International Journal on Robotics, Automation and Sciences

An Edge Convolution Neural Network Model for Plant Health Classification Using Camera

Charis Teoh Yi En and Kok Beng Gan *

***Abstract* – As per the Food and Agricultural Organization (FAO), plant diseases infect approximately 1.3 billion tonnes of crops. Historically, farmers relied on visual inspection for disease detection and classification. In this study, a Convolutional Neural Network (CNN) with five convolutional layers was used to accurately recognize plant diseases. A deployable CNN model was developed for classifying plant diseases, integrated into a web application with a camera, forming a vision system integrated with CNN model. The CNN model was trained using a public dataset comprising 19,384 images of potatoes, peppers, and tomatoes, collected under controlled conditions. These plants were chosen due to their common occurrence in Malaysia. The evaluation metrics F1 score were used to assess the model's performance. The accuracy and F1-score of the trained model were 97.2% and 97%, respectively.**

Keywords—Plant Health, Edge, Classification, Convolution Neural Network.

I. General

A study by a scientist from UC Agriculture and Natural Resources indicates that crop losses have surged from 10% to 40% due to plant diseases. In developing nations, over 80% of agricultural output is produced by small-scale farmers, making them particularly vulnerable to these losses. Climate change exacerbates this problem, leading to a significant crop loss from insects and diseases due to harmful viruses and bacteria. This plant health can devastate yields, limiting food availability and accessibility, and driving up prices. However, advancements in computer vision present a prospect to expand the use of artificial intelligence in agricultural [1]. Early detection of plant health can lead to effective treatments and boosting crop yields. The classification of plant diseases is challenging due to the vague symptom boundaries on leaves. Image analysis of plant diseases can be complicated by poor-quality images affected by factors such as lighting, resolution, and weather conditions. Moreover, different plant diseases may exhibit similar symptoms.

A multitude of literatures review have been studied and analyzed, focusing on the typical workflow of plant disease classification [2]. This approach encompasses several steps: acquiring images, pre-processing them, segmenting the images, extracting features, and finally classifying the images. Plant disease classification is accomplished by extracting features from images of the plants [3] and subsequently classifying these features [4]. Yang [5] suggests that while a lot of focus is on deep learning, specifically DCNN, CNN is adequate for image processing and can extract sufficient information. The author suggests that using a combination of a shallow CNN and a traditional machine learning algorithm can be more effective in detecting and classifying plant diseases. This approach requires lesser parameters compared to

*Corresponding Author email: kbgan@ukm.edu.my ORCID: 0000-0002-8776-5502

Charis Teoh Yi En is with Department of Electrical and Electronic Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia.Universiti Kebangsaan Malaysia (e-mail: A165100@siswa.ukm.edu.my).

Kok Beng Gan is with Department of Electrical, Electronic and Systems Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia. (phone: +603-8921-7149; fax: +603-8911-8359; e-mail: kbgan@ukm.edu.my).

International Journal on Robotics, Automation and Sciences (2025) 7, 1:1-6
https://doi.org/10.33093/ijoras.2025.7.1.1
Manuscript received: 26 Nov 2024 | Revised: 3 Jan 2025 | Accepted: 11 Jan 2025 | Published: 31 Mar 2025
Published by MMU PRESS. URL: http://journals.mmupress.com/ijoras

deep learning models. This model can achieve high precision, recall, and F1-score. Srivastava [6] also implemented CNN model with an accuracy of 88% for plant disease detection and classification. Toda [7] reduced 75% parameters by eliminating unnecessary layers by applying neuron-wise and layer-wise visualization techniques for plant disease detection and classification. Sladojevic [8] employed a deep CNN approach for plant disease detection and classification with 96.3% accuracy.

Gogul [9] explains that unique feature extraction method makes CNNs particularly effective for image classification, as they extract features from lower to higher levels in an image. ANNs can easily overfit when dealing with large images, which also require more powerful processors due to their complex vectors. When comparing the performance of CNN, ANN, and SVM in classifying vegetation species, they achieved accuracies of 99%, 94%, and 91% respectively [10].

A lack of sufficient leaf image samples can result in overfitting. To address this, Arsenovic [11] carried out a study employing conventional augmentation techniques and GAN to expand the dataset. The model trained in this manner achieved an accuracy of 93.67%. Future research should focus on detecting and categorizing various plant diseases in different geographic area. Harte [12] trained a CNN model with augmentation and transfer learning, achieving 97.2% accuracy and an F1 score over 96.5%. This model has been deployed as a web application. Rishiikeshwer [13] showed that CNN model with 3600 augmented datasets can achieve 98% accuracy, but with 400 leaf images the accuracy dropped to 95%.

A significant number of deep learning models don't perform well when applied to independent data. As a result, numerous research efforts have been made to explore how the use of segmented images can enhance model accuracy. Both Chowdhury [14] and Paul Sharma [15] have addressed this issue by training CNN models with segmented image data. Sharma's [16] showed that the S-CNN model with segmented images achieved an accuracy of 98.6% on independent data, surpassing the F-CNN model trained with full images. Chowdhury's modified U-net segmentation model [14] achieved an accuracy of 98.66%, an IoU of 98.5%, and a Dice score of 98.73%. Additionally, EfficientNet-B7 achieved accuracies of 99.12% for six category classification and 99.95% for binary classification.

The EfficientNet-B4 model can classify 10 classes with segmented images at 99.89%. However, the quality of image segmentation remains an issue. Hassan [17] employed models such as InceptionV3, MobileNetV2, InceptionResnetV2, and EfficientNetB0 for the detection and classification of plant diseases. The dataset was divided into an 80-20 split, with 80% of the images used for training and 20% for testing. This division helped reduce both the computational cost and the number of parameters. The EfficientNetB0 model achieved an accuracy of 99.6%. Training the images on the MobileNetV2 and EfficientNetB0 architectures required 565 and 545 seconds per epoch, respectively, when using colored images.

In CNN, the feature maps are repeatedly extracted through convolution for image classification [18]. The network then produces a label representing the predicted class. The CNN network parameters can be optimized using gradient descent and back-propagation methods [19]. Tabbakh and colleagues [20] have shown the effectiveness of using Vision Transformer to extract deep features from leaves, but they have not yet implemented the model in a mobile application.

Petrellis [21] has successfully created a mobile application for plant disease monitoring with accuracy over 90%. Rishiikeshwer [13] has utilized a CNN which is integrated into an IoT Web Application designed to acquire, process and display the predicted name of the plant disease. Ramcharan [22] has trained a CNN model that is implemented in a mobile application. However, the F1-score performance dropped by 32% when applied to real-world images and videos. Therefore, it's crucial to supply ample images for training the CNN model to ensure its applicability in real-world scenarios. Numerous applications of Convolutional Neural Networks (CNN) are centered around image recognition, including systems for attendance tracking through masked face recognition [23], as well as biomedical applications [24-25].

The aim is to develop an edge CNN model to categorize plant diseases in real-time at the edge using a camera. The system's disease classification model, based on CNN, was trained with OpenCV and Python using 20,638 images of healthy and diseased potatoes, peppers, and tomatoes. The model can identify and categorize plant diseases by analyzing the leaves' conditions in real time using a camera.

II. METHODOLOGY

The system overview for plant disease classification system using deep learning is shown in Figure 1. The system is divided into two main components: model training and testing, and the real-time vision system. The system consists of Pixy2 camera web applications that use a CNN model to detect and classify plant diseases. The construction of the sequential model was accomplished using Python with the Tensorflow 2.3.0 and Keras frameworks. A CNN model was trained as part of this work, with the aim of detecting and classifying plant diseases.

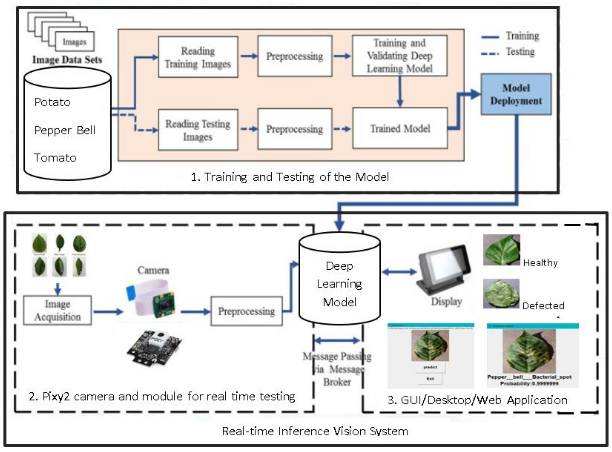

FIGURE 1. System overview

A. *Plant Disease Model Training*

A dataset, publicly accessible and comprising 20,638 images of both healthy and diseased specimens of potatoes, peppers, and tomatoes, was downloaded from Kaggle. These images were collected under controlled conditions. Given the decision to use a sequential model, there was no requirement for a validation set. The training dataset contained 19,384 images, while the testing dataset included 1,254 images. These images were distributed across 15 distinct categorical classes. Sample images of the plant diseases are displayed in Figure 2.

FIGURE 2. Sample images of plant disease

The model employed 10-Fold cross validation in this study. The data was randomly split into 10 equal-sized subsets, following a 90-10 split. To address GPU memory limit issues while maintaining a larger batch size (64 in this case), each image was resized to 48×48 and segmented as part of the preprocessing step. The number of train and test plant disease images is shown in Figure 3.

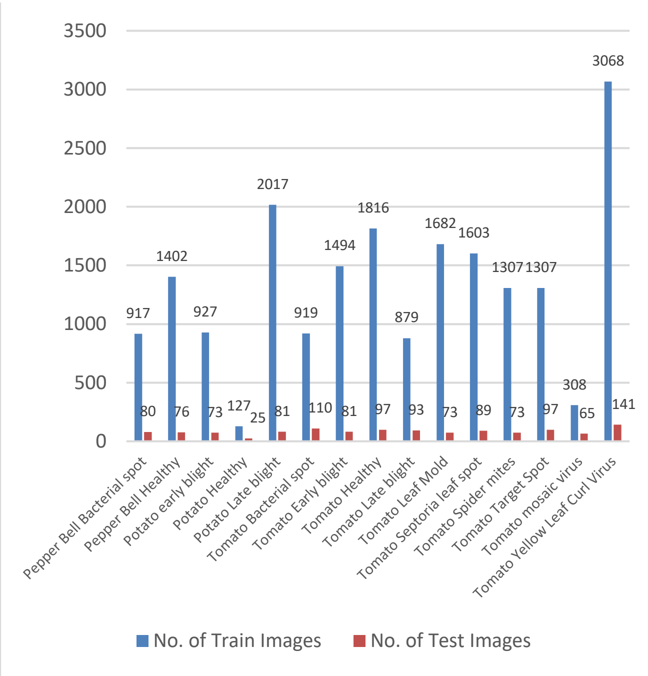

FIGURE 3. Number of train and test plant disease images

The augmentation was utilized to expand the dataset, mitigate overfitting, and equalize the image distribution. To artificially diversify the samples, realistic transformations were applied to the training images. This not only curbed overfitting but also exposed the model to various facets of the training data. It was crucial to design models that accept raw data as input, as opposed to preprocessed data. If a model is designed to expect preprocessed data, the same preprocessing pipeline would need to be replicated when exporting the model for use in web or mobile applications.

The architecture of the CNN model employed in this study is depicted in Figure 4. The model's top layer was subjected to training, followed by a comprehensive fine-tuning of the entire model. The initial five layers of the model consisted of Conv2D layers, which processed the input images. The first layer had 64 filters, the second layer had 128 filters, the third layer had 256 filters, and the fourth and fifth layers each had 512 filters. Each layer of filters was designed to detect patterns.

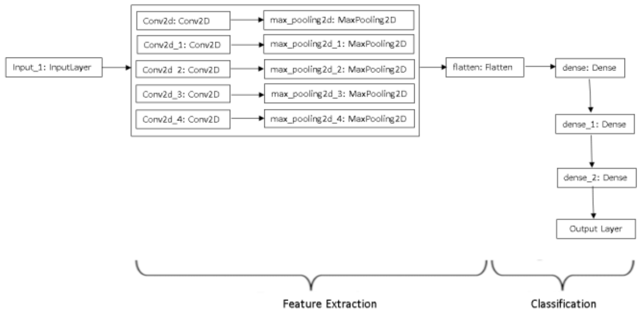

FIGURE 4. Block diagram of CNN model architecture

The convolutional layer was subsequently handed to max pooling, which facilitated the extraction of low-level features such as edges and points, thereby enhancing accuracy. During the pooling process, the maximum element in the feature map covered by the filter was selected. The output of feature map is the most significant feature. Each convolutional layer was combined with the ReLU (Rectified Linear Unit) activation function, as CNNs with ReLUs demonstrated faster and more reliable training, enabling models to learn more efficiently and function optimally. The images were processed through the layers multiple times to ensure superior feature extraction.

The model was trained with an epoch of 25. Increasing the number of epochs generally improved accuracy and reduced loss, thereby enhancing the precision of the plant disease classification model. The batch size was established at 64, as larger batch sizes can expedite training and potentially yield superior generalization performance. However, it's important to note that an increase in batch size would also necessitate more GPU memory.

A dropout layer was incorporated prior to the classification layer as a regularization measure to avoid overfitting in the model. When a dropout rate of 0.25 was applied, it resulted in a random elimination of 25% of the nodes from the neural network. The implementation of dropout gradually improved accuracy and reduced loss. The loss function employed was categorical cross entropy, with a lower score signifying superior model performance. The Adam Optimizer was selected with a learning rate of 0.05 due to its efficiency and ability to train the neural network in less time.

Classification involved fully connected classifiers, which were developed based on the model's various learnings. This CNN model utilized three dense layers to identify and classify these features. In a dense layer, all outputs from the preceding layer are connected to all its neurons, ensuring full connectivity. The 'Softmax' activation function was used, providing a probability for each predicted class. The pooled images were flattened, converting them into single-dimension vectors.

B. Evaluation metrics

The performance of the trained CNN model was assessed by the training accuracy, test accuracy, training loss, and test loss. Additionally, graphs were plotted to show the relationship between training accuracy and test accuracy over epochs, as well as training loss and test loss over epochs. Precision serves as an indicator of the classifier's accuracy.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

Recall is the number of positive sample images that are classified correctly.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

The F1-score, ranging from 0.0 (the worst) to 1.0 (the best), represents the weighted harmonic mean of recall and precision. It is used to compare different classifier models through its weighted average. Because the F1-score considers both false positives and false negatives, it may be lower but is generally more informative than simple accuracy measures.

$$F1 - score = \frac{2*(Recall*Precision)}{Recall+Precision} \tag{3}$$

III. Results And Discussions

The training and testing model achived similar accuracy of 97.2%. The model training loss is 8.5% while test loss is 11.9%. Table 1 showed the key predictive analytics metrics for 15 distinct categorical classes. The model's performance improves with a higher F1-score. Precision indicates the model's capability to classify plant diseases, while the computation of recall reveals its limitations. For accurate detection and classification of plant diseases, it's crucial to minimize false negatives due to their potential adverse effects. The model demonstrates commendable performance, achieving 0.97 average accuracy and class-weighted accuracy.

TABLE 1. Key predictive analytics metrics for 15 classes of plant disease images

Class ID	Class	Precisi-on	Recall	F1-score	Support
1	Pepper bell bacterial spot	1.00	0.94	0.97	80
2	Pepper bell healthy	0.97	0.93	0.95	76
3	Potato early blight	1.00	0.97	0.99	73
4	Potato late blight	0.95	1.00	0.98	81
5	Potato healthy	0.96	0.88	0.92	25
6	Tomato bacterial spot	1.00	0.98	0.99	110
7	Tomato early blight	0.96	0.90	0.93	81
8	Tomato late blight	1.00	0.98	0.99	93
9	Tomato leaf mold	1.00	0.99	0.99	73
10	Tomato septoria leaf spot	0.92	1.00	0.96	89
11	Tomato spider mites	0.94	1.00	0.97	73
12	Tomato target spot	0.90	0.95	0.92	97
13	Tomato yellow leaf curl virus	0.99	1.00	1.00	141
14	Tomato mosaic virus	1.00	0.98	0.99	65
15	Tomato healthy	1.00	1.00	1.00	97
Accuracy				0.97	1254
Macro average		0.97	0.97	0.97	1254
Weighted average		0.97	0.97	0.97	1254

Figure 5 illustrates the plot of training accuracy in relation to the epoch, while Figure 6 depicts the plot of training loss against the epoch. The model test accuracy and loss were 97.2% and 11.9% respectively.

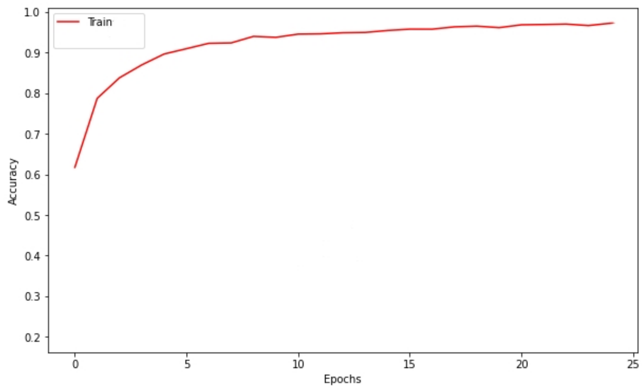

FIGURE 5. Model accuracy

Finally, the developed model was integrated into a web application built with HTML and CSS. A Pixy2 camera was used to interface with the web application, creating a real-time inference vision system for detecting and classifying plant diseases. Figure 7 illustrates this real-time inference vision system using the CNN model.

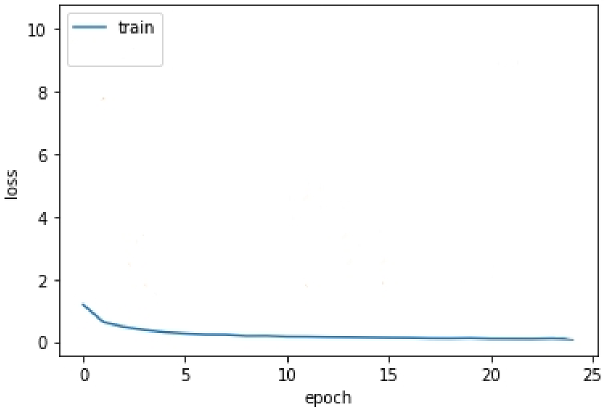

FIGURE 6. Model loss

FIGURE 7. Camara system with CNN model for detecting and classifying plant diseases.

Author details are listed in the footnote provided on the first page of manuscript. Use Arial font with size of 8 for the texts in the foot note. Corresponding author email should be provided in the first row of foot note. It's then followed by all author's affiliation information. Author details must show the full name of author (should be same with name stated under the article title), Department name of organization (of affiliation), Name of organization (of Affiliation), City, Country, and author email. Please refer to the footnote for examples.

IV. CONCLUSION

In conclusion, the CNN model achieved 97.2% accuracy and 97% average F1 score, indicating its effectiveness in detecting and classifying plant diseases. The model has been deployed on web applications and with a camera to create a real-time system. This system can offer valuable insights for farmers, helping them apply pesticides more effectively to treat plant diseases, thus benefiting the agricultural sector. Future efforts will focus on further improving the model's accuracy.

The current GPU faces memory constraints, limiting its ability to perform advanced deep processing. To fully leverage deep CNNs and extract a broader range of plant disease features, a more powerful GPU with increased memory is needed. The current system can classify three classes and 15 types of plant diseases. Expanding the system's training to include a wider variety of plants and diseases is proposed to broaden its scope. To improve the system's accuracy, more images of different plants are needed for the extraction of additional plant features. In future, capturing diverse array of plant images dataset can enrich dataset to develop superior models. Future improvements in accuracy are anticipated with the use of more advanced algorithms.

ACKNOWLEDGMENT

This research was partially funded by "Ministry of Higher Education under Fundamental Research Grant Scheme, grant no. FRGS/1/2020/TK0/UKM/02/14".

AUTHOR CONTRIBUTIONS

Charis Teoh Yi En: Conceptualization, Data Curation, Methodology, Validation, Writing – Original Draft Preparation;

Kok Beng Gan: Project Administration, Supervision, Writing – Review & Editing.

CONFLICT OF INTERESTS

No conflict of interests were disclosed.

ETHICS STATEMENTS

Our publication ethics follow The Committee of Publication Ethics (COPE) guideline. https://publicationethics.org/

REFERENCES

[1] Y.M. Alsakar, N.A. Sakr and M. Elmogy, "Plant Disease Detection and Classification Using Machine Learning and Deep Learning Techniques: Current Trends and Challenges," *Lecture Notes in Networks and Systems*, pp. 197-217, 2023. DOI: https://doi.org/10.1007/978-981-99-4764-5_13

[2] C. Sarkar, D. Gupta, U. Gupta and B.B. Hazarika, "Leaf disease detection using machine learning and deep learning: Review and challenges," *Applied Soft Computing*, vol. *145*, *no. 110534*, p. *110534*, 2023. DOI: https://doi.org/10.1016/j.asoc.2023.110534

[3] E.O. Asani, Y.P. Osadeyi, A.A. Adegun, S. Viriri, J.A. Ayoola, and E.A. Kolawole, "mPD-APP: a mobile-enabled plant diseases diagnosis application using convolutional neural

network toward the attainment of a food secure world," *Frontiers in Artificial Intelligence*, vol. *6*, pp. *1227950*, 2023. DOI: https://doi.org/10.3389/frai.2023.1227950

[4] V. Singh, N. Sharma and S. Singh, "A review of imaging techniques for plant disease detection," *Artificial Intelligence in Agriculture*, vol. *4*, pp. *229–242*, 2020. DOI: https://doi.org/10.1016/j.aiia.2020.10.002

[5] A. Nayak, S. Chakraborty and D.K. Swain, "Application of smartphone-image processing and transfer learning for rice disease and nutrient deficiency detection," *Smart Agricultural Technology*, vol. *4*, *no. 100195*, pp. *100195*, 2023. DOI: https://doi.org/10.1016/j.atech.2023.100195

[6] V.K. Trivedi, P. Shukla and A. Pandey, "Plant Leaves Disease Classification Using Bayesian Regularization Back propagation Deep Neural Network," *Journal of Physics: Conference Series*, vol. *1998*, *no. 1*, pp. *012025*, 2021. DOI: https://doi.org/10.1088/1742-6596/1998/1/012025

[7] J. Lu, L. Tan and H. Jiang, "Review on Convolutional Neural Network (CNN) Applied to Plant Leaf Disease Classification," *Agriculture*, vol. *11*, *no. 8*, pp. *707*, 2021. DOI: https://doi.org/10.3390/agriculture11080707

[8] K.P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. *145*, pp. *311–318*, 2018. DOI: https://doi.org/10.1016/j.compag.2018.01.009

[9] S. Sachar and A. Kumar, "Automatic Plant Identification Using Transfer Learning," *IOP Conference Series: Materials Science and Engineering*, vol. *1022*, *no. 1*, pp. *012086*, 2021. DOI: https://doi.org/10.1088/1757-899X/1022/1/012086

[10] A. Sabat-Tomala, E. Raczko and B. Zagajewski, "Comparison of Support Vector Machine and Random Forest Algorithms for Invasive and Expansive Species Classification Using Airborne Hyperspectral Data," *Remote Sensing*, vol. 12, no. 3, pp. 516, 2020. DOI: https://doi.org/10.3390/rs12030516

[11] J. Ma, K. Du, F. Zheng, L. Zhang, Z. Gong and Z. Sun, "A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network," *Computers and Electronics in Agriculture*, vol. *154*, pp. *18–24*, 2018. DOI: https://doi.org/10.1016/j.compag.2018.08.048

[12] A. Abade, P.A. Ferreira and F. de Barros Vidal, "Plant diseases recognition on images using convolutional neural networks: A systematic review," *Computers and Electronics in Agriculture*, vol. *185*, p. *106125*, 2021. DOI: https://doi.org/10.1016/j.compag.2021.106125

[13] V.K. Perumal, S. T, S. P R and D. S, "CNN based plant disease identification using PYNQ FPGA," *Systems and Soft Computing*, vol. 6, no. 200088, pp. 200088, 2024. DOI: https://doi.org/10.1016/j.sasc.2024.200088

[14] B.S. Sukhadeo, Y.D. Sinkar, S.D. Dhurgude and S.V. Athawale, "Plant disease detection using machine learning techniques based on internet of things (IoT) sensor network," *Internet Technology Letters*, pp. e546, 2024. DOI: https://doi.org/10.1002/itl2.546

[15] M. Jung, J.S. Song, A. Shin, B. Choi, S. Go, S. Kwon, J. Park, S.G. Park and Y. Kim, "Construction of deep learning-based disease detection model in plants," *Scientific Reports*, vol. 13, no. 1, 2023. DOI: https://doi.org/10.1038/s41598-023-34549-2

[16] S.S. Chouhan, U.P. Singh, U. Sharma and S. Jain, "Leaf disease segmentation and classification of Jatropha Curcas L. and Pongamia Pinnata L. biofuel plants using computer vision based approaches," *Measurement*, vol. 171, no. 108796, pp. 108796, 2020. DOI: https://doi.org/10.1016/j.measurement.2020.108796

[17] V. Tanwar and S. Lamba, "Classification of Multiple Maize Leaf Diseases Using a Blended Convolutional Neural Network," *3rd International Conference on Intelligent Technologies,* pp. *1–7*, 2023. DOI: https://doi.org/10.1109/CONIT59222.2023.10205797

[18] A.D. Coelho, B.G. Dias, W. de Oliveira Assis, F. de Almeida Martins and R.C. Pires, "Monitoring of Soil Moisture and Atmospheric Sensors with Internet of Things (IoT) Applied in Precision Agriculture," *2020 XIV Technologies Applied to Electronics Teaching Conference*, pp. *1–8*, 2020. DOI: https://doi.org/10.1109/TAEE46915.2020.9163766

[19] J. Chen, H. Yin and D. Zhang, "A self-adaptive classification method for plant disease detection using GMDH-Logistic model," *Sustainable Computing: Informatics and Systems*, vol. *28*, *no. 100415*, pp. *100415*, 2020. DOI: https://doi.org/10.1016/j.suscom.2020.100415

[20] P. Gole, P. Bedi, S. Marwaha, M.A. Haque, and C.K. Deb, "TrIncNet: a lightweight vision transformer network for identification of plant diseases," *Frontiers in Plant Science*, vol. *14*, pp. *1221557*, 2023. DOI: https://doi.org/10.3389/fpls.2023.1221557

[21] N. Petrellis, "Plant Disease Diagnosis for Smart Phone Applications with Extensible Set of Diseases," *Applied Sciences*, vol. 9, no. 9, pp. 1952, 2019. DOI: https://doi.org/10.3390/app9091952

[22] A. Ramcharan, P. McCloskey, K. Baranowski, N. Mbilinyi, L. Mrisho, M. Ndalahwa, J. Legg and D.P. Hughes, "A Mobile-Based Deep Learning Model for Cassava Disease Diagnosis," *Frontiers in Plant Science*, vol. 10, pp. 272, 2019. DOI: https://doi.org/10.3389/fpls.2019.00272

[23] J. J. How, S. C Tan and K. S. Liew, "Masked Face Recognition Attendance System Using A Modified Convolutional Neural Network," *International Journal on Robotics, Automation and Sciences*, vol. 4, pp. 23–29, 2022. DOI: https://doi.org/10.33093/ijoras.2022.4.4

[24] J.S.V. Leong and K.B. Gan, "Cuffless Non-invasive Blood Pressure Measurement Using CNN-LSTM Model: A Correlation Study," *International Journal on Robotics, Automation and Sciences*, vol. 5, no. 2, pp. 25–32, 2023. DOI: https://doi.org/10.33093/ijoras.2023.5.2.3

[25] J.P. Cheng and S.C. Haw, "Mental Health Problems Prediction Using Machine Learning Techniques," *International Journal on Robotics, Automation and Sciences*, vol. 5, no. 2, pp. 59–72, 2023. DOI: https://doi.org/10.33093/ijoras.2023.5.2.7