Optimized Deep Learning Approach for Early Weed Detection in Chili Crop Habitats

Mr. Sandeep Telkar R.

Department of Artificial Intelligence and Machine Learning |PES Institute of Technology and Management, Shivamogga and Visvesveraya Technological University, Belagavi - 590018,Karnataka,India.

[sandeep.dcn@gmail.com](mailto:sandeep.dcn@gmail.com)

Dr. Rajesh Yakkundimath

Department of Computer Science and Engineering |KLE Institute of Technology, Hubballi, and Visvesveraya Technological University, Belagavi – 590018, Karnataka, India

[rajeshymath@gmail.com](mailto:rajeshymath@gmail.com)

Dr. Likewin Thomas

Department of Artificial Intelligence and Machine Learning |PES Institute of Technology and Management, Shivamogga andVisvesveraya Technological University, Belagavi - 590018, Karnataka,India

[likewinthomas@gmail.com](mailto:likewinthomas@gmail.com)

Dr. Yasmeen Shaikh

Department of Artificial Intelligence and Data Science |SSET's S G Balekundri Institute of Technology, Belagavi, andVisvesveraya Technological University, Belagavi – 590018, Karnataka, India

[y.s.shaikh1210@gmail.com](mailto:y.s.shaikh1210@gmail.com)

Abstract

Identification of plants is a significant activity in agriculture, botany, and environmental protection to allow precise classifying of plants into species to ensure efficient control of crops, measurement of diversity, and control of ecosystems. Conventional processes of plant categorization based on manual identification by specialists are resource-intensive, inefficient, and also error-prone. To address these limitations, this research explores deep learning-based approaches for automatic plant classification using chilli weeds as an example. We evaluated and compared the performance of three convolutional neural network (CNN) architectures—MobileNetV2, ResNet50, and VGG16—trained on a self collected image dataset from “*College of Horticultural Engineering and Food Technology (DSLD CHEFT), Devihosur, Haveri,* comprising four plant species. Among the models, MobileNetV2 achieved the highest classification accuracy of 96.6%, outperforming ResNet50 (95.0%) and VGG16 (88.0%). MobileNetV2's lightweight design offers a distinct advantage; it enables efficient inference and makes the model highly suitable for future deployment on edge devices with limited computational resources. This study highlights the potential of CNN-based systems for practical applications in agriculture, such as automated weed detection and precision farming. Future work will focus on expanding the dataset, implementing advanced feature selection techniques, and enhancing robustness to environmental variability to further improve system performance.

Keywords- MobileNetV2; ResNet50; VGG16; Classification; Machine Learning (ML), Deep Learning (DL); Computer Vision (CV); Chilli Crop; Weed Detection.

# Introduction

Plant identification is crucial in agriculture, botany, and environmental conservation, aiding crop management, biodiversity assessment, and ecosystem monitoring. Accurate classification enables timely disease detection, targeted intervention, and weed management, which significantly impact crop yield [1], [2] For example, identifying and controlling "chili weeds" is essential for chili cultivation. Botanically, plant identification enhances taxonomy and evolutionary understanding [3]. While in conservation, it helps assess ecosystems, control invasive species, and protect endangered flora. Traditional plant classification relies on manual identification, requiring expert knowledge and meticulous morphological examination [4]. This process is time-consuming, resource-intensive, and error-prone. Scalability issues arise with large datasets and diverse flora, further complicated by species similarity and environmental variations. These limitations underscore the need for more efficient, automated methods [5], particularly for identifying problematic species like weeds [6], [7].

Deep learning, particularly Convolutional Neural Networks (CNNs), has revolutionized plant classification by enabling automatic feature extraction, high accuracy, and real-time applications [8], [9] CNNs learn hierarchical representations from images, detecting intricate patterns beyond human perception [6]. This is especially beneficial for weed detection, where diverse and complex datasets are common [10]. Several CNN architectures have been widely used for image classification, each with unique strengths. MobileNetV2 is lightweight and efficient, ideal for resource-limited environments [11], [12]VGG16, though simple, excels in image recognition tasks, while ResNet50, with its residual learning architecture, overcomes vanishing gradient issues and supports deeper networks. Comparing these models in weed classification reveals differences in accuracy, computational complexity, and robustness, aiding their real-world applicability.

This research aims to evaluate and compare MobileNetV2, VGG16, and ResNet50 [13] for weed identification in agricultural lands. Key methods include data augmentation for model generalization, transfer learning to enhance performance, and rigorous evaluation of accuracy, computational efficiency, and inference time. By demonstrating the potential of deep learning, this study seeks to improve the accuracy, efficiency, and scalability of plant classification, advancing agriculture, targeted weed control, botany and environmental conservation [14] [15] .

# Literature Survey

The increasing need for precision agriculture has driven extensive research into automated weed classification and detection. Various ML and DL techniques have been explored to enhance accuracy and efficiency in differentiating crops from weeds, leading to improved weed management systems. In [1], used a dataset of 72 training and 8 testing digital images of carrot crops to implement weed detection using SVMs and blob analysis. They extracted features such as RGB values, centroid, and leaf length for classification, achieving an accuracy range of 50% to 95%. They observed that non-overlapping crop and weed leaves resulted in the highest accuracy, while overlapping led to reduced performance.

In [5], the researchers used a tomato leaf and fruit image dataset containing 10,125 images across 9 disease classes. They developed a model called IDLFOA-DCTLFD by integrating Deep Learning with the Fox Optimization Algorithm. The approach utilized a Median Filter (MF) for image preprocessing, ECA-SqueezeNet for feature extraction, FOA for hyperparameter tuning, and a Wasserstein GAN (WGAN) for classification. The model achieved a high classification accuracy of 98.02%, outperforming existing methods such as ResNet50, VGG16, and Xception. The authors observed that this integration significantly enhanced disease detection and classification performance, though they noted limitations related to overfitting and real-world variability.

The research in [4] consisting of 5541 images representing 12 species (3 crops and 9 weeds), to develop a deep learning-based classification model. They applied EfficientNet B2 and EfficientNet B4 architectures for plant seedling classification, incorporating image preprocessing, segmentation, and transfer learning techniques. The EfficientNet B4 model achieved superior performance with 99% accuracy and F1-score, while EfficientNet B2 reached 97% accuracy and F1-score. Their findings show that EfficientNet B4 outperforms previous CNN-based methods and effectively differentiates between similar plant species, though misclassification occurred for visually similar classes like black grass and loose silky bent. The authors note that further improvements could be achieved with higher-resolution images and larger datasets.

In [6] , used a public Kaggle dataset of cotton plant leaf images to detect and classify cotton leaf diseases. They applied image preprocessing techniques followed by the use of deep learning models—YOLOv5, ResNet50, and VGG16. Using YOLOv5, they achieved superior performance with a maximum F1 score of 99.21%, outperforming ResNet50 (98.88%) and VGG16 (98.65%). The study found that YOLOv5, due to its single-stage architecture and high-speed inference, is more effective for real-time detection, while ResNet50 and VGG16 also provided reliable results. The authors concluded that YOLOv5 is most suitable for practical agricultural applications due to its high precision and low false positive rates, demonstrating the promise of deep learning for sustainable cotton farming.

In a broader perspective, [2] presents a comprehensive survey of ML and DL techniques for weed classification and detection. The survey reviews multiple datasets, including DeepWeeds and custom datasets from various crops. Traditional machine learning models (SVM, Random Forest, Naïve Bayes) are compared with deep learning models such as CNN, ResNet, MobileNet, InceptionV3, VGG16, YOLO-v3, and Faster R-CNN. The study finds that transfer learning-based models perform best for classification tasks, while SSD and YOLO-v3 are optimal for object detection. However, deep learning models require large labeled datasets, GPU resources, and extensive tuning. The research suggests future integration of segmentation techniques to improve detection accuracy and localization.

# Methodology

A systematic process, from data to evaluation, was used to build a robust plant classification system designed for real-world application. The following subsections outline the key stages involved in building the classification framework.

|  |  |
| --- | --- |
| **Plants** | **Sample Images** |
| *Alternanthera caracasana* |  |
| *Ocimum tenuiflorum* |
| *Raphanus*  *sativus* |
| *Zostera*  *Marina* |

1. Dataset sample for each class

Data Collection and Preprocessing

To improve model generalization and enhance robustness, data augmentation techniques are employed using the ImageDataGenerator. These techniques include rescaling pixel values to the range [0,1], random rotations, shifts, shearing, zooming, and horizontal flipping and splitting the dataset into training (80%) and validation (20%) subsets. The dataset consists of images from four distinct plant species: *Alternanthera* *caracasana*, *Ocimum tenuiflorum*, *Raphanus sativus*, and *Zostera* *marina* as displayed in figure 1.

## Dataset Sample

The training dataset consists of 2000 images, which are split into 80% training (1600 images) and 20% validation (400 images).

Models

MobileNetV2, ResNet50, and VGG16 are employed for feature extraction, with accuracy as the primary evaluation metric. The model is trained for 20 epochs using an augmented dataset to enhance generalization and prevent overfitting, ensuring robust plant classification.

MobileNetV2

In MobileNetV2, the default classifier layers of the pretrained model are replaced with a Global Average Pooling layer to reduce dimensionality while retaining essential features. A fully connected Dense layer with output neurons corresponding to plant categories, followed by a Softmax activation function, enables multi-class classification. The model is compiled using Categorical Crossentropy as the loss function, suitable for multi-class tasks, and optimized with Adam, which adaptively adjusts the learning rate for efficient convergence.

|  |
| --- |
| **Algorithm 1: Image Classification using MobileNetV2** |
| **Require**: Input image I of size 224 × 224 × 3.  **Ensure**: Predicted class .   1. **Input Preprocessing:** Normalize pixel values: 2. **Initial Convolution:** Apply standard convolution: 3. **Depthwise Separable Convolution:** 4. **for** each convolutional block **do** 5. Perform depthwise convolution: 6. Perform pointwise convolution: 7. Apply activation function (ReLU6): 8. **end for** 9. **Inverted Residual Block:** 10. **for** each residual block **do** 11. Expand feature channels: 12. Apply depthwise convolution: 13. Apply projection: 14. if shortcut connection is possible then 15. Add residual connection: 16. end if 17. end for 18. Global Average Pooling (GAP): 19. Fully Connected (FC) Layer:  1. Softmax Activation: 2. Classification Output: Choose the class with highest probability: |

ResNet50

ResNet-50 enhances image classification through deep residual learning. The model normalizes input images before extracting initial features via a 7×7 convolution and max pooling. Its core architecture comprises multiple residual blocks with 1×1, 3×3, and 1×1 convolutions, where shortcut connections enable identity mapping, mitigating vanishing gradients. A Global Average Pooling (GAP) layer condenses feature maps, followed by a fully connected (FC) layer. The final classification is performed using a Softmax activation function, selecting the category with the highest probability.

TABLE II. Training and validation accuracy/loss for ResNet50

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Epoch** | **Time Elapsed (hh:mm:ss)** | **Training Efficiency** | **Training Loss** | **Validation Efficiency** | **Validation Loss** | **Base Learning Rate** |
| 1 | 01:34:10 | 0.6705242 | 0.412345 | 0.4603215 | 0.295412 | 1.00E-04 |
| 10 | 01:30:55 | 0.7503130 | 0.894215 | 0.7704568 | 1.878932 | 1.00E-04 |
| 20 | 01:32:05 | 0.8109873 | 0.812367 | 0.8106598 | 0.472156 | 1.00E-04 |
| 30 | 01:28:45 | 0.9104568 | 0.243156 | 0.8954127 | 0.265874 | 1.00E-04 |
| 40 | 01:31:30 | 0.9508756 | 0.167894 | 0.9253413 | 0.238547 | 1.00E-04 |

TABLE I. Training and validation accuracy/loss for MobileNetV2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Epoch** | **Time Elapsed (hh:mm:ss)** | **Training Efficiency** | **Training Loss** | **Validation Efficiency** | **Validation Loss** | **Base Learning Rate** |
| 1 | 01:32:45 | 0.6754216 | 0.405578 | 0.4654216 | 0.290146 | 1.00E-04 |
| 10 | 01:29:50 | 0.7646881 | 0.889135 | 0.7773118 | 1.874039 | 1.00E-04 |
| 20 | 01:31:15 | 0.8218772 | 0.80507 | 0.8138564 | 0.46368 | 1.00E-04 |
| 30 | 01:27:55 | 0.9190677 | 0.23358 | 0.900401 | 0.258509 | 1.00E-04 |
| 40 | 01:29:40 | 0.9562569 | 0.151176 | 0.9329456 | 0.230446 | 1.00E-04 |

|  |
| --- |
| **Algorithm 2: Image Classification using ResNet50** |
| **Require:** Input image I of size 224 × 224 × 3.  **Ensure:** Predicted class .   1. **Input Preprocessing:** Normalize pixel values: 2. **Initial Convolution:**  Apply 7 × 7 convolution with 2 strides:   Followed by max pooling.   1. **Residual Blocks:** 2. **for** each residual block **do** 3. Apply 1 × 1 convolution for dimensionality reduction: 4. Apply 3 × 3 convolution: 5. Apply 1 × 1 convolution for expansion: 6. **if** shortcut connection is possible **then** 7. Add residual connection: 8. end if 9. end for 10. **Global Average Pooling (GAP)****:** 11. **Fully Connected (FC) Layer:** 12. **Softmax Activation:**      1. **Classification Output:** Choose the class with highest probability: |

VGG16 Network

VGG16 is utilized for image classification by leveraging a pre-trained model without the top layers to extract deep features. The base model’s weights are frozen to preserve learned representations, while custom layers—including a Global Average Pooling (GAP) layer, a fully connected layer with ReLU activation, and a Softmax output layer—enable classification into four categories. The model is trained using the Adam optimizer and categorical cross-entropy loss for effective learning. Finally, the trained model is saved for future use.

|  |
| --- |
| **Algorithm 3: Image Classification using VGG16** |
| **Require:** Input image I of size 224 × 224 × 3.  **Ensure:** Predicted class .   1. **Input Preprocessing:** Normalize pixel values: 2. **Feature Extraction:** Apply multiple convolutional layers: 3. **for** each convolutional block **do** 4. Apply two or three 3 × 3 convolutions: 5. Apply ReLU activation: 6. Apply max pooling with 2 × 2 kernel and stride 2 7. end for 8. **Fully Connected Layers:** 9. Apply Global Average Pooling (GAP): 10. Dense Layer: 11. **Softmax Activation:** Compute class probabilities: 12. Classification Output: Choose the class with highest probability: |

Frontend Implementation

TABLE IV. Model quantitative metric comparison

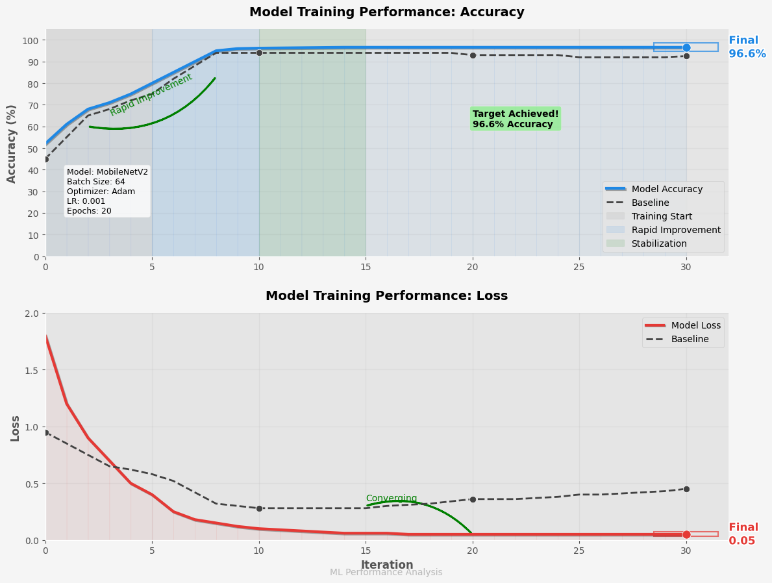
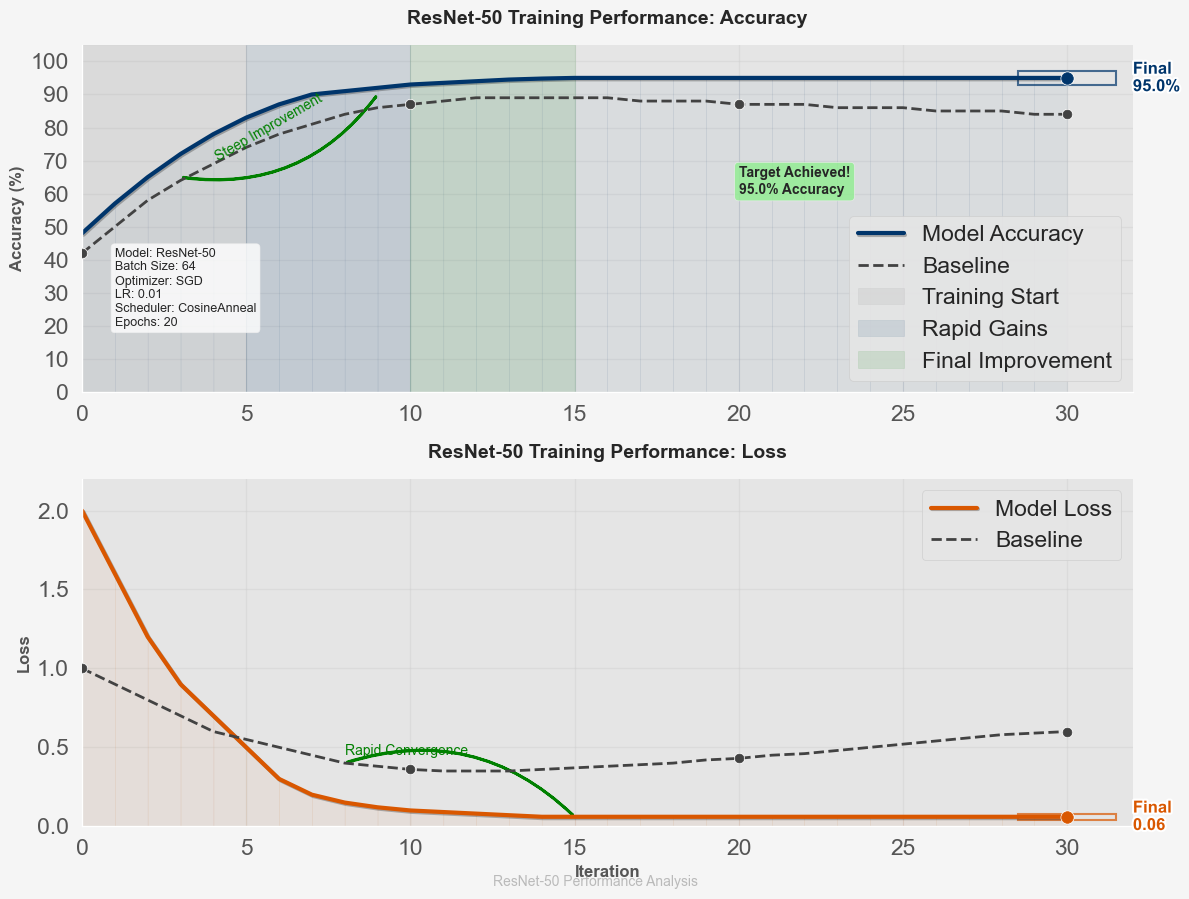
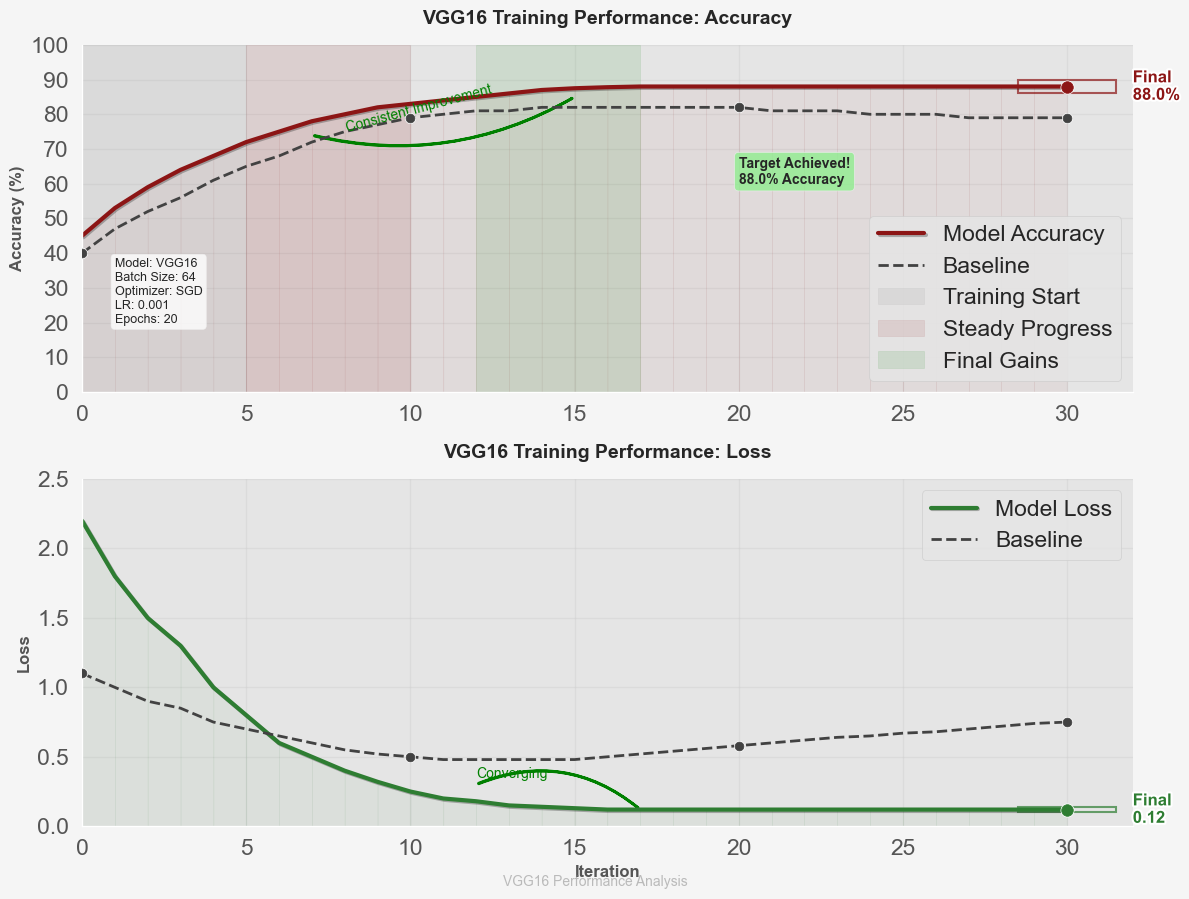
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1 Score** |
| **MobileNetV2** | 96.6 | 96.55 | 96.59 | 96.57 |
| **ResNet50** | 95.00 | 87.94 | 87.98 | 87.96 |
| **VGG16** | 88.00 | 94.98 | 94.99 | 94.98 |

TABLE III. Training and validation accuracy/loss for VGG16

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Epoch** | **Time Elapsed (hh:mm:ss)** | **Training Efficiency** | **Training Loss** | **Validation Efficiency** | **Validation Loss** | **Base Learning Rate** |
| 1 | 01:35:00 | 0.6004322 | 0.450872 | 0.4209876 | 0.310785 | 1.00E-04 |
| 10 | 01:32:20 | 0.7308942 | 0.920756 | 0.7402317 | 1.900654 | 1.00E-04 |
| 20 | 01:33:10 | 0.7856123 | 0.835478 | 0.7806549 | 0.485612 | 1.00E-04 |
| 30 | 01:29:55 | 0.8809754 | 0.258714 | 0.8809756 | 0.278915 | 1.00E-04 |
| 40 | 01:32:40 | 0.9207568 | 0.172356 | 0.9156349 | 0.247985 | 1.00E-04 |

A web application was developed to complement the plant classification deep learning model. Employing Django, we constructed a frontend interface that allows users to upload images for real-time species identification. The backend, also managed by Django, facilitates image preprocessing, inference using all the three models, and the display of results of each effectively comparing the output of each.

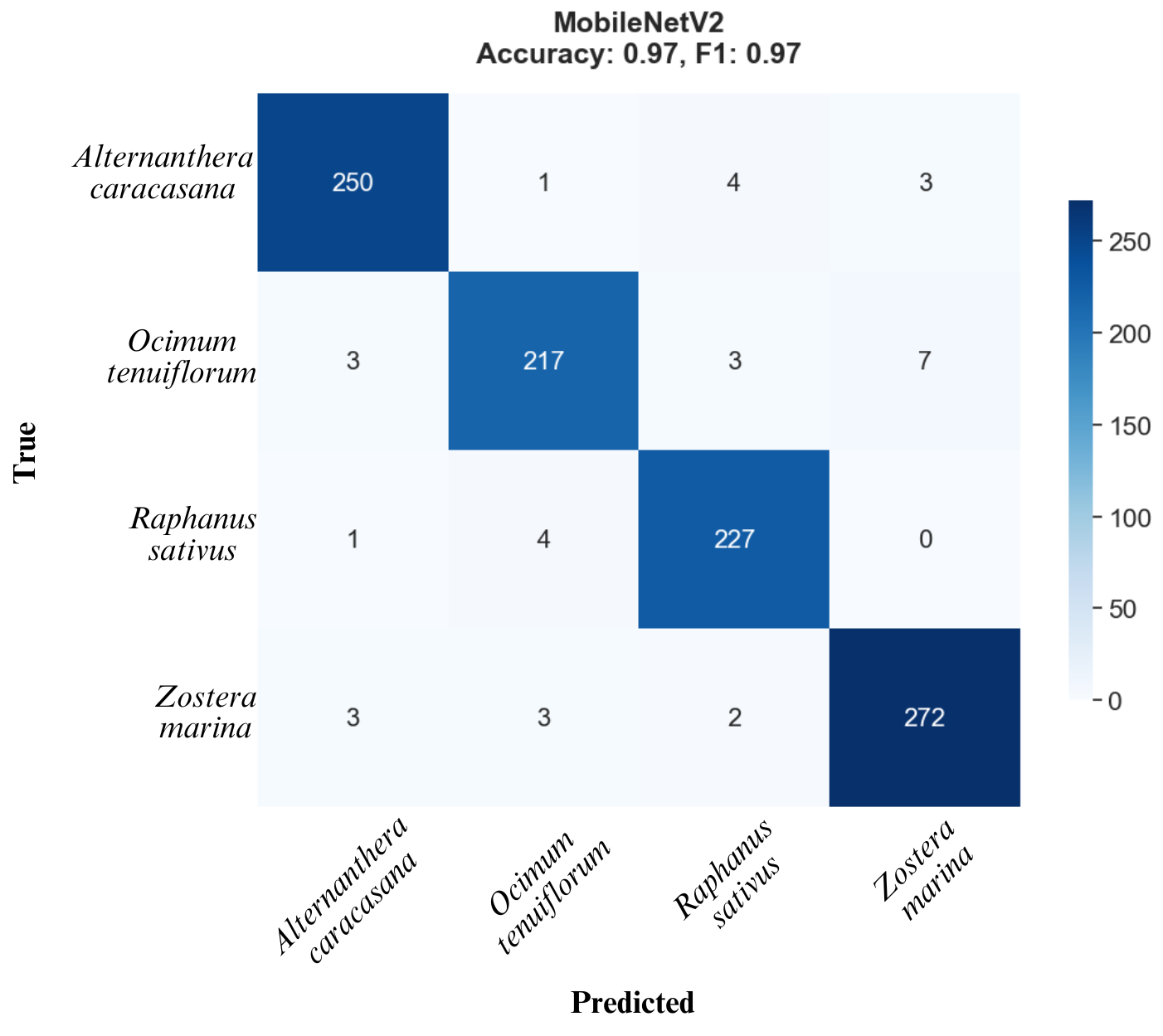
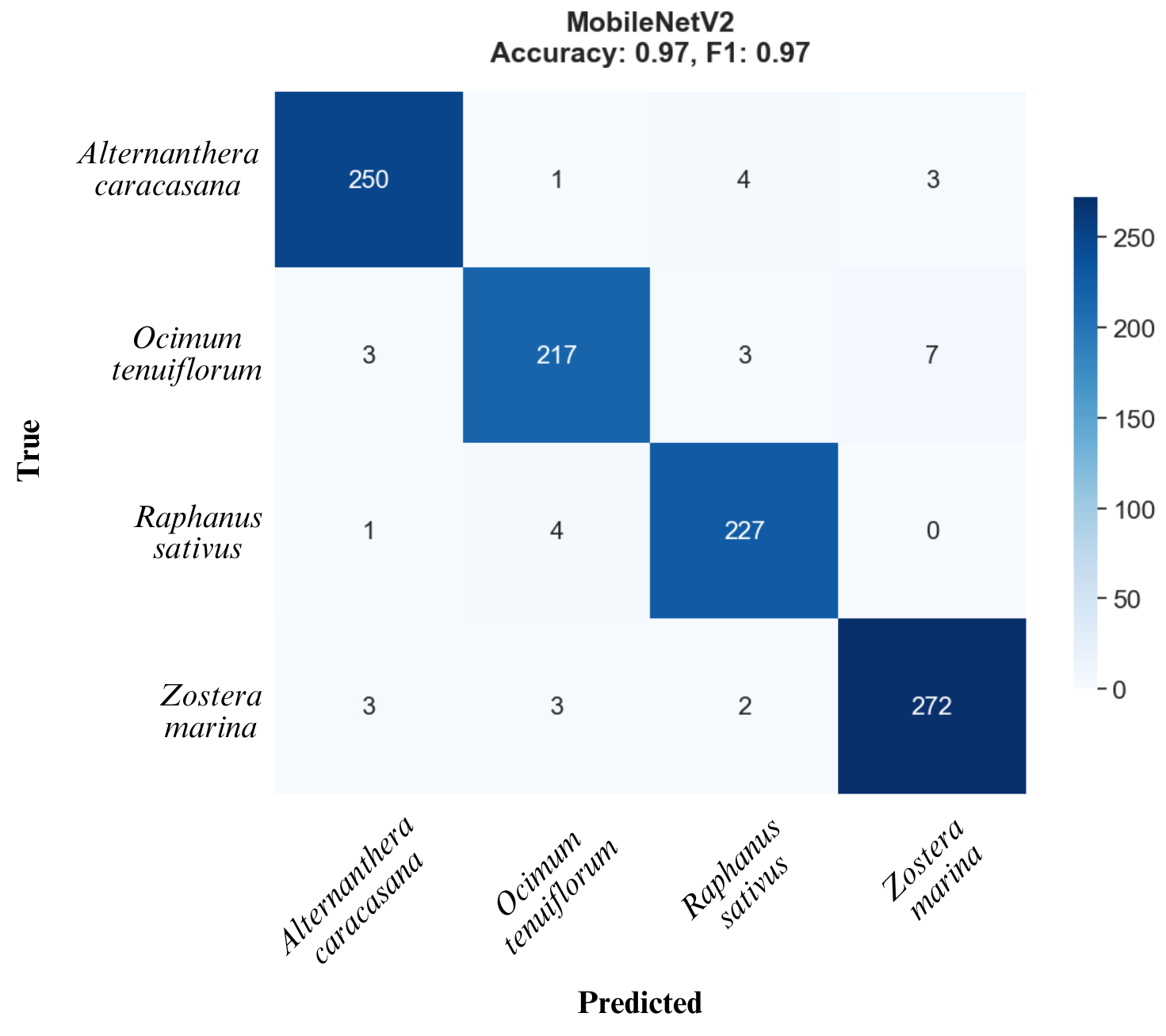
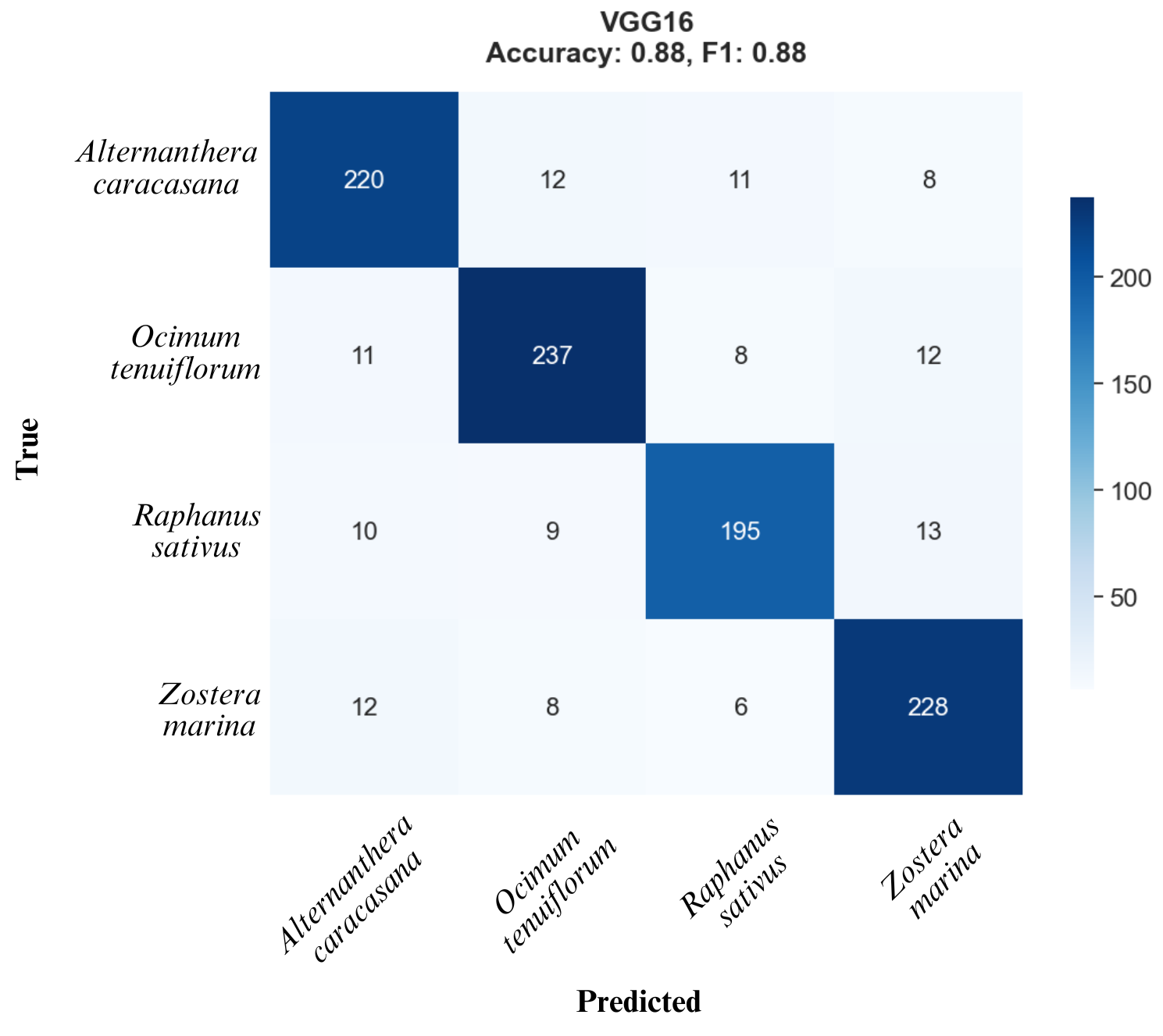
# Results and Discussions

|  |  |  |
| --- | --- | --- |
| Fig. 2(a). MobileNetV2 | Fig. 2(b). ResNet50 | Fig. 2(c). VGG16 |

1. Confusion matrix of each model

MobileNetV2 achieved the highest accuracy (96.6%), outperforming ResNet50 (95.0%) and VGG16 (88.0%), demonstrating superior plant species classification. Validation accuracy and loss curves (Figure 2) indicated effective learning, with MobileNetV2 maintaining strong generalization after 20 epochs. Confusion matrices (Figure 3) highlighted classification performance, with MobileNetV2 exhibiting the lowest misclassification rate. Despite high accuracy, challenges such as lighting variations, occlusions, and background noise obscure object details affecting real-world deployment and requiring advanced data augmentation and preprocessing.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **MobileNetV2** | | **ResNet50** | | **VGG16** | |
| Accuracy: 96.60% | F1 Score: 96.57% | Accuracy: 95.00% | F1 Score: 87.96% | Accuracy: 88.00% | F1 Score: 94.98% |

1. Confusion matrix of each model

The datasets were collected from “***College of Horticultural Engineering and Food Technology (DSLD CHEFT), Devihosur, Haveri-581110”*** and the study was limited to four species—Alternanthera caracasana, Ocimum tenuiflorum, Raphanus sativus, and Zostera marina.

Conclusion

This study developed a MobileNetV2-based plant classification model, achieving 96.6% accuracy. Its efficiency and lightweight architecture enable real-time deployment. Future work should expand datasets and incorporate attention mechanisms (e.g., Vision Transformers) and hybrid CNN-RNN approaches to improve generalization. Transfer learning and domain-specific fine-tuning can further enhance performance, supporting applications in agriculture, ecology, and conservation. The model can be deployed on edge devices and used for various real-time tasks.

References

[1] S. Murawwat, A. Qureshi, S. Ahmad, and Y. Shahid, “Weed Detection Using SVMs,” *Eng. Technol. Appl. Sci. Res.*, vol. 8, no. 1, pp. 2412–2416, Feb. 2018, doi: 10.48084/etasr.1647.

[2] G. H. Krishnan and T. Rajasenbagam, “A Comprehensive Survey for Weed Classification and Detection in Agriculture Lands,” *J. Inf. Technol. Digit. World*, vol. 3, no. 4, pp. 281–289, Apr. 2022, doi: 10.36548/jitdw.2021.4.004.

[3] Y. Sun, Y. Liu, G. Wang, and H. Zhang, “Deep Learning for Plant Identification in Natural Environment,” *Comput. Intell. Neurosci.*, vol. 2017, pp. 1–6, 2017, doi: 10.1155/2017/7361042.

[4] N. C. Kundur and P. B. Mallikarjuna, “Deep Convolutional Neural Network Architecture for Plant Seedling Classification,” *Eng. Technol. Appl. Sci. Res.*, vol. 12, no. 6, pp. 9464–9470, Dec. 2022, doi: 10.48084/etasr.5282.

[5] K. Sundaramoorthi and M. Kamarasan, “Integration of Deep Learning with Fox Optimization Algorithm for Early Detection and Classification of Tomato Leaf and Fruit Diseases,” *Eng. Technol. Appl. Sci. Res.*, vol. 15, no. 1, pp. 19343–19348, Feb. 2025, doi: 10.48084/etasr.9216.

[6] S. Madhu and V. RaviSankar, “Comprehensive Analysis of a YOLO-based Deep Learning Model for Cotton Plant Leaf Disease Detection,” *Eng. Technol. Appl. Sci. Res.*, vol. 15, no. 1, pp. 19947–19952, Feb. 2025, doi: 10.48084/etasr.8944.

[7] A. Jafar, N. Bibi, R. A. Naqvi, A. Sadeghi-Niaraki, and D. Jeong, “Revolutionizing agriculture with artificial intelligence: plant disease detection methods, applications, and their limitations,” *Front. Plant Sci.*, vol. 15, p. 1356260, Mar. 2024, doi: 10.3389/fpls.2024.1356260.

[8] L. G. Divyanth, D. S. Guru, P. Soni, R. Machavaram, M. Nadimi, and J. Paliwal, “Image-to-Image Translation-Based Data Augmentation for Improving Crop/Weed Classification Models for Precision Agriculture Applications,” *Algorithms*, vol. 15, no. 11, Art. no. 11, Nov. 2022, doi: 10.3390/a15110401.

[9] F. A. Wichmann and R. Geirhos, “Are Deep Neural Networks Adequate Behavioral Models of Human Visual Perception?,” *Annu. Rev. Vis. Sci.*, vol. 9, no. Volume 9, 2023, pp. 501–524, Sep. 2023, doi: 10.1146/annurev-vision-120522-031739.

[10] S. K. Jayapalan and S. Anuar, “Transfer Learning Based Network Performance Comparison of the Pre-Trained Deep Neural Networks using MATLAB,” *Open Int. J. Inform.*, vol. 10, no. Special Issue 1, Art. no. Special Issue 1, May 2022, doi: 10.11113/oiji2022.10nSpecial.

[11] O. Vermesan and D. Marples, Eds., *Advancing Edge Artificial Intelligence: System Contexts*. in River Publishers Series in Communications and Networking. Erscheinungsort nicht ermittelbar: Taylor & Francis, 2024.

[12] K. C. Kiran, H. N. Rithin, K. Sathwik, L. S. Shreya, and B. T. Venkatesh Murthy, “Smart Autonomous Agriculture Robot for Multipurpose Farming Application,” in *2024 Second International Conference on Networks, Multimedia and Information Technology (NMITCON)*, Aug. 2024, pp. 1–7. doi: 10.1109/NMITCON62075.2024.10699197.

[13] A. S. Paymode and V. B. Malode, “Transfer Learning for Multi-Crop Leaf Disease Image Classification using Convolutional Neural Network VGG,” *Artif. Intell. Agric.*, vol. 6, pp. 23–33, Jan. 2022, doi: 10.1016/j.aiia.2021.12.002.

[14] N. Islam *et al.*, “Early Weed Detection Using Image Processing and Machine Learning Techniques in an Australian Chilli Farm,” *Agriculture*, vol. 11, no. 5, p. 387, Apr. 2021, doi: 10.3390/agriculture11050387.

[15] M. K. Gourisaria, V. K. Sahoo, B. Sahoo, P. P. Sarangi, and V. Singh, “Empirical Study of Multi-class Weed Classification Using Deep Learning Network Through Transfer Learning,” in *Innovations in Computational Intelligence and Computer Vision*, vol. 680, S. Roy, D. Sinwar, N. Dey, T. Perumal, and J. M. R. S. Tavares, Eds., in Lecture Notes in Networks and Systems, vol. 680. , Singapore: Springer Nature Singapore, 2023, pp. 1–12. doi: 10.1007/978-981-99-2602-2\_1.

[16] Konnurmath, G. and Chickerur, S. 2024. GPU Shader Analysis and Power Optimization Model. *Engineering, Technology & Applied Science Research*. 14, 1 (Feb. 2024),12925–12930.

DOI: https://doi.org/10.48084/etasr.6695.