



## ANÁLISIS DE ARQUITECTURAS DE REDES NEURONALES CONVOLUCIONALES APLICADAS A LA DETECCIÓN DE CENICILLA POLVORIENTA EN CULTIVOS DE JITOMATE

Deny Lizbeth Hernández Rabadán<sup>1</sup>, Juan Paulo Sánchez Hernández<sup>\*1</sup>, Sandra Elizabeth León Sosa<sup>1</sup>, Juan Frausto Solis<sup>2</sup>, Juan Javier González Barbosa<sup>2</sup>

<sup>1</sup>Universidad Politécnica del Estado de Morelos, Departamento de Ingeniería en Tecnologías de la Información, Cuauhnáhuac 566, Lomas del Texcal, C.P. 62574 Jiutepec, Morelos

[juan.paulosh@upemor.edu.mx](mailto:juan.paulosh@upemor.edu.mx) (\*Autor de correspondencia)

<sup>2</sup>Tecnológico Nacional de México/Instituto Tecnológico de Ciudad Madero, Cd. Madero 89440, México

### Resumen

La detección de enfermedades en cultivos es una de las preocupaciones mayores para los agricultores de jitomate, pues una identificación incorrecta o tardía podría tener graves consecuencias en la producción de uno de los cultivos más demandados en México. Una de las enfermedades más comunes en los cultivos de jitomate es la cenicilla polvorienta, que si no es tratada en tiempo puede afectar la calidad de la producción representando pérdidas económicas en los agricultores. En este sentido, para detectar enfermedades en cultivos se ha aplicado con gran éxito las redes neuronales convolucionales (RNC). En este artículo abordamos el concepto de transferencia de conocimiento, debido a que se presenta un análisis de seis arquitecturas de redes neuronales convolucionales tales como: Resnet18, AlexNet, VGG11\_bn, Squeezenet, Densenet e Inception v3 para detección de la enfermedad cenicilla polvorienta en imágenes de cultivo de jitomate adquiridas en ambientes no controlados. Las imágenes son sometidas a un proceso de ajuste en color y brillo aplicando el espacio de color YIQ y enseguida son utilizadas para entrenar a cada una de las arquitecturas. Los resultados del análisis muestran que todas las arquitecturas evaluadas muestran muy buen desempeño en el entrenamiento, siendo las de mejor desempeño Squeezenet y Densenet.

**Palabras claves:** Transferencia de aprendizaje, redes neuronales convolucionales, cenicilla polvorienta



## Introducción

Una de las actividades humanas más trascendentes es la agricultura, debido a que representa básicamente el medio por el que los humanos y animales se alimentan adecuadamente. Con los años, la agricultura ha ido mejorando en todos sentidos con la inclusión de nuevas tecnologías, particularmente, la **inteligencia artificial** ha estado incursionando con fuerza en distintos procesos de la agricultura con el fin de apoyar en obtener una mejor producción y mejorar la calidad de los alimentos. La visión computacional y el procesamiento de imágenes son las áreas de la inteligencia artificial que más han destacado, proporcionando soluciones no destructivas y de bajo costo. Por ejemplo, se han desarrollado soluciones para la detección de enfermedades y plagas (Shruthi *et al.*, 2019), predicción de producción agrícola (Khan *et al.*, 2020), control de calidad de los frutos o verduras (Kalia *et al.*, 2019), entre otras soluciones.

La detección de enfermedades y plagas en cultivos es de gran interés en la comunidad científica, por lo que se han desarrollado varias investigaciones donde se han aplicado modelos de aprendizaje profundo, específicamente de arquitecturas de Redes Neuronales Convolucionales (RNC) (Lu *et al.*, 2001). Muchos de los trabajos realizados hacen uso de la base de datos pública llamada PlantVillage (Hughes y Salathé, 2015) que recolecta alrededor 54303 imágenes de hojas, clasificadas en 38 categorías por cultivo y enfermedad. Dentro de estos trabajos se encuentra el realizado por (Mohanty *et al.*, 2016) quienes utilizaron la base de datos PlantVillage y realizaron transferencia de aprendizaje utilizando la arquitectura de la RNC AlexNet para lograr identificar 14 especies de cultivo y 26 enfermedades (o ausencia de ella) en las imágenes. El modelo entrenado logró alcanzar una eficacia de 99.35%. En el artículo presentado por (Rangarajan, 2018) se analizaron las arquitecturas de AlexNet y VGG16 para clasificación de 6 enfermedades en hojas de tomate, las cuales se obtuvieron de PlantVillage. Los resultados reportados indican que la arquitectura AlexNet obtuvo mejor eficacia con un 97.49% en comparación con la arquitectura VGG16 la cual alcanzó un 97.29%.

En el trabajo de (Mohameth *et al.*, 2020) aplicaron transferencia de aprendizaje y evaluaron las siguientes arquitecturas de RNC: VGG16, ResNet 50, Google Net para clasificación de las enfermedades de los cultivos que se integran en la base de PlantVillage. Los resultados que se obtuvieron demostraron que la arquitectura VGG16 fue mejor que las demás, pues alcanzó una eficacia en la clasificación de un 97.82%.

El trabajo realizado por Kamal (Kamal, 2019), presentan dos modelos de arquitecturas (MobileNet modificado y MobileNet reducido) en las cuales realizaron convoluciones separables en profundidad y fueron probados con el modelo convencional de VGG. Las pruebas se realizaron con la base de imágenes de PlantVillage y mostraron que los modelos MobileNet modificado y MobileNet reducido obtuvieron porcentajes de eficacia cercanos a la arquitectura convencional VGG. El modelo MobileNet reducido alcanzó un 98.34% con un menor número de parámetros comparado con VGG, lo que lo hace ideal para ser utilizado en dispositivos embebidos.



Uno de los cultivos más importantes en México, es el jitomate, debido a que representa uno de los ingredientes básicos en la dieta del mexicano y además es fuente de vitaminas, minerales y antioxidantes que lo hacen muy importante en la alimentación. El jitomate, para una correcta producción debe tener cuidados muy particulares desde su siembra hasta su recolección, de lo contrario puede verse afectado por diversas enfermedades o plagas.

Una de las enfermedades que comúnmente ataca a este cultivo es la cenicilla polvorienta, la cual presenta afectaciones en diversas partes de la planta con son: las hojas, el tallo o en el fruto. Los síntomas de la presencia de cenicilla son visualmente perceptibles en las hojas, ya que se presentan manchas amarillas en el haz que se vuelven necróticas en el centro observándose un fieltro blanquecino en el envés si el agente es *Leveillula taurica*, en caso de que el agente sea *Oidium lycopersicum* se muestra micelio superficial de color blanco con ocasiones bordes amarillos en las hojas (De hortalizas, 2006).

En este trabajo de investigación se eligió la cenicilla polvorienta debido a que sus características visuales pueden ser caracterizadas como un patrón de la enfermedad. Además, se introduce el concepto de transferencia de aprendizaje al entrenar un conjunto de imágenes de cultivos de jitomate con las arquitecturas de redes neuronales convolucionales, tales como: Resnet18, AlexNet, VGG11\_bn, Squeezenet, Densenet e Inception v3. Con lo anterior, se realiza una comparativa y verificar cuál de las arquitecturas propuestas es la que tiene mejor desempeño.

Como se mencionó anteriormente, la base de datos PlantVillage se ha utilizado en varias investigaciones que tienen como objetivo el diagnóstico de enfermedades en cultivos; sin embargo, esta base de imágenes se caracteriza por constar de imágenes controladas, es decir, imágenes que fueron tomadas en condiciones de laboratorio, por lo que la eficacia en los resultados de las metodologías propuestas puede ser pobre al ser probados en un ambiente real, como lo es un campo de cultivo. En este trabajo, se propone utilizar imágenes del cultivo de jitomate adquiridas en un ambiente real, carente de control de iluminación, por lo que se construyó una pequeña base de datos de imágenes de plantas con presencia visual de cenicilla polvorienta e imágenes de plantas sanas. La metodología propuesta integra un proceso de homogeneización de brillo y color en las imágenes para mejorar los resultados arrojados por las arquitecturas de Redes neuronales convolucionales.

Este artículo se organiza de la siguiente manera: en la sección uno se presenta la introducción en donde se introduce brevemente la investigación y los elementos importantes, así como se presentan algunos de los trabajos relevantes que tienen relación a nuestra investigación; en la sección dos se presentan los materiales y métodos, es decir, se presenta el concepto de transferencia de aprendizaje y las arquitecturas propuestas; en la sección cinco se mostrarán los resultados y pruebas realizadas; finalmente, se presentan las conclusiones y algunos de los trabajos que se van a continuar desarrollando.

## Materiales y Métodos

### Redes neuronales convolucionales

Las redes neuronales son técnicas de inteligencia artificial que en la actualidad se han estado utilizando con mayor frecuencia, debido a que los nuevos modelos denominados redes neuronales convolucionales y se han aplicado con gran éxito en una gran variedad de problemas de clasificación, segmentación, seguimiento, pronóstico, entre otros. La arquitectura básica de las redes neuronales convolucionales consta de tres etapas, una entrada, una capa de extracción de características y una capa de clasificación (figura 1), las cuales se describen a continuación.

**Etapas de entrada:** Esta etapa se refiere al conjunto de datos de entrada que puede ser una imagen, si el problema que se desea resolver las involucra, sin embargo, puede ser diferentes tipos de entradas las que alimenten a la red neuronal convolucional. En esta etapa es muy importante que el conjunto de datos sea sustancial.

**Etapas de extracción:** En esta etapa la extracción de características es el objetivo y para poder lograrlo se aplican esencialmente dos operaciones: convoluciones y submuestreo. La convolución se refiere a la aplicación de un conjunto de kernels u operadores que permiten resaltar características propias de las imágenes y se pueden aplicar los operadores que sean necesarios para ir extrayendo las características que ayuden al modelo a tener un mejor desempeño y por cada convolución se aplica un submuestreo o reducción de la matriz de características.

**Etapas de clasificación:** En esta etapa de clasificación se reciben los datos extraídos en la etapa anterior y son entrenados en una red neuronal multiconectada según el número de clases que se establezca para finalmente obtener un modelo entrenado que puede ser utilizado para clasificar las clases mencionadas.

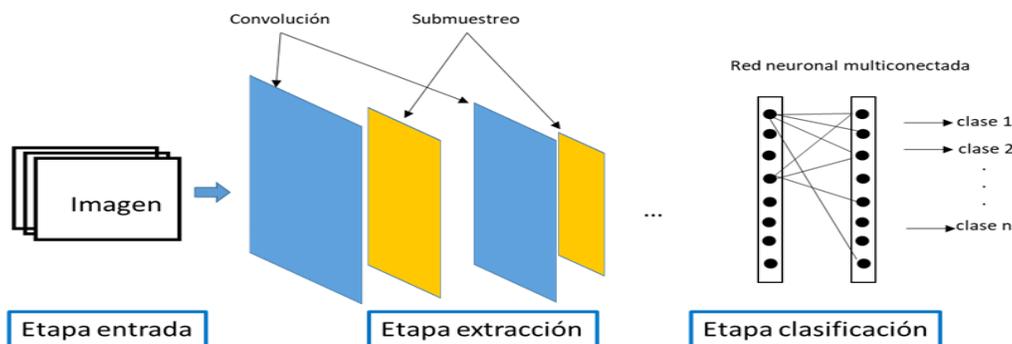


Figura 1. Etapas básicas de una red neuronal convolucional

Construir la arquitectura de una red neuronal convolucional puede ser un proceso muy tardado, debido al tiempo de experimentación y sintonización de los hiperparámetros. Derivado de esta problemática, se han desarrollado una gran cantidad de arquitecturas, las cuales se pueden utilizar para clasificar diversos patrones de objetos, como autos,

pelotas, aves, edificios, entre otros. Además, esas arquitecturas se pueden reentrenar e incluir nuevos patrones de objetos para clasificar a ese concepto se le denomina transferencia de aprendizaje (Transfer Learning en inglés). La transferencia de aprendizaje es bastante útil para los investigadores que no cuentan con equipos tan sofisticados para poder generar su propia arquitectura de red neuronal convolucional, debido a que el tiempo de entrenar un nuevo patrón se reduce sustancialmente. Algunas de las arquitecturas pre entrenadas son las presentadas en el cuadro 1.

**Cuadro 1.** Listado de arquitecturas de redes neuronales convolucionales preentrenadas

<b>Nombre arquitectura</b>	<b>Descripción</b>	<b>Autores</b>
<b>Resnet18</b>	Arquitectura de red neuronal con 18 capas de profundidad con la primera capa kernel de 7x7 y las siguientes 17 con kernels de 3x3.	(He et al, 2016)
<b>VGG11_bn</b>	Arquitectura de red neuronal con 11 capas de profundidad con kernels de 3x3 y con normalización de batch.	(Simonyan y Zisserman, 2015)
<b>Densenet</b>	Arquitectura de red neuronal que combina operaciones de convolución, pooling, con bloques densos y capas de transición.	(Huang et al, 2017)
<b>Alexnet</b>	Arquitectura de red neuronal con cinco capas convolucionales y 3 densas multi conectadas. Las capas convolucionales tienen tamaños de kernels distintos, por ejemplo, la primera tiene 11x11, la segunda 5x5 y las restantes 3x3.	(Krizhevsky, 2014)
<b>Squeezenet</b>	Arquitectura de red neuronal compacta que reemplaza los filtros de 3x3 por 1x1. Lo anterior permite aumentar la capacidad de compresión sin perder eficacia.	(Iandola, 2016)
<b>InceptionV3</b>	Arquitectura de red neuronal convolucional desarrollada en Google la cual combina bloques simétricos y asimétricos, convoluciones, reducciones máximas, reducciones promedio, normalización de batch, concatenaciones y capas multi conectadas.	(Szegedy, 2015)

### **Descripción del conjunto de imágenes**

El conjunto de imágenes utilizadas corresponde a plantas capturadas en un invernadero, en donde no se controló la iluminación o fondo. Se capturaron 65 imágenes sanas y 65 imágenes enfermas con cenicienta polvoriento, las cuales se dividieron en 20% para validación y 80% para entrenamiento. Se puede observar en la figura 2 un ejemplo de las imágenes capturadas.



a) Planta de jitomate sana



b) Planta de jitomate enferma

**Figura 2.** Un ejemplo de plantas de jitomate sanas (a) y enfermas (b) con cenicilla polvorienta

El dinamismo de la iluminación durante la captura de imágenes provocó que la información de color y brillo en las imágenes fuese diferente lo que podría afectar los resultados de la predicción de la enfermedad, por lo que se aplicó un algoritmo para ajustar las imágenes en ambos parámetros tomando como referencia una imagen del mismo conjunto que mostrará mejor estabilidad. La figura 3a y 3b, se observa la diferencia entre iluminación y brillo y al aplicarle el procesamiento se observa que la iluminación y brillo en la figura 3c y 3d se homogeneizan.



a)



b)



c)



d)

**Figura 3.** Imágenes de plantas enfermas y sanas. a) y b) son las imágenes originales, c) y d) resultados de aplicar homogeneización de color y brillo.

En las imágenes a) y b) mostradas en la figura 3, se puede observar una diferencia en color y brillo entre ellas debido a que fueron tomadas bajo condiciones de iluminación diferentes. Después de aplicar el algoritmo para homogeneización, en las figuras c) y d) se aprecia visualmente que las imágenes tienen tonalidades cercanas en color y el brillo se nota más homogéneo entre ambas.

Las imágenes del cultivo se adquirieron en formato *RGB*, sin embargo, el algoritmo para homogeneización propuesto por (Meunkaewjinda *et al*, 2005) utiliza el modelo de color

YIQ, por lo que se aplicaron las siguientes ecuaciones (1-3) para realizar la conversión de RGB a YIQ:

$$Y = 0.299 * R + 0.587 * G + 0.114 * B \quad (1)$$

$$I = 0.596 * R - 0.275 * G - 0.321 * B \quad (2)$$

$$Q = 0.212 * R - 0.523 * G + 0.311 * B \quad (3)$$

Donde  $R, G$  y  $B$  representan los valores de cada uno de los componentes del rojo, verde y azul para cada píxel de la imagen.

El algoritmo utiliza una imagen de referencia la cual debe ser selecta del dataset, eligiendo aquella que muestre visualmente tener mejor estabilidad en color y brillo. Una vez elegida la imagen de referencia y contando con la correspondiente transformación en YIQ tanto para la imagen de referencia como para las imágenes de entrada, se aplican las ecuaciones de la 4 a la 6.

$$\mu_{im} = \frac{1}{m \times n} \sum_{n=1}^{m \times n} I(n) \quad (4)$$

$$\mu_{im} = \mu_{tar} - \mu_{im} \quad (5)$$

$$I_{new} = I(n) + \mu_{new} \quad (6)$$

Donde:

- $I(n)$  es el valor de intensidad para cada canal  $Y, I, Q$  de la imagen de entrada.
- $\mu_{im}$  es el valor de la media para cada canal de la imagen de entrada.
- $\mu_{tar}$  es el valor de la media para cada canal de la imagen de referencia con una matriz de tamaño  $m \times n$ .
- $I_{new}$  es la nueva imagen con ajuste en color y brillo para cada canal YIQ

Como último paso del procedimiento, la imagen nueva  $I_{new}$  es convertida a RGB.

## Resultados y Discusión

En la experimentación realizada se utilizaron las arquitecturas Squeezenet, DenseNet, Resnet18, Alexnet, VGG11\_bn, e InceptionV3. Cada una de ellas con los mismos hiperparámetros, es decir, batch de 8, dos clases y 20 épocas para el entrenamiento. Se decidieron 20 épocas, debido a que todos los modelos alcanzaron valores cercanos a 1 muy rápidamente, por lo que incrementar más épocas, podría generar sobreentrenamiento. En la figura 3 se puede observar el comportamiento de la precisión en la etapa de entrenamiento del conjunto de datos, en donde los modelos Squeezenet y Densenet fueron los que obtuvieron mejores resultados. Para el caso de la figura 4, se observa el comportamiento de la pérdida y en este caso Squeezenet fue el que mejor se comportó.

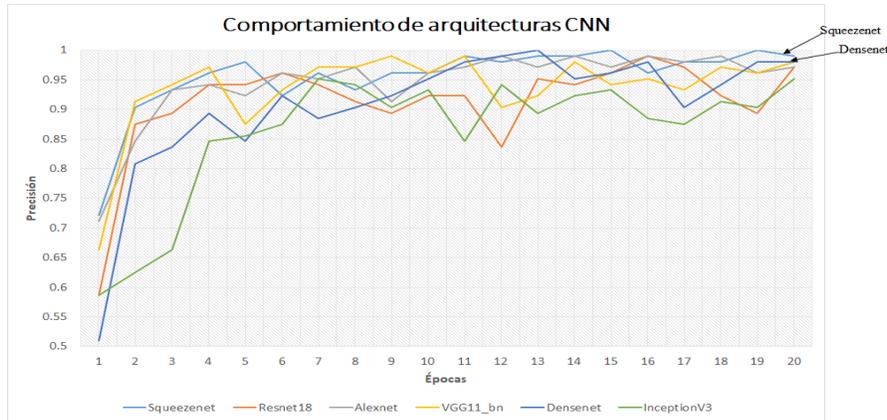


Figura 3. Comportamiento del entrenamiento de las arquitecturas CNN.



Figura 4. Comportamiento de la pérdida de las arquitecturas CNN

En la figura 5, se presenta el mejor valor alcanzado en la etapa de entrenamiento. En la figura 5a, se observa que la mayoría de las arquitecturas alcanza el 0.99 de precisión y en el caso de la figura 5b, la arquitectura que alcanzó el mejor desempeño en pérdida fue Alexnet, seguido por SqueezeNet.

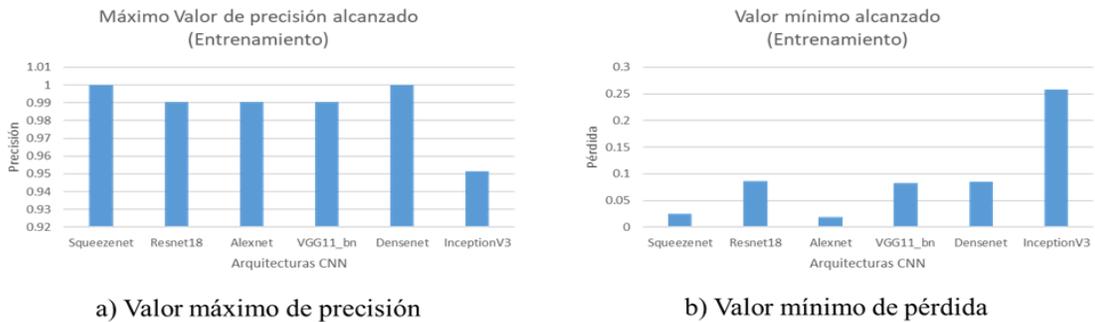
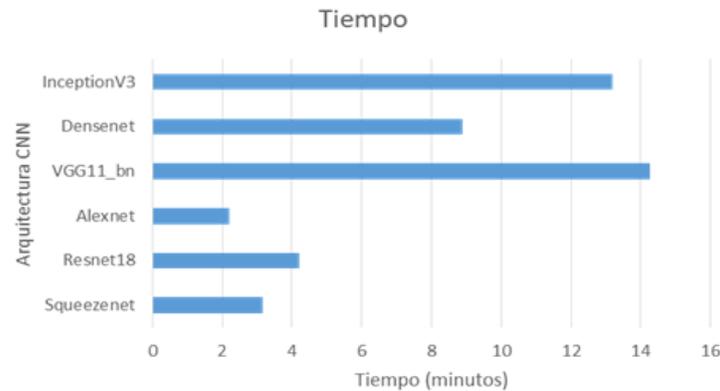


Figura 5. Valores máximos (a) y mínimos (b) alcanzados durante las 20 épocas realizadas

En el caso del tiempo de entrenamiento, en la figura 6 se observa el comportamiento del tiempo de entrenamiento de cada una de las arquitecturas y se puede ver que Alexnet fue la de menor tiempo de entrenamiento seguido por Squeezenet.



**Figura 6.** Tiempos de entrenamiento en minutos

## Conclusiones

Los sistemas inteligentes en la actualidad son cada vez más utilizados, tanto en la vida cotidiana como en actividades de agricultura. El uso de las arquitecturas de redes neuronales convolucionales mediante la transferencia de aprendizaje permite el uso de modelos de clasificación más robustos y facilitan enormemente la etapa de sintonización de hiperparámetros y entrenamiento de nuevas clases.

En este trabajo se presenta una comparativa de las arquitecturas Squeezenet, DenseNet, Resnet18, Alexnet, VGG11\_bn, e InceptionV3. Los resultados mostraron que DenseNet y Squeezenet tuvieron mejor desempeño en la etapa de entrenamiento con el conjunto de datos elegido. En el caso de Squeezenet alcanzó una precisión de 99.04% de porcentaje de precisión y DenseNet alcanzó un porcentaje 98.04% de precisión.

En el caso de la aplicación de la mejora en iluminación y brillo consideramos que ayudó a que las arquitecturas alcanzaran rápidamente buenos resultados de desempeño, sin embargo, no se realizaron pruebas suficientes para poder comprobar que la mejora impacta en el entrenamiento de manera significativa.

Finalmente, como trabajo futuro, se incrementará el número de arquitecturas pre entrenadas, el conjunto de imágenes y los tipos de enfermedades, esto con el fin de realizar un estudio más amplio y corroborar la efectividad de los modelos en imágenes tomadas en entornos no controlados.



## Referencias Bibliográficas

- Shruthi, U., Nagaveni, V., & Raghavendra, B. K. (2019, March). A review on machine learning classification techniques for plant disease detection. In 2019 5th International conference on advanced computing & communication systems (ICACCS) (pp. 281-284). IEEE.
- Khan, T., Qiu, J., Qureshi, M. A. A., Iqbal, M. S., Mehmood, R., & Hussain, W. (2020). Agricultural fruit prediction using deep neural networks. *Procedia Computer Science*, 174, 72-78.
- Kalia, P., Garg, A., & Kumar, A. (2019). Fruit quality evaluation using Machine Learning: A review. In 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT) (Vol. 1, pp. 952-956). IEEE.
- Lu, J., Tan, L., & Jiang, H. (2021). Review on convolutional neural network (CNN) applied to plant leaf disease classification. *Agriculture*, 11(8), 707.
- Hughes, D. P., and Salathé, M. (2015). An open access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv:1511.08060.
- De hortalizas, P. (2006). Plagas y Enfermedades del Tomate. Guía de Identificación y Manejo. Recuperado el 28 de septiembre de 2015. [http://vegetablemdonline.ppath.cornell.edu/NewsArticles/Tomato\\_Spanish.pdf](http://vegetablemdonline.ppath.cornell.edu/NewsArticles/Tomato_Spanish.pdf).
- Meunkaewjinda, A., Srikaew, A., Kumsawat, P., Attakitmongcol, K., & Boonjung, H. (2007). Hybrid artificial intelligence approach for grape leaf disease diagnosis system from color imagery. In The 9th International Conference on Precision Agriculture (ICPA) (pp. 20-23).
- Mohameth, F. , Bingcai, C. and Sada, K. (2020) Plant Disease Detection with Deep Learning and Feature Extraction Using Plant Village. *Journal of Computer and Communications*, 8, 10-22. doi: 10.4236/jcc.2020.86002.
- Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in plant science*, 7, 1419.
- Rangarajan, A. K., Purushothaman, R., & Ramesh, A. (2018). Tomato crop disease classification using pre-trained deep learning algorithm. *Procedia computer science*, 133, 1040-1047.
- Kamal, K. C., Yin, Z., Wu, M., & Wu, Z. (2019). Depthwise separable convolution architectures for plant disease classification. *Computers and Electronics in Agriculture*, 165, 104948.



- Simonyan, K., Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. [arxiv.org/abs/1409.1556](https://arxiv.org/abs/1409.1556).
- He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.
- Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q. (2017). Densely connected convolutional networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708.
- Krizhevsky, A. (2014). One weird trick for parallelizing convolutional neural networks. [arxiv.org/abs/1404.5997](https://arxiv.org/abs/1404.5997).
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. (2015). Rethinking the Inception Architecture for Computer Vision. [arxiv.org/abs/1512.00567](https://arxiv.org/abs/1512.00567).
- Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. [arxiv.org/abs/1602.07360](https://arxiv.org/abs/1602.07360).