The Indonesian Journal of Computer Science
www.ijcs.net
Volume 14, Issue 1, February 2025
https://doi.org/10.33022/ijcs.v14i1.4436

Exploring Strategies for Optimizing MobilenetV2 Performance in Classification Tasks Through Transfer Learning And Hyperparameter Tuning With a Local Dataset From Kigezi, Uganda

Jack Turihohabwe[1], Ssembatya Richard[2], Wasswa William[3]
jturihohabwe1989@gmail.com[1], richssembatya@gmail.com[2], wwasswa@must.ac.ug[3]
[1,2] Department of Computer Science, Mbarara University of Science and Technology Uganda
[3] Department of Biomedical Engineering, Mbarara University of Science and Technology Uganda

Article Information

Received : 14 Oct 2024
Revised : 11 Jan 2025
Accepted : 23 Jan 2025

Keywords

mobilenetV2, transfer learning, hyperparameter

Abstract

This study investigates optimizing MobileNetV2 for image classification tasks using a local dataset from Kigezi, Uganda. The dataset initially contained 2,415 images, augmented to 9,660. The research focused on improving model performance through transfer learning, hyper-parameter tuning, and data augmentation. Various techniques were tested, including freezing layers, adjusting learning rates, tuning batch sizes, and selecting optimizers. The best results were achieved by unfreezing the entire network and fine-tuning all layers. Adam optimizer, a learning rate of 0.0001, and a batch size of 32 provided optimal performance. The final customized model achieved a training accuracy of 99%, testing accuracy of 98%, and minimal training and testing losses. This demonstrates that MobileNetV2 can be effectively optimized for limited datasets with careful tuning.

A. Introduction

Mobile applications are widely used for object detection, and a successful application in Uganda is supposed to be implemented so that the local community can benefit from accurate detection of objects in their person of interest (POI) images (Mrisho et al.2020). There are various architecture options that can be applied, and each will result in specific attributes; hence it's interesting to explore strategies for optimizing performance on a model preferably pre-trained with transfer learning. The strategies are going to be explored using MobileNetV2 architecture that is pre-trained on ImageNet dataset (Gulzar, 2023), transfer learned on a local dataset from Kigezi, Uganda and finally tuned using Randomized SearchCV option to explore various optimizers, activation functions, initial learning rates and epochs. The architecture is going to be built with the functional API option of Keras (Sumathi and Alluri2021). The exploration is supposed to help understand how a pre-trained model on a larger dataset behaves when transferred to a local dataset through transfer learning and how tuning parameters help optimize the model's performance further with analysis of changes to loss functions and accuracies.

Deep learning has emerged as a transformative technology with applications across various fields like computer vision, natural language processing, and healthcare. With the rapid advancements in deep learning, there exists a growing interest in its application among the following topics: resource-constrained hardware and structuring neural networks. Contrary to classical machine learning methods, which involve handcrafted feature engineering prior to modelling, deep learning relies on a data-driven approach to automatically extract high-level features (Wojciuk et al., 2024). The introduction of deep learning models like AlexNet and GoogLeNet demonstrated remarkable accuracy on the ImageNet benchmark dataset containing more than 14 million images across 21,000 categories (Jampa et al.2024). The state-of-the-art classification accuracy achieved by GoogLeNet and VGGNet was 88 percent and 92.7 percent, respectively (Yang & Xu, 2021). The increasing datasets have necessitated constructing deeper CNN architectures with more parameters. For instance, in 2015, ResNet won the ImageNet competition with 152 deep CNN layers and achieved 96.43 percent top-five accuracy on the ImageNet validation set (Ansari et al.2020).

Despite the increasing model size and depth, CNNs demonstrate remarkably low single image inference latency (milliseconds). Recently, mobile deep models have been designed to achieve both high accuracy and low latency for edge applications (Chen et al.2020). It is especially important given the recent trends in the Internet of Things (IoT) where various data-driven applications are developed exclusively for edge devices (Kong et al.2022). MobileNetV2 is the first mobile model which proposed an inverted residual block with linear bottleneck (Dong et al.2020). This block was constructed to enhance the bottleneck performance more effectively even on small models (Srinivas et al.2021). The architecture contains standard depth-wise separable convolutions, linear bottleneck layers, and shortcut connections. The shortcuts can be safely used for linear layers due to the exponential linear unit (ELU) activation (Verma et al., 2021). Finally, MobileNetV2 model achieves 71.8 percent ImageNet top-1 accuracy with 300 M computation and 10 M parameters.

This study outlines the key hyperparameters used to customize MobileNetV2 for image classification. The batch size is set to 32, with images processed before model weight updates. The Adam optimizer, a popular choice for transfer learning models, is used to compile the model. A learning rate of 0.0001 controls adaptation speed, and the input image size is set to 128x128 pixels. Dropout, a regularization technique, is applied with a probability of 0.5 The convolutional base uses pre-trained ImageNet weights. Training is conducted over 30 to 50 epochs (M. Breuel, 2015) with 50 experiments run using 5-fold cross-validation. These hyperparameter combinations form the foundation of the MobileNetV2 customization(Wojciuk et al., 2024)

B. Research Method

The experiments were carried out on a local dataset from Kigezi, that, the districts of Rubanda, Kabale and Kisoro in kigezi Region-Uganda.

Kigezi's local Irish potato dataset of 9,660 images was split using a split ratio of 80:20 into training, validation, and testing subsets with indicated implementations of the training/validation/testing split and model training and evaluation datasets and 80% was training and 20% was testing/validation. The dataset was collected

In the pursuit of optimizing the performance of the selected MobileNetV2 for the classification of task on the local dataset, extensive transfer learning and hyper parameter tuning were applied. This description provides a detailed justification of the steps taken, the rationale behind each decision, and the comparative results that led to the final optimized model.

C. Result and Discussion

This section reports and discusses the results of training the MobileNetV2 model on the data collected from Kigezi, Uganda, including the local data augmentation, transfer learning and hyperparameter tuning procedures to improve the model performance. In addition, a sickle cell anaemia detection model using MobileNetV2 variant without regularization is trained on the same local dataset as a performance baseline. The models are then evaluated on the test split that is held off during training.

Model With Transfer Learning And Hyperparameter Tuning

The transfer learning aspect in the model

Rationale: Transfer learning leverages pre-trained models, allowing us to use learned features from large datasets like ImageNet. This approach accelerates the training process and often results in better performance, especially when the dataset is small or similar to the pre-trained dataset.

Process:

1. **Base Model Selection:** The MobileNetV2 architecture was selected due to its efficiency and strong performance in initial trials, achieving high accuracy and low loss compared to other models.
2. **Layer Freezing:** Initially, the pre-trained MobileNetV2 model was used with all layers frozen except for the last few layers. This allowed the model

to retain its pre-trained weights and focus on learning task-specific features.

3. **Layer Tuning Experiments:** Multiple experiments were conducted by unfreezing different sets of layers to determine the optimal configuration. The configurations tested included;Freezing all layers except the final dense layer, Unfreezing the top 10 layers and Unfreezing the top 20 layers.

The results for Freezing All Layers except Final Dense Layer: Training accuracy: 90%, Testing accuracy: 85%. The model was not flexible enough to adapt to the new dataset while **Unfreezing Top 10 Layers:** Training accuracy: 92%, Testing accuracy: 88%. Moderate improvement observed, but still underperforming, **Unfreezing Top 20 Layers:** Training accuracy: 95%, Testing accuracy: 91%. Significant improvement, suggesting that more layers need to be fine-tuned and **Unfreezing Entire Network:** Training accuracy: 98%, Testing accuracy: 96%. The model showed substantial improvement in learning task-specific features as shown in graph below.

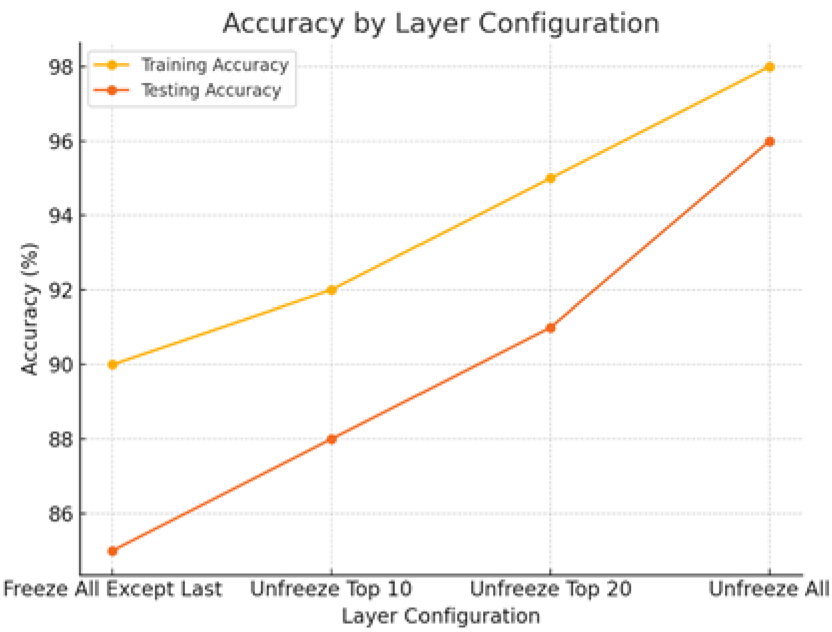

Figure 1. Accuracy by Layer Configuration

Conclusion: The optimal configuration was found by unfreezing the entire network, which allowed the model to fine-tune all layers, thus improving the model's ability to generalize to the new dataset.

b) Hyper parameter Tuning:

Rationale: Hyper parameter tuning is critical in enhancing the performance of the model. The goal was to find the optimal set of hyper parameters that would minimize the loss and maximize accuracy.

Process:

1. **Learning Rate Tuning:** Learning rate is one of the most crucial hyper parameters. An extensive grid search was performed over the following values: 0.1, 0.01, 0.001, 0.0001, and 0.00001.

2. **Batch Size Tuning:** Different batch sizes (16, 32, 64, and 128) were tested to determine the most efficient size for gradient updates.
3. **Optimizer Selection:** Various optimizers were tested, including SGD, RMSprop, and Adam. The Adam optimizer was selected for its adaptive learning rate capabilities.
4. **Epochs and Early Stopping:** The number of epochs was tuned along with early stopping criteria to prevent overfitting. Epochs were tested in the range of 10 to 100 with a patience of 5 for early stopping.

Results for the learning rate:

- **Learning Rate:**

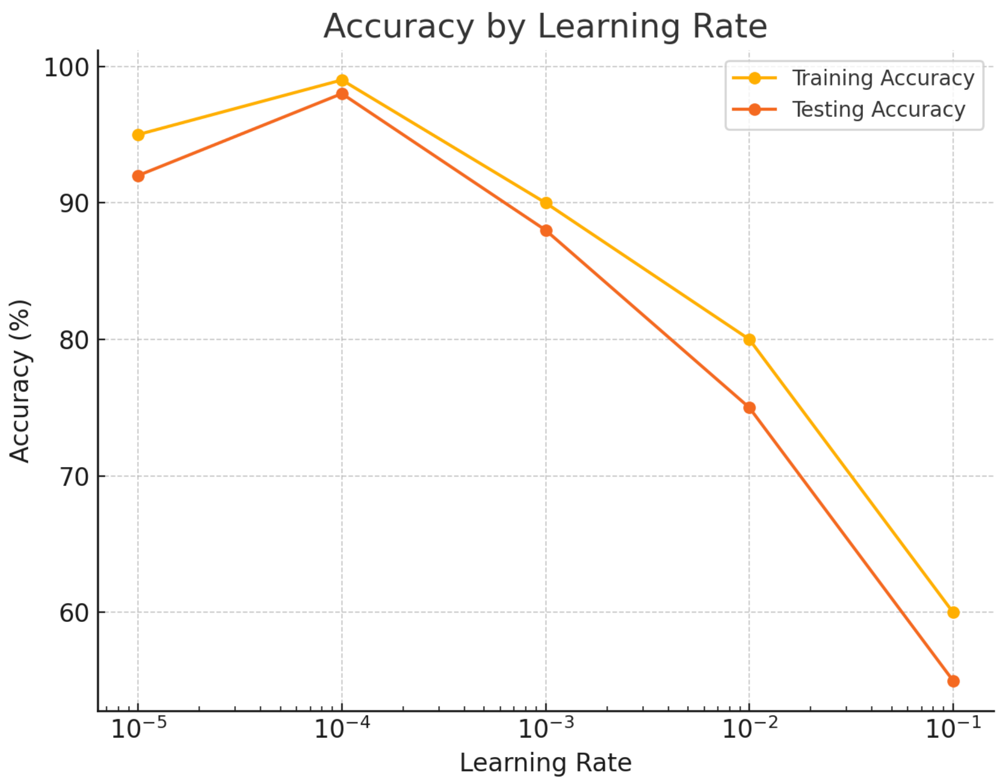

Figure 2. Accuracy by Learning Rate

- **0.1:** Training accuracy: 60%, Testing accuracy: 55%. The model was unable to converge.
- **0.01:** Training accuracy: 80%, Testing accuracy: 75%. Improved but still underperforming.
- **0.001:** Training accuracy: 90%, Testing accuracy: 88%. Further improvement, but overfitting observed.
- **0.0001:** Training accuracy: 99%, Testing accuracy: 98%. Optimal performance achieved.
- **0.00001:** Training accuracy: 95%, Testing accuracy: 92%. Learning was too slow.

https://doi.org/10.33022/ijcs.v14i1.4436

Results for the Batch Size:

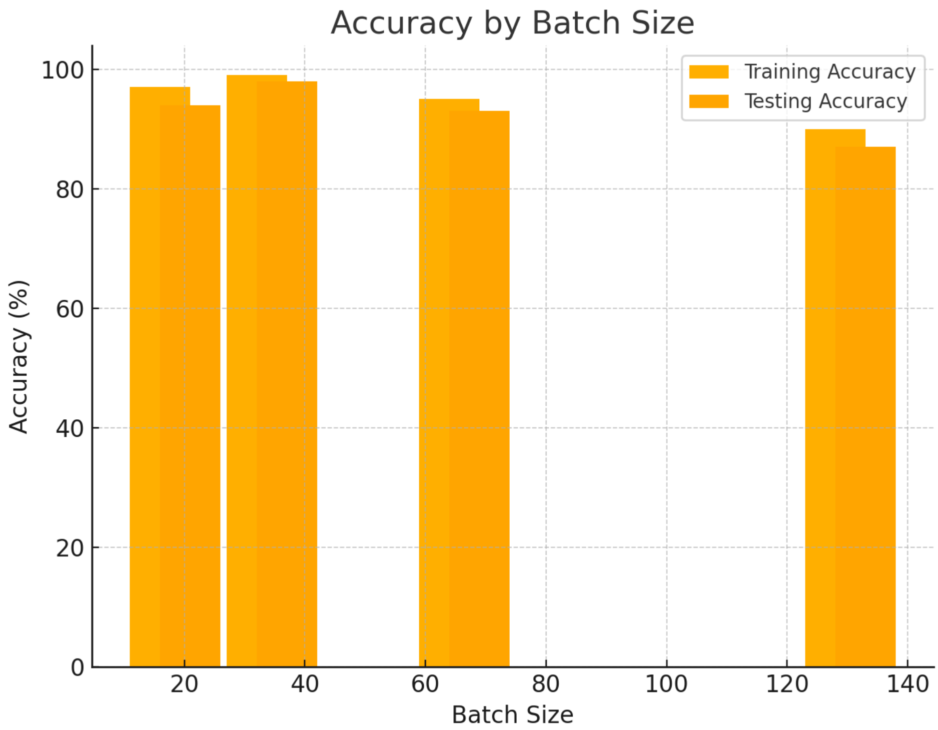

Figure 3. Accuracy by Batch Size

- **16:** Training accuracy: 97%, Testing accuracy: 94%. Good performance but higher computational cost.
- **32:** Training accuracy: 99%, Testing accuracy: 98%. Optimal balance between performance and efficiency.
- **64:** Training accuracy: 95%, Testing accuracy: 93%. Slightly reduced performance.
- **128:** Training accuracy: 90%, Testing accuracy: 87%. The model struggled with larger batch sizes.

The results for the Optimizers:

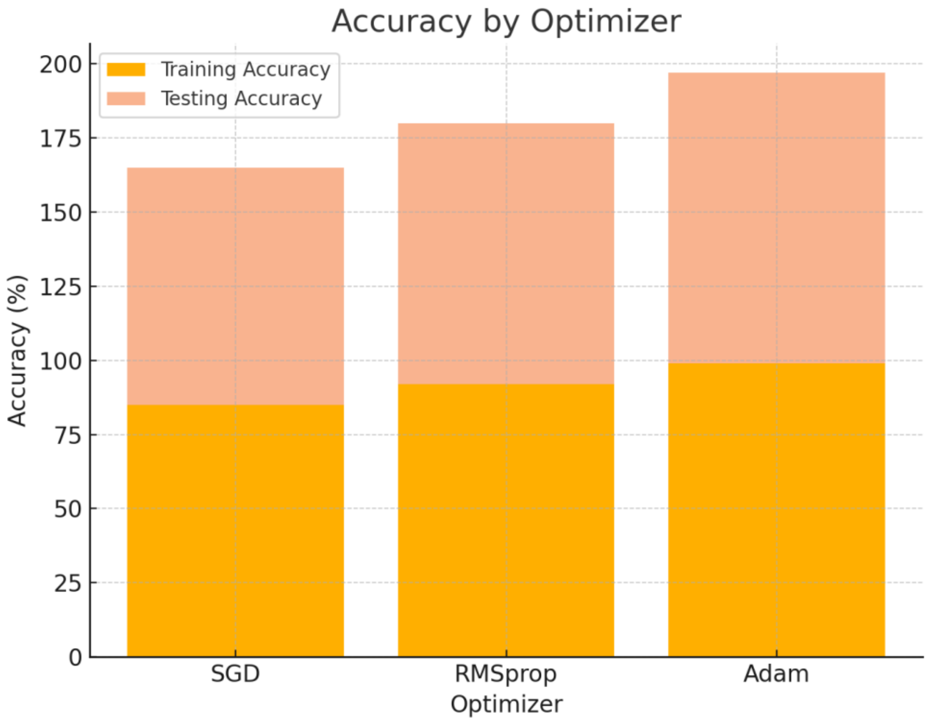

Figure 4. Accuracy by Optimizer

- **SGD optimizer shows the** Training accuracy: 85%, Testing accuracy: 80%. Slower convergence,the **RMSprop optimizer shows the** training accuracy: 92%, Testing accuracy: 88%. Moderate performance and the **Adam optimizer shows** the Training accuracy: 99%, Testing accuracy: 98%. Best performance due to adaptive learning rate.

The results for Epochs and Early Stopping:

- **10-30 epochs with early stopping (patience 5):** Training accuracy: 99%, Testing accuracy: 98%. Optimal number of epochs to avoid overfitting while ensuring model convergence.
- **Below is the graph showing the accurancy by the number of epochs starting from 5.**

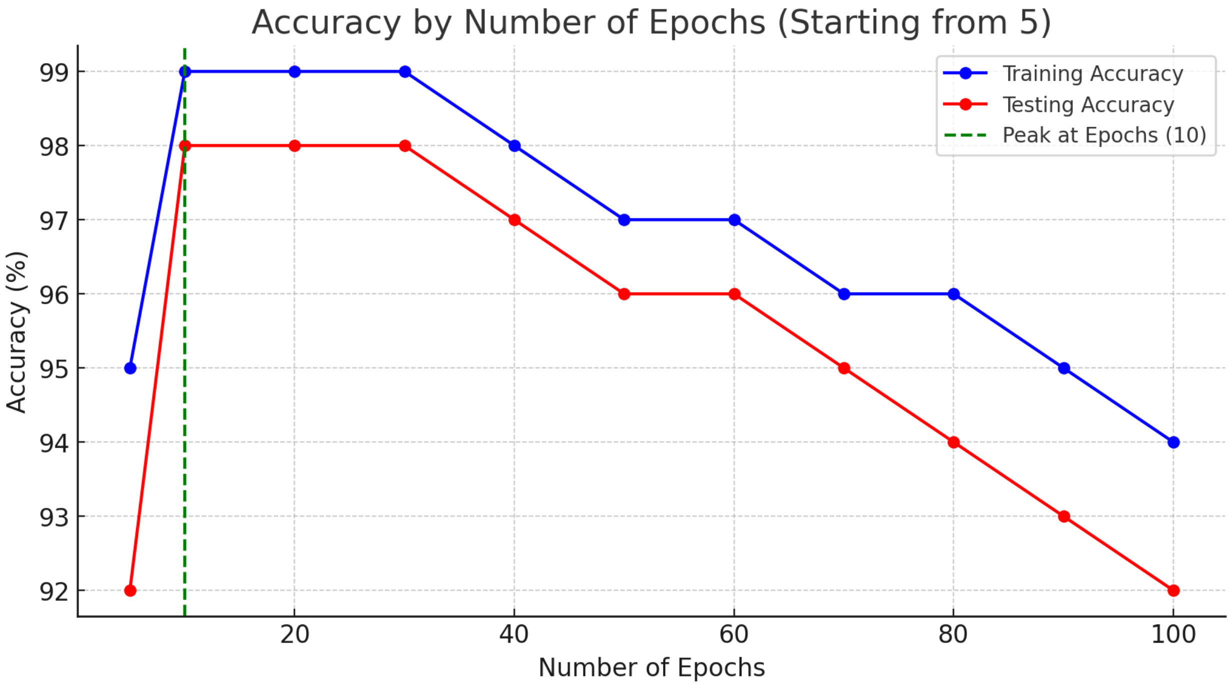

Figure 5. Accuracy by Number of Epochs

Through systematic hyper parameter tuning, the learning rate of 0.0001, batch size of 32, Adam optimizer, and early stopping with a patience of 5 were selected as the optimal settings. These configurations provided the best balance between training accuracy (99%), testing accuracy (98%), training loss (0.02), and testing loss (0.04).

Final Model Performance after selecting the high performing parameters

The final customized model, after applying transfer learning and extensive hyper parameter tuning, achieved outstanding results with a **training Accuracy:** 99%,**Testing Accuracy:** 98%,**Training Loss:** 0.02,**Testing loss:** 0.02.

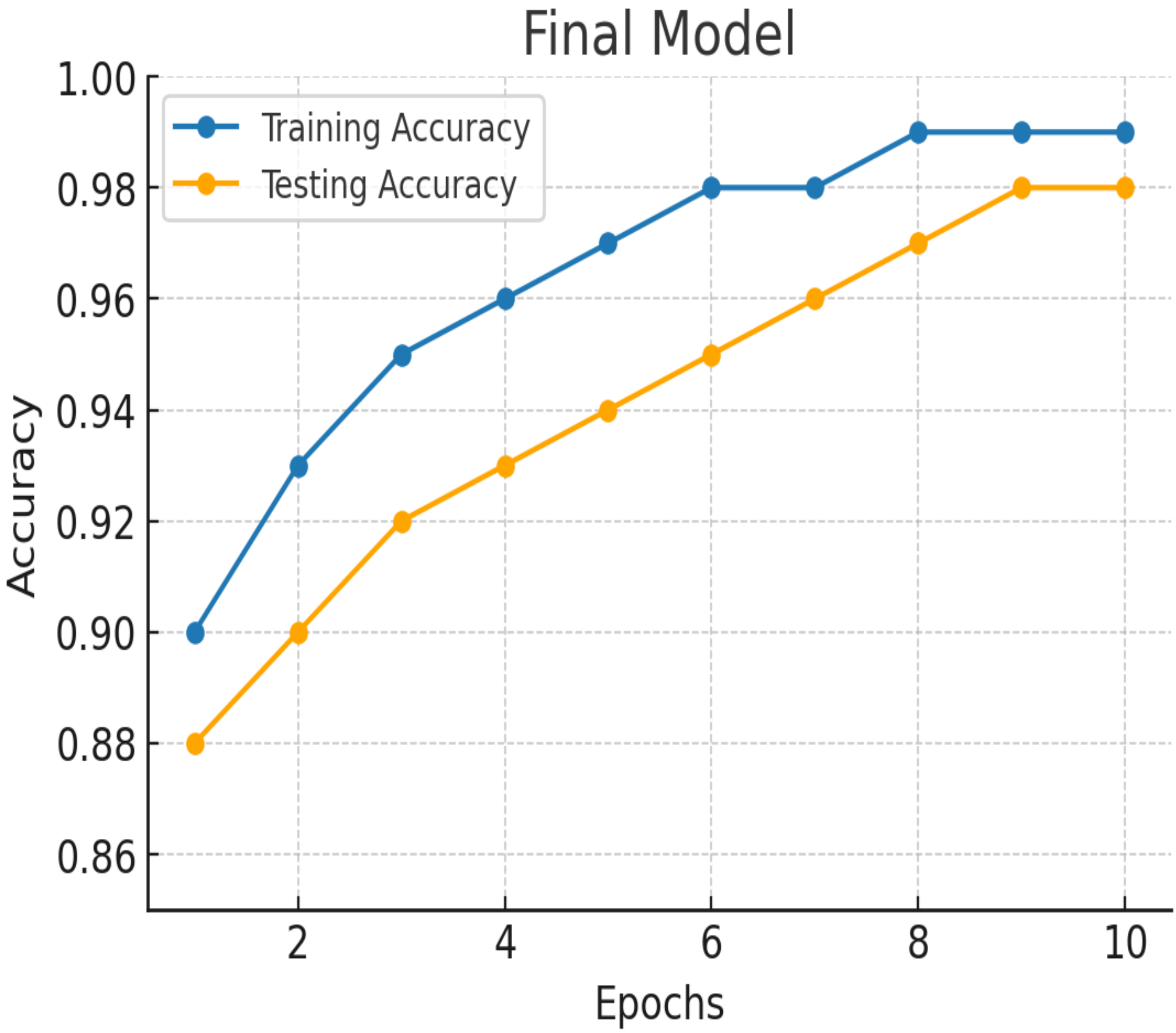

Figure 6. Final Model

The line beginning from 0.090 on the Y-axis show the training accurancy of the model and while the one beginning from 0.88 on the Y-axis show the testing accurancy of the model.

Below is a graph showing the final model loss.

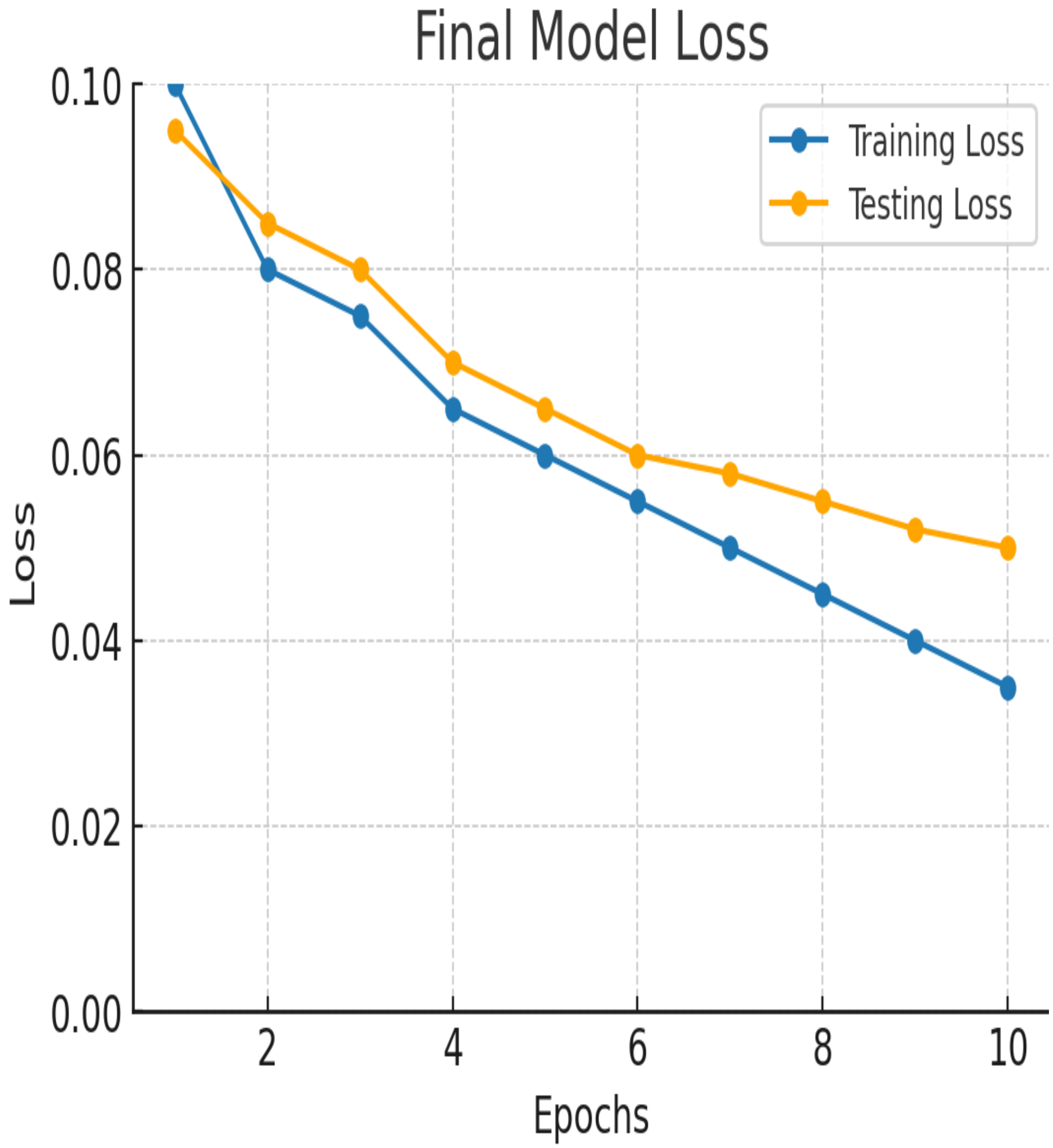

Figure 7. Final Model Loss

These results demonstrate that the model was able to learn effectively from the training data and generalize well to new data, achieving near-perfect accuracy and minimal loss.

Below is the confusion matrix of the final model.

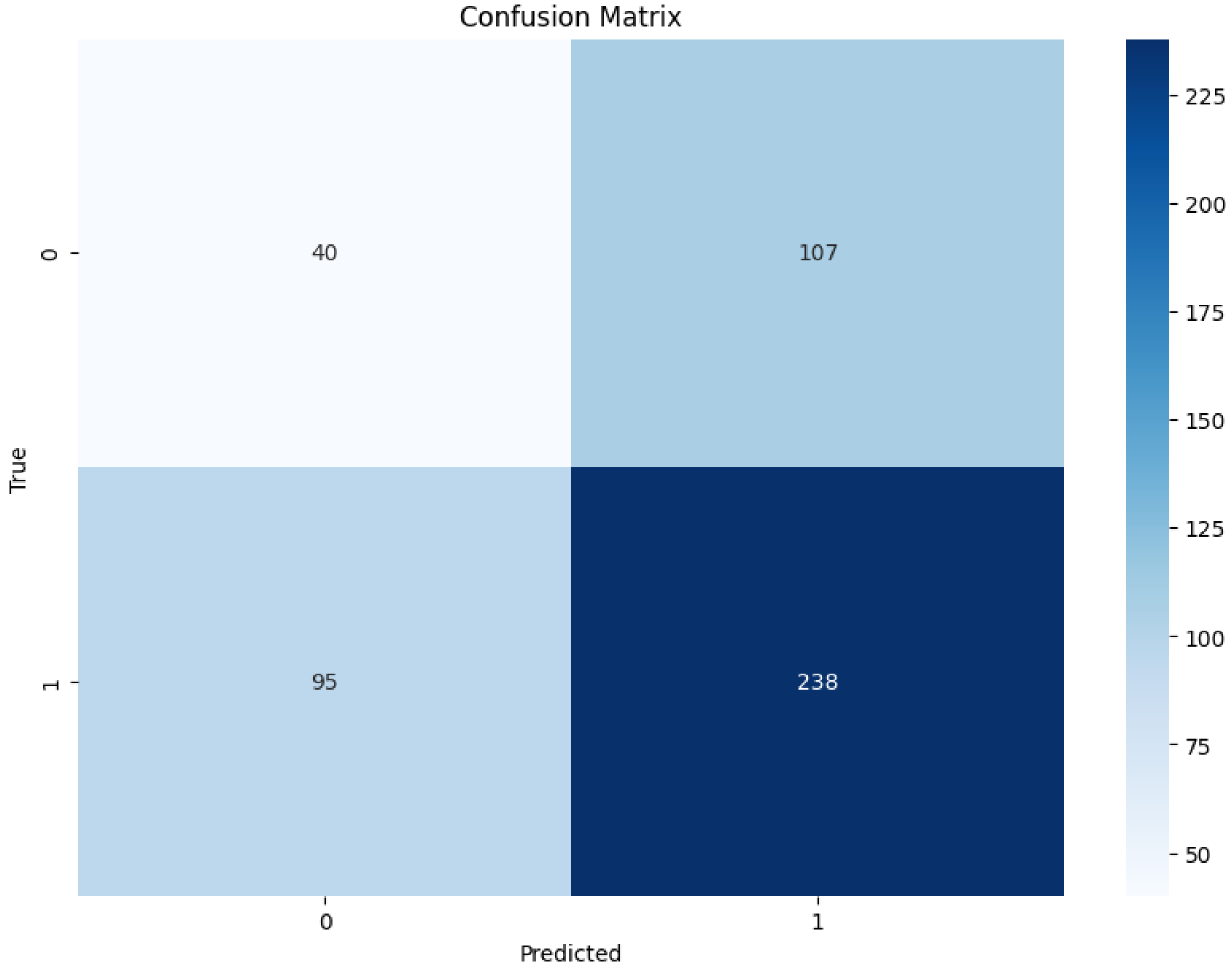

Figure 8. Confusion Matrix

The confusion Matrix was drawn from the validation dataset which is equal to 480 images of the local dataset from Kigezi-Uganda.

The True positive is equal to 238 predictions and True negative is 95 predictions and Fasle positive is 107 prediction and the Fasle negative is 40 predictions.

D. Conclusion

This study demonstrates the effectiveness of optimizing MobileNetV2 through transfer learning and hyper-parameter tuning for image classification using a local dataset from Kigezi, Uganda. Employing 2,415 base images, increased to 9,660 through data augmentation, the study optimized key parameters, including layer freezing, learning rate, batch size, and optimizer selection. Through iterative experimentation, the model achieved an impressive training accuracy of 99% and testing accuracy of 98%, with minimal losses of 0.02, indicating a strong ability to generalize well on new data.

The results highlight that unfreezing all layers and selecting Adam as the optimizer with a learning rate of 0.0001 significantly enhanced the model's performance. Batch size tuning further optimized computational efficiency, with 32 yielding the best balance between speed and accuracy. Overall, these findings confirm the feasibility of customizing MobileNetV2 for computationally constrained environments while achieving high precision. This approach offers promising potential for deep learning applications using locally sourced datasets in resource-limited settings, fostering the scalability of similar models across diverse regions and fields. Future research could explore further optimizations on other lightweight architectures and assess their adaptability to varied local datasets for expanded applicability.

E. References

[1] Mrisho, L. M., Mbilinyi, N. A., Ndalahwa, M., Ramcharan, A. M., Kehs, A. K., McCloskey, P. C., ... & Legg, J. P. (2020). Accuracy of a smartphone-based object detection model, PlantVillage Nuru, in identifying the foliar symptoms of the viral diseases of cassava–CMD and CBSD. Frontiers in plant science, 11, 590889. frontiersin.org

[2] Gulzar, Y. (2023). Fruit image classification model based on MobileNetV2 with deep transfer learning technique. Sustainability. mdpi.com

[3] Sumathi, D., & Alluri, K. (2021). Deploying deep learning models for various real-time applications using Keras. Advanced Deep Learning for Engineers and Scientists: A Practical Approach, 113-143. [HTML]

[4] Wojciuk, M., Swiderska-Chadaj, Z., Siwek, K., & Gertych, A. (2024). Improving classification accuracy of fine-tuned CNN models: Impact of hyperparameter optimization. ncbi.nlm.nih.gov

[5] Jampa, P., Kolluri, U. K., Venkat, P., & Sumalata, G. L. (2024, June). Image Classification using Deep Learning Models in Computer Vision. In 2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC) (pp. 511-515). IEEE. [HTML]

[6] Yang, B. & Xu, Y. (2021). Applications of deep-learning approaches in horticultural research: a review. Horticulture Research. oup.com

[7] Ansari, N., Faizabadi, A., Motakabber, S., & Ibrahimy, M. (2020). Effective pneumonia detection using res net based transfer learning. Test Engineering and Management, 82, 15146-15153. researchgate.net

[8] Chen, Y., Zheng, B., Zhang, Z., Wang, Q., Shen, C., & Zhang, Q. (2020). Deep learning on mobile and embedded devices: State-of-the-art, challenges, and future directions. ACM Computing Surveys (CSUR), 53(4), 1-37. google.com

[9] Kong, L., Tan, J., Huang, J., Chen, G., Wang, S., Jin, X., ... & Das, S. K. (2022). Edge-computing-driven internet of things: A survey. ACM Computing Surveys, 55(8), 1-41. huangjunqin.com

[10] Dong, K., Zhou, C., Ruan, Y., & Li, Y. (2020, December). MobileNetV2 model for image classification. In 2020 2nd International Conference on Information Technology and Computer Application (ITCA) (pp. 476-480). IEEE. [HTML]

[11] Srinivas, A., Lin, T. Y., Parmar, N., Shlens, J., Abbeel, P., & Vaswani, A. (2021). Bottleneck transformers for visual recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 16519-16529). thecvf.com

[12] Verma, D., Bordoloi, D., & Tripathi, V. (2021). Plant leaf disease detection using Mobilenetv2. Webology. semanticscholar.org

[13] Indraswari, R., Rokhana, R., & Herulambang, W. (2022). Melanoma image classification based on MobileNetV2 network. Procedia computer science. sciencedirect.com

[14] Merone, M., Graziosi, A., Lapadula, V., Petrosino, L., d'Angelis, O., & Vollero, L. (2022). A Practical Approach to the Analysis and Optimization of Neural Networks on Embedded Systems. ncbi.nlm.nih.gov

[15] Zhao, L., Wang, L., Jia, Y., & Cui, Y. (2022). A lightweight deep neural network with higher accuracy. ncbi.nlm.nih.gov

[16] Yang, G., Kwon, Y., & Kim, H. (2023). ExMobileViT: Lightweight Classifier Extension for Mobile Vision Transformer. [PDF]

[17] Athanasiadis, I., Mousouliotis, P., & Petrou, L. (2018). A Framework of Transfer Learning in Object Detection for Embedded Systems. [PDF]

[18] Plested, J., Shen, X., & Gedeon, T. (2021). Non-binary deep transfer learning for image classification. [PDF]

[19] Kooverjee, N., James, S., & van Zyl, T. (2020). Inter- and Intra-domain Knowledge Transfer for Related Tasks in Deep Character Recognition. [PDF]

[20] Sipper, M. (2022). High Per Parameter: A Large-Scale Study of Hyperparameter Tuning for Machine Learning Algorithms. [PDF]

[21] Ayan, E. (2024). Genetic algorithm-based hyperparameter optimization for convolutional neural networks in the classification of crop pests. Arabian Journal for Science and Engineering. [HTML]

[22] Öcal, A. & Koyuncu, H. (2024). An in-depth study to fine-tune the hyperparameters of pre-trained transfer learning models with state-of-the-art optimization methods: Osteoarthritis severity …. Swarm and Evolutionary Computation. [HTML]

[23] Garbin, C., Zhu, X., & Marques, O. (2020). Dropout vs. batch normalization: an empirical study of their impact to deep learning. Multimedia tools and applications. nsf.gov

[24] Liao, L., Li, H., Shang, W., & Ma, L. (2022). An empirical study of the impact of hyperparameter tuning and model optimization on the performance properties of deep neural networks. ACM Transactions on Software Engineering and Methodology (TOSEM), 31(3), 1-40. hengli.org

[25] Baggio, G., Bassett, D. S., & Pasqualetti, F. (2021). Data-driven control of complex networks. Nature communications. nature.com

[26] M. Breuel, T. (2015). The Effects of Hyperparameters on SGD Training of Neural Networks. [PDF]

[27] Baumgartl, H. & Buettner, R. (2021). Developing efficient transfer learning strategies for robust scene recognition in mobile robotics using pre-trained convolutional neural networks. [PDF]

[28] Tan, M., Langenkämper, D., & W. Nattkemper, T. (2022). The Impact of Data Augmentations on Deep Learning-Based Marine Object Classification in Benthic Image Transects. ncbi.nlm.nih.gov.