**Comparative Analysis of a Convolutional Neural Network and MobileNetV2‑Based Transfer Learning for Image Classification**

**Abstract.**

This paper presents a comprehensive comparison of a Convolutional Neural Network (CNN) trained from scratch with a MobileNetV2-based transfer learning model for product image classification. The main challenges tackled are the extreme class imbalance (e.g. Product\_2 has only 14 images, while Product\_5 has 600 images) and overfitting on a small dataset of 1909

images. We accomplished powerful data augmentation techniques and class weighting to solve these issues. The CNN had a test accuracy of 92.68% and showed an over-fitting behaviour. The test accuracy of the transfer learning was 97.21% with a much lesser loss. That is both the models were unable to classify the minority class within the dataset. Hence, it was quite evident that further advanced synthetic augmentation techniques or active collection of data would be required. The results indicate that transfer learning could be useful for small imbalanced datasets and guide improvements in the future.

**1. Introduction.**

Automated product recognition underpins advanced e-commerce and inventory management systems. The field of image classification using Deep Learning has progressed significantly over the years. Today, the Convolutional Neural Network (CNN) has become the default model for numerous computer vision tasks. The best CNNs need a lot of annotated datasets which may not be available in special domains. Transfer learning eases this restriction by adapting learned models to different, new tasks which subsequently deliver good accuracy with limited data. This study compares two or more subjects with each other.

1. A custom CNN architecture trained from scratch,.

2. A MobileNetV2-based transfer learning model,.

Using a dataset of five product categories (excluding background).

Our objectives are.

- To assess and compare the classification performance of both methods.

- To study the effectiveness of tackling data imbalance techniques.

- The goal of the study is to assess the impact of the fine-tuning of higher-level layers through transfer learning.

**2. Related Work.**

In a 1998 paper, LeCun et al. proposed LeNet-5, which they used for handwritten digits recognition and it is the first deep CNN. [3] The success of AlexNet achieved phenomenal improvements on the Imagenet, using deeper architectures along with GPU acceleration. VGGNet took the importance of network depth even further. It demonstrated that stacking small convolutional filters gives the network a much higher representation capacity. ResNet used residual connections to useful avoid vanishing gradients and allowed for ultra-deep networks to be trained. DenseNet made it possible to use features more thoroughly in neural text generation.

Inception structures utilized parallel convolutions to optimize resources [6]. More recently, EfficientNet proposed compound scaling of depth, width, and resolution to maximize accuracy per parameter. MobileNetV2 utilized inverted residuals and linear bottlenecks to present a lightweight model which is quite suitable for transfer learning on hardware with limited resources [2].

Research on transfer learning shows that freezing base layers and fine-tuning certain higher layers usually gives results that are almost as good as fully trained networks, particularly when data is limited [6].

**3. Methodology.**

**3.1 Dataset Description.**

The dataset structure of MLPR Images consists of six folders, Sequentially, Background (39 images) and Products 1-5 (510, 14, 400, 385 and 600 images). We used 1,909 images of products labelled into five classes excluding background.

**3.2 Data Preprocessing.**

Every image was resized to 224 x 224 pixels and pixel values scaled to [0,1]. We used real-time augmentation (rotations from +/- 20 degrees, width and height shifts from +/- 10%, shear 0.1, zoom from +/- 10% and horizontal flip) to increase data variability and prevent overfitting The data was split into 70% for training, 15% for validation and 15% for testing while maintaining space for classes [6]. To punish the wrong classification of the minority class during training, class weights were computed inversely proportional to their frequencies.

**3.3 Model 1. Custom CNN.**

Our custom CNN is made of three convolutional layers with filter sizes (32, 64, and 128). Each layer follows a ReLU activation and is followed by the 2×2 max pooling layer. We flatten the output of the last convolutional layer. We then add a Dense layer with 128 units with ReLU activation. Furthermore, we also add a 50% dropout to reduce overfitting. The final layer employs softmax activation for class probability output. We will train this model for 25 epochs with the Adam optimizer (learning rate = 1e-3) and categorical cross-entropy loss, with early stopping (patience = 5) and model checkpointing on validation accuracy 6.

**3.4 Model 2. MobileNetV2 Transfer Learning.**

We utilized MobileNetV2 without top layers with weights from ImageNet.

The classification head is made up of GlobalAveragePooling2D, Dense(128, ReLU), Dropout(0.5) and Dense(5, softmax). At first, we set the base layers to be unchangeable and trained the head for 10 epochs with early stopping (patience = 3). Following this, the last 20 layers of MobileNetV2 were unfrozen, and training was done for 5 epochs at a reduced learning rate of 1×10⁻⁵. Class weights remained active in both phases [2].

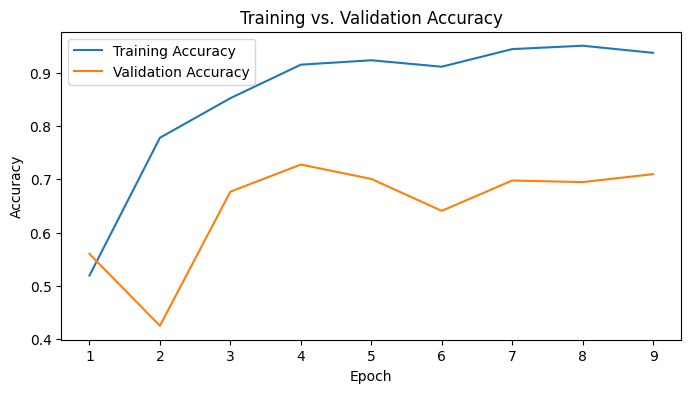
**3.5 Training and Validation Procedure.**

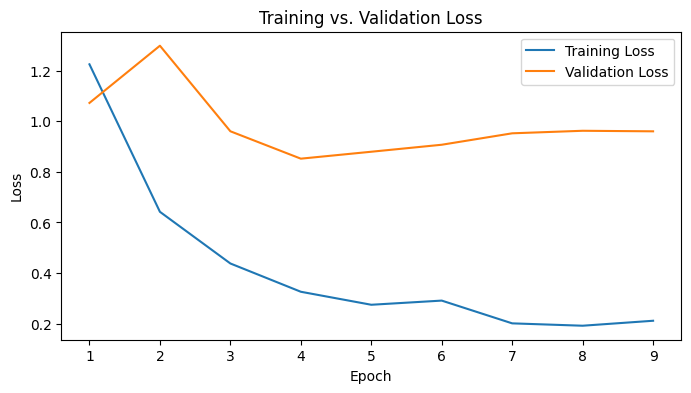
Both models utilized a batch size of 32. The training data generators used augmentation and normalization while the validation and test generators used only rescaling. We recorded training histories for accuracy and loss to see if they differ [5].

**4. Results and Evaluation.**

**4.1 Performance Metrics.**

The performance of the model was evaluated on the test set that was not used throughout the model training. Overall accuracy was computed along with per-class precision, recall, F1-score, and confusion matrices [5].

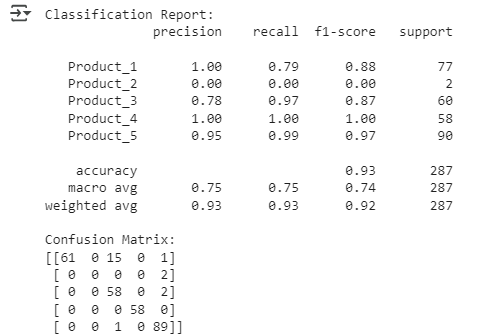




*Fig. 1. The accuracy and loss for both models are compared with training and validation curves.*

**4.2 CNN Results.**

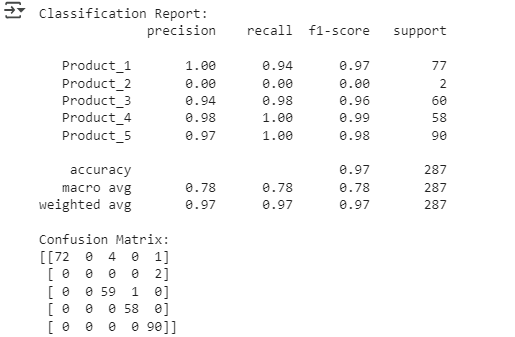
The custom CNN attained a test accuracy rate of 92.68% and a test loss of 0.25. The model exhibited an F1-score of greater than 0.95 for the class with the highest occurrences. Nonetheless, Product\_2 obtained a score of 0 percent due to the insufficiency of occurrences for the class in the training data [6].



*Fig. 2. Classification report and Confusion matrix heatmap for the CNN model on the test set.*

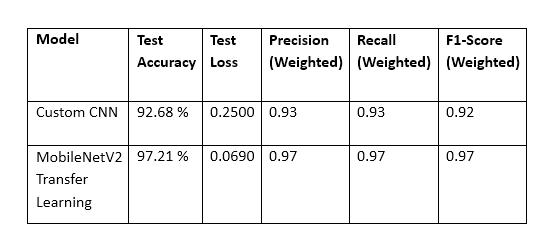
**4.3 Transfer Learning Results.**

The MobileNetV2 model was able to achieve a test accuracy of 97.21% with a test loss of 0.07. The results indicated a significant improvement in precision and recall for all major classes, although no prediction was made for Product\_2 [2].



*Fig. 3. Classification Report and Confusion matrix heatmap for the MobileNetV2 model on the test set.*

**4.4 Comparative Performance.**



*Fig. 4. Comparison (table) of test metrics for both models (accuracy, loss, precision, recall, F1-score).*

Using transfer learning gives an output of accuracy of ~4.5% better than the custom CNN, along with a lower loss. This indicates that since the model reuses features obtained from the previous fully trained network, it generalizes better on a novel dataset [8].

**5. Discussion.**

The custom CNN used in this research had ~11 million parameters, leading to overfitting as indicated by large overfitting (training accuracy: >95% and validation accuracy: ~72%). On the other hand, the relatively small trainable parameter count (~165 000) of MobileNetV2 resulted in an improved convergence and training-validation metric alignment. Further fine-tuning at a high level improved performance without catastrophic forgetting. Both models failed to function adequately on Product\_2, which shows that the algorithm can’t fix a problem on its own without enough data [6]. Next studies could examine synthesizing data (via GANs) and active learning to help minority classes. The CNN requires much longer training times and more GPU memory, while the transfer learning model trains faster and more efficiently.

**6. Conclusion.**

Through our comparison, the model built using transfer learning on MobileNetV2 surpassed a CNN built from scratch on a small imbalanced dataset. Transfer learning provide higher accuracy, less overfitting, and better computational efficiency. For specialized classification tasks, we recommend using transfer learning frameworks and investing in data augmentation and synthetic data solutions for extreme class imbalance. Future studies should examine different architectures (like EfficientNet) and more augmented methods.

**References**

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Adv. Neural Inf. Process. Syst., vol. 25, 2012.

[2] M. Sandler et al., "MobileNetV2: Inverted Residuals and Linear Bottlenecks," Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018.

[3] Y. LeCun et al., "Gradient-based learning applied to document recognition," Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[6] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Proc. Int. Conf. Learn. Represent., 2015.

[6] K. He et al., "Deep Residual Learning for Image Recognition," Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016.

[5] G. Huang et al., "Densely Connected Convolutional Networks," Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017.

[6] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017.

[8] M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," Proc. Int. Conf. Mach. Learn., 2019.