prasad 2016[6] | GWT, GLCM | K-NN | 93% |
| Mokhtar et al 2015[9] | GLCM | SVM | 99.80% |
| Semary et al 2015[10] | Color Momentos, GLCM, wAVELETS | SVM | 92% |
| Dandawate 2015[11] | SIFT | SVM | 93.79% |
| Raza et al 2015[17] | Estadísticas globales y locales | SVM | 89.93% |
| Brahimi et al [8] | GLCM | CNN | 99.19% |
| Esta propuesta. | GLCM | CNN | 99.19% |

trabaja en un sistema de relación “espacialmente invariante” se seleccionan cuatro direcciones N, NE, E y SE y posteriormente se promedian, ver figura 12.

El rendimiento del clasificador entrenado se comparó con el de otros clasificadores desarrollados en proyectos de detección de enfermedades vegetales. La tabla 4 presenta información general sobre estos proyectos. Además, se incluyen gráficos que analizan el número de imágenes (figura 13), el número de enfermedades (figura 14) y la eficiencia de clasificación de cada estudio (figura 15), incluido el clasificador desarrollado, nombrado SIET (Sistema de Identificación de Enfermedades del Tomate), para pronta referencia.

La tabla 5 muestra la comparación particular con el mejor resultado documentado en Brahimi et al. [8], dado que dicha investigación comparte objetivos y alcances similares con esta propuesta.

6. Implementación del sistema Web

El clasificador se puso a disposición de usuarios finales a través de un sistema Web (ver figura 16) con las características siguientes:

- Requisitos funcionales. Selección de imágenes desde el sistema de archivos local, clasificación en alguna de las diez categorías referidas indicando la probabilidad, la eficiencia de clasificación debe ser mayor al 90%.
- Requisitos mínimos de hardware: 1GB de RAM, 1 CPU Intel core i3, 30 GB de espacio en disco.
- Tecnologías de desarrollo. Java, OpenCV, Maven, Spring boot, JSF y PrimeFaces.
- Características de equipos clientes. Es suficiente con poder ejecutar un navegador Web con soporte vigente, lo que implica contar con al menos un procesador de 1 GHz y 2 GB de RAM.

7. Conclusiones

En esta propuesta se implementó una solución de aprendizaje profundo como alternativa al problema del reconocimiento de enfermedades del tomate. Este tipo de solución demostró tener mejor desempeño en comparación con trabajos similares que proponen un algoritmo de aprendizaje automático como solución, así como

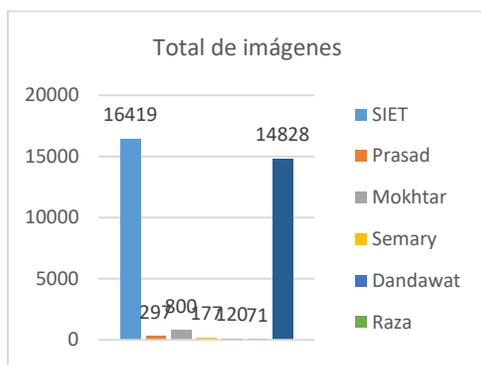


Fig. 13. Comparación del tamaño del conjunto de datos

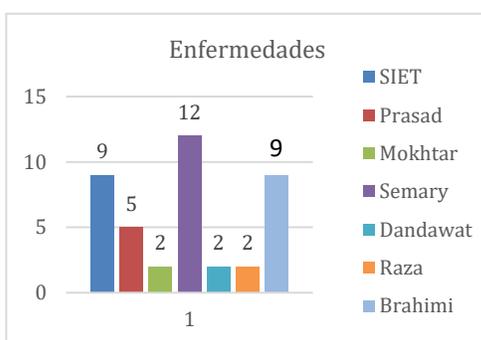


Fig. 14. Análisis comparativo del número de enfermedades identificables

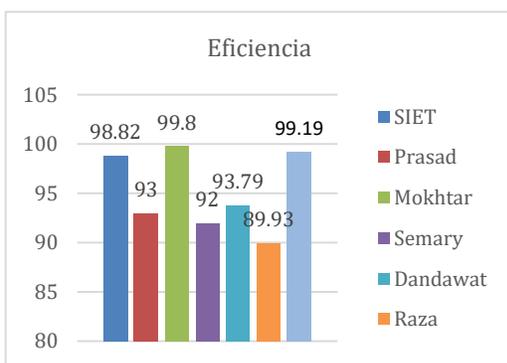


Fig. 15. Comparación de la eficiencia de la clasificación

resultados similares en comparación con trabajos que utilizan otros algoritmos de aprendizaje profundo.

Adicionalmente, con respecto al tiempo de procesamiento, la etapa más demandante es la calibración. En dicha etapa se notó el beneficio que equipos con mayores recursos que el

disponible para esta propuesta, en virtud de que se podría reducir el tiempo de calibración y, por tanto, permitiría hacer pruebas con otros modelos y seleccionar el mejor.

Por otra parte, para fines de esta propuesta, la efectividad de clasificación, mayor al 98%, fue suficiente para el sistema Web.

7.1. Contribuciones

Las principales contribuciones de este trabajo son:

- 1.- Se aplicó una transferencia de aprendizaje (Transfer Learning) usando una red pre-entrenada para reconocimiento de imágenes y se desarrolló una arquitectura propia basada en una propuesta anterior [8].
- 2.- Se tiene un desempeño equivalente con un 99.19% de eficiencia, sin embargo, al usar menos capas se tiene un reconocimiento en menor tiempo en sistemas equivalentes.
- 3.- Se agregó una clase extra de hojas sanas la cual no se usaba en otras propuestas.
- 4.- Se implementó en un sistema Web para ser usado por agricultores que la necesiten.

7.2. Trabajo a futuro

El sistema desarrollado es susceptible de ser mejorado en los siguientes aspectos:

- Implementar y comparar la eficiencia de clasificadores de aprendizaje automático y clasificadores de aprendizaje profundo que consideren arquitecturas distintas, con el objetivo de determinar cuál es la alternativa que mejor desempeño ofrece.
- Ampliar el número de plantas y las enfermedades identificables.
- Almacenar las imágenes que los usuarios proporcionen para incorporarlas al conjunto de datos, con el objetivo de mejorar la eficiencia de clasificación.
- El desarrollo de una aplicación móvil en la que la clasificación de imágenes se haga localmente, sin el uso de un servidor Web, lo que permitiría que las personas puedan usar la aplicación en lugares sin conexión a internet. Lo anterior se considera factible dado

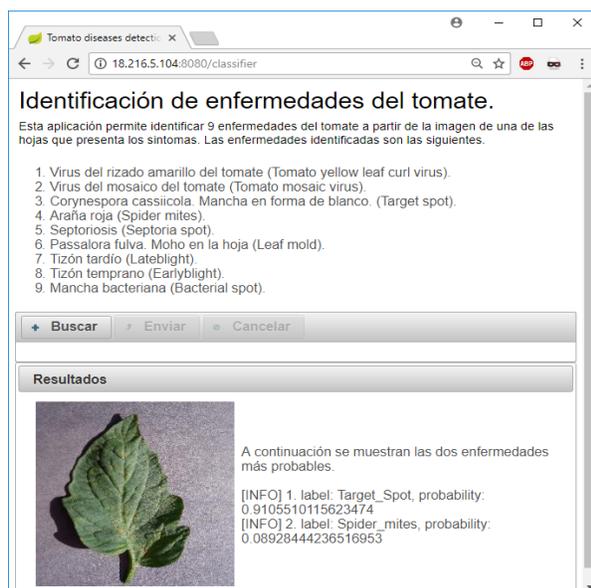


Fig. 16. Sistema Web para la identificación de enfermedades del tomate

el bajo costo computacional que requiere la clasificación de imágenes.

- La viabilidad de usar redes neuronales más avanzadas como Transformers para visión computacional.
- La posibilidad de aplicar técnicas de aumento de datos para mejorar la robustez del modelo.

Reconocimientos

Los autores agradecen al Instituto Politécnico Nacional (IPN), especialmente a la Escuela Superior de Cómputo (ESCOM) y al Centro de Innovación y Desarrollo Tecnológico en Cómputo (CIDETEC) por el apoyo recibido y las facilidades otorgadas para el desarrollo de este trabajo.

Referencias

1. **Servicio de Información Agroalimentaria y Pesquera. Dirección de Análisis Estratégico. (2024).** Comportamiento del PIB Agroalimentario al cuarto trimestre de 2023. (2022:IV-2023:IV). https://www.gob.mx/cms/uploads/attachment/file/895676/Analisis_PIB_4to_trim_2023.pdf.
2. **Servicio de Información Agroalimentaria y Pesquera. Dirección de Análisis Estratégico. (2024).** Datos abiertos de producción agrícola. <https://www.gob.mx/siap/acciones-y-programas/produccion-agricola-33119>.
3. **Andrade-Bustamante, G., Aispuro Hernández, E., Hernández-Montiel, L.G., Holguín Peña, R. J., Rueda-Puente, E. O. (2019).** Antisuero Vs Hongos fitopatógenos En El Cultivo De Tomate En Sonora, México. *Revista Mexicana de Ciencias Agrícolas*, Vol. 10, No. 4, pp. 873–84. Doi: 10.29312/remexca.v10i4.1706.
4. **Backes, A.R., Odemir Martinez, B. (2010).** Plant leaf identification using color and multi-scale fractal dimension. *Lecture Notes in Computer Science*. DOI: 10.1007/978-3-642-13681-8_54.
5. **Itheri Yahiaoui, M., Boujemaa, N. Zagrouba, E. (2013).** Advanced tree species identification using multiple leaf parts image queries. *IEEE International Conference on Image Processing*, Doi: 10.1109/ICIP.2013.6738817.
6. **Yanikoglu, B., Aptoula, E., Tirkaz, C. (2014).** Automatic plant identification from photographs. *Journal of Machine Vision and Applications*. Doi: 10.1007/S00138-014-0612-7.
7. **Prasad, S., Peddoju, S. K., Ghosh, D. (2016).** Multi-resolution mobile vision system for plant leaf disease diagnosis. *Signal, image and video processing*, Vol. 10, No. 2, pp. 379–388. Doi: 10.1007/s11760-015-0751-y.
8. **Mohamed Brahimi, Kamel Boukhalfa, Abdelouahab Moussaoui. (2017).** Deep Learning for Tomato Diseases: Classification and Symptoms Visualization. *Applied Artificial Intelligence*, Doi: 10.1080/08839514.2017.1315516.
9. **Mokhtar, U., El Bendary, N., Hassenian, A. E., Emary, E., Mahmoud, M. A., Hefny, H., Tolba, M. F. (2015).** SVM-based detection of tomato leaves diseases. In *Intelligent Systems' 2014: Proceedings of the 7th IEEE International Conference Intelligent Systems IS'2014, Warsaw, Poland*, Vol. 2, pp. 641–652.

- Springer International Publishing. Doi: 10.1007/978-3-319-11310-4_55.
10. **Semary, N. A., Tharwat, A., Elhariri, E., Hassanien, A. E. (2015).** Fruit-based tomato grading system using features fusion and support vector machine. In *Intelligent Systems' 2014: Proceedings of the 7th IEEE International Conference Intelligent Systems IS'2014*, Vol. 2, pp. 401–410, Springer International Publishing. Doi: 10.1007/978-3-319-11310-4_35.
 11. **Dandawate, Y., Kokare, R. (2015).** An automated approach for classification of plant diseases towards development of futuristic Decision Support System in Indian perspective. In *2015 International conference on advances in computing, communications and informatics (ICACCI)*, pp. 794–799. Doi: 10.1109/ICACCI.2015.7275707.
 12. **Ahmed Foysal, F., Shakirul Islam, M., Abujar, S., Akhter Hossain, S. (2020).** A Novel Approach for Tomato Diseases Classification Based on Deep Convolutional Neural Networks. *Algorithm for Intelligent systems*. Doi: 10.1007/978-981-13-7564-4_49.
 13. **Muchtar, K., Chairuman F., Muhammad Yusuf Kardawi, M., Febriana, A., Zarima, N., Chih-Yang Lin, (2020).** Embedded-based Tomato Septoria Leaf Detection with Intel Movidius Neural Compute Stick. *IEEE 10th Global Conference on Consumer Electronics (GCCE)*. Doi: 10.1109/GCCE53005.2021.9621829.
 14. **Brdar, S., Panic, M., Hogeveen-van Echtelt, E., Mensink, M., Grbović, Ž., Woltering, E.J., Chauhan, A. (2021).** Predicting sensitivity of recently harvested tomatoes and tomato sepals to future fungal infections. *Scientific Reports*. Doi: 10.1038/S41598-021-02302-2.
 15. **Jie Di, Qing Li. (2022).** A method of detecting apple leaf diseases based on improved convolutional neural network. *PLoS ONE*. Doi: 10.1371/JOURNAL.PONE.0262629.
 16. **Khandakar Rabbi Ahmed, Towfika Salam, Subarna Nandi, Nazmul Hasan, Omor Faruk Al Rashid, Shafiun Miraz, Md. Shafi Reza, Farjana Rahman, (2024).** An Identification of Guava Fruit Disease Using ML. *International Conference on Advancement in Electrical and Electronic Engineering (ICAEEE)*. Doi: 10.1109/ICAEEE62219.2024.10561747.
 17. **Raza, S-e-A., Prince, G., Clarkson, J.P., Rajpoot, N.M. (2015).** Automatic Detection of Diseased Tomato Plants Using Thermal and Stereo Visible Light Images. *PLoS ONE*, Vol. 10, No. 4, e0123262. Doi: 10.1371/journal.pone.0123262.
 18. **BVLC reference cafenet (2025).** Disponible en https://github.com/BVLC/caffe/tree/master/models/bvlc_reference_caffenet.

Article received on 09/01/2025; accepted on 24/07/2025.

**Corresponding author is J. Félix Serrano-Talamantes.*