

Prototipo de Aplicación para la Detección de Enfermedades en Cultivos de Tomate

Edgar Rodrigo Arredondo Basurto¹, Eduardo Gutiérrez Aldana², José Félix Serrano Talamantes³

Escuela Superior de Cómputo I.P.N México D.F.

Tel. 57-29-6000 ext 52000 y 52021. Email: ¹edgar_arredondo@outlook.com, ²eduardogutierrezaldana@gmail.com, ³jfserrano@ipn.mx

Resumen – El trabajo terminal presentado consiste en el prototipo de aplicación de reconocimiento de imágenes que permite identificar enfermedades en cultivos de tomate con base en el reconocimiento de imágenes a través de técnicas de aprendizaje profundo. Se logra una eficiencia de clasificación del 98.82%.

Palabras clave – Clasificador, descriptor, aprendizaje automático, aprendizaje profundo, red neuronal artificial.

I. INTRODUCCIÓN

La producción de tomate en México creció a una tasa promedio anual de 3.3% entre 2005 y 2015 para ubicarse en 3.1 millones de toneladas [1]. Sin embargo, las enfermedades son consideradas uno de los principales factores limitantes en el cultivo de tomate. Por ejemplo, el tizón temprano representa una de las enfermedades más comunes en el mundo y puede generar un decremento significativo en la producción de tomate y lesiones serias en los frutos. De forma similar, el tizón tardío es la enfermedad más seria que afecta al tomate. Puede rápidamente, en menos de una semana, arruinar plantaciones enteras y generar fuentes de infección para otras plantas [2].

De esta forma, es importante proveer una identificación y detección temprana en el momento en que una enfermedad se presenta. Lo anterior puede marcar la diferencia para elegir el tratamiento correcto y evitar que la enfermedad se propague. Para lograr esto, el agricultor debe contar con una formación académica, un extenso conocimiento de varias disciplinas, experiencia en los síntomas de las enfermedades así como conocimiento de la gran cantidad y diversidad de causas de las enfermedades.

La arquitectura general de sistemas de clasificación de enfermedades de plantas a través del análisis imágenes consiste de tres etapas: pre-procesamiento, extracción de descriptores y clasificación.

Las etapas de pre-procesamiento y extracción de descriptores implican operaciones que suelen ser complejas y consumir tiempo considerable. Por esta razón es común solicitar asistencia al usuario lo que disminuye el grado de automatización del sistema.

En este proyecto se plantea el uso de un modelo de aprendizaje profundo, específicamente de una Red Neuronal Convolutiva (CNN) como una alternativa a la clasificación de enfermedades en plantas. Este modelo permite que los descriptores sean definidos y obtenidos de una forma completamente automática. Además, permite realizar la etapa de entrenamiento y clasificación directamente sobre las imágenes, sin incluir operaciones de pre-procesamiento. Las características anteriores permiten desarrollar un sistema completamente automatizado en el que la única acción que el usuario debe realizar es proporcionar la imagen de la hoja enferma. Además, esta solución permite agregar más clases al algoritmo clasificador únicamente proporcionando la cantidad suficiente de imágenes durante la etapa de entrenamiento, gracias a que la ingeniería de descriptores es completamente automática.

II. METODOLOGÍA

La metodología propuesta consiste de dos etapas principales (figura 1), el clasificador y el sistema Web. En dichas etapas se realizan las acciones siguientes.

1. Entrenamiento (calibración fina)
2. Identificación de enfermedades (clasificación)

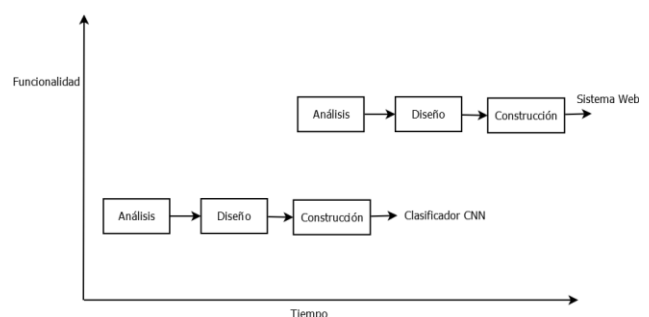


Figura 1. Metodología.

A. Calibración fina de la AlexNet

Una red CNN está formada por dos tipos principales de capas: convolucionales y completamente conectadas. En

las capas convolucionales se realiza la extracción de descriptores y en las capas completamente conectadas la clasificación. El entrenamiento se realiza con el algoritmo de propagación hacia atrás (back propagation). Realizar el entrenamiento desde cero requiere de un conjunto de datos de tamaño considerable (miles de imágenes) para obtener resultados aceptables. Existe otra alternativa de entrenamiento que consiste en la transferencia de aprendizaje desde redes pre-entrenadas, realizando una calibración fina de los parámetros originales de la red. Esto disminuye el tiempo de cómputo y el número de imágenes necesarias.

En la etapa de entrenamiento de este proyecto se realizó un proceso de calibración fina sobre una red pre-entrenada en ImageNet. Esto implica el reemplazo de la capa de salida de la red por una nueva capa que contiene diez clases (nueve enfermedades y planta sana). Dicho procedimiento se realizó con el framework de aprendizaje profundo Caffe [3]. Este framework proporciona modelos de distintas arquitecturas entrenados sobre ImageNet. La arquitectura seleccionada fue la ganadora del *ImageNet Large Scale Visual Recognition Competition (ILSVRC) 2012*, la AlexNet. Caffe cuenta su versión de dicha red que es llamada CaffeNet [4]. Esta red contiene cinco capas convolucionales y tres capas completamente conectadas. La función de activación ReLU es aplicada después de cada capa convolucional y completamente conectada.

El conjunto de datos usado para el proceso de calibración fina de la CaffeNet está formado por 16,419 imágenes distribuidas en diez clases, como se muestra en la tabla 1. Estas imágenes forman parte del repositorio de acceso abierto de PlantVillage.org y fueron compartidas por Brahimi M. [5].

Clase	Número de imágenes
Virus del rizado amarillo del tomate	4032
Virus del mosaico del tomate	325
Corynespora cassiicola. Mancha en forma de blanco	1,356
Araña roja	1,628
Septoriosis	1,723
Passalora fulva. Moho en la hoja	904
Tizón tardío	1,781
Tizón temprano	952
Mancha bacteriana	2,127
Hoja sana	1,591
Total	16,419

Tabla 1. Conjunto de datos.

Del total de imágenes se usó el 80% para realizar el entrenamiento de la red y el restante 20% para pruebas. En la etapa de pruebas se realizó la medición de eficiencia mediante el uso de la matriz de confusión.

La red se entrenó durante 8097 iteraciones, equivalentes a siete días de cómputo en una máquina con un procesador Intel core i3 y 4 GB de RAM. La eficiencia de clasificación obtenida en la etapa de pruebas fue de 98.82%. En la tabla siguiente se muestra la eficiencia por clase.

Clase	Eficiencia [%]
Araña roja	100
Septoriosis	98.59
Mancha bacteriana	99.76
Virus del rizado amarillo	99.91
Planta sana	98.74
Virus del mosaico	100
Mancha en forma de blanco	91.45
Tizón temprano	98.00
Tizón tardío	99.21
Moho en la hoja	100

Tabla 2. Eficiencia de clasificación por clase.

B. Identificación de enfermedades.

Una vez entrenado el modelo hay una variedad de alternativas de lenguajes de programación en los cuales puede realizarse la identificación de enfermedades: Python, C++, Java y Matlab en combinación con Caffe u OpenCV.

Las tecnologías de desarrollo de la aplicación web fueron OpenCV, Java, Spring boot, Maven, JavaServer Faces y Primefaces.

Se generó la aplicación como un jar el cual contiene las dependencias necesarias para la ejecución de la aplicación, incluyendo un servidor Web. Sin embargo, también son necesarias ciertas bibliotecas nativas de OpenCV que no pueden ser comprimidas en el jar. Por tal motivo, además de Java, también es necesario tener OpenCV instalado.

En la figura 2 se muestra la interfaz Web del sistema con la cual el usuario puede identificar la enfermedad que sufre la planta de la imagen que proporcione.



Figura 2. Interfaz Web del sistema.

C. Pruebas del sistema.

Se evaluó el desempeño general del sistema Web mediante la ejecución de las siguientes pruebas.

Id	Descripción de la prueba	Resultado extensión esperado
1	Seleccionar una imagen del sistema de archivos local del dispositivo.	Es posible seleccionar imágenes *.jpg, *.png y *.bmp.
2	Seleccionar un archivo con extensión distinta a las permitidas.	No es posible seleccionar tipos de archivos distintos a los permitidos.
3	Seleccionar una imagen cuya clase sea previamente conocida (etiquetada) y clasificarla.	El sistema clasifica la imagen en alguna de las nueve enfermedades o en hoja sana.
4	Seleccionar una imagen que no corresponda con ninguna de las diez clases válidas.	El sistema debe indicar que la imagen no corresponde a ninguna de las enfermedades identificables u a una hoja sana.
5	Medir la eficiencia de clasificación con imágenes del conjunto de prueba.	La eficiencia de clasificación es superior al 90%.
6	Medir el tiempo de respuesta del sistema Web.	El tiempo promedio es menor a cinco segundos.

Tabla 3. Pruebas del sistema.

III. RESULTADOS.

Los resultados obtenidos son los siguientes:

1. Se permitió la selección de archivos con extensiones jpg, png y bmp.
2. No se permitió la selección de tipos de archivos distintos a los permitidos.
3. El sistema clasificó la imagen y mostró los dos resultados más probables.
4. El sistema clasificó algunas de las imágenes en clases válidas.
5. La eficiencia de clasificación fue del 98.82%.
6. El tiempo promedio de clasificación fue de 582 ms de los cuales 115 ms corresponden únicamente al proceso de clasificación. El resto corresponde a la transmisión de información.

También se comparó el desempeño del clasificador entrenado con otros desarrollados en proyectos de detección de enfermedades en plantas. En la tabla 4 se muestra información general de estos proyectos.

Estudio	Descriptor	Clasificador	Eficiencia [%]
Prasad, 2016 [6]	GWT, GLCM	KNN	93.00
Mokhtar et al. 2015 [7]	GLCM	SVM	99.80
Semary et al. 2015 [8]	Momentos de color, GLCM, descomposición wavelet	SVM	92.00

Dandawate 2015 [9]	SIFT	SVM	93.79
Raza et al. 2015 [10]	Estadísticas locales, globales	SVM	89.93

Tabla 4. Estudios de referencia.

En las figuras siguientes se muestran gráficas que comparan el número de imágenes, el número de enfermedades y la eficiencia de clasificación de cada estudio incluyendo el desarrollado, al cual se hace referencia como SIET (Sistema de Identificación de Enfermedades del Tomate).

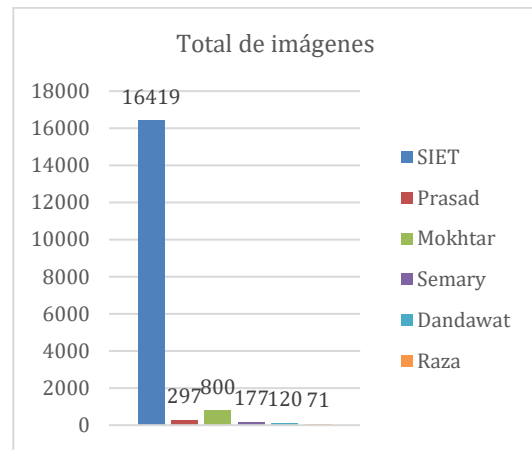


Figura 3. Comparativa del tamaño del conjunto de datos.

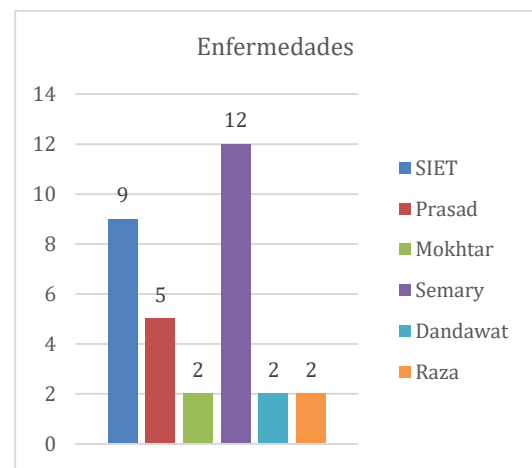


Figura 4. Comparativa del número de enfermedades identificables.

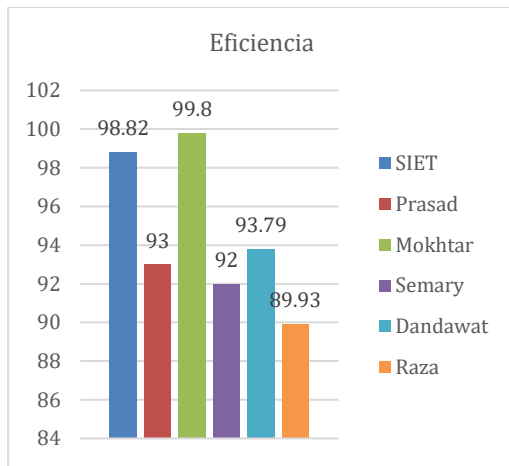


Figura 5. Comparativa de la eficiencia de clasificación.

IV. CONCLUSIONES.

Con el desarrollo de este prototipo 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.

Sin embargo, para realizar esta comparación de forma completamente objetiva es necesario que todas las soluciones consideren el mismo cultivo, las mismas enfermedades y el mismo conjunto de datos. La limitante en este proyecto, para desarrollar este tipo de estudio fue el equipo de cómputo disponible con el que se desarrolló el proceso de entrenamiento.

De esta forma, a continuación se enumeran una serie de directivas propuestas que sirvan de guía para el trabajo a futuro del prototipo desarrollado.

1. 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.
2. Ampliar el número de plantas y las enfermedades identificables.
3. Almacenar las imágenes que los usuarios proporcionen para incorporarlas al conjunto de datos, con el objetivo de mejorar la eficiencia de clasificación.
4. El desarrollo de una aplicación móvil.

RECONOCIMIENTOS

Los Autores agradecen a la Escuela Superior de Cómputo del Instituto Politécnico Nacional por el apoyo recibido y las facilidades otorgadas para el desarrollo del presente trabajo terminal.

REFERENCIAS

- [1] Fideicomisos Instituidos en Relación con la Agricultura (FIRA). Dirección de Investigación y Evaluación Económica y Sectorial. Panorama Agroalimentario. Tomate Rojo 2016 [En línea] Disponible en: <https://www.gob.mx/>.
- [2] CAB International .Plantwise Knoldge Bank. [En línea] Disponible en <https://www.plantwise.org/KnowledgeBank/>.
- [3] Caffe. Deep Learning Framework. [En línea] Disponible en <http://caffe.berkeleyvision.org/>.
- [4] BVLC reference caffe net. [En línea]. Disponible en https://github.com/BVLC/caffe/tree/master/models/bvlc_reference_caffenet.
- [5] Mohamed Brahimi. Computer Science Department. Mohamed El Bachir El Ibrahimi University.
- [6] Prasad S. et al. Multi-resolution mobile vision system for plant leaf disease diagnosis. 2015.
- [7] Mokhtar U. et al. SVM-Based detection of tomato leaves diseases. 2015.
- [8] Semary N. et al. Fruit-based tomato grading system using features fusion and support vector machine. 2015.
- [9] Dandawate Y., Kokare R. An automated approach for classification of plant diseases towards development of futuristic decision support system in Indian perspective 2015.
- [10] Raza S. et al. Automatic detection of diseased tomato plants using thermal and stereo visible light images. 2015.