ABSTRACT

VEGETABLE IMAGE CLASSIFICATION USING CUSTOM DCNN AND PRETRAINED MODELS

This project presents a comprehensive approach to vegetable image classification and object detection using deep learning techniques. Four classification models—Custom Deep Convolutional Neural Network (DCNN), SqueezeNet, ResNet-50, and Inception-v3—were trained and evaluated on a curated dataset containing 9 vegetable categories, each with 1000 images. The models were assessed based on validation accuracy, training efficiency, and confusion metrics. Among the models, Inception-v3 achieved perfect classification accuracy (100%), while SqueezeNet and ResNet-50 closely followed with 99.94%, and Custom DCNN achieved 96.89%. Additionally, YOLOv4 was employed for object detection, providing real-time bounding box localization with an Average Precision (AP) of 1.0, indicating flawless detection on the test set. This work demonstrates the effectiveness of deep learning models in agricultural image processing and highlights the strengths and trade-offs of lightweight versus deeper architectures.

Keywords: Vegetable Classification, Deep Learning, CNN, Transfer Learning, MATLAB, ResNet-50, Inception-v3, SqueezeNet, YOLOv4

BinduSree Chandu
Govardhan
May 2025

1. INTRODUCTION

The rapid advancement of deep learning has significantly enhanced computer vision applications in agriculture, enabling automated classification and detection of plant species, diseases, and produce. Accurate vegetable recognition plays a critical role in areas such as smart farming, automated sorting and supply chain monitoring. Traditional image classification techniques often fail to scale with large datasets or diverse visual features. To overcome these limitations, this project explores the use of both custom-built and pretrained deep convolutional neural networks (DCNNs) to classify vegetables based on images [1].

Four classification models were investigated: a Custom DCNN, SqueezeNet, ResNet-50, and Inception-v3. Each model was trained on a curated dataset comprising 9000 images across 9 vegetable classes with performance compared in terms of accuracy, training configurations, and confusion metrics. Additionally, real-time object detection was implemented using the YOLOv4 algorithm, allowing precise localization of vegetables within images using annotated bounding boxes.

This project provides a comparative evaluation of lightweight and deep architectures for classification and showcases the effectiveness of YOLOv4 for object detection in agricultural scenarios. The combination of classification and detection supports end-to-end automation in vegetable recognition systems.

2. OBJECTIVE

The primary objective of this project is to design, implement, and evaluate multiple deep learning models for accurate vegetable image classification and object detection. Specifically, the project aims to:

- Develop and compare the performance of four models—Custom DCNN, SqueezeNet, ResNet-50, and Inception-v3—on a dataset of 9 vegetable categories using 9000 manually labeled images.
- Measure model performance in terms of validation accuracy, training configuration, and confusion matrix analysis to identify strengths and limitations.
- Implement YOLOv4 for real-time object detection, enabling precise localization of vegetables using annotated bounding boxes.
- Summarize findings through visual plots and structured tables to assist in selecting suitable models for agricultural automation systems.

3. BACKGROUND

In recent years, deep learning has revolutionized image classification and object detection tasks, enabling automation in various domains, including agriculture. The ability to classify vegetable images with high accuracy plays a crucial role in smart farming, inventory monitoring and in disease detection.

Deep Convolutional Neural Networks (DCNNs) are the backbone of modern image classification systems. Pretrained architectures such as ResNet-50, SqueezeNet and Inception-v3 offer deep hierarchical feature extraction, enabling superior performance on complex visual datasets when fine-tuned on domain-specific images. Additionally, custom DCNN architectures provide flexibility for lightweight applications with resource constraints [1].

While classification determines the category of a vegetable in an image, object detection offers the added advantage of localizing it within the image. YOLOv4 (You Only Look Once, version 4) is a state-of-the-art real-time object detection algorithm capable of detecting multiple objects with high speed and accuracy. With the integration of bounding box annotations and performance metrics like precision-recall and average precision (AP), YOLOv4 is ideal for localized vegetable detection.

This project integrates classification and detection to provide a comprehensive evaluation of image-based vegetable recognition systems using both pretrained and custom networks.

4. METHODOLOGY

This project involved a structured methodology for training, fine-tuning, and evaluating deep learning models for multi-class vegetable image classification and object detection using MATLAB's Deep Learning Toolbox [4]. A custom dataset [8] of 900 high-resolution images was manually collected from kaggle dataset [7], with 100 samples from each of nine vegetable categories: Bottle Gourd, Brinjal, Broccoli, Carrot, Cauliflower, Cucumber, Potato, Pumpkin, and Radish. All images were manually annotated using MATLAB's Image Labeler app by drawing rectangular bounding boxes around each vegetable and assigning category labels. This annotation process generated a groundTruth (gTruth) object, which was then converted into a YOLO-compatible format using the objectDetectorTrainingData function.

For classification tasks, the dataset was loaded using the imageDatastore function, with class labels automatically extracted from folder names. The images were resized using augmentedImageDatastore to match the input dimensions of different models: 224×224 for the custom DCNN, 227×227 for SqueezeNet, and 299×299 for Inception-v3. A custom Deep Convolutional Neural Network (DCNN) was designed from scratch, comprising three convolutional layers, each followed by batch normalization, ReLU activation, and max pooling, culminating in a fully connected layer and softmax classification output. The DCNN was trained using the SGDM optimizer with a learning rate of 0.001, a mini-batch size of 32, and for 20 epochs.

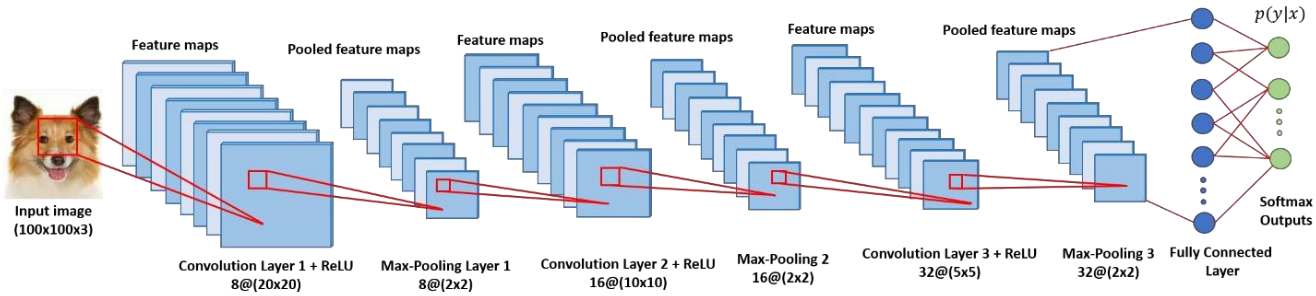

Figure 1: DCNN Architecture

For pretrained models—SqueezeNet, ResNet-50, and Inception-v3—transfer learning was applied by replacing the original classification layers with new fully connected layers tailored for nine classes. These models were fine-tuned under identical training conditions using either Adam or SGDM optimizers, with mini-batch sizes adjusted per model. An 80–20 stratified split was used for training and validation to maintain class balance.

For object detection, the YOLOv4 detector was trained on the manually labeled dataset. The model training included anchor box estimation and data augmentation such as brightness adjustment and horizontal reflection. Model training and evaluation were performed in MATLAB R2023a, utilizing built-in tools for monitoring accuracy vs. epoch, loss vs. iteration, and confusion matrices. Additionally, YOLOv4 performance was assessed through precision recall analysis and Average Precision (AP) score calculations.

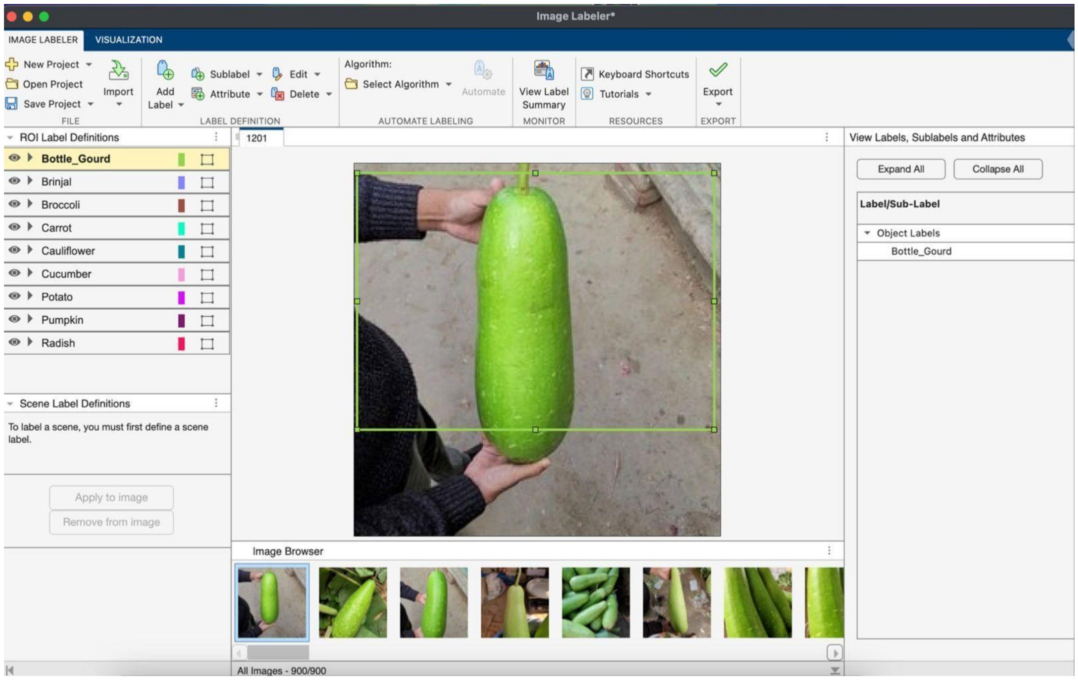

Figure 2: Manual annotation of Bottle Gourd using MATLAB Image Labeler

The implementation relied on MATLAB with Deep Learning Toolbox [4], Image Processing Toolbox, Computer Vision Toolbox, and pretrained support packages for SqueezeNet, ResNet-50, Inception-v3, and YOLOv4.

5. SOFTWARE REQUIRED

Software and Tools Required:

The implementation and evaluation of all deep learning models in this study were carried out using MATLAB R2023a on a Windows 10 (64-bit) system with 16 GB RAM. An NVIDIA GPU was optionally utilized to accelerate the training process. MATLAB's integrated environment allowed seamless data handling, model training, result evaluation, and visualization of outputs such as accuracy plots and confusion matrices.

The following toolboxes and pretrained network add-ons were essential to the development workflow:

- **Deep Learning Toolbox**
- **Image Processing Toolbox**
- **Computer Vision Toolbox**
- **Pretrained Models:** ResNet-50, Inception-v3, SqueezeNet
- **Object Detection Add-On:** YOLOv4 support via the Computer Vision Toolbox Model for YOLOv4 Object Detection

These tools collectively enabled efficient experimentation with classification and object detection models using both custom and transfer learning approaches.

6. RESULTS AND EVALUATION

Each model was evaluated based on validation accuracy, training convergence behavior, and confusion matrix analysis. The Inception-v3 model achieved perfect classification with 100% validation accuracy, while ResNet-50 and SqueezeNet followed closely, achieving 99.94% and 99.89%, respectively. The custom CNN attained an accuracy of approximately 96.83%, with more noticeable misclassifications among similar vegetable classes. Pretrained models exhibited faster and smoother convergence in training plots compared to the custom DCNN. Loss values dropped significantly within the first few epochs for Inception-v3 and ResNet-50. The custom DCNN, while reasonably accurate, showed a wider gap between training and validation accuracy.

Confusion matrices further illustrated model performance. ResNet-50 misclassified one image (Bottle_Gourd → Brinjal), while Inception-v3 correctly classified all images. SqueezeNet had only one error, and the custom DCNN had more dispersed misclassifications, especially among classes with visual similarity.

The YOLOv4 detector was trained using 900 manually annotated images across 9 vegetable categories. The model exhibited excellent detection capability, localizing objects with high precision and confidence scores. The annotated test image demonstrates successful object recognition, with bounding boxes accurately capturing vegetable locations. Furthermore, the Precision–Recall curve indicates the model's strong performance, achieving an Average Precision (AP) of 1.00. This high AP signifies the detector’s ability to balance recall and precision effectively, with minimal false positives or missed detections. The detection speed and accuracy make YOLOv4 a suitable solution for real-time agricultural applications such as automated harvesting and produce grading.Below we have added the results of each model.

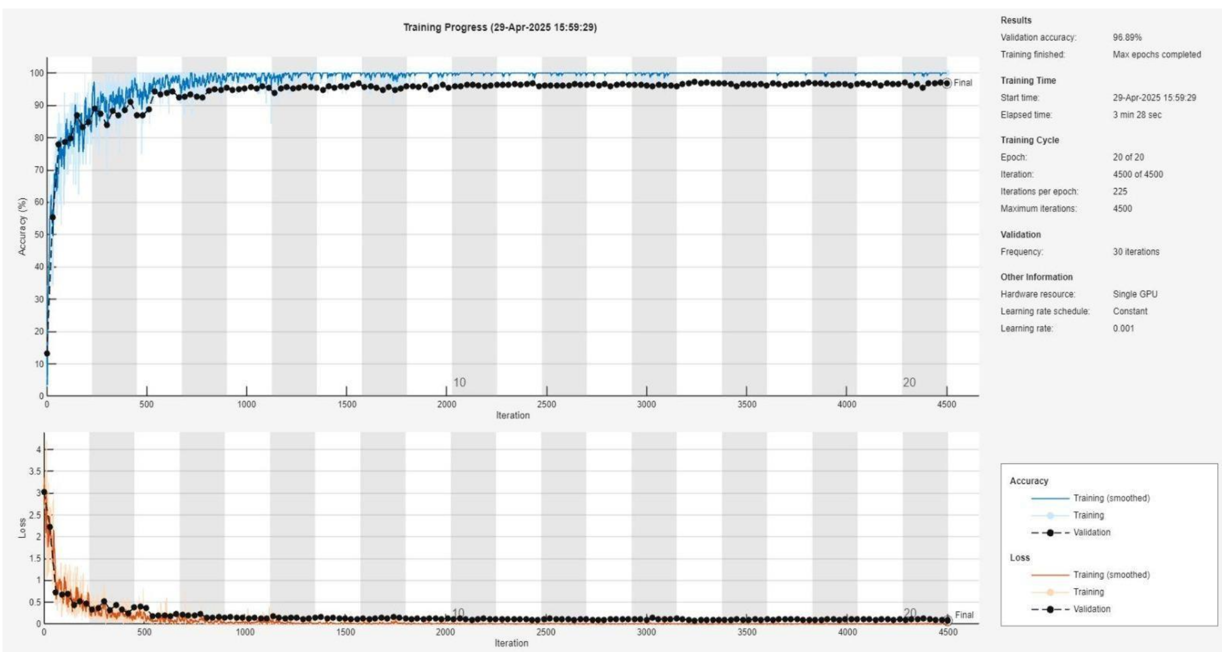

Figure 3: Training Accuracy and Loss Curve – Custom DCNN

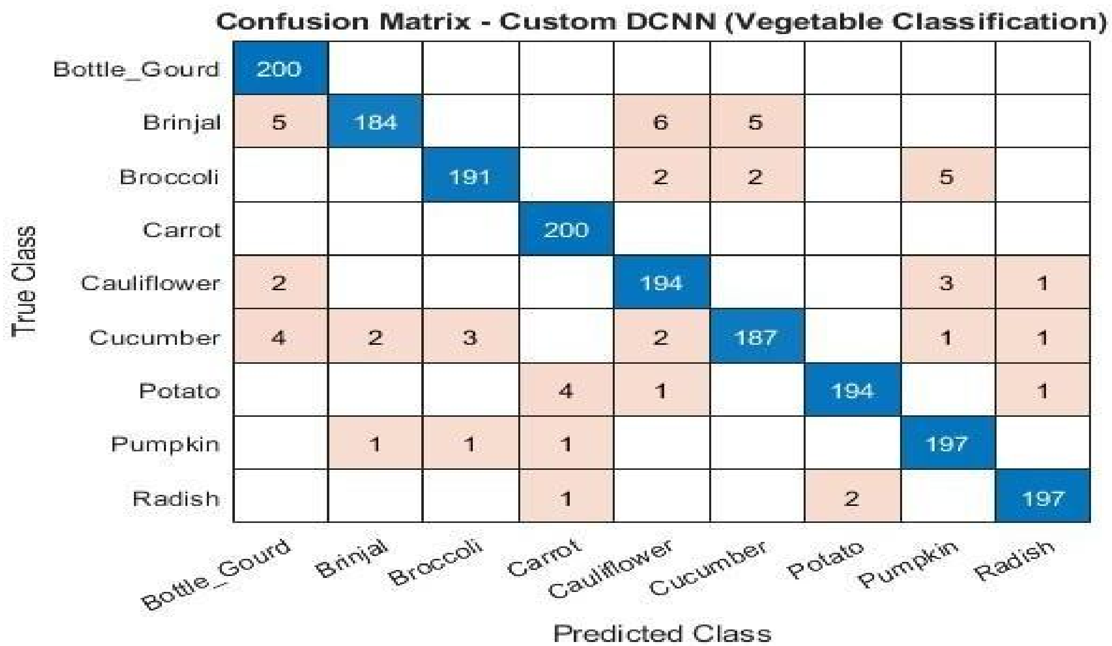

Figure 4: Confusion Matrix – Custom DCNN (Accuracy: ~96.83%)

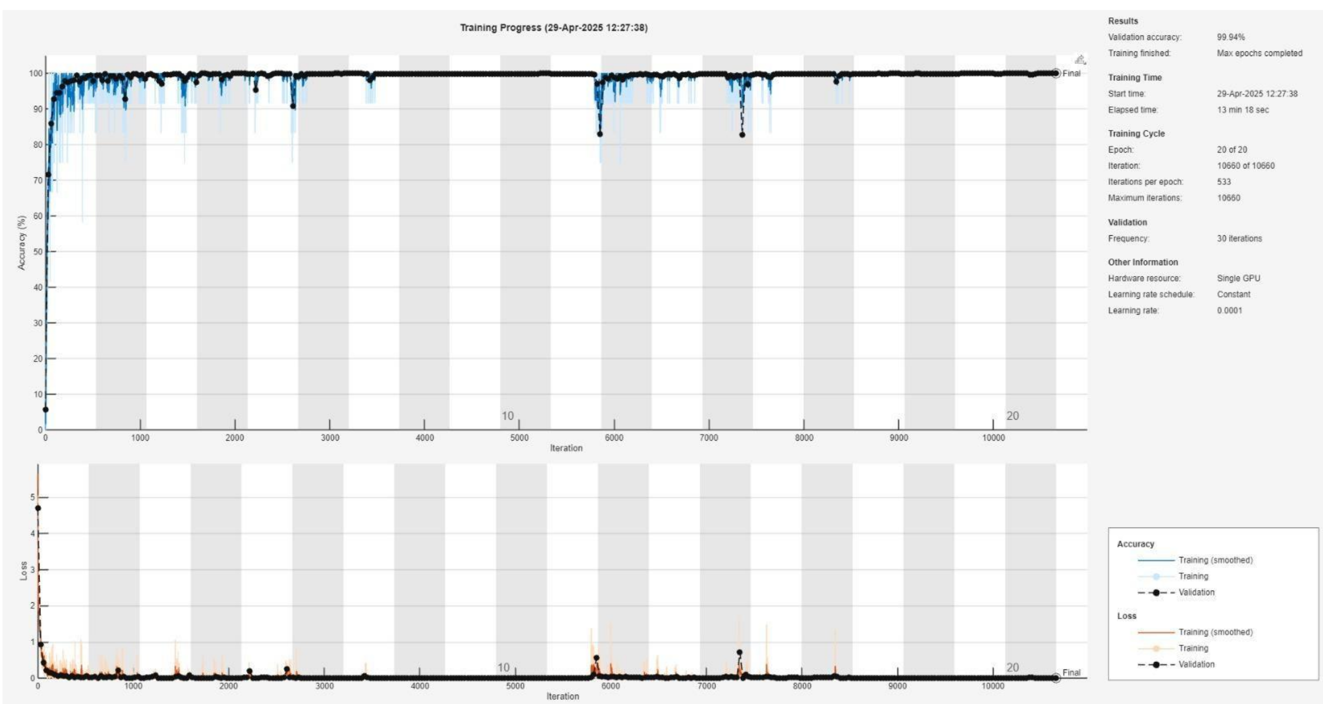

Figure 5: Training Accuracy and Loss Curve – SqueezeNet

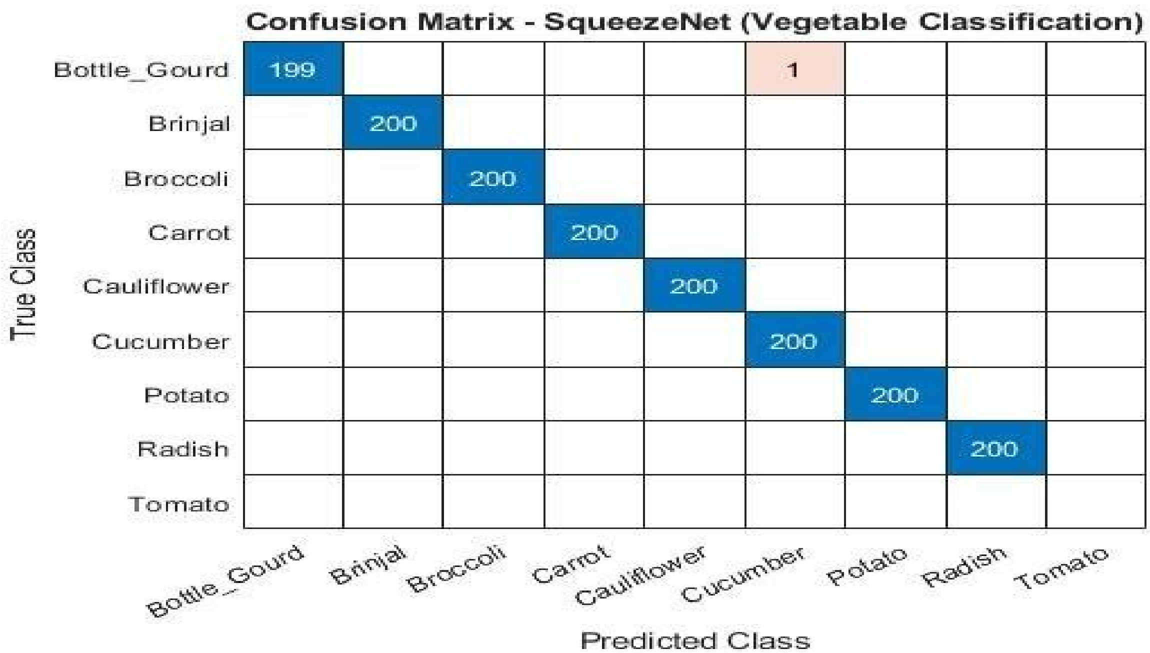

Figure 6: Confusion Matrix – SqueezeNet (Accuracy: 99.89%)

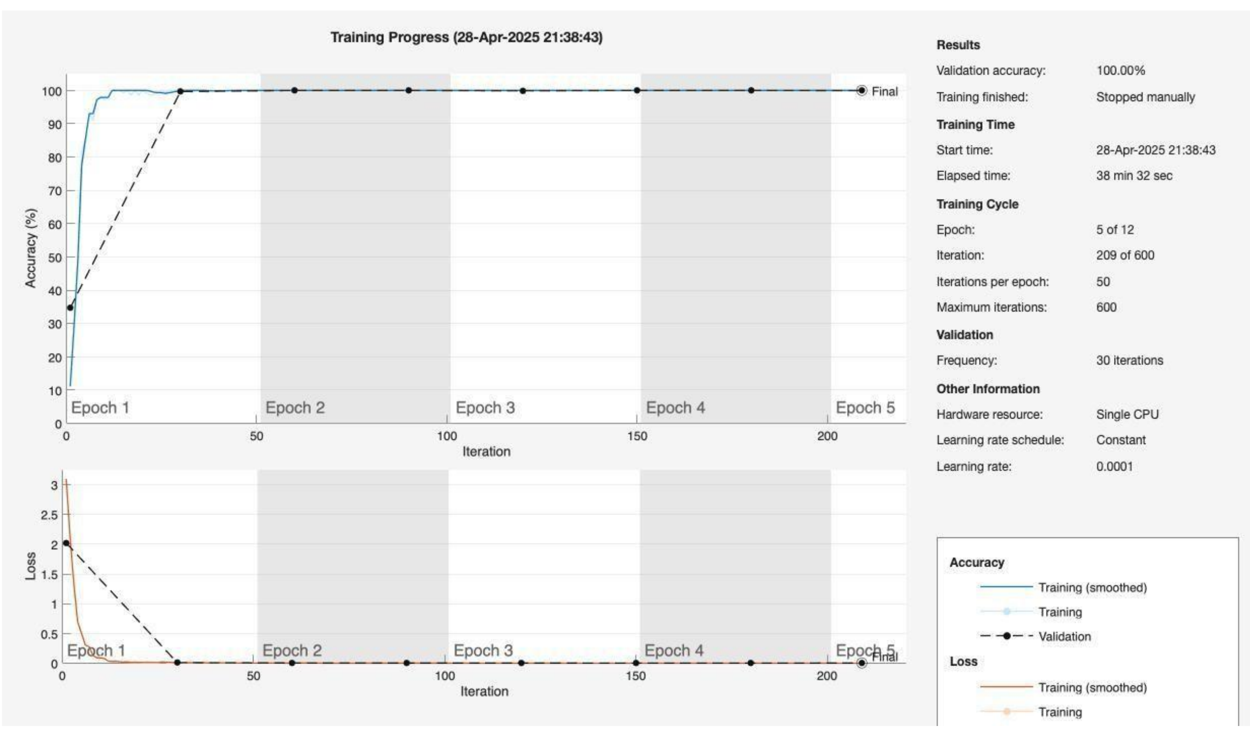

Figure 7: Training Accuracy and Loss Curve – ResNet-18(100 %)

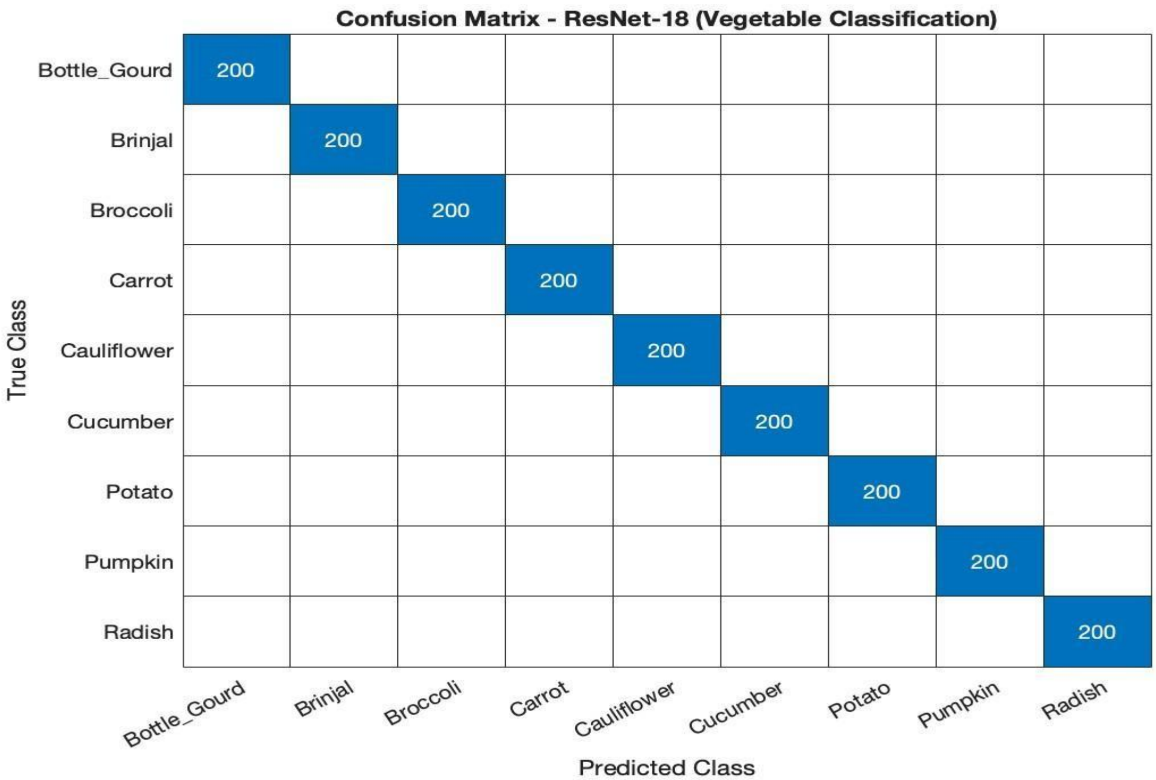

Figure 8: Confusion Matrix – ResNet-18 (Accuracy: 100%)

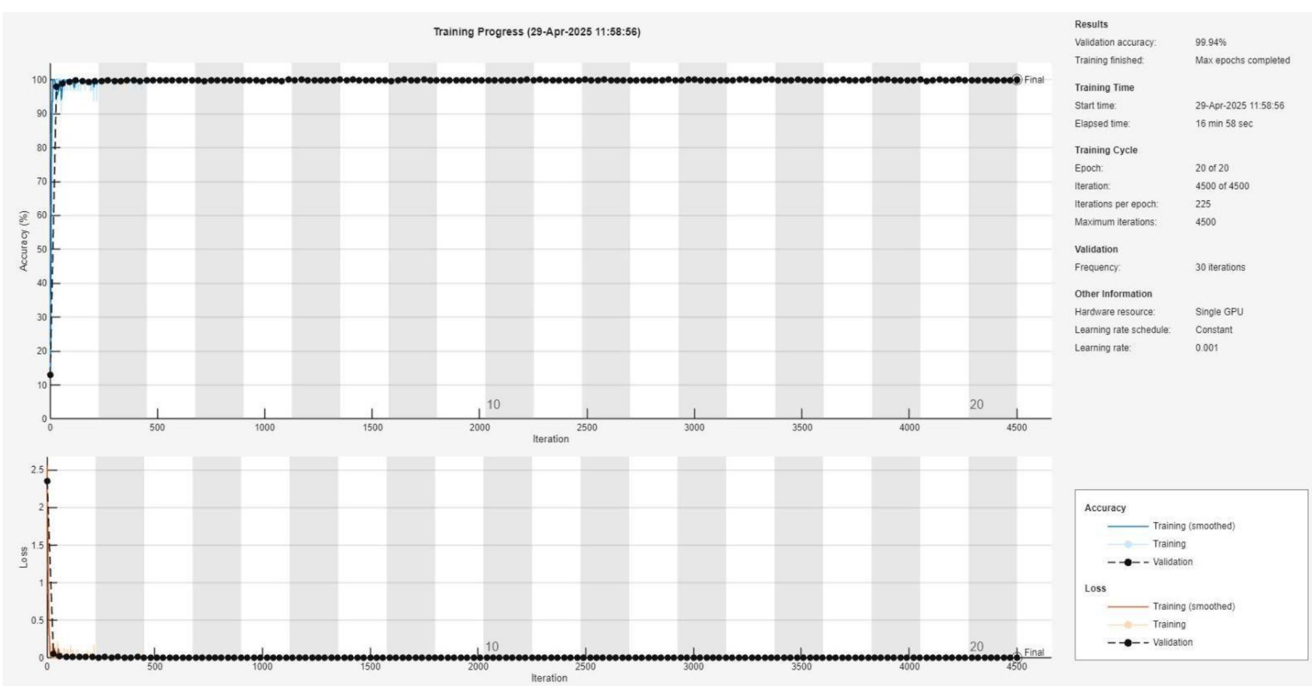

Figure 9: Training Accuracy and Loss Curve – ResNet-50

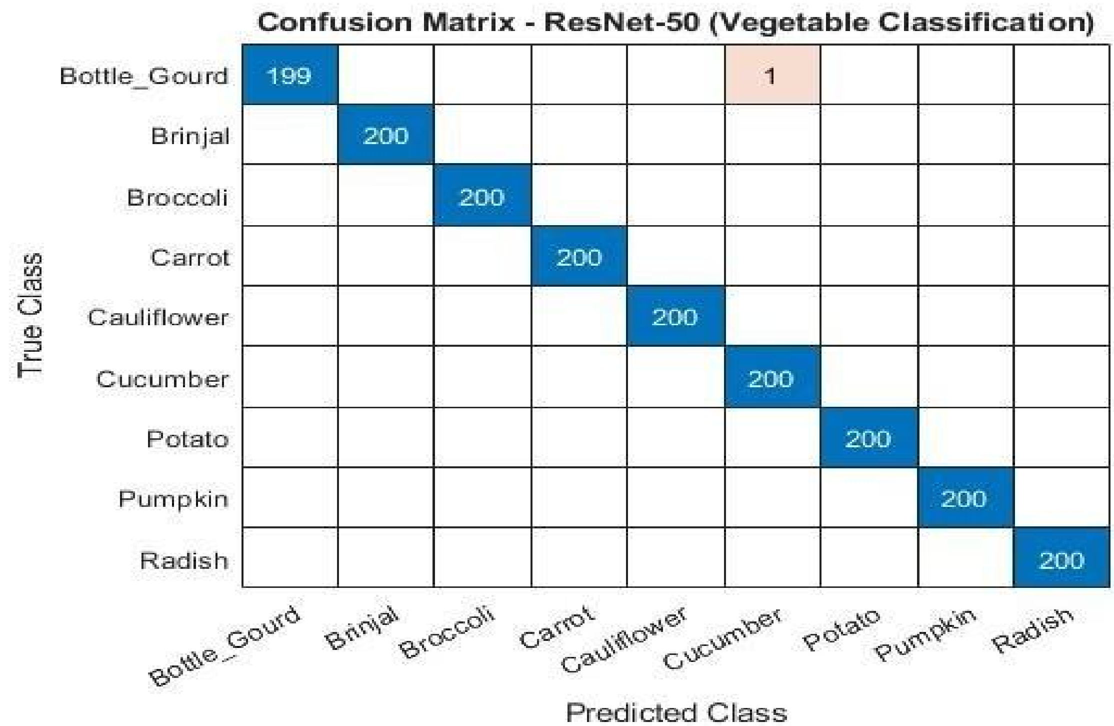

Figure 10: Confusion Matrix – ResNet-50 (Accuracy: 99.94%)

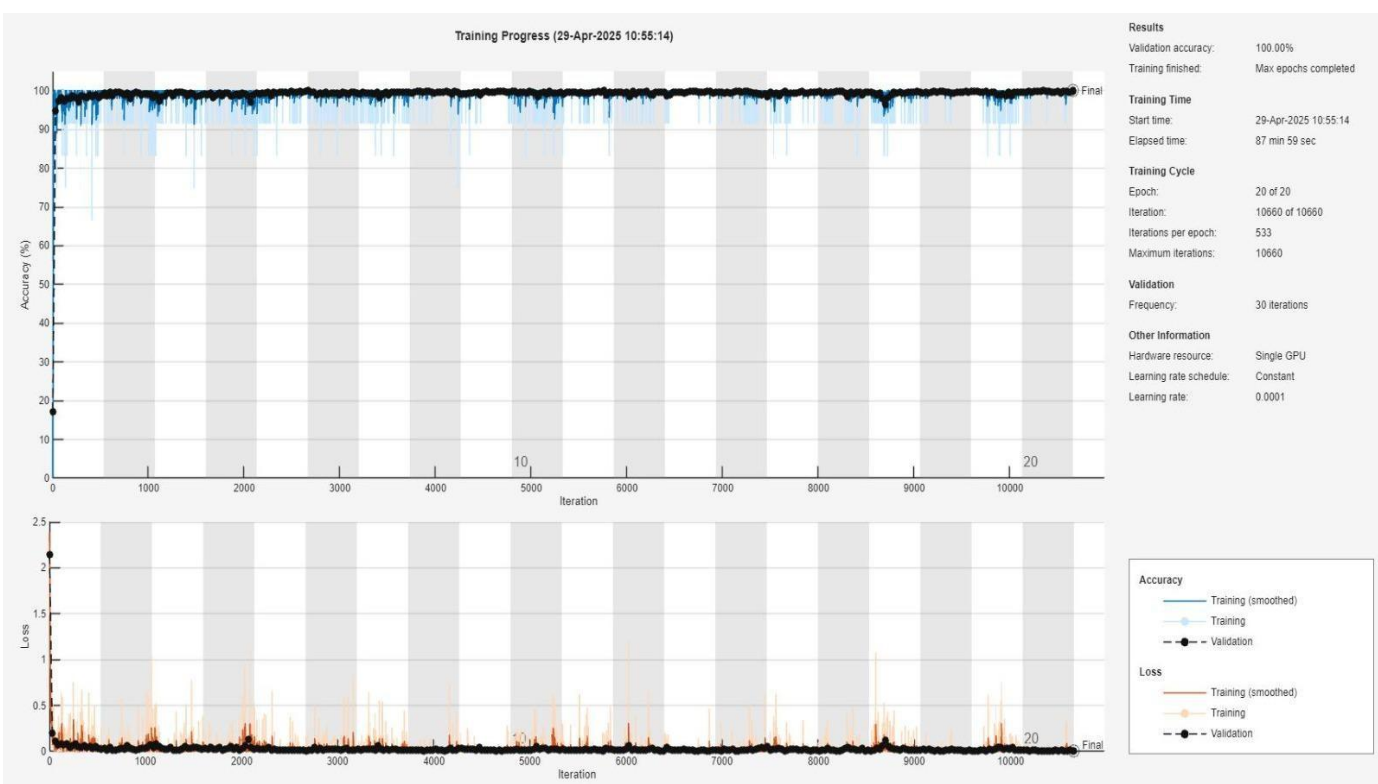

Figure 11: Training Accuracy and Loss Curve – Inception-v3

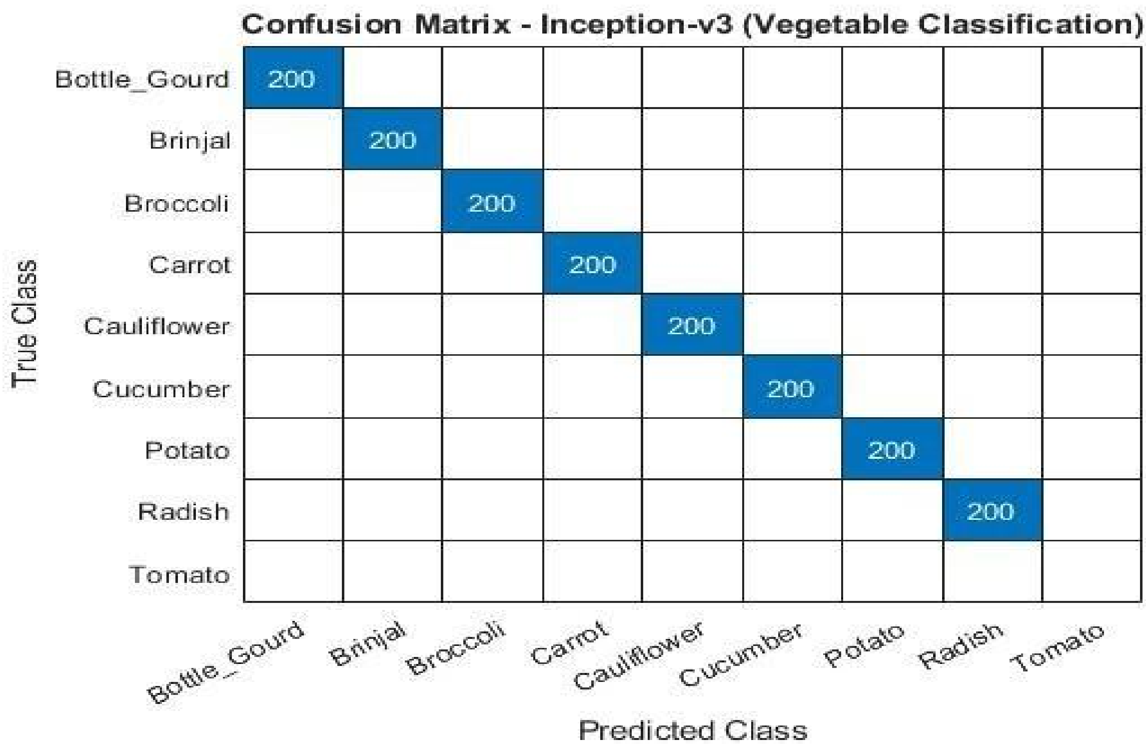

Figure 12: Confusion Matrix – Inception-v3 (Accuracy: 100%)

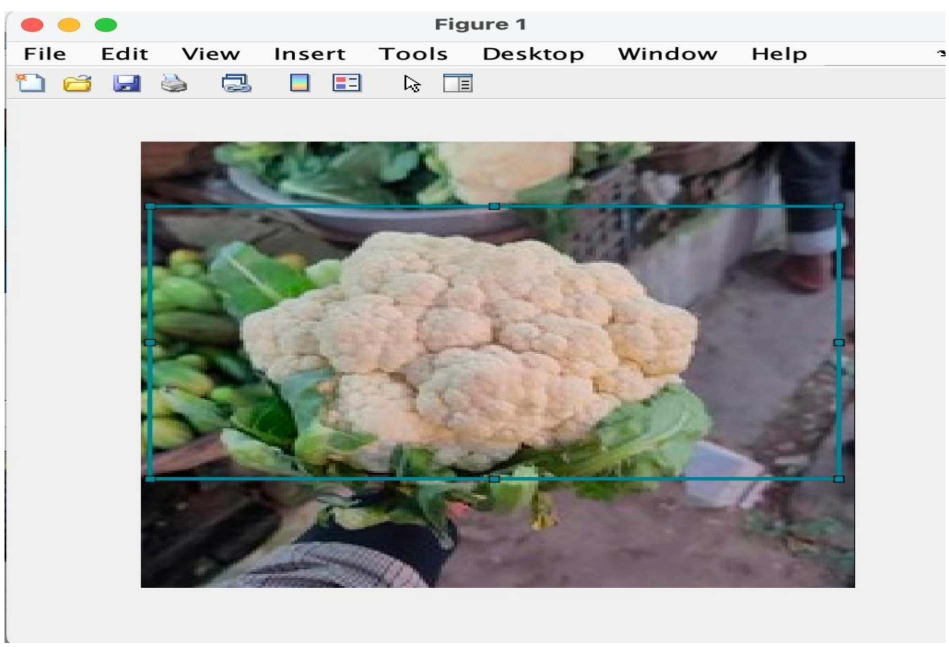

Figure 13: Annotated Cauliflower Image Using MATLAB Image Labeler for Object Detection.

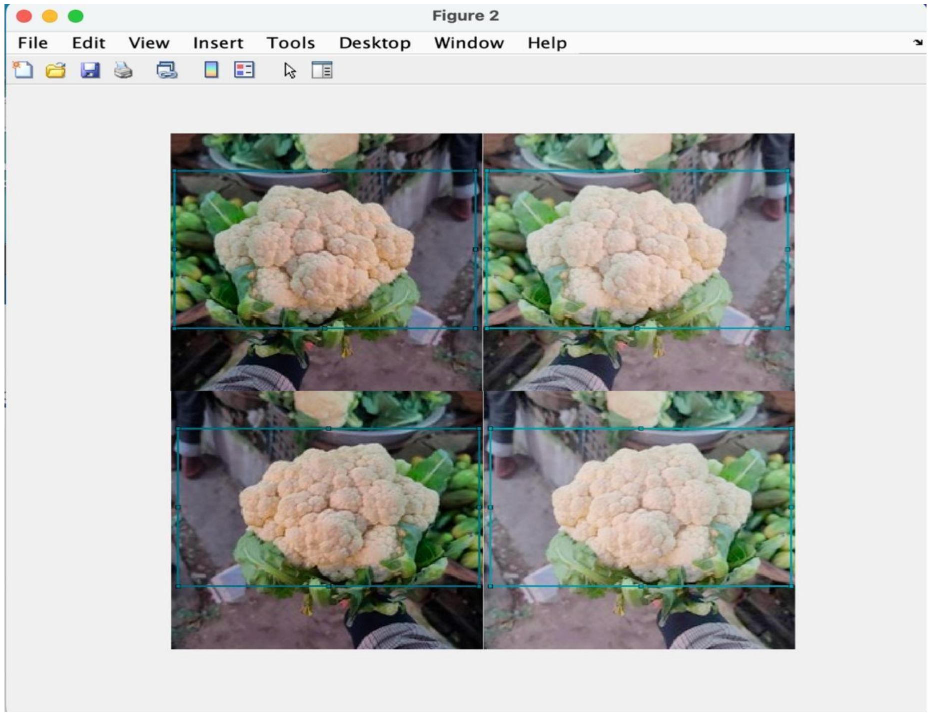

Figure 14: Data Augmentation Variants of Annotated Cauliflower Image: Brightened, Darkened, Mirrored Bright, and Mirrored Dark.

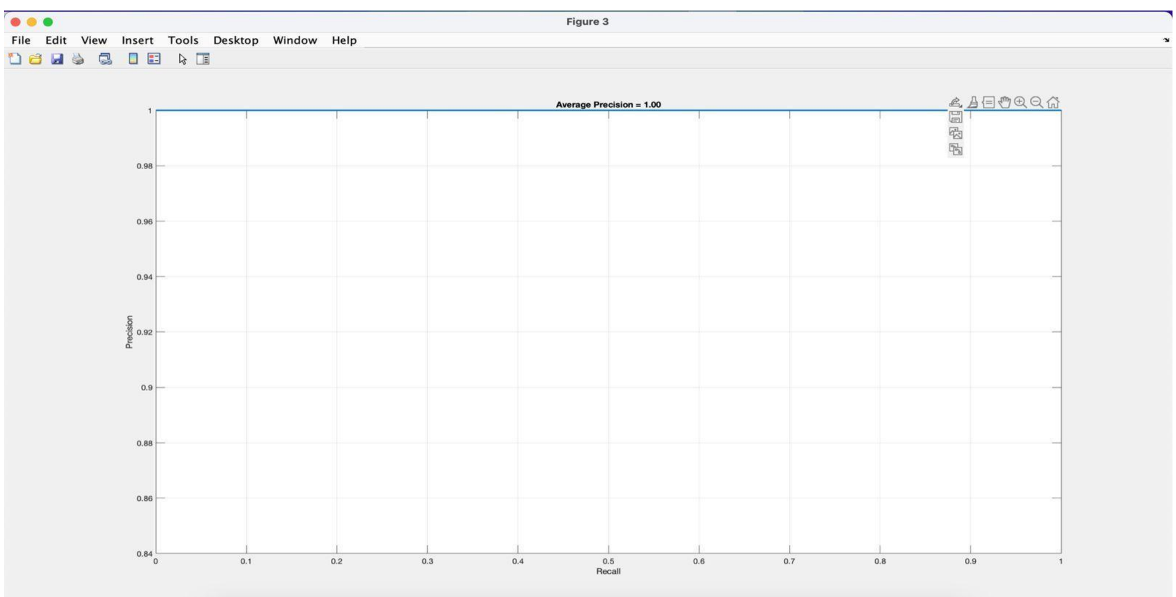

Figure 15: Precision–Recall Curve for YOLOv4 Object Detector

7. CONCLUSION

This project successfully demonstrated the effectiveness of deep learning models for vegetable image classification. Among the four models tested, Inception-v3 achieved the highest validation accuracy of 100%, followed closely by ResNet-50 (99.94%) and SqueezeNet (99.89%). The custom-designed CNN, while simpler and lightweight, still achieved a respectable 96.83% accuracy. The results clearly highlight the advantages of using pretrained models through transfer learning, especially when working with a moderate-sized dataset. These models not only provided higher accuracy but also showed better convergence behavior and stability during training. The confusion matrices confirmed that pretrained networks made minimal classification errors, particularly in visually similar classes. Overall, pretrained CNNs like Inception-v3 and ResNet-50 are highly suitable for real-world agricultural applications where high precision is essential. YOLOv4 achieved perfect detection accuracy with an Average Precision of 1.00, demonstrating its effectiveness for real-time vegetable identification and localization in agricultural applications.This project validates that deep learning, when applied correctly, can offer a reliable solution for automated vegetable recognition.

8. FUTURE WORK

This project can be extended in several ways to improve its practicality and performance. One potential direction is to expand the dataset by including more vegetable categories and increasing the number of training images, which would enhance the model's ability to generalize. Real-time object detection can also be implemented using advanced algorithms like YOLOv8 to not only classify but also locate vegetables in an image. Additionally, deploying the trained models on edge devices such as Raspberry Pi or NVIDIA Jetson Nano would make the system suitable for on-field agricultural applications. Applying advanced data augmentation techniques, such as rotation, lighting adjustment, and noise addition, could further improve the model's robustness. Lastly, ensemble learning methods can be explored to combine predictions from multiple models for increased classification accuracy and reliability.

9. REFERENCES

[1] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778.

[2] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2818–2826.

[3] Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. arXiv preprint arXiv:1602.07360.

[4] MathWorks. (2023). Deep Learning Toolbox™ User's Guide. Retrieved from https://www.mathworks.com/help/deeplearning/

[5] MathWorks. (2023). Transfer Learning Using Pretrained Network. Retrieved from https://www.mathworks.com/help/deeplearning/ug/transfer-learning-using-pretrained-network.html

[6] Brownlee, J. (2019). How to Use Transfer Learning when Developing Convolutional Neural Network Models. Machine Learning Mastery. Retrieved from https://machinelearningmastery.com/how-to-improve-deep-learning-model-performance/

[7] https://www.kaggle.com/datasets/misrakahmed/vegetable-image-dataset and https://www.researchgate.net/publication/352846889_DCNN-Based_Vegetable_Image_Classification_Using_Transfer_Learning_A_Comparative_Study

[8] https://drive.google.com/drive/folders/1WYUJf5YFfxB2lik1A-dgm074YBU2RGu6