Hybrid MobileNetV2-ResNet Architecture for Image Classification: Tsinelas and Sapatos

Sean Vergel Labnotin[1]

University of Science and Technology of Southern Philippines, Lapasan Cagayan de Oro 9000, Philippines

Abstract. This paper presents a robust solution for the challenging binary classification of Filipino footwear, distinguishing between Tsinelas (slippers) and Sapatos (shoes) on an minimal, custom-collected dataset (24 training images). To address data scarcity and computational constraints, we propose a novel MobileNetV2-ResNet Hybrid architecture. The core innovation lies in a strategic feature-level fusion: the pre-trained, efficient MobileNetV2 backbone [1] is frozen to leverage high-quality, generic features learned from ImageNet, while custom ResNet Blocks [2] are added and trained specifically to refine these features via residual learning. This approach facilitates deep, task-specific feature extraction without overfitting. Training was conducted using aggressive data augmentation and an Adam optimizer, resulting in a promising test accuracy of 90.00%. Although the model showed strong generalization, training metrics exhibited volatility due to the extremely limited batch size (1). The results validate the hybrid architecture's efficacy as an accurate and resource-efficient method for specialized computer vision tasks where data is sparse.

Keywords: Hybrid Neural Network · MobileNetV2 · ResNet · Image Classification · Deep Learning · Tsinelas · Sapatos · Data Augmentation.

1 Introduction

1.1 Problem being solved

The core problem addressed is the binary image classification of Filipino footwear, specifically distinguishing between Tsinelas (slippers or flip-flops) and Sapatos (closed shoes). This involves developing an accurate and computationally efficient deep learning model capable of categorizing input images into these two distinct classes.

1.2 Why it is relevant

Accurate and automated image classification of common objects like footwear has significant practical relevance in various sectors, from retail inventory to fine-grained scene understanding. This project serves as a valuable case study in applying advanced transfer learning and network fusion techniques, specifically the MobileNetV2-ResNet hybridization, to solve a challenging computer vision task, especially when limited by a small, custom-collected dataset.

2 Dataset Description

2.1 Source

The dataset was gathered by the author himself. In which he takes photos of tsinelas and sapatos within his and his family's household.

2.2 Size

There are a total of 34 data gathered for this study, 17 of which are sapatos and 17 for tsinelas. For training both sapatos and tsinelas will be using 12 data each, and as for testing both will be using 5 data each class.

2.3 Sample Images

Fig. 1. sapatos 1

Fig. 2. sapatos 2

Fig. 3. sapatos 3

Fig. 4. sapatos 4

Fig. 5. tsinelas 1

Fig. 6. tsinelas 2

Fig. 7. tsinelas 3

Fig. 8. tsinelas 4

3 Methodology

3.1 Architectures used

The proposed model is a custom MobileNetResNetHybrid architecture, designed for efficient and accurate image classification.

MobileNetV2 Backbone: The model utilizes a pre-trained MobileNetV2 [1] as the primary feature extraction backbone. MobileNetV2 is selected for its efficiency, low computational cost, and ability to output rich feature maps.

ResNet-Style Refinement: The high-level features are subsequently refined using two sequential ResNetBlock modules. Each block implements the standard residual connection (x+F(x)) [2], which is beneficial for deep networks as it helps mitigate the vanishing gradient problem and allows for effective feature processing.

Classification Head: A final classification head uses an AdaptiveAvgPool2d(1) layer for global feature pooling, followed by a fully connected layer (nn.Linear(1280, 512)), a Dropout(0.5) layer for regularization, and a final linear layer to project the features to the two output classes.

3.2 Explanation of the Fusion Strategy

The network employs a feature-level fusion strategy via transfer learning:

Frozen Backbone: The parameters of the pre-trained MobileNetV2 feature extractor are explicitly frozen. This strategy preserves the powerful, general-purpose features learned from the large ImageNet dataset [1], which is critical for preventing overfitting when training on a very small target dataset.

Residual Refinement and Task-Specific Learning: The subsequent trainable ResNet Blocks and the classification head are the only parts of the network that are trained. This allows the model to learn the specific, fine-grained residual mappings and classification logic required to distinguish between Tsinelas and Sapatos, effectively fusing the efficiency of MobileNetV2 with the optimization benefits of ResNet [2].

3.3 Preprocessing and Training Details

Since the size of the dataset is small images were loaded and preprocessed using distinct transformation pipelines for training and testing. All images were first converted to PyTorch Tensors and normalized using the standard ImageNet means and standard deviations.

To artificially increase the data variability and reduce overfitting, the training images underwent extensive data augmentation, including RandomRotation(20) (up to 20 degrees), RandomHorizontalFlip(), RandomResizedCrop(300) (to a 300x300 pixel output), and RandomAffine(20).

The testing images were only subjected to deterministic resizing (Resize(300)) followed by a central crop (CenterCrop(300)) to ensure consistency without introducing random variations.

Training Configuration The model was trained for 20 epochs. Given the small dataset, a highly constrained batch size of 1 was used, contributing to the volatility observed in the training metrics. The training was conducted on a GPU (CUDA) specifically on an RTX 4060 desktop. The network was optimized using the Adam optimizer with a fixed, small learning rate of $1 * 10^{-4}$. The classification objective was measured using the standard nn.CrossEntropyLoss() function. Only the parameters in the custom ResNet blocks and the final classification head were updated, while the MobileNetV2 feature weights remained fixed as part of the transfer learning approach.

4 Results and Visualizations

4.1 Accuracy

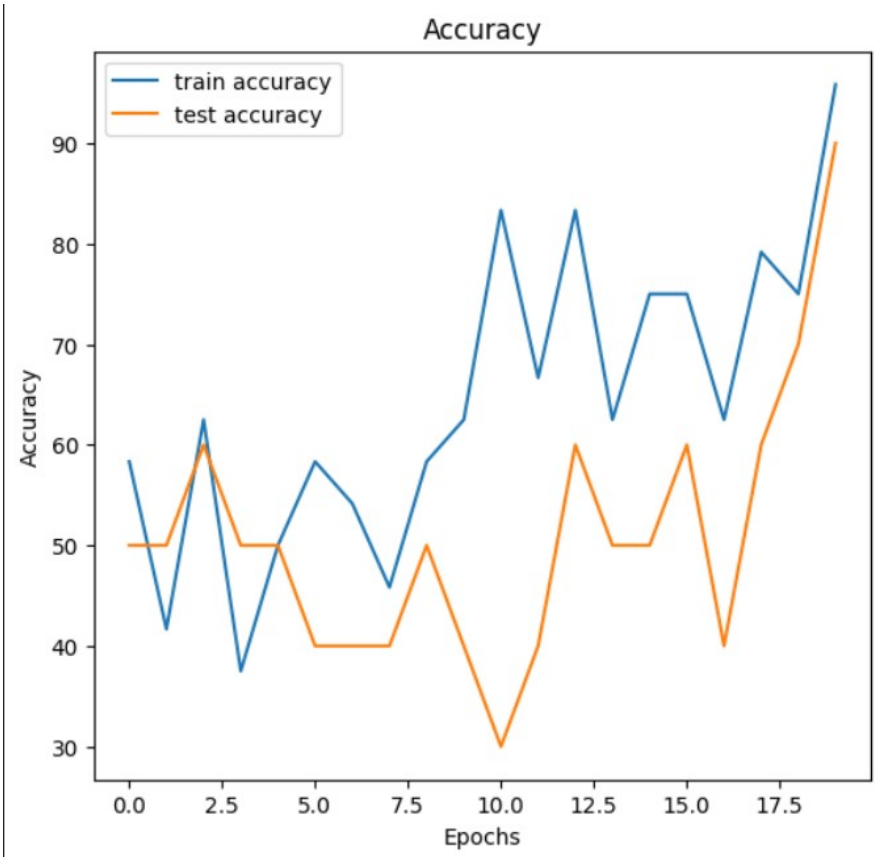

Fig. 9. Training and Testing Accuracy throughout epoch

As shown in 9, the accuracy of the training and testing improves as the number of epoch increase and it reaches it's peak at the 20th epoch where train accuracy is 95.83% and test accuracy is 90.00%. It also shown that the accuracy is highly volatile ranging from 37.50% to 95.83% . When the model was evaluated after the training, it has a test accuracy of 90.00%.

4.2 Loss

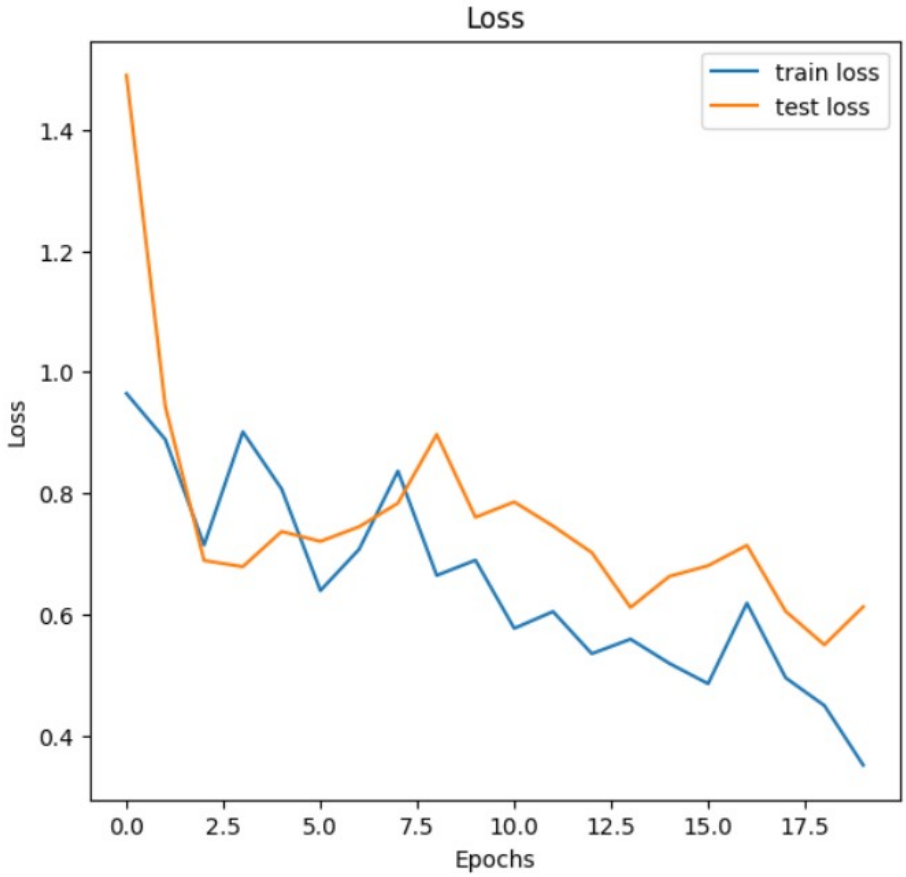

Fig. 10. Training and Testing loss throughout epoch

As shown in 10, both training and testing loss decreases as the number of epoch increase. This suggests that the training is improving the higher the epoch. Both the test loss and training loss generally trended downwards but exhibited minor fluctuations.

4.3 Sample Predictions

Fig. 11. Sapatos Sample Predictions

Fig. 12. Tsinelas Sample Predictions

For the sapatos sample predictions shown on 11. there were no errors and all 5 samples were completely predicted by the model. However for the tsinelas sample predictions shown on 12 3rd sample was incorrectly predicted by the model.

4.4 Visual Explanation of the Architecture

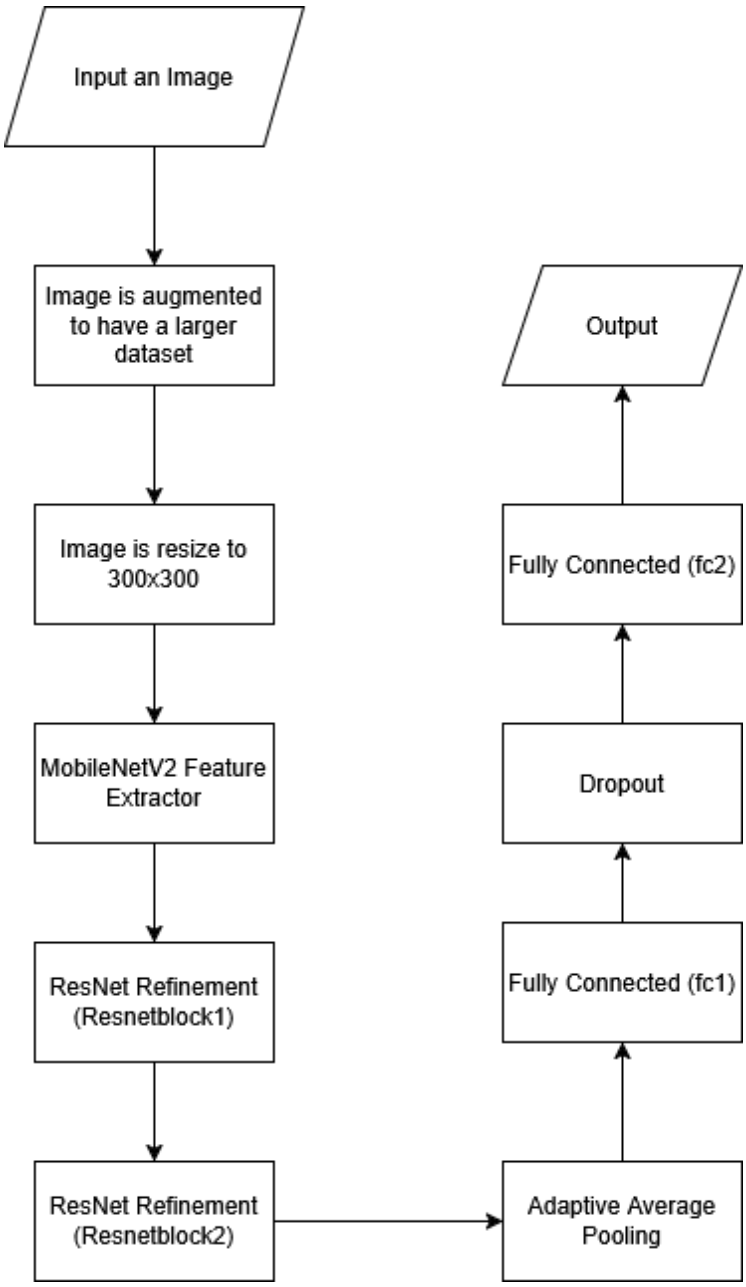

Fig. 13. Architecture

5 Discussion

5.1 What your fusion contributed

The MobileNetV2-ResNet fusion was essential for the project's success, allowing the model to gain maximum performance from minimal data.

Efficient Transfer Learning By leveraging and freezing the pre-trained MobileNetV2 backbone [1], the model gained immediate access to high-quality, efficient feature representations, minimizing the data required for effective learning.

Feature Refinement The subsequent trainable ResNet Blocks [2] provided the network with the depth and residual mechanism needed to fine-tune these generic features into highly discriminative, task-specific representations for the Tsinelas vs. Sapatos classification.

5.2 What worked, what didn't

Prediction performance was good since it achieved high test accuracy of 90.00% despite a minimal training set size, validating the hybrid transfer learning approach. The combination of data augmentation and backbone freezing was also effective in training a functional model on extremely limited data. Increasing the epoch was also quite effective as it gives more chances for the training model to improve.

Although some work, there were also some things I tried that didn't work, strictly due to the limited amount of data gathered. I tried using a higher batch size, but in most cases it would often result in overfitting. I also tried early stopping, but the results were inconsistent and there were times when it stopped too early.

6 Conclusion

The MobileNetResNetHybrid network is an effective and robust solution for the low-resource footwear classification problem. The strategic fusion of a frozen, efficient MobileNetV2 feature extractor with trainable ResNet-style residual blocks successfully leverages transfer learning to achieve a notable 90.00% test accuracy. Future work should focus on addressing the size of datasets by increasing its size as this would help in reducing overfitting and enables in using higher batch sizes.

References

1. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4510–4520. https://doi.org/10.48550/arXiv.1801.04381

2. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. https://doi.org/10.48550/arXiv.1512.03385