Journal of Engineering Research, Volume 29, Number 3, September 2024

Development of a Tomato Leaf Disease Detection System Using Convolutional Neural Networks

S.O Owoeye[2], F.O. Durodola, M.I Adekunle[1], A.T. Oyelami[1], B.M. Bisiriyu[1]
[1]Department of Mechatronics Engineering, Federal University of Agriculture, Abeokuta, Nigeria.
Email: owoeyeso@funaab.edu.ng

Abstract
The study and early detection of plant diseases are important because they harm plants and essential products like food, clothing, furniture, and housing. Plant diseases such as late blight, early blight, and septoria leaf spot significantly reduce tomato crop yields, posing major challenges to sustainable agriculture and food security. This paper aims to develop a cloud-based software system to accurately diagnose and recommend diseases in tomato leaves from images, providing recommended treatments for the detected diseases. The system integrates three neural networks: two Convolutional Neural Networks for object detection and feature extraction, and a third neural network for classification. These networks were trained on images from the Plant Village dataset, with preprocessing to enhance image quality and annotation accuracy. The system was tested and verified, achieving a training accuracy of 98% and a loss of less than 7.6% on the test set. However, real-world testing indicated an accuracy of around 80% for non-test-set images. The developed system has the potential to significantly aid in the early detection and management of tomato plant diseases, improving crop yields and agricultural sustainability. The software is accessible via an Application Programming Interface developed using TypeScript, and a frontend web application serves as a human-machine interface.

Keywords*:* Disease detection, Machine Learning, Tomato Leaf, Convolutional Neural Networks, TensorFlow.

1.0 **INTRODUCTION**

The tomato (*Solanum lycopersicum L.*) is a crop plant from the nightshade family (Solanaceae) that originated in the Andes of South America. It was introduced to Europe in the 16th century and has since become an essential crop cultivated worldwide (Gerszerg *et al.,* 2015). Tomatoes are among the most popular and economically significant vegetable crops globally, serving as a key ingredient in various dishes, particularly in West Africa (Zhenhua *et al*., 2017; Jeffrey, 2022). They can be marketed in fresh form or processed into products like pastes, concentrates, juice, and ketchup. Additionally, tomatoes are a rich source of vital nutrients such as lycopene, β-carotene, and vitamin C, all of which contribute positively to human health (Bergougnoux, 2014).

With the global population projected to reach 9 billion by 2037 (Pison, 2022), there is a pressing need to enhance tomato production to meet the rising demand. Advances in molecular biology and plant biotechnology have enabled the genetic engineering of tomatoes to improve resilience against biotic stresses (e.g., diseases, pests, weeds) and abiotic stresses (Gerszerg *et al.,* 2015; Tian, 2020). Despite these advances, no cultivar can completely withstand these stresses, underscoring the importance of early detection and mitigation of symptoms.

Traditionally, symptom detection involves visual inspection or laboratory analysis of leaf samples. However, these methods are often inefficient for large-scale adoption or lack reliability. Therefore, there is a critical need for a more efficient and reliable system for disease identification (Sanjeela & Jaswinder 2023). This paper aims to develop a cloud-hosted artificial intelligence system that utilizes Convolutional Neural Networks (CNNs) for detecting common diseases in tomato leaves. The CNN-based system offers superior efficiency and reliability compared to traditional methods of visual inspection or laboratory analysis.

Agarwal *et al*. (2020) proposed a novel technique for identifying diseases in tomato crops using a Convolutional Neural Network (CNN). The CNN architecture comprised three convolution and max-pooling layers with varying numbers of filters in each layer. The model was trained on an NVIDIA DGX v100 machine equipped with 40600 CUDA cores, 5120 tensor cores, 128 GB RAM, and 1000 TFLOPS. The proposed CNN model was trained on tomato leaf samples from the Plant Village dataset, and data augmentation techniques were employed to address class imbalance. These techniques included changing orientation, cropping, and modifying the colour scheme of images using the Python Augmenter package.

Xie *et al*. (2020) proposed a Faster Disease Recognition with Improved Attention Convolutional Neural Network (DR-IACNN) architecture based on the Faster Regional Convolutional Neural Network (Faster R-CNN) algorithm for disease detection in grape leaves. The architecture incorporated a Squeeze-Extraction block (SE block) and Inception modules into the ResNet backbone network of Faster R-CNN, resulting in an improved feature extraction method named INSE-ResNet. The experiments demonstrated that the Faster DR-IACNN achieved a mean Average Precision (mAP) of 81.1% and a detection speed of 15.01 FPS, outperforming Faster R-CNN in terms of precision and real-time detection capability. However, it should be noted that the dataset used for training was generated in a laboratory, which may lead to biased model performance when applied to real-world leaf images.

Aanis *et al*. (2023) provided an overview of 70 studies on deep-learning applications for disease diagnosis and management in agriculture. The literature review encompassed deep learning trends and methodologies, with a focus on plant disease diagnosis. The study highlighted major datasets used for training deep learning systems, including the PlantVillage dataset, Digipathos dataset, NLB dataset, RoCoLe, and Rice disease dataset. Findings from the study suggested an effective approach for early detection, identification, and severity estimation of crop diseases, emphasizing the potential for developing automated and efficient plant disease management systems.

Current tomato leaf disease detection systems are mostly academic or used by organizations with more resources, making it difficult for local farmers. This paper aims to provide a cost-effective and practical solution for the both local and urban farmers.

2.0 METHODOLOGY

The accessibility gap between rural farmers and well-equipped agricultural facilities necessary for optimal crop yield is one of the targeted aims of this paper, by providing an easy-to-use and efficient system for detecting common diseases in tomato leaves and offering recommended treatments for the detected diseases. Farmers can simply take a picture of a tomato leaf via a web application that transmits the image data to a cloud-hosted system that will determine if the picture is that of a tomato leaf or not. Valid tomato leaf images undergo analysis by a pretrained Convolutional Neural Network (CNN) to ascertain their health status. If diseased, the system provides recommended treatments. The CNN model would be deployed as a cloud-hosted system to facilitate easy training and updates, considering the project's target demographic may lack familiarity with technology. Additionally, the benefit of having a cloud-hosted solution is that; experts would be in charge of training the models and providing up-to-date treatments for the diseases.

The system is comprised of two components:

1. A cloud-hosted Convolutional Neural Network (CNN) accessible via an Application Programming Interface (API) written in the Typescript programming language. The API includes An internet-accessible API endpoint providing a list of detected classes, and an internet-accessible API endpoint that accepts image files and returns a successful response if the image is that of a tomato leaf with additional metadata if the image contains a diseased tomato leaf. Otherwise, the endpoint returns an error with a specific error message indicating that a non-tomato leaf image was uploaded.

2. A web application for farmers to take pictures of tomato leaves, transmitting them to the cloud-hosted backend for real-time analysis and treatment recommendations.

The process flow chart illustrating the system operation during image processing is depicted in Figure 1.

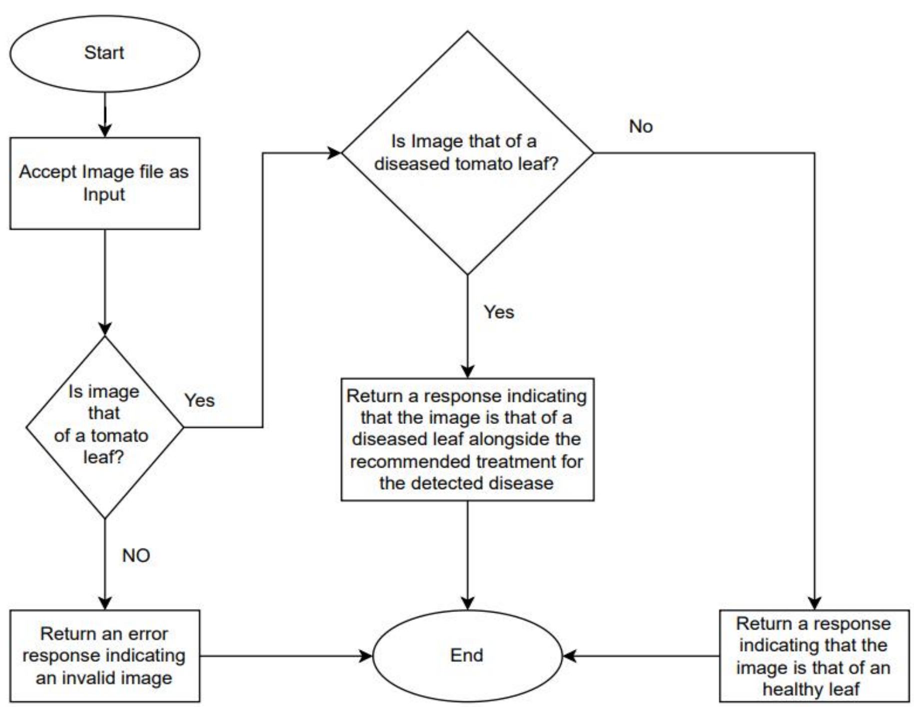

Figure 1: The system operation during image processing

The cloud-hosted backend was written in the Typescript programming language as there is a machine learning framework called "Tensorflow" available for the programming language. The Typescript programming language was also used to develop the frontend component of the cloud-hosted application as well as train machine learning models which significantly reduced the technological complexity of the software system. The web application for farmers would be developed using standard frontend development tools including HTML, CSS, Typescript, and React. A flow chart depicting the project flow from problem definition to system deployment is depicted in Figure 2.

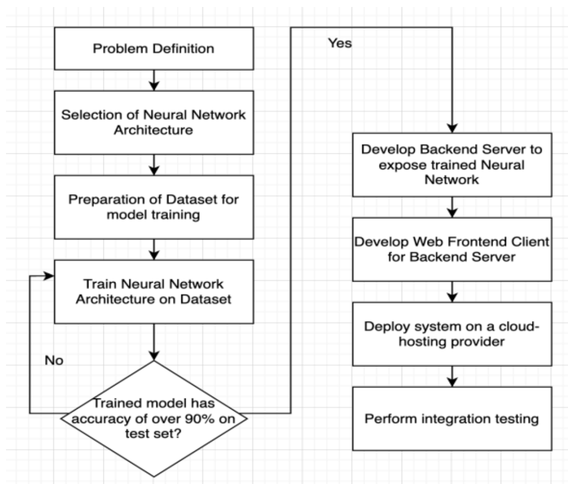

Figure 2: Project flow from problem definition to system deployment

2.1 Problem Definition

There is a significant disparity between local farmers in Nigeria and the essential agricultural facilities required for optimal crop yields. Additionally, many of these farmers may lack comprehensive and up-to-date knowledge about various tomato diseases and their corresponding treatments, which have been extensively researched and documented. This knowledge gap often leads to losses after planting seasons due to low crop yields resulting from diseased crops. This project aims to empower local farmers in Nigeria by providing them with a reliable and efficient means of identifying common diseases in their tomato plantations. Instead of relying on advanced facilities like agricultural laboratories, farmers can simply take a picture of a tomato leaf via a web application and receive real-time analysis results. The system should provide probability scores for each identified disease along with treatment recommendations.

2.2 Selection of Neural Network Architecture

The system is expected to identify tomato leaves, identify certain features on the detected tomato leaf and classify the leaf as healthy or not within an uploaded image. This is an object detection task that encompasses feature detection and classification. For this project, transfer learning is used in combination with the Imagenet and ResNet V2 50 model to train a feature extraction and classification model to recognize tomato leaves in images along with any distinguishing features on the tomato leaves for disease detection.

2.3 Preparation of Dataset for Model Training

The tomato leaf images are sourced from the PlantVillage dataset (Kaggle dataset) which contains over twenty-five thousand combined images of healthy tomato leaves and tomato leaves with the following diseases; Bacterial spot, Early blight, Late blight, Leaf mould, Septoria leaf spot, Spider mites: Two-spotted spider mite, Target spot, Tomato yellow leaf curl virus, Tomato mosaic virus, and Powdery mildew. Since the model is expected to differentiate between tomato leaf images, whether diseased or not, and non-tomato leaf images, it was necessary to train the model on non-tomato leaf images as well. For this purpose, a subset of the ImageNet dataset used to train the Mobilenet neural network architecture is used to train the CNN to recognize unknown objects (Andrew *et al*., 2017; Joseph, 2023). The dataset was split into two groups; the training set and the test set in a ratio of four to one.

2.4 Training the Neural Network

The model was trained on an Apple Silicon M2 Pro chip with 12 computing cores available in the chip coupled with 16 gigabytes of unified LPDDR5-6400 memory, using the TensorFlow machine learning framework. The model was trained over 30 epochs of the training dataset with a batch size of 50, the test dataset was used to calculate the model accuracy and loss.

2.5 Development of Backend Server and Frontend Web Client

The backend server and frontend web client were both developed with the typescript programming language. The backend server was developed using the express web application framework while the frontend web client was developed using the ReactJS web framework. The system was deployed on a virtual machine with an ARM chip running on the AWS platform (a cloud hosting provider). It is accessible anywhere there is a stable internet connection.

3.0 RESULTS AND DISCUSSION

The Loss against Epoch graph is shown in Figure 3. It can be seen that loss decreases over time as a result of the model learning patterns in the dataset and making fewer errors in classifications.

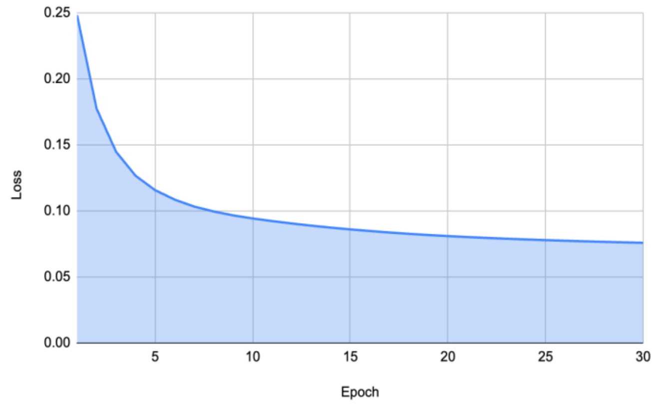

Figure 3: Graph of Loss against Epoch

Figure 4 represents the accuracy against epoch. It can be seen that the model accuracy increases over time as a result of the model learning patterns in the dataset and accurately predicting the image classes.

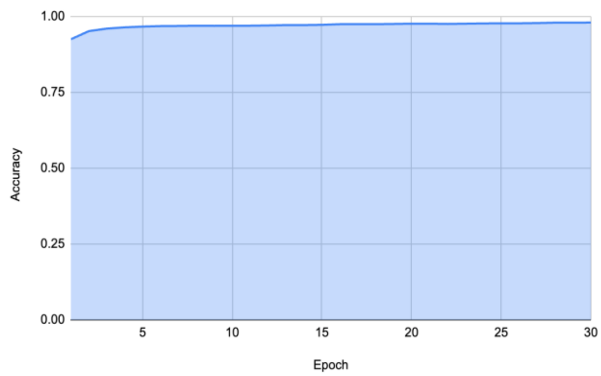

Figure 4: Graph of Accuracy against Epoch

Figures 5, 6, and 7 show the graphs of Precision against Epoch, Recall against Epoch and F1 Score against Epoch respectively. It can be seen that the graphs plateau after the 19th epoch indicating that further training may not result in significant improvements in the model's performance and this can be observed as improvements in model accuracy and loss start to decline after the 19th epoch

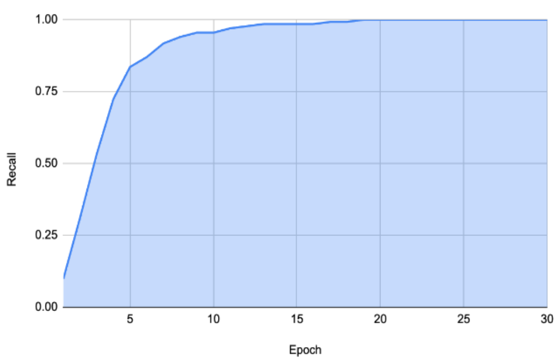

Figure 5: Precision against Epoch

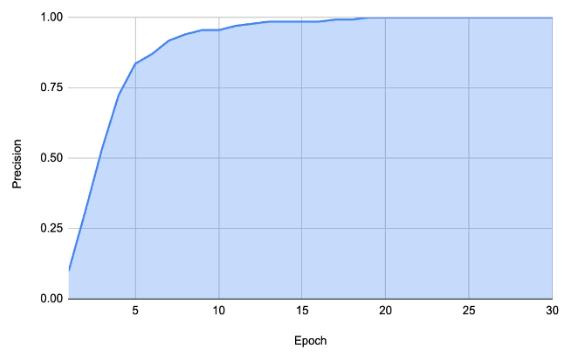

Figure 6: Recall against Epoch

The model performs well across different evaluation metrics. It maintains a low loss value of **0.075 (7.5%)**, which helps in minimizing errors during training. With an accuracy of **0.9804 (98%)**, it accurately predicts class labels for most samples. Notably, it achieves perfect **Precision**, **Recall**, and **F1 score** of 1, indicating its ability to identify positive instances without any false positives or negatives. Overall, these results suggest that the model is reliable for its intended task.

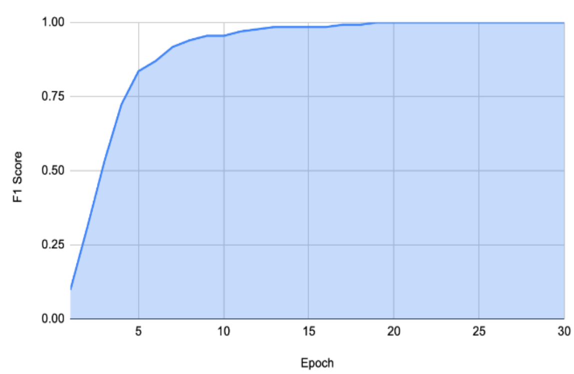

Figure 7: F1 score against Epoch

3.1 Testing on Samples from Training Dataset

The image of a tomato leaf with the label "Bacterial Spot" was selected from the training dataset to determine how well the system recognizes images from the training dataset. The result is shown in Figure 8.

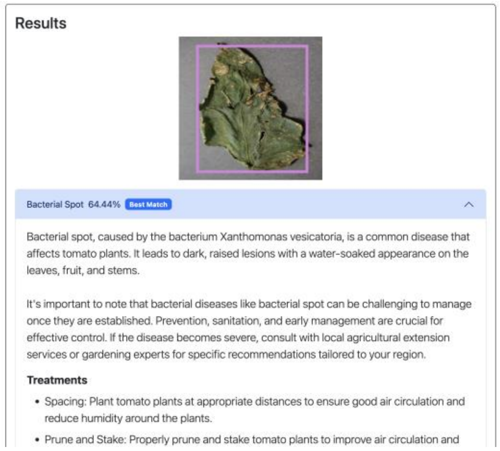

Figure 8: Detection results for an image from the training dataset

As seen above, the model predicts correctly that there is a **64.44%** chance that the image is that of a tomato leaf with the bacteria spot disease. It also gives more information on the detected disease and the possible treatments.

3.2 Testing On Real World Tomato Leaf Images

The image of a tomato leaf sourced from a vegetable garden suspected to be afflicted with either the Septoria leaf spot disease or the leaf mould disease was analyzed by the system. The result is shown in Figure 9.

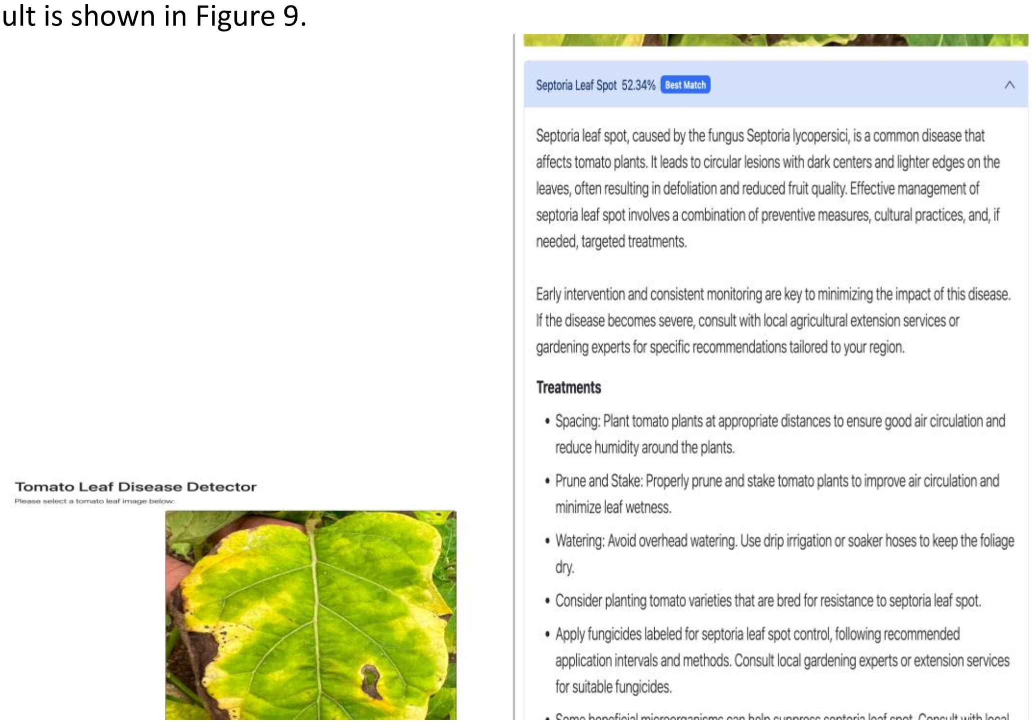

Figure 9: Detection results for an image from the training dataset

It can be seen that the model gives a prediction of 52.34% for septoria leaf spot which is recognized by small circular spots, yellowing of surrounding tissue, coalescing spots, and dark centres. It also gives a close prediction of 41.6% for leaf mould which is recognized by yellowing and chlorosis; pale spots with dark borders and leaf curling.

3.3 Testing on Non-Tomato Leaf Images

To test how well the model generalizes on non-tomato leaf images, some indoor objects were used to test its accuracy for the "unknown" object class. The result for a prediction on a shoe is shown in Figure 10.

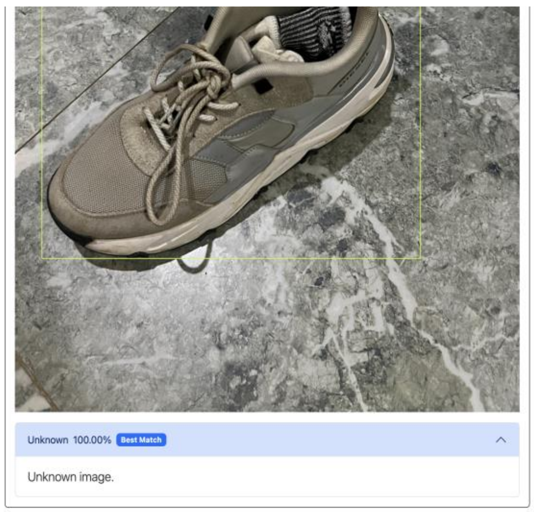

Figure 10: Detection results for the image of a shoe

As seen above, the model can generalize on non-tomato leaf images and label them as unknown since they are out of the scope of the model's operations.

3.4 Discussion

The model was trained using a data-augmented dataset comprising approximately one hundred thousand images. Its performance yielded an accuracy score of 98% on the test set, with a loss of less than 7.6%. However, subsequent real-world testing revealed an accuracy level lower than anticipated. Approximately 80% of non-test-set images were accurately predicted, indicating a slight overfitting of the model. This potential overfitting may stem from the limited quality and quantity of the training dataset. It is notable that production-grade object detection systems typically undergo training on datasets containing millions of images. This extensive dataset enhances the quality of extracted features for more accurate identification.

4.0 CONCLUSION

In conclusion, the developed system for detecting tomato leaf diseases has been successfully implemented. This system includes a cloud-hosted web application, offering end-users the convenience of uploading tomato leaf images from their internet-connected devices for real-time disease analysis. Moreover, the application provides recommendations for treating the identified diseases on their tomato plantations. A notable feature of the system is its ability to undergo remote training and updates without necessitating end-users to update the application on their devices, owing to its web-hosted nature. Furthermore, there is potential for continued enhancement of treatment quality and the object detection algorithm through the engagement of domain experts.

REFERENCES

Aanis, A., Dharmendra, S., & El Gamal, A. (2023). A survey on using deep learning techniques for plant disease diagnosis and recommendations for the development of appropriate tools. Smart Agricultural Technology, 3. https://doi.org/10.1016/j.atech.2022.100083

Agarwal, M., Singh, A., Arjaria, S., Sinha, A., & Gupta, S. (2020). ToLeD: Tomato leaf disease detection using convolution neural network. Procedia Computer Science, 167, 293-301. https://doi.org/10.1016/j.procs.2020.03.225

Andrew, H., Menglong, Z., Bo, C., Dmitry, K., Weijun, W., Tobias, W., Marco, A., & Hartwig, A. (2017, April). MobileNets: Efficient convolutional neural networks for mobile vision applications.

Bergougnoux, V. (2014). The history of tomato: From domestication to biopharming. Biotechnology Advances, 32(1), 170-189. https://doi.org/10.1016/j.biotechadv.2013.11.003

Gerszberg, A., Hnatuszko-Konka, K., Kowalczyk, T., & Kononowicz, A. K. (2015). Tomato (Solanum lycopersicum L.) in the service of biotechnology. Plant Cell, Tissue and Organ Culture, 120, 881–902. https://doi.org/10.1007/s11240-014-0664-4

Jeffrey, O. (2022). Explain artificial intelligence and the history of artificial intelligence. International Journal of Computer Science, International Journal of Computer Science, 9(1), pp 56-61

Joseph, V. R. (2023). Optimal ratio for data splitting. Statistical Analysis and Data Mining: The ASA Data Science Journal, 15(4), 531-538. https://doi.org/10.1002/sam.11583

Kaggle dataset. (n.d.). Retrieved March 17, 2024, from https://www.kaggle.com/datasets/emmarex/plantdisease

Pison, G. (2022). World population: 8 billion today, how many tomorrows? Population and Societies, (604). https://doi.org/10.3917/popsoc.604.0001

Sanjeela, S., & Jaswinder, S. (2023). An experimental study of tomato viral leaf disease detection using machine learning classification techniques. Bulletin of Electrical Engineering and Informatics, 12(1), 451-461. https://doi.org/10.11591/eei.v12i1.4385

Tian, T. (2020). Artificial intelligence image recognition method based on convolutional neural network algorithm. IEEE Access, 8, 125731-125744. https://doi.org/10.1109/ACCESS.2020.3006097

Xie, X., Ma, Y., Liu, B., He, J., Li, S., & Wang, H. (2020). A deep-learning-based real-time detector for grape leaf diseases using improved convolutional neural networks. Frontiers in Plant Science, 11, 529357. https://doi.org/10.3389/fpls.2020.00751

Zhenhua, L., Yun, Z., & Yunhao, L. (2017). Towards a full-stack DevOps environment (Platform-as-a-Service) for cloud-hosted applications. Tsinghua Science and Technology, 22(1), 1-9. https://doi.org/10.1109/TST.2017.7830891