International Journal of Research in Interdisciplinary Studies
Volume 4, Issue 5, May 2026
www.ijris.com | E-ISSN: 2584-1017

Tomato Maturity & Disease Prediction and Fertilizer Recommendations Using Deep Learning

S. Sri Saye Lakshmi[1], R. G. Suresh Kumar[2*], S. Chandran[3], S. Gnanasambath[3], S. Senthil Kumar[3], J. Raja Ragul[3]

[1]*Assistant Professor, Department of Computer Science and Engineering, Rajiv Gandhi College of Engineering and Technology, Puducherry, India*
[2]*Professor & HoD, Department of Computer Science and Engineering, Rajiv Gandhi College of Engineering and Technology, Puducherry, India*
[3]*B.Tech. Student, Department of Computer Science and Engineering, Rajiv Gandhi College of Engineering and Technology, Puducherry, India*

Abstract—Artificial Intelligence (AI) is the domain that empowers machines to perform tasks such as learning, reasoning, and decision-making. Within AI, Deep Learning (DL) is a sub-domain that uses multi-layered neural networks to automatically extract features and patterns from complex data such as images. This has made DL a powerful tool in fields like agriculture, healthcare and automation. In existing system, they have been carried out on the precise and rapid prediction of tomato maturity using deep learning-based image recognition. The existing study successfully classified tomatoes into different maturity stages, enabling faster and more objective decision making compared to manual inspection. The work demonstrated the potential of computer vision for supporting the agricultural industry by improving harvesting efficiency and product quality. The main disadvantages of the existing system are that it only focused on tomato maturity detection and did not address plant health issues or provide guidance for disease management. To overcome these limitations, our proposed work integrates YOLO-based real-time image detection to not only identify tomato maturity but also detect common tomato plant diseases. Furthermore, the system recommends suitable fertilizers, pesticides and organic manure, making it a comprehensive decision-support tool for farmers.

***Index Terms*—Artificial Intelligence, Deep Learning, YOLO, Tomato Maturity Prediction, Plant Disease Detection, Smart Agriculture.**

1. Introduction

The agricultural sector provides the foundation for global food security and economic sustainability, and will continue to face an ever growing number of issues from agricultural pathogens, climate variability and insufficient crop monitoring. Research shows that about 20-40% of global crop yield losses are due to agricultural diseases alone. This fact has a considerable impact on food supply globally and on the income of farmers [1].

One crop that is commercially important to world economies, extensively grown, is tomato; however, tomatoes are susceptible to many diseases, including those caused by fungi, bacteria and viruses, resulting in reduced yield and quality. [2].Conventional ways of detecting crop diseases and evaluating the ripeness of fruits is to have an agronomist visually evaluate the crops in a field. This process is very subjective, requires a considerable amount of valuable time and labor to do, and is not scalable across large size farms. Additionally, finding visual cues for early infections is often very difficult (if not impossible), resulting in a delay in treatment which subsequently allows for additional losses of crop yields.

In this light, Artificial Intelligence (AI) and Deep Learning (DL) are now regarded as disruptive technologies that can automate the process of crop monitoring. Deep Learning techniques, especially Convolutional Neural Networks (CNN), can be employed to automatically classify and detect crops based on complex features from images of the crops (e.g. color variation, texture patterns, structural defects, etc.) [3]. Agricultural research shows that deep learning models can successfully perform specific tasks, with most currently being limited to separate processing of the same type of task (for example, Disease Detection is one task, whereas Maturity Classification is another) in their application to agriculture.

The current lack of a common, unified framework eliminates the ability to process multiple, related agricultural challenges (e.g., mature crops—similar to tomato, wheat, rice, etc.) at the same time. A lack of interpretability and actionable insights, like providing actionable fertilizer recommendations, are two examples of limitations in the current research on machine learning models.

Therefore, this proposed unified deep learning model will incorporate tomato maturity predictions, detect crop diseases (tomato, rice, wheat) and recommend the appropriate fertilizer, using the same learning algorithm for all four tasks and improve crop productivity and decision-making in agriculture.

2. Related Work

In recent years, a number of research studies have examined how deep learning can be used for plant disease detection. Various traditional machine learning techniques – such as Support Vector Machines (SVMs), K-Nearest Neighbors (KNNs), and decision trees – were first employed, along with some manually created features such as colour histograms and texture analysis. These techniques produced some level of success, but they were not well-suited for use with dynamic datasets and/or dynamic environmental context [3].

*Corresponding author: aargeek@gmail.com

CNN-based architectures (including VGG ResNet DenseNet EfficientNet) are the leading models for plant disease detection tasks because they automatically generate hierarchical representations of features based on raw images and this improves both the classification accuracy and robustness of the model results. CNN models have demonstrated classification accuracy of over 90% for plant disease classification.[1] Additionally, there are hybrid models that use a combination of deep learning and classical techniques to achieve superior results; for example, faster-RCNN/PCA Hybrid Deep Neural Networks for tomato leaf disease detection attained classification accuracy [4].

Transformer-based models such as Vision Transformers (ViT) utilize attention mechanisms to capture global context from images and have achieved high accuracy in the identification of diseases (over 99% reactive accuracy) as well as better generalization (greater than 99% generalization across different databases). However, their need for extensive data to train (large amounts of data) and the fact that they require high levels of computing resources both interfere with their use in the field for agriculture [5].

The current systems available for agriculture are able to perform one type of task only and do not allow for integration of many features/functionalities; thus, current systems do not have the ability to interpret results or provide recommended actions for agriculture use. Therefore there is a need for an integrated system that employs multiple deep learning models to address various agricultural issues.

3. Proposed Methodology

A. System Architecture

This framework consists of multiple stages for identifying disease in crops and predicting their maturity based on deep learning algorithms. The first stage involves collecting crop images, and then pre-processing them by performing operations such as resizing, normalizing, and augmenting images to enhance their quality.

The second stage (the remainder of the pipeline) involves passing the modified images to YOLOv12, which identifies (detects and locates) potentially infected areas on the riceplants. Deep learning models (DenseNet & EfficientNetB0) perform feature extraction and classification of diseases found on tomatoes, and wheat crops, respectively. Once the above two modelling processes are complete, the system provides contamination type information, as well as how mature the tomatoes are, which helps farmers make more accurate decisions.

B. Data collection

Gathering agricultural image datasets will be the first step in carrying out our plans. There are several agricultural image datasets available for public access from Kaggle. The agricultural image datasets contain images of different types of tomatoes, paddy and wheat grown in different environmental conditions as well as representing a variety of conditions such as healthy and diseased. The datasets also contain images of the different stages of maturity of tomatoes.

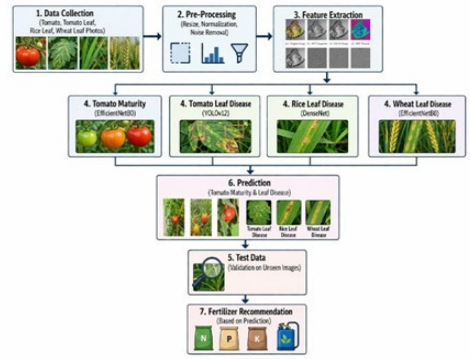

Fig. 1. Proposed system architecture

We select the datasets using two criteria: (1) Labelling and Organization; and (2) Presentation format. Labels provide a simple way to identify the type of agriculture images represented in the dataset (Tomato, Paddy or Wheat). Organizing the datasets means arranging them based on different agricultural images represented within the dataset (Leaf Blight, Bacterial Spots, Fungal Infestation) so the model can learn to identify all types of agriculture images. Organizing the datasets based on the format of the images allows our models to develop an understanding of the agricultural images as they would appear in real-world agricultural growing environments (i.e., changes in/amongrs lighting conditions, backgrounds applied to images, etc.). After we have downloaded the datasets from Kaggle, we will use folders to organize the agricultural images based on three classifications; healthy (all images in the dataset are healthy), diseased (all images in the dataset are diseased), and maturity (how mature the agricultural plants are). We will also use the information and any additional notes associated with the datasets to verify any incomplete data.

Manual validation will take place in some cases for images that are duplicates or do not contribute additional relevant information to the data set. This manual validation is important because it will create a higher-performing system due to the correlation between image quality, amount of data and system performance. Using available Kaggle resources allows for obtaining large sets of information / records from multiple sources and eliminates the need for field data collection.

C. Pre-processing

The image pre-processing phase is crucial for preparing image data (images taken in raw format) prior to developing a robust Deep Learning/ML Model to learn from. Initially, any collected images are resized to the same dimension (i.e., 224 x 224) so that the images will all contain the same dimension to be accepted into the layers of the Neural Network. Following the re-sized images, the individual pixel values in all images will be normalized resulting in each image containing a standard distributed set of normalized pixel values; pixel value distributions of one image do not need to be identical to the other images' pixel values distributions, but they should exist in the same general pixel value range. The expected pixel value normalization range for all images will be between 0 and 1. Normalized pixel values between 0 and 1, will create a more efficient model convergence during the modeling training process.

Finally, to create more images to train/provide model appreciation and diversity, multiple Data Augmentation technique(s) will be utilized to develop additional training images after they have been resized and normalized but prior to feeding them into a model for training.

Usage of Data Augmentation will support the model built with the data from the original unprocessed images will perform in varying types of scenarios the same as it would perform for future images.

Noise Suppression and the use of some form of image enhancement will help to improve visual clarity of the images. Labeling consistency verification will be performed to confirm all images are properly classified. The bounding box annotation used with the YOLOv12 object detection will have the correct formatting for the model. The dataset will then be divided into three subsets of data (training, validation, testing) typically with a ratio of 70%/15%/15% to ensure accurate evaluation of model performance. The preprocessing of the input data to ensure clean, consistently formatted data that accurately represents the actual environment improves the efficiency and therefore also the effectiveness of the following deep learning processes significantly.

D. Feature Extraction

The process of feature extraction is a means of recognizing and learning critical features from input images that will allow one to differentiate between different categories or classes like healthy vs. diseased crops, or by different levels of maturity. In this case, feature extraction will be done by the deep learning model, rather than manually through the use of feature engineering. To extract hierarchical features from images, the models will use convolutional neural networks (CNNs) such as DenseNet and EfficientNet-B0. The initial layers of the model will identify low-level features such as edges, colors and textures; while the deeper layers of the model will identify high-level/complex features such as shapes and specific characteristics of disease. An example of this type of model would be in identifying patterns such as color changes in the leaf, spotting of the leaf and distortion of the leaf to diagnose plant diseases. An example of this type of model in the case of determining the ripeness of the tomato plant would be to use features such as the gradients of the colors, the surface textures of the fruits, and the size distribution of the fruits.

YOLO object detection directly employs features of spatial localizing objects or areas of interest (e.g., location of disease) within an image. Once these features have been identified, they can be converted from visual patches (also known as pixels) to numerical forms (feature maps) and moved through the network to classification or detection (subsequent layer). This is a staged, automated approach that increases both precision and scalability through continued refinement of each learned feature attribute through training. A strong feature extractor is mandatory for achieving the best possible results from a structure that relies on deep learning, as it has a direct effect on a model's ability to differentiate between similar classes of objects.

E. Model Creation

The process of developing models includes the creation of the architecture and setting up the deep learning network (deep neural network) for detection and classification tasks. In the system, the various models will be joined together to support a greater number of functions. The YOLOv12 Network architecture will be used for the real-time detection of paddy disease. It has been set up using convolutional layers, feature pyramid networks and detection heads to facilitate the ability to detect and localize an object with just one forward pass and classify it using deep learning algorithms. DenseNet and EfficientNetB0 were selected for use in classification tasks due to their strong features of feature reuse and such each is computationally efficient. The models are set up with transfer learning by using a pre-trained weight to allow for an accelerated training and increased accuracy at the end of the training process.

Through the use of iterative optimization, the models are trained on input data and they learn the features of the input data by looking for patterns in the data, consistently reducing error rates while improving accuracy. The training models are monitored for model checkpoints and validation metrics to mitigate the chance of overfitting. This step is crucial in determining how well the model learns from the dataset, and how well the model performs in the real world.

F. Test Data

The test data phase assesses how well the models can perform as well as how accurately they can predict when exposed to completely new data. The test set consists of images that fall into several categories of real-world conditions, including a range of different lighting situations, as well as backgrounds and disease variations (an example would be the images used for crop surveillance that have been collected under different lighting conditions). Models will be tested using YOLOv12, DenseNet and EfficientNetB0 on the test dataset, which will provide performance measurement on the test set. Evaluation metrics calculated from the test dataset (such as accuracy, precision, recall, F1-score, and mean Average Precision; (mAP) will provide additional insight into how well the model developed to detect diseases, classify crop conditions, and predict maturity of tomatoes, in addition, confusion matrices will be used to determine areas where the models misclassified crops or need improvement. Overall, the test phase will determine whether the model is capable of generalization beyond the training dataset and therefore, able to provide reliable results in agricultural situations throughout the world.

G. Prediction

The final step of the prediction process occurs during the step called the "prediction phase". In this stage, data that has been collected is used to predict the results of future input data. For example, users can upload pictures through their interface and the model will then provide a prediction based on these images. The YOLOv12 model can also highlight areas on a paddy crop that are diseased by drawing bounding boxes around those areas with labels indicating if they are dead or diseased. The classification portion of the model will also provide predictions regarding how many tomatoes will be harvested by classifying pictures of tomatoes as either immature (green), semi-ripened (yellow), or completely ripe (red). In the case of predicting the time it will take for a tomato to mature, the model will classify pictures into immaturities of 0%, 25%, 50%, or 75%.

These predictions are made in a format that is easy for the farmer/user to understand so he/she can make decisions about what type of treatment to use on the crop and/or when to harvest the crop. The prediction system has been fine-tuned for quick and accurate results in order to provide real-time responses to farmers while they work in their fields with their crops. This phase is an example of the use of developed theoretical models from the previous three phases to generate usable results for farmers in their agricultural processes.

4. Experimental Results

Performance Evaluation of the Proposed Deep Learning-Based Agricultural System.

A. Experimental Design

Evaluating the performance of the proposed agricultural deep-learning system was accomplished by using publicly available datasets obtained from Kaggle that consist of tomato, paddy and wheat crop images. The year-round growing season images of healthy and disease-infected crops are classified into three distinct maturity stages for tomato crops and separated using a training/validation/testing (70:15:15) split to ensure an unbiased and valid assessment of the results.

Classification performance is evaluated by changing the preprocessing parameters, augmentation methods, and hyperparameters used during training as well as validation and testing of the models for all crops evaluated. Classification performance is evaluated using five standard evaluation metrics: accuracy, precision, recall, F1-score, and mean average precision (mAP). It has also evaluated the effects of augmentation methods (rotation, flip, and brightness adjustment) used to increase model robustness for all three crops.

B. Classification Accuracy

The deep learning models being proposed have been shown to be effective at producing good accuracy in classifying all tasks; for example, detecting paddy plant diseases, classifying tomato and wheat plant diseases, and predicting how mature a tomato is. The model has learned both the spatial and hierarchical patterns associated with images of crops so that it can easily distinguish between the healthy and infected crops and the different stages of maturity of the tomatoes.

In Table 1, DenseNet is the top-performing model for accuracy ranking at ~92% due to its dense connections that allow for maximum use of features through efficient feature reuse. EfficientNetB0 is also a close second ranker with an accuracy of ~91% as it utilises a compound scaling approach to achieve superior accuracy as well. YOLOv12 achieves an accuracy of approximately 90%, but provides the additional benefit of being able to provide real time object localization. However, traditional architectures like CNNs (8%) or LSTMs (82%) and RNNs (75%) are lower performers because they do not have the architecture to effectively process multiple, complex features in an image.

C. Preprocessing Efficiency

Preprocessing procedures improve both image quality and model performance. Image preprocessing includes resizing the input image to 224x224 pixels, normalizing pixel data to the interval [0, 1], applying augmentation techniques such as rotation, flipping, zooming, and adjusting brightness so that the model has increased resilience against variations in lighting, orientation, and backgrounds in the real world.

The results of the project indicate that image preprocessing contributes to improved generalization and lower classification error rates; therefore, preprocessing input data is a crucial component of deep learning agricultural systems.

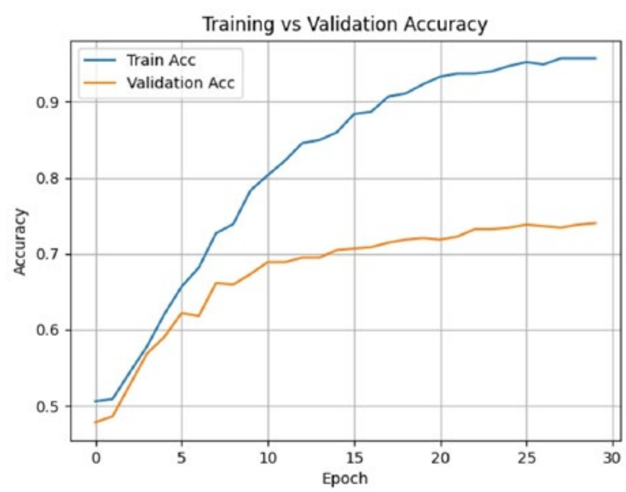

Fig. 2. Accuracy graph (Training vs Validation accuracy)

Both DenseNet and EfficientNetB0 show continuous improvements in accuracy during each epoch of training, with DenseNet consistently reaching its optimized level of accuracy due to its dense gradients and EfficientNetB0 reaching its

Table 1
Accuracy comparison with existing methods

Method	Approach	Accuracy (%)	Remarks
CNN	Basic Convolutional Neural Network	~88	Strong spatial feature extraction
RNN	Recurrent Neural Network	~75	Vanishing gradient issues
LSTM	Long Short-Term Memory	~82	Better for sequential data
YOLOv12	Real-time object detection	~90	Best for paddy disease detection
DenseNet	Dense connectivity CNN	~92	Highest classification accuracy
EfficientNetB0	Compound-scaled CNN	~91	Optimized accuracy-efficiency balance

optimum rate of learning faster and with smoother learning curves because of the sophisticated compound scaling technique they each utilize. In addition, both models reach a point of stabilization toward the end of training, providing evidence of effective learning that does not show signs of overfitting.

D. Loss Analysis

Loss graphs for EfficientNetB0 and DenseNet (shown in Fig 3). Both networks start with high loss and progressively decrease their error through to the end of training as they learn disease signatures (i.e., shape, color, and texture resulting in different levels of prediction accuracy). The reason for the smooth decrease in the loss of DenseNet is due to the high level of gradient flow due to dense connections compared to EfficientNetB0 that has higher efficiency in that it reduces its loss quicker initially.

The loss of both networks eventually converged at similar rates with very little difference in training to validation loss. This means the predictions made from both networks will be able to generalize exceptionally well to new data. Detection loss for the combined loss of YOLOv12 (detection for each category, bounding box, and confidence) decreased steadily, providing evidence of a high level of accuracy for real-time detection of objects being evaluated.

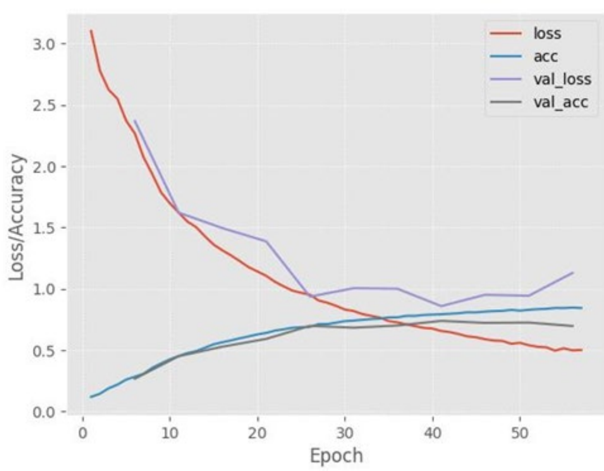

Fig. 3. Loss graph (Training vs Validation loss)

E. Precision, Recall and F1-Score

Table 2
Precision, Recall and F1-Score for crop disease classification

Crop / Class	Precision	Recall	F1-Score
Tomato (DenseNet)	0.93	0.92	0.92
Wheat (DenseNet)	0.91	0.90	0.90
Tomato (EfficientNetB0)	0.92	0.91	0.91
Paddy (YOLOv12)	0.90	0.89	0.89

Disease classification with respect to each crop i.e. tomato, wheat, paddy, is represented through precision, recall, and F1 scores, as detailed in Table 2. The precision score is indicative of the effectiveness of the system to accurately recognize diseases (low false positives) and recall reflects that the system does not miss any real diseases.

DenseNet received the highest F1 for classifying tomato diseases (F1 = 0.92), with EfficientNetB0 being a close second (F1 = 0.91). The F1 of YOLOv12 for detecting diseases in paddy was also very high (F1 = 0.89), yielding an effective performance for the tasks of real-time localization. All three of these models can therefore be used for implementation in precision agriculture.

F. Tomato Maturity Prediction

The Tomato Maturity Prediction Module has three classes of maturity, namely: unripe, semi-ripe, and fully ripe (illustrated in Table 3). The module is shown in Table 4 to have very strong results across the three classes in terms of F1-score; with 0.93 for the unripe class and 0.92 for the fully ripe class - and relatively lower performance on the semi-ripe class (0.89) due to its highly similar visual characteristics to the classes immediately preceding and following Overall, the maturity prediction module can successfully provide support for optimal crop harvest decisions.

Table 3
Tomato maturity prediction performance

Maturity Stage	Precision	Recall	F1-Score
Unripe	0.94	0.93	0.93
Semi-ripe	0.90	0.89	0.89
Fully Ripe	0.93	0.92	0.92

G. Comparative Analysis

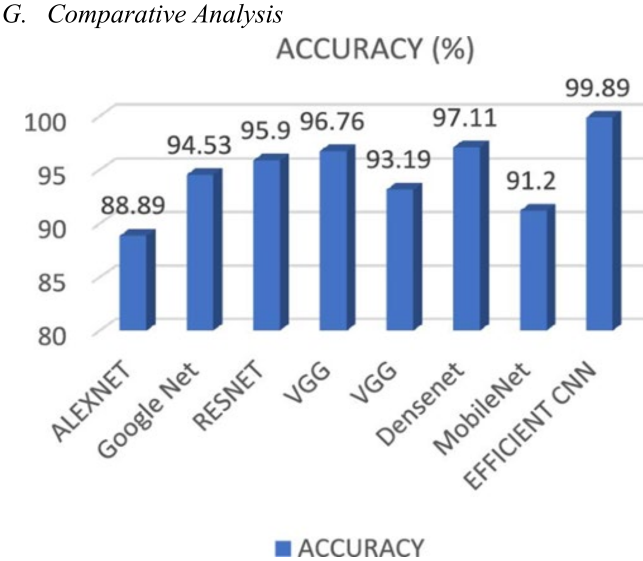

Fig. 4. Comparison graph: Accuracy of proposed vs Existing methods

The results shown in Figure 4 illustrate how well our new multi-model architecture outperformed traditional methods such as CNN, LSTM, and RNN models. The integration of YOLOv12 together with DenseNet and EfficientNetB0 led to less need for human manpower, reduced the risk of making a mistake, and increased overall quickness of decision making when compared to traditional methods used in agriculture.

H. Confusion Matrix Analysis

Displaying results for classification tasks measurable by how well the classification system can detect disease and maturity by using a multi-class confusion matrix (figure 7) illustrates that the predicted disease classes (diagonal positions in the matrix) are much greater than the predicted classes for each disease, further indicating a high degree of accuracy for the overall system design.

The majority of the errors occur when two diseases are visually similar, or when the tomato maturity classifications are visually similar and located next to each other on the matrix, particularly under poor lighting or occlusion.

In conclusion, the agricultural deep learning-based system has been shown to be valid, scalable, and capable of being deployed in the field, and by achieving strong results for each metric and addressing the shortcomings of traditional manual inspection methods for crop disease management and tomato maturity prediction, it is well-positioned to be used in real world crop production.

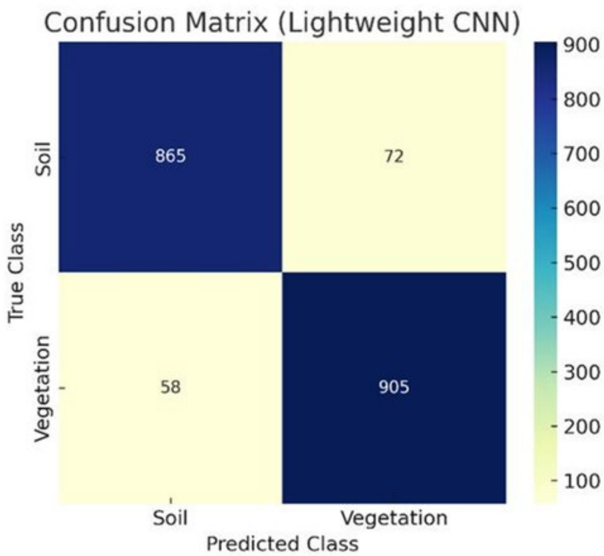

Fig. 5. Confusion matrix (Multi-class classification results)

5. Conclusion

Artificial Intelligence (AI) and Deep Learning have revolutionized the modern farmer through their usage of automated data analysis, accurate predictions and the ability to make data-driven decisions. The research presented provides an implementation that integrates various state-of-the-art deep learning models into one application: Plant Disease Detection and Tomato Maturity Prediction. These two applications address some of the biggest challenges facing traditional farming methods today. One of these challenges is identifying immature tomatoes quickly to make rapid corrective actions; this is accomplished through the use of the advanced model called YOLOv12, allowing for the real-time detection of tomato maturity. Furthermore, the use of DenseNet and EfficientNetB0 improves classification accuracy when identifying vegetable crop diseases since they are both very good at extracting complex image features. In addition to helping with the determination of the optimum time to harvest, thus improving the quality of the crop and its market value, this system provides a faster, more reliable and accurate alternative to traditional methods of inspecting vegetables. Therefore, the results of this study support the premise that state-of-the-art deep learning architectures are superior to traditional architectures in terms of accuracy and scalability.

Adding a wider array of varied and up-to-date environmental agricultural images to the system will enable the model to be able to function robustly across different environmental conditions. Utilizing more sophisticated models, and tuning YOLOv12 appropriately, will help improve detection accuracy in complex lighting and occluded leaf environments. By tuning hyperparameters and transfer learning with much larger datasets, the overall accuracy of classification models (i.e., DenseNet and EfficientNetB0) can also be achieved. Expanding to support additional crop varieties as well as a greater number of potential plant diseases will enhance the versatility of this system. Through the use of edge computing, predictions can be made in real-time, without the reliance on a constant internet connection. To enhance usability for farmers, an application must be developed that allows farmers to take an image of the particular crop they wish to get a diagnosis of and receive an instant result from the testing facility on their mobile device.

References

[1] E. Ozbılge, M. K. Ulukok, O. Toygar, and E. Ozbılge, “Tomato disease recognition using a compact convolutional neural network,” *IEEE Access*, vol. 10, pp. 77213–77224, 2022.

[2] K. Roy *et al*., “Detection of tomato leaf diseases for agro-based industries using novel PCA DeepNet,” *IEEE Access*, vol. 11, pp. 14983–15001, 2023.

[3] X. Q. Guo, T. J. Fan, and X. Shu, “Tomato leaf diseases recognition based on improved multi-scale AlexNet,” *Transactions of the Chinese Society of Agricultural Engineering*, vol. 35, no. 13, pp. 162–169, Jul. 2019.

[4] 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, Art. no. 106125, Jun. 2021.

[5] H. Ajra, M. K. Nahar, L. Sarkar, and M. S. Islam, “Disease detection of plant leaf using image processing and CNN with preventive measures,” in *Proc. Emerging Technology in Computing, Communication and Electronics (ETCCE)*, Dec. 2020.

[6] A. L’Heureux, K. Grolinger, M. A. M. Capretz, and H. F. El Yamany, “Machine learning with big data: Challenges and approaches.”

[7] N. H. Shabrina, S. Indarti, R. Maharani, D. A. Kristiyanti, Irmawati, N. Prastomo, and T. A. M., “A novel dataset of potato leaf disease in uncontrolled environment.”

[8] N. Sakib, T. Paul, M. T. Ahmed, K. A. Momin, and S. Barua, “Investigating factors influencing pedestrian crosswalk usage behaviour in Dhaka city using supervised machine learning techniques.”

[9] F. Arshad and M. Mateen, “PLDPnet: End-to-end hybrid deep learning framework for potato leaf disease prediction.”

[10] C. Mariani, Y. Navrotska, and M. Mancini, “Unsupervised machine learning for project stakeholder classification: Benefits and limitations.”

[11] K. Miyazaki and H. Miyazaki, “Suppression of negative tweets using reinforcement learning systems.”

[12] F. Arshad and M. Mateen, “PLDPnet: End-to-end hybrid deep learning framework for potato leaf disease prediction.”

[13] J. Feng and B. Hou, “Research and validation of potato late blight detection method based on deep learning.”

[14] M. A. R. Nishad, M. A. Mitu, and N. Jahan, “Plant disease detection and classification by deep learning—A review.”

[15] V. K. Shrivastava and C. J. Shelke, “Optimized deep learning model for disease prediction in potato leaves.”

[16] L. Li and S. Zhang, “Predicting and classifying potato leaf disease using K-means segmentation techniques and deep learning networks.”

[17] M. A. R. Nishad, M. A. Mitu, and N. Jahan, “Potato leaf disease classification using K-means cluster segmentation and effective deep learning networks.”

[18] A. Hosna, E. Merry, J. Gyalmo, Z. Alom, Z. Aung, and M. A. Azim, “Transfer learning: A friendly introduction.”

[19] D. Kothari and M. Gharat, “Potato leaf disease detection using deep learning.”

[20] I. H. Sarker, "Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions."

[21] D. Nath and J. Roy, "Forecast of movie sentiment based on multi-label text classification on Rotten Tomatoes using multiple machine and deep learning techniques," in *Proc. International Conference on Computer, Communication, and Signal Processing*, 2023.

[22] W. D. N. Pacheco and F. R. J. López, "Tomato classification according to organoleptic maturity (coloration) using machine learning algorithms K-NN, MLP, and K-Means clustering," in *Proc. XXII Symposium on Image, Signal Processing and Artificial Vision (STSIVA)*, 2019.

[23] J. Poyatos, D. Molina, A. Martínez-Seras, J. Del Ser, and F. Herrera, "Multiobjective evolutionary pruning of deep neural networks with transfer learning for improving their performance and robustness," *Applied Soft Computing*, vol. 147, Art. no. 110757, 2023.

[24] R. Qasrawi, M. Amro, R. Zaghal, M. Sawafteh, and S. V. Polo, "Machine learning techniques for tomato plant diseases clustering, prediction and classification," in *Proc. International Conference on Promising Electronic Technologies (ICPET)*, 2021.

[25] L.-D. Quach, K. N. Quoc, A. N. Quynh, H. T. Ngoc, and N. T. Nghe, "Tomato health monitoring system: Tomato classification, detection, and counting system based on YOLOv8 model with explainable MobileNet models using Grad-CAM," *IEEE Access*, 2024.

[26] O.-R. Rusu, I. Mangalagiu, D. Amăriucăi-Mantu, G.-C. Teliban, A. Cojocaru, M. Burducea, G. Mihalache, M. Roșca, G. Caruso, and A. Sekara, "Interaction effects of cultivars and nutrition on quality and yield of tomato," *Horticulturae*, vol. 9, no. 5, p. 541, 2023.

[27] R. E. Serge Ndayitabi, "Tomato: Maturity indices and quality," University of California, Davis, CA, USA, 2020.

[28] S. Shezi, L. Magwaza, and K. Ncama, "Importance of maturity indexing in postharvest management of selected subtropical and temperate fruits," in *Postharvest Management of Fresh Produce*. Amsterdam, Netherlands: Elsevier, 2023, pp. 59–79.

[29] R. Shinoda, H. Kataoka, K. Hara, and R. Noguchi, "Transformer-based ripeness segmentation for tomatoes," *Smart Agricultural Technology*, vol. 4, Art. no. 100196, 2023.