**Desarrollo de una aplicación móvil para la detección y clasificación de enfermedades en la planta del Tomate utilizando Swin Transformer**

Oscar David Valencia - 20192182074

Maria José Capera Firigua – 20201188262

Facultad de Ingeniería, Universidad Surcolombiana

Electiva Ciencias de la computación II - Visión Artificial

Ferley Medina Rojas

Agosto de 2023

**TABLA DE CONTENIDOS**

[**INTRODUCCIÓN 3**](#_tycrd2ytbpn5)

[**PROBLEMA 4**](#_76696e9e9q2l)

[**OBJETIVOS 5**](#_5wmsknvme73h)

[Objetivo general 5](#_f6dlx77kymqv)

[Objetivos especificos 5](#_d6r42owmwevb)

[**REVISIÓN DE LITERATURA 6**](#_r2lsqc1cu1it)

[Machine Learning Approach towards Tomato Leaf Disease Classification. International Journal of Advanced Trends in Computer Science and Engineering, 9(1). 6](#_xa2sjoqh2ody)

[Tomato plant disease detection using transfer learning with C-GAN synthetic images 7](#_nhdykxx5j871)

[Tomato leaf disease classification using supervised learning techniques contrasting. 9](#_93hzfrq8tjbd)

[Artificial intelligence in tomato leaf disease detection 11](#_1pm0ksx7pk1k)

[Computer-aided fusion-based neural network in application to categorize tomato plants. Springer-Verlag London Ltd., part of Springer Nature 2023 14](#_obxgds67bnog)

[**MARCO TEÓRICO 17**](#_osxtvooujh3s)

[**REFERENCIAS 22**](#_3irfze7s0g8d)

# INTRODUCCIÓN

La detección temprana y precisa de enfermedades en el tomate se ha convertido en una necesidad imperante para mitigar su impacto devastador. Aunque los avances en la visión artificial y el aprendizaje automático han proporcionado soluciones potenciales, estos métodos han sido insuficientes en términos de velocidad, costo, y precisión. Las prometedoras técnicas de Deep Learning, particularmente las redes neuronales convolucionales (CNN), no han vuelto a plantear nuevas arquitecturas ni tampoco un cambio de paradigma, lo que exige un replanteamiento innovador.

Este estudio se propone explorar y validar un enfoque innovador mediante la aplicación del "Swin Transformer", una arquitectura novedosa en el dominio de la visión por computadora. A través de un análisis sistemático y una metodología rigurosa, se buscará desarrollar un modelo que supere las limitaciones existentes en la detección y clasificación de enfermedades del tomate.

Se enfocará en responder preguntas críticas relacionadas con el desarrollo y validación de este nuevo modelo, así como en su comparación con los métodos basados en CNN. La investigación abordará aspectos esenciales como el aumento de datos, la integridad de los resultados, la configuración y entrenamiento óptimo de la arquitectura Swin Transformer, y la interpretación visual de las decisiones del modelo.

La investigación propuesta tiene un potencial significativo para transformar la práctica actual en la detección y clasificación de enfermedades del tomate, ofreciendo un modelo más preciso y eficiente.

El trabajo se estructura en introducción, problema, objetivos, estado del arte, marco teórico, metodología, materiales y métodos, resultados, conclusiones y trabajo futuro.

# PROBLEMA

El cultivo del tomate es esencial para la alimentación y la economía agrícola global debido a que es una de las hortalizas más producidas mundialmente. De acuerdo con la FAO (2021), la producción mundial superó los 189 millones de toneladas, abarcando más de 5 millones de hectáreas cultivadas. Nutricionalmente es una fuente rica de fitoquímicos, nutrientes, antioxidantes y compuestos fenólicos (Collins et al., 2022). Se ha determinado que su consumo aporta beneficios anticancerígenos y protección contra enfermedades cardiovasculares y neurodegenerativas (Collins et al., 2022).

No obstante, el tomate es vulnerable a una amplia variedad de enfermedades causadas por hongos, bacterias, fitoplasmas, virus y viroides (Panno et al., 2021) Tabla 1. Esta vulnerabilidad se ve incrementada por factores como su limitada diversidad genética resultado de la selección intensiva, la tendencia al monocultivo, el intercambio de material infectado a nivel internacional y el cambio climático (Panno et al., 2021). Estas enfermedades representan pérdidas económicas que superan los 30.000 millones de dólares al año (Caruso et al., 2022). Dichas pérdidas repercuten en la economía agrícola, la salud pública y la sostenibilidad ambiental, al afectar el rendimiento y la calidad del cultivo, desestabilizar ecosistemas y elevar los costos de producción por el uso intensivo de pesticidas y otras medidas de control (Panno et al., 2021).

La detección temprana y precisa de enfermedades en el tomate es primordial para mitigar su impacto, pero a pesar de los avances significativos en visión artificial en la última década, estos han presentado limitaciones. En el ámbito del machine learning, los métodos suelen recurrir a la extracción manual de características y algoritmos heurísticos, lo que puede ser lento, laborioso, costoso y susceptible a errores (Zahangir Alom et al., 2018). Estos enfoques han sido superados por técnicas de Deep Learning como las redes neuronales convolucionales (CNN) (Rawat et al. 2022; Thangaraj et al. 2022). Sin embargo, estas últimas parecen haber llegado a un punto de estancamiento tras el desarrollo de arquitecturas como ResNet, DenseNet y EfficientNet (Li, 2020).

Es por ello, que la presente investigación sugiere un enfoque innovador basado la arquitectura "Swin Transformer", el cual ha demostrado ser altamente efectivo en tareas de visión por computador pero que nunca ha sido aplicado en la clasificación de enfermedades del tomate. El objetivo es desarrollar y validar un modelo más preciso que las soluciones anteriores utilizando el dataset de PlantVillage, el cual contiene nueve clases de enfermedades del tomate y una clase de la planta sana. Posteriormente, se comparará su rendimiento con estudios previos basados en CNN.

En este contexto, se plantea las siguientes preguntas:

* ¿Cómo desarrollar y validar un modelo de detección y clasificación de enfermedades del tomate utilizando la arquitectura Swin Transformer que supere la precisión de las redes neuronales convolucionales (CNN)?
* ¿Cómo se puede construir y preparar un dataset balanceado en las diferentes clases de enfermedades del tomate que garantice la integridad de las predicciones del modelo?
* ¿Cómo configurar y entrenar de manera óptima la arquitectura Swin Transformer?
* ¿Cómo evaluar e interpretar las predicciones del modelo Swin Transformer para cada clase?
* ¿Cómo se compara la exactitud (accuracy) del modelo Swin Transformer con los modelos basados en redes neuronales convolucionales?

# OBJETIVOS

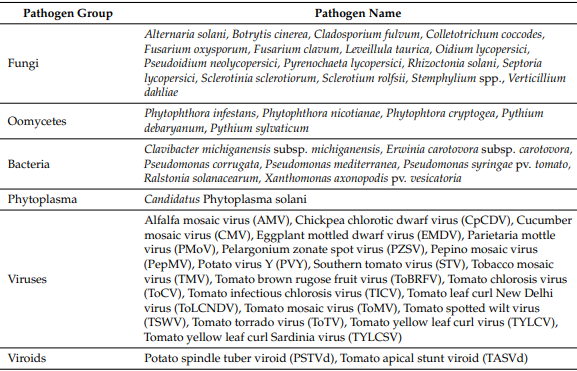
## Objetivo general

Desarrollar y validar un modelo de detección y clasificación de enfermedades de plantas de tomate utilizando la arquitectura Swin Transformer, con el propósito de conseguir una precisión superior en comparación con las redes neuronales convolucionales.

## Objetivos específicos

1. Construir y preparar un dataset balanceado en las diferentes clases de enfermedades del tomate que asegure la integridad de las predicciones del modelo.
2. Configurar y entrenar de manera óptima la arquitectura Swin Transformer.
3. Evaluar e interpretar las predicciones del modelo Swin Transformer para cada clase.
4. Comparar y analizar la exactitud (accuracy) del modelo Swin Transformer con modelos basados en redes neuronales convolucionales.

**Tabla 1.** Lista de patógenos de plantas de tomate (Panno et al., 2021).



**ESTADO DEL ARTE**

En el artículo ***"Machine Learning Approach towards Tomato Leaf Disease Classification"***, Gadade y Kirange, (2020) presentan un enfoque exhaustivo para la identificación y clasificación de las enfermedades de las hojas del tomate. El estudio sigue meticulosamente un proceso que comienza con la recopilación de datos, donde aprovecharon 9.000 imágenes de hojas de tomate del dataset PlantVillage. Esta recopilación se centró especialmente en siete clases de imágenes enfermas y una clase de imagen sana.

A continuación, los autores llevaron a cabo una etapa de preprocesamiento, en la que se minimizó el ruido de las imágenes de la enfermedad de la hoja de tomate utilizando un filtro de mediana. Después se realizó la extracción de características mediante tres técnicas: GLCM (Gray-Level Co-Occurrence Matrix), Gabor y SURF (Speeded Up Robust Features). Para clasificar estas imágenes de hojas de tomate en categorías normales o enfermas, los investigadores emplearon una serie algoritmos de clasificación. Entre ellos, la máquina de soporte vectorial (SVM), k-Nearest Neighbors (KNN), Naïve Bayes (NB) y árboles de decisión.

Los resultados de la investigación se centraron en métricas clave como accuracy, precision, recall y f1-score. Al emplear árboles de decisión con características extraídas mediante Gabor, se registraron las siguientes métricas: 'accuracy' de 0,6497, 'precision' de 0,2047, 'recall' de 0,861 y 'f1-score' de 0,2772. Sin embargo, al combinar el clasificador SVM con Gabor, los resultados mejoraron considerablemente: 'accuracy' de 0,7339, 'precision' de 0,2525, 'recall' de 0,9492 y 'f1-score' de 0,3989. Utilizando el clasificador KNN y Gabor, las métricas alcanzaron 'accuracy' de 0,732, 'precision' de 0,2555, 'recall' de 0,9831 y 'f1-score' de 0,4056. Finalmente, con Naïve Bayes y Gabor, se observaron valores de 'accuracy' de 0,675, 'precision' de 0,2187, 'recall' de 0,9695 y 'f1-score' de 0,3568.

Acorde a los resultados obtenidos, el estudio destaca que Gabor fue el método de extracción más eficaz, y que SVM resultó ser el algoritmo de clasificación más competente. Consecuentemente, Gadade y Kirange (2020) respaldan la combinación de SVM y Gabor como una herramienta robusta para la clasificación de enfermedades en hojas de tomate, subrayando su aplicabilidad en escenarios en tiempo real. Aun así, sugieren que técnicas más avanzadas, como Adaptive neuro fuzzy, Neural Networks y algoritmos genéticos, podrían ampliar las capacidades de clasificación.

No obstante, este trabajo evidencia que los métodos tradicionales de aprendizaje automático presentados aquí muestran métricas inferiores en comparación con las técnicas modernas de Deep learning basadas en redes neuronales convolucionales. Esta observación subraya una posible obsolescencia de los métodos tradicionales en el ámbito de la clasificación de enfermedades en hojas de tomate a través de técnicas de visión por computadora.

En el estudio titulado ***"ToLeD: Tomato Leaf Disease Detection using Convolution Neural Network"*** de Mohit Agarwala et al. (2020) se propone un enfoque basado en una Red Neuronal Convolucional (CNN) para la detección y clasificación de enfermedades en las hojas del tomate. El trabajo parte de la importancia de los tomates como cultivo global, abordando el problema de la disminución en la calidad y cantidad debido a diversas enfermedades.

El estudio utiliza el dataset de Plant Village que contiene imágenes de hojas de tomate afectadas por nueve tipos de enfermedades, además de una clase de hojas sanas. Para equilibrar el dataset, se aplicaron técnicas de aumento de datos resultando en 10,000 imágenes para el entrenamiento, 7,000 para la validación y 500 para las pruebas.

Se diseñó una arquitectura CNN específica que incorpora 3 capas convolucionales, 3 capas de max-pooling y 2 capas completamente conectadas. Además, se utilizaron modelos CNN preentrenados mediante transferencia de aprendizaje y utilizando el mismo dataset, con el fin de comparar su eficiencia con el modelo propuesto.

Los resultados de las métricas evaluadas varían en un rango de precisión del 76% al 100% en función de las clases, con una precisión promedio del 91,2%, superando a los modelos pre-entrenados, como VGG16 con un accuracy promedio del 77.2%, InceptionV3 del 63.4% y MobileNet del 63.7%.

El modelo propuesto presenta un número menor de parámetros entrenables en comparación con los modelos pre-entrenados, lo que sugiere que podría ser más eficiente en términos de almacenamiento y computación. Sin embargo, la variabilidad en el ‘accuracy’ por clase y su bajo puntaje promedio comparado con el de estudios más recientes son un indicador claro de que, aunque el modelo propuesto es en general eficaz, todavía hay margen para mejorar la predicción por clases y plantea como trabajo futuro el ajuste de la arquitectura para incrementar la precisión.

El artículo titulado ***"Tomato plant disease detection using transfer learning with C-GAN synthetic images"***, escrito por Amreen Abbas, Sweta Jain y Mahesh Gour, aborda una metodología de aprendizaje profundo para detectar enfermedades en las plantas de tomate mediante el análisis de imágenes de sus hojas. La metodología se basa en dos fases principales: la generación de imágenes sintéticas utilizando Redes Antagónicas Generativas Condicionales (C-GAN) para la ampliación del conjunto de datos, seguida de la clasificación de enfermedades utilizando un modelo DenseNet121 previamente entrenado.

## Title: Tomato plant disease detection using transfer learning with C-GAN synthetic images

**Authors:** Amreen Abbas, Sweta Jain, Mahesh Gour

**Methodology**

In this research work, we have developed a deep learning based framework to recognize the tomato plant diseases by investigating tomato leaf images. This method would help farmers in classification of diseases affecting tomato cultivation by simply taking an image of diseased leaves, instead of going after costly expert analysis. In the proposed method, we generated synthetic images using Conditional Generative Adversarial Network (C-GAN) (Mirza and Osindero, 2014) and augmented these images to the dataset for training purpose.

Thereafter a DenseNet model (Huang et al., 2017) was trained on tomato plant images and tomato synthetic images (generated by the C-GAN model) for the detection of tomato disease from the leaf images.

In this section, we will discuss the proposed method. The Block diagram of the proposed method is shown in Fig. 1. The proposed approach can be divided into two parts; in the first part, synthetic images have been generated using C-GAN for data augmentation. In the second part, a pre-trained DenseNet121 model has been fine-tuned on tomato leaf images for disease classification. The detailed description is given in the following subsections:

***C-GAN model as synthetic image generator***

In order to prevent the network from over-fitting, Conditional Generative Adversarial Network (C-GAN) (Mirza and Osindero, 2014) can be used as data augmentation technique to enhance the size of dataset. In GAN, traditional convolutional layers are applied to form an image matrix from random noise. GAN consists of a discriminator model and a generator model. The work of generator is to produce fake images and the work of discriminator is to distinguish between fake and real images. The generator and discriminator train simultaneously and try to outdo each other. The discriminator makes sure that the images generated by the generator are as close to the real images as possible.

The C-GAN model (Mirza and Osindero, 2014) consists of two adversarial networks: the generator model and discriminator model. The C-GAN discriminator model consists of an input layer, an embedding layer, and a dense layer followed by an input layer, reshape layer, and concatenate layer, which is in turn followed by four convolutional layers. Each convolutional layer is followed by a Leaky ReLU layer. The last Leaky ReLU layer is followed by a flatten layer, a dropout layer, and a dense layer. The model consists of a total of 771,454 trainable parameters. The network layers and parameters of discriminator model are represented in Table 2.

The C-GAN generator model consists of input layers and a dense layer followed by an embedding layer and a Leaky ReLU layer which is in turn followed by dense, reshape and concatenate layers. The concatenate layer is followed by four convolutional layers where each convolutional layer is followed by a Leaky ReLU layer. The last Leaky ReLU layer is followed by a final convolutional layer. The model consists of a total of 1,735,904 trainable parameters. The network layers and parameters of discriminator model are represented in Table 3. The C- GAN generator model (G) takes latent points and represented in Table 3. The C- GAN generator model (G) takes latent points and random noise as input and generates synthetic images while the work of discriminator model (D) is to differentiate between real images and synthetic images generated by the G model. Suppose *I* represents data and *L* represents the additional class label information where *I* and *L* are fed as input to the discriminative function and the input noise distribution in G model is given by *qz*(*z*). The D model tries to maximize the probability of assigning the labels correctly to the original images as well as synthetic images generated by the generator model while the G model tries to minimize the generator loss. The objective function of C-GAN is similar to a two player minimax game. In the proposed approach, we have used C-GAN for generating synthetic tomato leaf images of various diseases. For generating synthetic images from the C-GAN model, we first trained it on the tomato images.

To train C-GAN, a real tomato image *I*(*n*) with corresponding label *L*(*n*) is given as input to the discriminator model; simultaneously, a noise and a label *L*(*n*) is given to the generator model. Then the generator model generates tomato fake image *If* and now generated fake image *If* is also given to the discriminator model.

Thereafter, the discriminator tries to distinguish between fake and real images. This way, C-GAN has been trained on the tomato leaf images. After successful completion of the training of C-GAN, we have obtained synthetic images of tomato leaves corresponding to each disease category.

*Tomato plant disease detection with DenseNet model*

The pre-trained Dense Convolutional Network (DenseNet) has been trained on CIFAR-10, CIFAR-100, SVHN, and ImageNet datasets (Huang et al., 2017). It has achieved promising results for object recognition tasks. In this study, we used the DenseNet121 model using transfer learning for tomato plant disease detection from tomato leaf images. The detailed description of DenseNet architecture is given as follows:

**DenseNet Architecture:** DenseNet is a deep architecture in which each layer is connected to every other layer in a feed-forward manner. DenseNet differs from ResNet (He et al., 2016) in the manner of a combination of features that take place at a layer before they are passed on to the next layers. In ResNet, the features are combined through summation while in DenseNet, the features are combined by concatenating them. Other convolutional networks have *l* connections for *l* layers whereas DenseNet has *l(l* + *1)/2* connections for *l* layers. In DenseNet, inputs of each layer are the feature-maps of previous layers, and subsequently, the feature-maps of a layer are used as inputs in the succeeding layers (Huang et al., 2017).

The architecture of DenseNet121 model is represented in Fig. 2 and the description of different layers of the model is represented in Table 4. It consists of 4 dense blocks that take 224 × 224 pixels image as input. The first convolution layer consists of 2000 convolutions of size 7 × 7 with stride 2, which is followed by a 3 × 3 max pooling layer with stride 2. The pooling layer is followed by three dense blocks with each dense block being followed by a transition layer. The fourth dense block is followed by a classification layer. The dense blocks consist of batch normalization, ReLU and convolutional layers. The transition layer consists of 1 × 1 convolution layer followed by 2 × 2 average pooling layer with stride 2.

**DenseNet Fine-tuning:** In order to fine-tune the pre-trained DenseNet model on the tomato leaf images, the top Fully-connected layer, and Softmax layer have been removed from the network, and two new Convolutional layers with ReLU activation function, an Average pooling layer, a Fully-connected layer, and a Softmax layer have been added in the network. The weights of the first 410 layers have been initialized with pre-trained imagenet weights, and the remaining layers’ weights have been learned during the training. The model has been trained for 100 epochs with a learning rate of 0.0001 and a batch size of 32, with the weights being updated using Adam’s optimizer (Kingma and Ba, 2014). We have experimentally found these are the best suitable values of hyperparameter for our application. For the training of the DenseNet model, generated synthetic images of tomato leaves and images from the tomato PlantVillage dataset (Hughes et al., 2015) have been combinedly used.

*Dataset*

In order to evaluate the performance of the proposed approach, a publicly available tomato PlantVillage dataset (Hughes et al., 2015) has been used in the experiments. PlantVillage dataset consists of 16,012 images of tomato plant leaves of ten classes. Out of ten classes, nine categories are of tomato plant leaf diseases, and one category is for healthy leaves. The classes of the PlantVillage dataset and class-wise image distribution is represented in Table 5. We have used abbreviations of class names, which are also shown in Table 5. All the images of the dataset have been resized to 224 × 224 for faster computations. We have divided the dataset into training set, validation set and test set in the ratio of 60:10:30 with the consideration of no overlapping between the three sets. The training set and the validation set have been used during the network training, and the test set has been used for performance assessment of the model.

## Experiment setup and evaluation metrics

In this study, two sets of experiments have been performed. In the first set of experiments, the C-GAN model has been trained on the training set for 300 epochs to generate synthetic leaf images of tomato for each of the ten categories. The weights of the generator and discriminator model were updated after each epoch to generate synthetic images as close to real images as possible. At the end of the training of the network, we have generated 4,000 synthetic images of tomato leaves from the C-GAN model. In the second set of experiments, the pre-trained DenseNet model has been trained on the original training set as well as on the combination of the training set (from the PlantVillage) and generated synthetic tomato leaf images.

To evaluate the performance of the proposed approach, we have used various performance metrics like accuracy, precision, recall, F1-score, and confusion matrix. In order to evaluate the quality of the synthetic images generated by C-GAN, we have used image quality measures like Peak Signal-to-Noise Ratio (PSNR) and Inception Score (IS).

Inception score uses the pre-trained Inception-v3 (Szegedy et al., 2016) model to classify the generated images. The model is used to classify a large number of generated images by predicting the probability of the image belonging to each class. These predictions are summed up into the inception score. The Kullback–Leibler (KL) divergence is calculated for each image as the conditional probability multiplied by the log of the conditional probability minus the log of the marginal probability.

**Results**

### *Performance of C-GAN*

The qualitative performance of C-GAN can be visually examined in Fig. 3, where it shows the original tomato leaf images from the dataset as well as the synthetic tomato leaf images generated by the C-GAN model. It can be seen in Fig. 3 that the generated synthetic images are looking closely similar to the real images. The PSNR values between original images and between original and synthetic images is represented in Table 6. The comparable values of PSNR show that the quality of images generated by C-GAN is very close to the quality of real images. The mean and standard deviation of inception scores have been calculated on original images as well as on the C-GAN synthetic images, which are represented in Table 7. The inception score of synthetic images is also very similar to the score of original images.

### *Performance of classification model*

We have evaluated the performance of proposed model for 5-class classification (YLCV, SptL, MscV, TLB and Hlth classes), 7-class classification (YLCV, SpdM, LMld, MscV, TTS, TLB and Hlth classes) and 10- class classification (YLCV, SptL, SpdM, Bctsp, LMld, MscV, TTS, TEB, TLB and Hlth classes) tasks. Table 2 represents the classification performance of the proposed method on the PlantVillage dataset and augmented dataset (PlantVillage + Synthetic images dataset). The proposed method achieved a classification accuracy of 99.51%, 98.65%, and 97.11% for 5-class classification, 7-class, and 10-class classification tasks, respectively. It can be observed from Table 2, DenseNet121 model with synthetic images have shown performance improvement in accuracy, precision, recall, and F1-score for all classes, as compared to the original dataset. This improvement in classification performance clearly indicates that data augmentation using the C-GAN model has prevented the network from over-fitting and helped the network to be more generalized. The class-wise performance of the proposed approach for 10-class classification task on augmented dataset with synthetic images is represented in Table 3.

*Performance comparison*

Performance comparison of the proposed DenseNet121 model with pre-trained networks, namely VGG19 (Krizhevsky et al., 2012), ResNet50, Inception-V3 (Szegedy et al., 2016), Xception (Chollet, 2017), and MobileNet (Howard et al., 2020) for 10-class classification, is shown in Table 10. All the models were trained first on the original PlantVillage dataset and then on augmented dataset consisting of synthetic images generated by the C-GAN model. The DenseNet121 model gave the best results among all the pre-trained models with an accuracy of 97.11% on the augmented dataset. The results of all the models clearly depict that an augmented dataset consisting of synthetic images generated by the C-GAN model has given better accuracy as compared to the original dataset.

In the past, several studies have been proposed for the tomato plant disease detection from tomato leaves images. We have compared our proposed method with some of the studies that are proposed on the tomato PlantVillage dataset. The performance comparison with existing methods is represented in Table 11. The studies presented in the table are mainly developed for three kinds of classification tasks, such as 5- class, 7-class and 10-class classification. For 5 class classification task, the performance of our approach is better than the performance in (Widiyanto et al., 2019). Likewise for 7 class classification task, the performance of our approach is comparable to the performance achieved in (Rangarajan et al., 2018). Similarly, for 10 class classification task, the proposed method has shown its superiority over the methods in (Agarwal et al., 2020; Durmus¸ et al., 2017; Elhassouny and Smarandache, 2019).

**Table 2. Accuracy, Precision, Recall, and F1-score for the model with and without augmentation. [10]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1- score** |
| **5 classes** |  |  |  |  |
| DenseNet121 | 98.16% | 0.83 | 0.97 | 0.98 |
| DenseNet121 + Synthetic images | 99.51% | 0.99 | 0.99 | 0.99 |
| **7 classes** |  |  |  |  |
| DenseNet121 | 95.08% | 0.94 | 0.94 | 0.94 |
| DenseNet121 + Synthetic images | 98.65% | 0.98 | 0.99 | 0.98 |
| **10 classes** |  |  |  |  |
| DenseNet121 | 94.34% | 0.95 | 0.92 | 0.93 |
| DenseNet121 + Synthetic images | 97.11% | 0.97 | 0.97 | 0.97 |

**Table 3. Precision, Recall and F1-score for different disease classes of tomato plant. [10]**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | **Precision** | **Recall** | **F1-score** |
| Yellow leaf curl virus | 0.97 | 0.97 | 0.97 |
| Septoria leaf spot | 0.97 | 0.98 | 0.97 |
| Two spotted spider mite | 1.00 | 0.97 | 0.98 |
| Bacterial spot | 1.00 | 0.97 | 0.98 |
| Leaf mold | 0.96 | 0.97 | 0.96 |
| Mosaic virus | 0.89 | 0.98 | 0.93 |
| Target spot | 1.00 | 0.99 | 0.99 |
| Early blight | 0.94 | 0.99 | 0.97 |
| Late blight | 0.99 | 1.00 | 0.99 |
| Healthy | 0.95 | 0.87 | 0.91 |
| **Mean** | **0.97** | **0.97** | **0.97** |

**Table 4. Comparison of performances of different Pre-trained Networks.**

|  |  |  |
| --- | --- | --- |
| **Method(s)** | **PlantVillage** | **PlantVillage + Synthetic images** |
| VGG19 | 89.60 | 90.90 |
| ResNet50 | 76.90 | 80.00 |
| Inception-V3 | 82.10 | 84.90 |
| Xception | 92.20 | 95.80 |
| MobileNet | 91.90 | 94.60 |
| DenseNet169 | 93.03 | 95.15 |
| DenseNet201 | 93.71 | 95.86 |
| **DenseNet121** | **94.34** | **97.11** |

**Conclusion**

Deep learning-based approaches have shown promising results in plant disease detection. In this study, we have proposed yet another deep learning-based method for tomato plant disease detection from tomato leaf images. In the proposed approach, synthetic images have been generated using Conditional Generative Adversarial Networks for data augmentation purposes. Thereafter a pre-trained DenseNet121 model is fine-tuned on the original tomato leaf images and synthetic images. The proposed data augmentation technique improves network generalizability and prevents it from the over-fitting problem. The proposed model achieves an accuracy of 98.16%, 95.08%, 94.34%, on the original PlantVillage dataset for 5-class classification, 7-class, and 10-class classification tasks, respectively, and it achieves an accuracy of 99.51%, 98.65%, 97.11% with the original PlantVillage + synthetic images dataset for 5-class classification, 7-class, and 10-class classification tasks, respectively. In addition, our experiment results show the proposed method’s superiority over the existing methods.

In future work, we wanted to extend this method for disease identification and classification to various parts of the plant, like fruits, stems, and branches. We also wanted to identify the different phases of the plant disease.

El artículo titulado ***"Tomato plant disease detection using transfer learning with C-GAN synthetic images"***, escrito por (Abbas et al., 2021) aborda una metodología de aprendizaje profundo para detectar enfermedades en las plantas de tomate mediante el análisis de imágenes de sus hojas. La metodología se basa en dos fases principales: la generación de imágenes sintéticas utilizando Redes Antagónicas Generativas Condicionales (C-GAN) para la ampliación del conjunto de datos, seguida de la clasificación de enfermedades utilizando un modelo DenseNet121 previamente entrenado.

La efectividad del método fue evaluada utilizando el conjunto de datos PlantVillage, que incluye imágenes de hojas de tomate con distintos tipos de enfermedades y hojas en estado saludable. Mediante el uso de C-GAN, se generaron imágenes sintéticas que se incorporaron al conjunto de datos existente. Posteriormente, el modelo DenseNet121 fue entrenado utilizando tanto las imágenes reales como las sintéticas.

Se llevaron a cabo dos tipos distintos de experimentos: el primero para el entrenamiento del modelo C-GAN y la generación de imágenes sintéticas, y el segundo para el entrenamiento del modelo DenseNet121 con el conjunto de datos original ampliado con las imágenes sintéticas. Se emplearon múltiples métricas de evaluación, como la precisión, la exactitud y el F1-score, para cuantificar el rendimiento del enfoque propuesto.

Un análisis comparativo con otros modelos preentrenados, tales como VGG19, ResNet50, Inception-V3, Xception y MobileNet, reveló que DenseNet121 obtuvo el máximo accuracy, alcanzando un 97.11% en el conjunto de datos ampliado para la clasificación 10 categorías de enfermedades. Estos resultados sobresalen en comparación con modelos de aprendizaje profundo alternativos.

No obstante, aunque el modelo DenseNet121 ajustado demostró una alta eficacia en la clasificación de enfermedades de la planta de tomate y la aplicación de C-GAN ayudó a evitar el sobreajuste, es importante evaluar de manera cualitativa la fidelidad de las imágenes sintéticas generadas, ya que podrían no tener el nivel de detalle que presentan las imágenes reales. Futuros estudios deben investigar la robustez del modelo bajo condiciones más variadas y con conjuntos de datos de mayor diversidad. Sería asimismo beneficioso realizar un análisis más exhaustivo sobre el impacto real de las imágenes sintéticas en el performance del modelo.

## Title: Tomato leaf disease classification using supervised learning techniques: contrasting analysis. International Conference on Advances in Computing, Communication and Materials (ICACCM).

**Authors:** Vandana Rawat, Neelam Singh, Bhavleen Kaur, Saksham Bora

**Methodology**

The methodology of the study consisted of the following components:

1. **Dataset**: They utilized all the images of tomatoes from the "Plant Village" collection, which contained 10 categories, among them pictures of healthy tomatoes.
2. **Preprocessing**: Obtaining and preprocessing images of tomato leaf disease, including noise removal.
3. **Machine Learning Techniques**: Applied various algorithms including SVM, K-Nearest Neighbor, Naïve Bayes, Decision Tree, Feed Forward Neural Network, Back Propagation Neural Network, Deep Neural Network, Conventional Neural Network, and Multi Kernel SVM.
4. **Evaluation Metrics**: Utilized Accuracy, Precision, F1-Score, and Recall for evaluating the performance of the learning algorithms.

**Results**

**Table 4. Comparison of ML Algorithms [11]**

|  |  |  |  |
| --- | --- | --- | --- |
| **Author and Year** | **Dataset** | **Algorithm** | **Accuracy (%)** |
| H. D. Gadade [2020] | PlantVillage dataset | Decision Tree | 67,00 |
| Jayanthi M.G [2020] | PlantVillage dataset | SVM | 90,00 |
| Nishant Vijay [2021] | PlantVillage dataset | KNN | 83.6 |
| Dr. D.K. Kirange [2020] | PlantVillage dataset | Naive Bayes | 67,00 |
| Dr. Sreelatha P [2021] | PlantVillage dataset | FFNN | 56.89 |
| Sridhar Udaya kumar [2021] | PlantVillage dataset | BPNN | 57.19 |
| K. Ch. Sri Kavya [2021] | PlantVillage dataset | Linear Kernel SVM | 59.28 |
| S. Karthick [2021] | PlantVillage dataset | RNN | 59.54 |
| Nishant Vijay [2021] | PlantVillage dataset | CNN | 98.5 |
| Shashikumar D.R. [2020] | PlantVillage dataset | MKSVM | 97.34 |

The results were presented as a comparative analysis of various machine learning techniques applied to the hydroponics dataset. The Convolution Neural Network (CNN) achieved the highest accuracy of 98.5%. Other algorithms' accuracies were also presented, highlighting the effectiveness of different methods in the tomato leaf disease classification. [11]

**Conclusión**

The study concluded that machine learning technologies are frequently utilized for tomato leaf disease prediction and prognosis. Among the various machine learning techniques compared, the Convolution Neural Network (CNN) method was identified as the best for detecting tomato leaf disease early on, with an accuracy of 98.5%. The study also emphasized the potential of hydroponics and the application of data-driven techniques in modern agriculture.

## Title: Computer-aided fusion-based neural network in application to categorize tomato plants. Springer-Verlag London Ltd., part of Springer Nature 2023

**Authors:** Rajyalakshmi Uppada · D. V. A. N. Ravi Kumar

**Methodology**

The methodology employed can be broken down into the following key steps:

1. **Image Pre-processing**: Utilizing Non-Subsampled Contourlet to acquire energy-detail components from the image dataset (Kaggle PlantVillage and Mendeley dataset).
2. **Cluster Extraction**: Modified K-means processing is used to extract colored clusters for tomato-leaf detection.
3. **Classification using SVM**: 38 extracted features are utilized by the Multiclass SVM (M-SVM) classifier for accurate categorization.
4. **Proposed M-CNN model**: A custom deep learning architecture is established for optimized unsupervised categorization. The model is based on the DenseNet-201 architecture and involves pre-processing stages followed by CNN categorization.

**Results**

**Table 6** Distinguished performance of NSC + K-Means + M-CNN classifier on specific tomato-leaves [13]

|  |  |  |  |
| --- | --- | --- | --- |
| **Category of tomato-leaf** | **Number of images** | **Average accuracy** | **M-CNN categorization** |
| Healthy Tomato-Leaf (D0) | 1528 | 99.32 | Healthy |
| Early-Blight (D1) | 1600 | 98.14 | Infected |
| Septoria-Leaf-Spot (D2) | 1701 | 99.17 | Infected |
| Leaf-Mold (D3) | 1523 | 99.39 | Infected |
| Bacterial-Spot (D4) | 1702 | 99.18 | Infected |
| Spider-Mites (D5) | 1610 | 99.32 | Infected |
| Late-Blight (D6) | 1527 | 99.27 | Infected |
| Target-Spot (D7) | 2247 | 98.53 | Infected |
| Mosaic-Virus (D8) | 1555 | 99.7 | Infected |
| Yellow-Leaf-Curl-Virus (D9) | 4286 | 99.55 | Infected |
| Powdery\_Mildew (D10) | 1004 | 99.09 | Infected |

**Table 6** Normalized confusion matrix for M-CNN classifier [13]

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Actual class | Predicted class | | | | | | | | | | |
|  | **D0** | **D1** | **D2** | **D3** | **D4** | **D5** | **D6** | **D7** | **D8** | **D9** | **D10** |
| **D0** | **98.98** | 0 | 0 | 0 | 0.13 | 0 | 0.13 | 0.76 | 0 | 0 | 0.13 |
| **D1** | 0.13 | **95.80** | 0.64 | 0.51 | 0 | 0.25 | 0.25 | 1.65 | 0.76 | 0 | 0.13 |
| **D2** | 0.64 | 0.91 | **94.48** | 0.82 | 0.78 | 0.67 | 0.64 | 0.80 | 0.51 | 0.76 | 0 |
| **D3** | 0.38 | 0.13 | 1.27 | **96.18** | 0.64 | 0.13 | 0.25 | 0.76 | 0.13 | 0.13 | 0.89 |
| **D4** | 0.25 | 0.25 | 1.27 | 0.25 | **96.82** | 0.13 | 0.64 | 0.25 | 0 | 0.13 | 0 |
| **D5** | 0.51 | 0.78 | 0.27 | 0.64 | 0.91 | **98.17** | 0.89 | 0.56 | 0.38 | 0.89 | 0.27 |
| **D6** | 0.16 | 0.13 | 0 | 0.13 | 0 | 0 | **99.95** | 0 | 0.13 | 0 | 0.27 |
| **D7** | 3.56 | 0.25 | 0.89 | 0.25 | 0.51 | 1.27 | 2.54 | **90.33** | 0.13 | 0.25 | 1.27 |
| **D8** | 0.13 | 1.40 | 0 | 0 | 0 | 0 | 0.51 | 0 | **97.96** | 0 | 0 |
| **D9** | 0 | 0 | 0.13 | 0 | 0 | 0 | 0.13 | 0.13 | 0.13 | **99.49** | 0.64 |
| **D10** | 0.64 | 0.91 | 0.82 | 0.78 | 0.67 | 0.64 | 0.80 | 0.51 | 0.76 | 0 | **94.48** |

These are the results of the proposed method:

* Achieved a model accuracy of 99.15% and an average precision of 95.6%.
* The combination of NSC, K-means, and M-CNN classifiers produced average sensitivity and specificity of 98.78% and 97.34% respectively.
* A detailed comparison with state-of-the-art approaches showed the model's superiority in terms of accuracy, precision, and handling more disease classes.

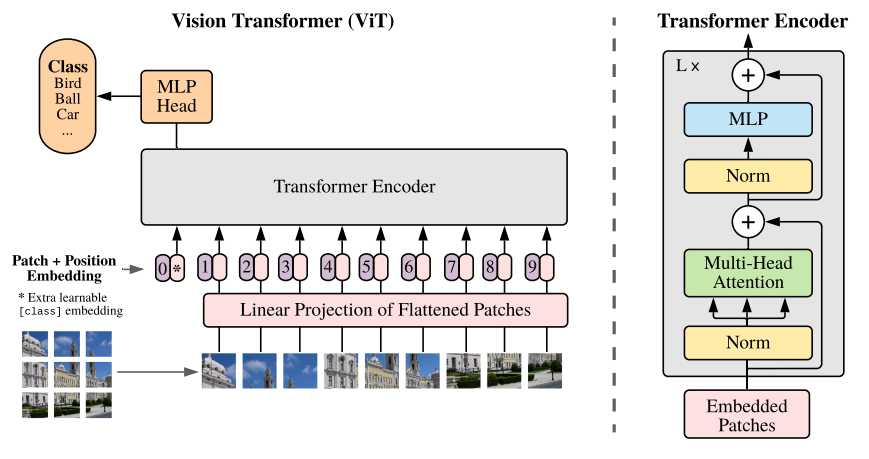
**Conclusion**

The study successfully introduces an efficient and accurate approach to detect and categorize diseased and healthy tomato plants. The proposed method, employing image pre-processing, clustering, SVM, and a customized CNN model, significantly outperformed traditional methods and other state-of-the-art approaches.

# THEORETICAL FRAMEWORK

**Introduction to the Vision Transformers (ViTs)**

Before getting into the Swin Transformer, it is essential to understand the concept of Vision Transformers (ViTs). These models adapt the transformer architecture, originally designed for natural language processing, for computer vision tasks. The key idea is to divide an image into patches, flatten them and feed them to a series of transformer layers, allowing the model to capture long-range relationships between different parts of the image [14].



**Figure 1:** Model overview of the first visual transformer [14]

**Limitations of Traditional ViTs**

Although ViTs have proven to be effective, they have certain limitations. Primarily, they require enormous amounts of data and computational power to train from scratch. In addition, the global attention used in traditional ViTs may not be optimal in terms of computational efficiency, especially for high-resolution images [14][15].

**General Architecture of the Swin Transformer: A Paradigm Shift**

The Swin Transformer, which stands for "Shifted Window Transformer," addresses the limitations of ViTs by introducing two key concepts: sliding windows and pyramid structure.

The Swin Transformer, like other Vision Transformers (ViTs), starts by dividing an image into non-overlapping patches using a splitting module. Each patch is considered a "token" and its feature is set by concatenating the RGB values of the raw pixels. In the Swin Transformer implementation, a patch size of 4x4 is used, resulting in a feature dimension of 48 (4 × 4 × 3). These features are subsequently projected to an arbitrary dimension (denoted as *C*) by means of a linear embedding layer.

The term "Stage" refers to the different stages of image processing through the architecture. In the Swin Transformer, the image resolution is successively reduced, from "Stage 1" to "Stage 4". This hierarchical structure is similar to traditional convolutional neural networks such as VGG and ResNet, where features are processed at different resolutions.

**Some detailed explanations:**

**MSA (Multi-Head Self-Attention):** a type of attention mechanism that allows the model to pay attention to different parts of the input simultaneously. It is essential to the transformer's ability to capture long-range relationships.

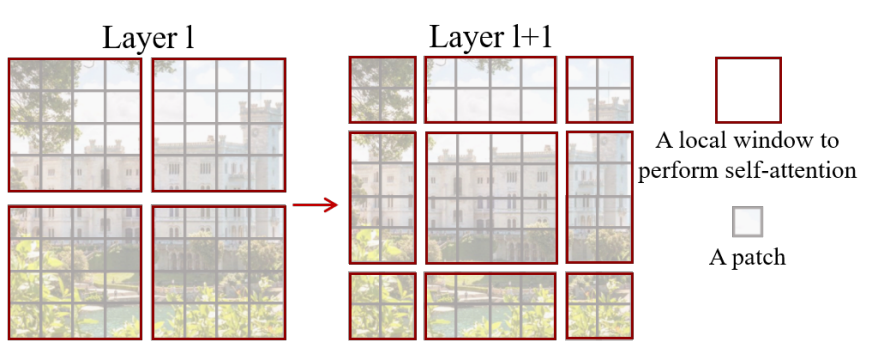
**MLP (Multi-Layer Perceptron):** Refers to a fully connected neural network. In the context of transformers, MLPs are used to transform features after layers of attention.

**GELU (Gaussian Error Linear Unit):** An activation function used in neural networks. It has been shown to have beneficial properties in transformer models.

**LN (LayerNorm or Layer Normalization):** It is a normalization technique used to stabilize and accelerate the training of neural networks.

**Ω(MSA) and Ω(W-MSA):** These are notations to represent the computational complexity of global attention and window-based attention, respectively.

**- Sliding Windows:** Instead of performing global attention over the entire image (which would have a quadratic complexity with respect to the number of tokens), the Swin Transformer introduces the concept of attention within non-overlapping local windows. This significantly reduces the computational complexity [16].



**Figura 2.** The shifted window approach to computing self-attention in the proposed Swin Transformer architecture.

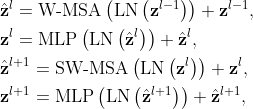
Equations (1) and (2) show the difference in complexity between global care and window-based care. Window-based attention is scalable and computationally more efficient [16].

(1) 

(2)

However, in order not to lose the ability to model relationships between patches in different windows, the Swin Transformer introduces the idea of windows shifted in successive blocks. This strategy alternates between two partitioning configurations in consecutive blocks of transformers, allowing each patch to relate to patches in neighboring windows.

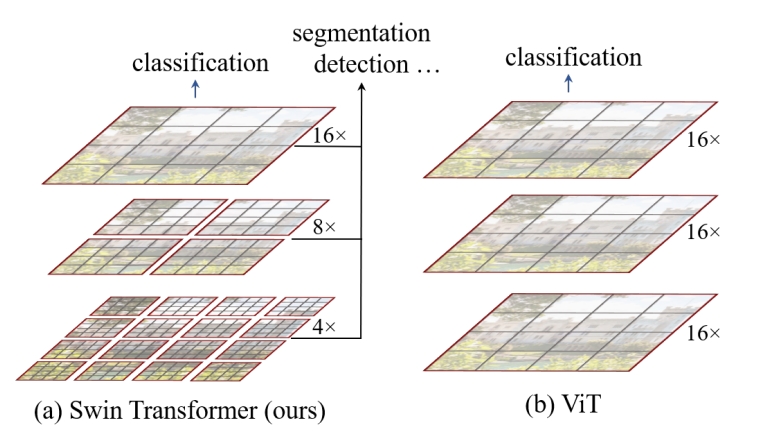
Equations (3) show how features are computed in successive blocks using different window-based attention configurations [16].

(3) 

In this context,  and  represent the resulting characteristics from the (S)W-MSA module and the MLP module corresponding to block . W-MSA and SW-MSA refer to window-based multi-head self-attention utilizing standard and shifted window partitioning setups, respectively.

Relative position bias is introduced to account for spatial relationships between patches within a window. This bias significantly improves model performance and is preferable to absolute position embeddings.

**- Pyramid Structure:** The Swin Transformer processes images in a series of resolutions, similar to a pyramid. It starts with small windows at a higher resolution and gradually clusters patches while reducing the resolution, allowing the model to capture features at different scales [16].



**Figure 3.** Comparison of the pyramidal structure method used by Swin tranformer and the method used by Vit.

**Architectural Details**

**- Tokenization and Embeddings:** Like other ViTs, the Swin Transformer starts by dividing the image into patches, which are then flattened and converted into embeddings through a linear layer [16].

**- Attention within Windows:** The attention layer in Swin Transformer operates only within sliding windows, using standard attention mechanisms. After each attention layer, the windows are scrolled to ensure coverage of the entire image [16].

**-Swin Transformer Block:** A Swin Transformer block is a fundamental unit in the architecture. It is where the shifted window-based attention occurs, which is essential for the efficiency and effectiveness of the model.

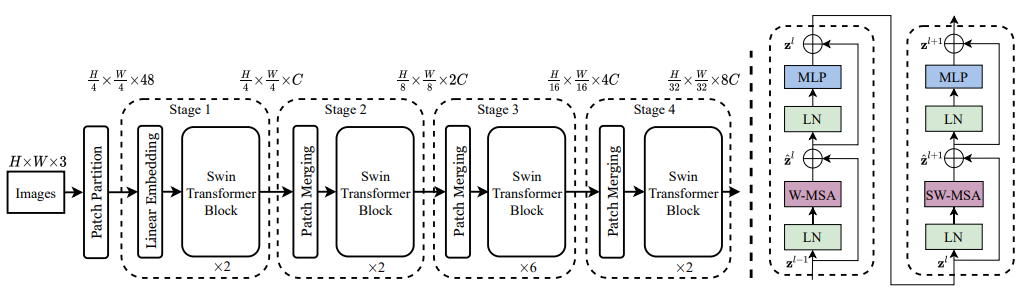
**-GELU:** GELU, or Gaussian Error Linear Unit, is a nonlinear activation function. It is used in neural networks to introduce nonlinearities into the model. In the Swin Transformer block, the GELU is used as an activation function in the 2-layer multilayer perceptron (MLP).

**-LN (LayerNorm):** LN, or Layer Normalization, is a normalization technique that is applied to features in a specific layer, rather than to a specific mini-batch. In Swin Transformer, LayerNorm is applied before each auto-tuned multi-head attention module (MSA) and each MLP. Normalization helps stabilize and accelerate training.

**-Residual Connection:** In deep networks, residual connections help avoid the gradient fading problem by allowing activations to skip one or more layers. In the Swin Transformer, a residual connection is applied after each attention module and MLP.

**- Resolution Reduction and Upscaling:** After processing the image at one resolution, the Swin Transformer reduces the resolution by grouping tokens and processing them in larger windows. This process is repeated several times, forming the pyramidal structure of the model [16].

**- Patch Merging:** "Patch Merging" is a technique used in the Swin Transformer to reduce the spatial resolution of the feature map and simultaneously increase the feature dimension, allowing the model to capture more abstract representations as it goes deeper. It works by aggregating information from neighboring patches and creating a new patch with a higher feature dimension.



**Figure 4.** On the left is the architecture of a Swin transformer (Swin-T) and on the right are two successive blocks of the Swin transformer W-MSA and SW-MSA which are multihead self-attention modules with regular and offset window configurations, respectively.

**Advantages of the Swin Transformer**

**- Computational Efficiency:** By avoiding global focus on the entire image, the Swin Transformer is significantly more efficient than traditional ViTs, especially for high-resolution images.

**- Modeling Capability:** The pyramidal structure allows the Swin Transformer to capture features at different scales, which can be crucial for tasks such as plant disease detection where patterns can vary in size.

**- Benchmarking:** Tests have shown that Swin Transformer performs very well and better on various data sets, outperforming other ViTs and CNNs in several benchmarks.

# REFERENCES

* Collins, E.J.; Bowyer, C.; Tsouza, A.; Chopra, M. Tomatoes: An Extensive Review of the Associated Health Impacts of Tomatoes and Factors That Can Affect Their Cultivation. Biology 2022, 11, 239.

Agarwal, Mohit, et al. “ToLeD: Tomato Leaf Disease Detection Using Convolution Neural Network.” *Procedia Computer Science*, vol. 167, no. 293–301, 2020, pp. 293–301, https://doi.org/10.1016/j.procs.2020.03.225.

Abbas, A., Jain, S., Gour, M., & Vankudothu, S. (2021). Tomato plant disease detection using transfer learning with C-GAN synthetic images. Computers and Electronics in Agriculture, 187(106279). https://doi.org/10.1016/j.compag.2021.106279

* Caruso, A. G., Bertacca, S., Parrella, G., Rizzo, R., Davino, S., & Panno, S. (2022). Tomato brown rugose fruit virus: A pathogen that is changing the tomato production worldwide. Annals of Applied Biology, 181(3), 258–274.
* Panno, S., Davino, S., Caruso, A. G., Bertacca, S., Crnogorac, A., Mandić, A., Noris, E., & Matić, S. (2021). A Review of the Most Common and Economically Important Diseases That Undermine the Cultivation of Tomato Crop in the Mediterranean Basin. Agronomy, 11(11), 2188. https://doi.org/10.3390/agronomy11112188
* Singh, V. K., Singh, A. K., & Kumar, A. (2017). Disease management of tomato through PGPB: current trends and future perspective. 3 Biotech, 7(4). https://doi.org/10.1007/s13205-017-0896-1
* Zahangir Alom, Taha, T. M., Yakopcic, C. G., Westberg, S., Sidike, P., Nasrin, Mst Shamima, Esesn, V., Abdul, & Asari, V. K. (2018). The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches. ArXiv E-Prints, 2. https://doi.org/10.48550/arxiv.1803.01164
* Boulent, J., Foucher, S., Théau, J., & St-Charles, P.-L. (2019). Convolutional Neural Networks for the Automatic Identification of Plant Diseases. Frontiers in Plant Science, 10. https://doi.org/10.3389/fpls.2019.00941
* Li, E. Y. (2020, August 1). 10 Papers You Should Read to Understand Image Classification in the Deep Learning Era. Medium. https://towardsdatascience.com/10-papers-you-should-read-to-understand-image-classification-in-the-deep-learning-era-4b9d792f45a7
* Rawat, V., Singh, N., Kaur, B., & Bora, S. (2022). Tomato Leaf Disease Classification Using Supervised Learning Techniques: Contrasting Analysis. 2022 International Conference on Advances in Computing, Communication and Materials (ICACCM), 1(2642-7354). https://doi.org/10.1109/icaccm56405.2022.10009617
* Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2. https://doi.org/10.1109/iccv48922.2021.00986
* Z. Liu et al., "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021, pp. 9992-10002, doi: 10.1109/ICCV48922.2021.00986.
* Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., ... & Asari, V. K. (2018). The history began from alexnet: A comprehensive survey on deep learning approaches.
* Boulent, J., Foucher, S., Théau, J., & St-Charles, P. L. (2019). Convolutional neural networks for the automatic identification of plant diseases. *Frontiers in plant science*, *10*, 941.
* Gadade, H. D., & Kirange, D. D. (2020). Machine learning approach towards tomato leaf disease classification. International Journal of Advanced Trends in Computer Science and Engineering, 9(1), 490-495.
* Abbas, A., Jain, S., Gour, M., & Vankudothu, S. (2021). Tomato plant disease detection using transfer learning with C-GAN synthetic images. Computers and Electronics in Agriculture, 187, 106279.
* Thangaraj, R., Anandamurugan, S., Pandiyan, P., & Kaliappan, V. K. (2022). Artificial intelligence in tomato leaf disease detection: a comprehensive review and discussion. Journal of Plant Diseases and Protection, 129(3), 469-488.
* Uppada, R., & Kumar, D. R. (2023). Computer-aided fusion-based neural network in application to categorize tomato plants.Springer-Verlag London Ltd., part of Springer Nature 2023, 1-9.
* Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale.
* Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021, July). Training data-efficient image transformers & distillation through attention. In International conference on machine learning (pp. 10347-10357). PMLR.