LOW-COST SENSOR AND DEEP LEARNING BASED METHOD TO SPRAY THE FARMS WITH CONTROLLED OR LIMITED PESTICIDES WITH EQUAL DISTRIBUTION

Project Reference No.: *49S_MCA_0265*

College	:	*St. Joseph's University, Bengaluru*
Branch	:	*Master of Computer Applications*
Guide(s)	:	*Dr. Deepa Nagalavi*
		Mr. Mohammed Mueen Pasha
Student(s)	:	*Ms. Jerusha Pamella Felix M*
		Ms. Divyashri R
		Ms. Devika J

Keywords:
Deep Learning, MobileNetV2, IoT, Plant Disease Detection, Pest Detection

Introduction:

1. Agriculture is a fundamental sector contributing to the economy and food security, especially in countries like India where a large population depends on farming. However, crop productivity is significantly affected by plant diseases and insect infestations, particularly in crops such as tomato. Early and accurate detection of these issues is essential to reduce yield loss and maintain crop quality.
2. Traditional methods rely on manual inspection, which is time-consuming, less accurate, and dependent on expert knowledge. With advancements in **deep learning**, image-based classification techniques have become more effective for detecting plant diseases and pests automatically.
3. In this project, datasets such as PlantVillage (for plant diseases) and IP102 (for insect classification) are used to build a unified detection system. A **MobileNetV2-based deep learning model** is applied using transfer learning to achieve efficient and accurate classification.
4. The system is further enhanced by integrating **Internet of Things (IoT)** concepts, where image data can be captured using smart devices and processed for real-time monitoring. This enables automated detection and supports timely decision-making for pesticide application.
5. Additionally, the model is optimized and converted into TensorFlow Lite format for deployment on mobile or embedded devices, making the system practical for field-level usage.

Objectives:

1. To develop a deep learning-based system for detection of tomato diseases and insect pests
2. To utilize PlantVillage and IP102 datasets for training and validation
3. To apply image preprocessing and data augmentation techniques
4. To implement transfer learning using MobileNetV2
5. To perform fine-tuning for improved model accuracy
6. To convert the trained model into TensorFlow Lite format
7. To enable prediction with confidence for decision support

Methodology:

The project begins with downloading the PlantVillage dataset for tomato diseases and the IP102 dataset for insect pest classification from Kaggle. The datasets are extracted and organized into separate folders for disease and insect classification.

For disease detection, the tomato dataset is loaded using ImageDataGenerator with rescaling and validation split. The dataset is divided into training and validation sets for model training. Similarly, for insect detection, the IP102 dataset is loaded and pre-processed using ImageDataGenerator with rescaling, and separate training and validation datasets are created.

A pretrained MobileNetV2 deep learning model is used as the base model for both disease and insect detection. The base model layers are frozen, and custom layers such as GlobalAveragePooling and Dense layers are added. The models are compiled using the Adam optimizer and categorical cross-entropy loss function. Both disease and insect models are trained separately using their respective datasets, and the trained models are saved.

In addition, both datasets are merged to create a combined dataset containing multiple classes. The merged dataset is pre-processed with augmentation techniques such as rotation, flipping, and zooming. A MobileNetV2-based model is then trained on the merged dataset for multi-class classification. Fine-tuning is performed by unfreezing selected layers to improve model performance.

The trained model is evaluated using accuracy and loss metrics. The final model is converted into TensorFlow Lite format for deployment on mobile or embedded devices.

The system supports IoT-based data collection, where plant images can be captured using mobile cameras or smart devices and sent to the model for real-time analysis. The system performs prediction on input images and provides class labels with confidence scores. Based on the prediction results, the system generates spray recommendations for effective pest and disease management.

Result and Conclusion:

1. In conclusion, the developed system successfully implements deep learning-based detection of plant diseases using image classification techniques. The disease detection model trained on the PlantVillage dataset achieved approximately 95% training accuracy and 92% validation accuracy, demonstrating strong performance in identifying tomato leaf diseases.
2. The training graphs indicate a steady improvement in accuracy and a reduction in loss, confirming effective learning of the model. Fine-tuning further enhanced the performance and stability of the model.
3. The system was tested using input images and successfully generated predictions including disease class along with confidence scores. It also provided severity levels and spray recommendations based on the detected condition.
4. Visualization through bar charts helped in understanding the top predicted classes and their confidence values. The model was successfully converted into TensorFlow Lite format, enabling deployment on mobile or embedded devices.
5. The system also supports IoT-based data capture, where plant images can be obtained using smart devices for real-time analysis and decision-making.
6. Overall, the project demonstrates a practical and efficient AI-based solution for plant protection, contributing to precision agriculture by enabling early detection and controlled pesticide usage.

References:

1. [1] D. P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *arXiv preprint arXiv:1511.08060*, 2015.
2. [2] X. Wu, Y. Chen, and L. Wang, "IP102: A large-scale benchmark dataset for insect pest recognition," *Proceedings of the IEEE Conference on Computer Vision Workshops*, pp. 878–887, 2019.
3. [3] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4510–4520, 2018.

4. [4] S. Mohanty, D. P. Hughes, and M. Salathé, “Using deep learning for image-based plant disease detection,” *Frontiers in Plant Science*, vol. 7, p. 1419, 2016.
5. [5] K. P. Ferentinos, “Deep learning models for plant disease detection and diagnosis,” *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018.
6. [6] M. Abadi et al., “TensorFlow: Large-scale machine learning on heterogeneous distributed systems,” *arXiv preprint arXiv:1603.04467*, 2016.
7. [7] F. Chollet et al., “Keras: Deep learning library for neural networks,” *Software available from keras.io*, 2015.
8. [8] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.

Future Scope:

1. **Database Storage for Captured Images:**

The future scope includes implementing a database system to store images captured through the camera. This will help in maintaining a structured dataset for further analysis and improvement of the model. Continuous storage of images will enable tracking of disease patterns and pest occurrences over time. It can also support periodic retraining of the model using updated real-world data. This enhancement will improve the accuracy, reliability, and learning capability of the system. Additionally, it will allow better monitoring and record-keeping for agricultural analysis.

2. **Extension to Multiple Crops:**

Another important future enhancement is extending the system to support multiple crops instead of limiting it to tomato plants. By incorporating datasets of other crops, the model can be trained to detect various diseases and pests across different plant types. This will make the system more versatile and applicable to a wider range of agricultural scenarios. It will also increase usability for farmers cultivating different crops. Overall, this extension will improve scalability and broaden the practical application of the system.