



## **Clasificación de Enfermedades en Tomates con Deep Learning: Un Modelo CNN para la Agricultura**

*1ª Ana María Vega Angarita*

*Estudiante*

*Departamento de ingeniería de sistemas*

*Universidad de Antioquia*

*Medellín, Colombia*

*ana.vega@udea.edu.co*

*2ª Maritza Tabarez Cárdenas*

*Estudiante*

*Departamento de ingeniería de sistemas*

*Universidad de Antioquia*

*Medellín, Colombia*

*maritza.tabarezc@udea.edu.co*

*3ª Johan Sebastian Henao Cañas*

*Estudiante*

*Departamento de ingeniería de sistemas*

*Universidad de Antioquia*

*Medellín, Colombia*

*johan.henao1@udea.edu.co*

### **Contexto de aplicación**

Las enfermedades en cultivos de tomate representan una amenaza significativa para la seguridad alimentaria y la economía agrícola, especialmente en regiones donde el acceso a expertos fitopatólogos es limitado. En este contexto, la inteligencia artificial y el deep learning ofrecen una solución escalable y accesible para agricultores, técnicos y empresas del sector agroindustrial.

Este proyecto se enfoca en desarrollar un modelo capaz de diagnosticar visualmente enfermedades en hojas de tomate a partir de imágenes, imitando la capacidad de un experto humano pero con la velocidad y escalabilidad de un sistema automatizado. Esta tecnología puede ser integrada en aplicaciones móviles para agricultores, sistemas de drones que monitorean cultivos, o incluso estaciones inteligentes de cultivo.

### **Objetivo de Machine Learning**

El objetivo principal de este proyecto es predecir la presencia de una enfermedad en una hoja de tomate mediante el uso de una red neuronal convolucional (CNN), dado un conjunto de imágenes que representan distintas clases de enfermedades, así como hojas saludables. La tarea de predicción es un problema de clasificación multiclase, donde cada imagen debe ser etiquetada en una de las 10 clases predefinidas.

**Objetivo específico:** Desarrollar un modelo de aprendizaje automático basado en CNN que, a partir de una imagen de una hoja de tomate, sea capaz de clasificarla correctamente en una de las 10 categorías posibles, que corresponden a diferentes enfermedades o a la categoría de "saludable". La CNN aprenderá a identificar patrones y características visuales en las imágenes para diferenciar entre hojas sanas y afectadas por diversas enfermedades.

**Dataset:** El conjunto de datos utilizado para este proyecto proviene de la competencia de Kaggle Tomato Leaf Disease Detection. Contiene imágenes de hojas de tomate clasificadas en distintas categorías que corresponden a enfermedades específicas, así como hojas saludables. Este dataset es ideal para problemas de clasificación multiclase usando redes neuronales convolucionales (CNN).



- **Tipo de datos:** imágenes en formato .jpg (11,000 archivos), con una sola imagen .jpeg adicional y un archivo de código en Python (cnn\_train.py) incluido.
- **Tamaño del dataset:** aproximadamente 186 MB en total.
- **Resolución de imágenes:** 256x256 píxeles. Este tamaño es óptimo para entrenamiento con redes convolucionales ya que ofrece suficiente información visual para aprender características relevantes sin generar una carga computacional excesiva.
- **Distribución:**
  - **Entrenamiento (train/):** 10,000 imágenes distribuidas equitativamente en 10 carpetas (1,000 imágenes por clase).
  - **Validación (val/):** 1,000 imágenes también distribuidas en las mismas 10 clases.
- **Clases (balanceadas):**
  - Tomato\_\_Bacterial\_spot
  - Tomato\_\_Early\_blight
  - Tomato\_\_Late\_blight
  - Tomato\_\_Leaf\_Mold
  - Tomato\_\_Septoria\_leaf\_spot
  - Tomato\_\_Spider\_mites Two-spotted\_spider\_mite
  - Tomato\_\_Target\_Spot
  - Tomato\_\_Tomato\_Yellow\_Leaf\_Curl\_Virus
  - Tomato\_\_Tomato\_mosaic\_virus
  - Tomato\_\_healthy

Cada clase tiene exactamente 1,000 imágenes tanto en el set de entrenamiento como en el de validación, lo cual resulta en un dataset balanceado, ideal para entrenar modelos de clasificación sin necesidad de técnicas de remuestreo.

- **Código base:** El dataset incluye un archivo Python (cnn\_train.py) con un modelo CNN básico construido con Keras. Este modelo:
  - Trabaja con imágenes reescaladas a **128x128 píxeles** (aunque las originales son de 256x256).
  - Utiliza una arquitectura secuencial con tres capas convolucionales seguidas por una capa densa y una de salida softmax con 10 neuronas (una por clase).
  - Aplica aumentos de datos (ImageDataGenerator) para robustecer el entrenamiento frente a variaciones comunes.
  - Compila el modelo con el optimizador Adam y la función de pérdida categorical\_crossentropy.



## **Métricas de desempeño**

### **Métricas de desempeño de Machine Learning**

Las métricas de desempeño de Machine Learning permiten evaluar qué tan bien un modelo está resolviendo una tarea, en este caso, la clasificación de enfermedades en hojas de tomate a partir de imágenes. Dado que el problema es de clasificación multiclase, se utilizan métricas que permiten evaluar el comportamiento del modelo tanto en general como por cada clase. En este caso, primero se utiliza la accuracy, que dice cuántas veces el modelo acierta. Por ejemplo, si se le dan 100 imágenes y le acierta a 90, tiene un 90% de accuracy.

Después, como este modelo clasifica varias enfermedades distintas, también es necesario saber cómo se comporta con cada una. Para eso el uso de precisión y recall por clase. La precisión dice qué tan confiable es el modelo cuando dice que una hoja tiene cierta enfermedad. El recall muestra si el modelo es capaz de encontrar todos los casos reales de esa enfermedad. Con estas dos métricas, se puede ver si el modelo es bueno detectando enfermedades difíciles o si se le escapan algunas.

También, el uso de F1-score ayuda a ver un equilibrio entre precisión y recall. Esto es útil sobre todo si algunas enfermedades aparecen poco y no se quiere que el modelo simplemente las ignore.

La matriz de confusión permite ver si el modelo se está confundiendo entre enfermedades parecidas. Por último, durante el entrenamiento también se revisa la pérdida (loss), que indica si el modelo está aprendiendo bien o se quedó estancado.

### **Métricas de desempeño de Negocio**

Más allá de que el modelo funcione bien, es necesario saber si realmente sirve para los agricultores. Una forma es ver si reduce errores humanos, comparando cómo lo hace el modelo frente a una persona que revisa las hojas.

También es importante saber cuánto tiempo ahorra. Por ejemplo, si un técnico tarda una hora en revisar 100 plantas, y el modelo lo hace en 1 minuto, ahí ya se presenta un impacto claro. Otra métrica es el dinero que se puede ahorrar si el modelo detecta una enfermedad a tiempo. Si no se detecta, el cultivo se puede perder. Entonces, si el modelo ayuda a detectarla antes, puede evitar esas pérdidas.

Además, se debe ver si el modelo cubre bien las enfermedades comunes del tomate. Si reconoce la mayoría, funciona. Pero si solo detecta unas pocas, no es tan útil. Y por último, si hay una enfermedad muy grave, es necesario que el modelo tenga mucha precisión justo en esa enfermedad, porque un error en ese caso podría salir caro.

## **Referencias**

- [1]<https://www.kaggle.com/datasets/kaustubhb999/tomatoleaf/>
- [2]<https://rramosp.github.io/2021.deeplearning/content/M04.html>
- [3]<https://acortar.link/iHg9tV>
- [4]<https://acortar.link/HmvO69>