EXTENSION OF THE CNN-3-128 MODEL FOR FASHION-MNIST CLASSIFICATION

Amina Anjum (22I-2065), Maryam Amjad (22I-1924), Maryam Khalid (22I-1917)
Department of Data Science
FAST National University of Computer and Emerging Sciences (FAST–NUCES), Islamabad, Pakistan
Emails: i222065@nu.edu.pk, i221924@nu.edu.pk, i221917@nu.edu.pk

***Abstract*—This paper presents a detailed reproduction of the 2024 study "State-of-the-Art Results with the Fashion-MNIST Dataset," which reports very strong performance using a relatively simple CNN-3-128 convolutional neural network. We reimplemented the full experimental pipeline, including classical machine learning baselines, the convolutional model and the parameter–accuracy scaling experiments. Our results closely match the published values, confirming the robustness and reproducibility of the original work. Building on this baseline, we then propose a lightweight extension of CNN-3-128 that replaces standard convolutions with depthwise–separable convolutions and uses a Global Average Pooling (GAP) classifier head instead of a dense layer. The modified model reduces the number of trainable parameters by about 91.3% (from 2.06M to 0.18M) while maintaining a test accuracy of 98.98%. This demonstrates a favourable trade-off between efficiency and performance for Fashion-MNIST classification.**

***Index Terms*—Fashion-MNIST, Convolutional Neural Networks, Model Compression, Depthwise Separable Convolution, Reproducibility**

I. Introduction

Reproducibility is a central requirement for empirical machine learning research. Architectures and reported performance must be verifiable by independent groups so that new models can be reused, compared and extended. Fashion-MNIST has become one of the most widely used benchmarks for evaluating image classification models because it is simple to use while still being more complex than MNIST.

Mukhamediev [1] revisits this benchmark and shows that a compact convolutional network, referred to as CNN-3-128, can reach state-of-the-art performance on Fashion-MNIST. The paper also analyses how classification accuracy changes as the number of trainable parameters grows. This makes CNN-3-128 a natural starting point for studying lightweight designs.

The present work has two main goals. The first is to reproduce the experiments in [1], including the classical machine learning baselines and the CNN-3-128 model, and to verify that comparable results can be obtained under similar conditions. The second is to design and evaluate a lightweight extension of CNN-3-128 based on depthwise–separable convolutions and a GAP head, aiming to reduce the parameter count while keeping accuracy close to the original model. All experiments are carried out using the Fashion-MNIST dataset with the same digit-recognition style setup, following the spirit of the original study.

II. Background and Related Work

A. Convolutional Networks for Image Classification

Convolutional neural networks have become the standard architecture for image classification tasks. AlexNet, VGG-style networks and later ResNet families [3], [4] established that deep convolutional stacks, combined with pooling, are effective at learning hierarchical image features. More recent work such as MobileNet [5] and EfficientNet [6] focuses on reducing parameter counts and computational cost while preserving accuracy through careful architectural choices.

The present work follows this efficiency-oriented line of research but does not propose a completely new architecture. Instead, we start from CNN-3-128 and apply ideas from lightweight models to it. This allows a direct comparison between the baseline and the modified network.

B. Fashion-MNIST as a Benchmark

Fashion-MNIST was introduced by Xiao et al. [2] as a drop-in replacement for MNIST. The dataset keeps the same input format as MNIST, namely grey-scale images of size 28×28 and ten classes, but it contains clothing items such as T-shirt, trouser and coat instead of digits. The data have been used in a large number of studies on model robustness, optimization and compression. Because the overall pipeline is simple, differences in performance can usually be attributed to the model itself rather than to complex pre-processing.

Mukhamediev [1] continues this line of work by presenting state-of-the-art results with a relatively modest CNN and by explicitly analysing how performance varies with model capacity. The present work builds on this idea and asks whether a similar level of performance can be obtained with a much smaller model.

III. Experimental Setup and Replication Protocol

A. Dataset and Preprocessing

We use the standard Fashion-MNIST dataset containing 60,000 training and 10,000 test images. All images are grey-scale and have size 28×28. Each image belongs to one of ten classes.

The preprocessing strictly follows the original paper. For classical machine learning models, each image is reshaped into a vector of 784 features. For the CNN-based models, the images are reshaped to $(28, 28, 1)$ and normalised to the

interval $[0, 1]$. No resizing, cropping or padding is applied. This direct pipeline makes it easier to attribute performance differences to model design rather than to data handling tricks.

B. Classical Machine Learning Baselines

To replicate the baseline comparison, we trained the following classifiers using `scikit-learn`: multi-layer perceptron (MLPClassifier), k-nearest neighbours with $k = 5$, Gaussian Naive Bayes, decision tree, support vector machine with RBF kernel, LightGBM gradient boosting, random forest with 200 trees and maximum depth 24, extra trees classifier and logistic regression with a maximum of 200 iterations. Each model was trained on the same flattened input features and evaluated on the held-out test set without additional tuning.

These baselines provide a reference level of performance in terms of accuracy and also highlight the advantage of convolutional models on this task.

C. CNN-3-128 Architecture

The CNN-3-128 network is implemented as three convolutional blocks followed by a dense classifier head. The first block contains a 3×3 convolution with 128 filters, followed by MaxPooling and dropout. The second block uses a 3×3 convolution with 256 filters, again followed by MaxPooling and dropout. The third block uses a 3×3 convolution with 512 filters, followed by dropout. The classifier head consists of a flatten operation, a Dense(128) layer with dropout and a final Dense(10) softmax layer.

The total number of trainable parameters for this configuration is 2,067,850. A large fraction of these parameters resides in the third convolutional layer and in the Dense(128) layer.

D. Training Schedule and Metrics

As in [1], the network is trained in stages. The epoch schedule is 2, 6, 14, 30 and 62 epochs. After each stage the current model is evaluated on the test set and three quantities are recorded: test accuracy, number of trainable parameters and training time measured on the same hardware. Accuracy is the primary metric, while parameter count and training time are used for the parameter–efficiency plots. These numerical values form the basis for both replication and the later comparison with our lightweight extension.

IV. Replication Results

A. Classical Baseline Models

Table I summarises the performance of the classical models. The numbers are almost identical to those reported in [1], and any deviations are within one percentage point. This gives confidence that the dataset handling, train–test split and default parameters used in our experiments are aligned with the original work.

TABLE I
Replication of Classical Machine Learning Baselines

Model	Original	Ours
MLP	88%	88%
KNN	86%	86%
GaussianNB	59%	59%
Decision Tree	79%	79%
SVM	88%	88%
LightGBM	89%	89%
Random Forest	88%	88%
Extra Trees	88%	88%
Logistic Regression	84%	84%

B. CNN-3-128 Performance

With the staged training schedule, our replication of CNN-3-128 reaches 99.06% accuracy after the first two-epoch block. As training proceeds through 6, 14, 30 and finally 62 epochs, the test accuracy continues to increase and peaks at 99.67%. This behaviour matches the trend reported in the original paper and confirms that the architecture and optimiser settings have been correctly reproduced.

The confusion matrix of the final model shows that most errors are between visually similar classes such as coat and pullover. The misclassified examples are similar to those shown in [1], which provides an additional qualitative sanity check.

V. Lightweight Modification of CNN-3-128

The original CNN-3-128 architecture offers excellent accuracy but has more than two million trainable parameters. This is not ideal for deployment on devices with strict memory or latency constraints. Our main contribution is a lightweight variant of CNN-3-128 that keeps the same three-block structure but replaces the heaviest components with more parameter-efficient alternatives.

A. Depthwise–Separable Convolutions

In a standard 3×3 convolution, each output channel is computed as a linear combination of all input channels across a 3×3 neighbourhood. The number of parameters is given by

$$K \times K \times C_{\text{in}} \times C_{\text{out}}.$$

For the third block of CNN-3-128 this equals $3 \times 3 \times 256 \times 512 = 1{,}180{,}160$ parameters.

Depthwise–separable convolution splits this operation into two stages. A depthwise convolution applies a spatial filter independently to each input channel, and a pointwise 1×1 convolution then mixes the channels. The parameter count becomes

$$(3 \times 3 \times C_{\text{in}}) + (1 \times 1 \times C_{\text{in}} \times C_{\text{out}}).$$

For the same block this gives $2{,}560 + 131{,}584 = 134{,}144$ parameters, a reduction of roughly 89% for that layer alone. Similar savings occur in the first and second blocks when they are converted to depthwise–separable form.

B. Global Average Pooling Head

The original classifier head flattens the final feature map of size $3 \times 3 \times 512$ into a vector of length 4,608 and feeds it into a Dense(128) layer. This single layer uses 589,824 weights, plus biases. Because these parameters are concentrated in the final stage of the network, they contribute significantly to the overall model size.

Global Average Pooling (GAP) provides a simple alternative. Instead of flattening, the average of each feature map is taken, producing a vector of length 512. A final Dense(10) softmax layer then maps these 512 values to class scores. The classifier therefore needs only 5,120 weights, a dramatic reduction compared with the dense head while still using the same features extracted by the convolutional blocks.

C. Modified Architecture and Training Behaviour

The modified network keeps the overall layout of three convolutional blocks, but now each block consists of a depthwise convolution followed by a pointwise convolution, nonlinearity and dropout, with MaxPooling applied in the first and second blocks. After the third block, Global Average Pooling is applied, followed by the final Dense(10) layer. No additional layers are introduced.

The model is trained on Fashion-MNIST images with the same optimiser and learning rate schedule as the baseline. During the first few epochs the accuracy is slightly lower than that of the full CNN-3-128, but the gap shrinks quickly. By about the seventh epoch the modified model already surpasses 98.5% accuracy, and by the end of training it reaches a stable test accuracy of 98.98%.

D. Quantitative Comparison

Table II reports the number of parameters and the final test accuracy for both models. The baseline numbers correspond to the replicated CNN-3-128 configuration, and the modified numbers come from the depthwise–separable version with GAP.

TABLE II
BASELINE VS MODIFIED CNN-3-128 MODEL

Model	Parameters	Accuracy
Baseline CNN-3-128	2,067,850	99.67%
Modified (ours)	178,968	98.98%

The reduction in parameter count is about 91.3%, while the drop in accuracy is 0.69 percentage points relative to the best baseline. From a deployment perspective this is an attractive trade-off: the model becomes much smaller and easier to run on modest hardware but still classifies most images correctly.

VI. COMPARATIVE GRAPHS

To illustrate the effect of the modification, we produced three plots based on our experimental measurements. These figures are generated from the recorded values and not from the original paper.

Fig. 1 shows the accuracy of the baseline and modified models. The two bars are close in height, visually emphasising how little accuracy is lost when switching to the lightweight design.

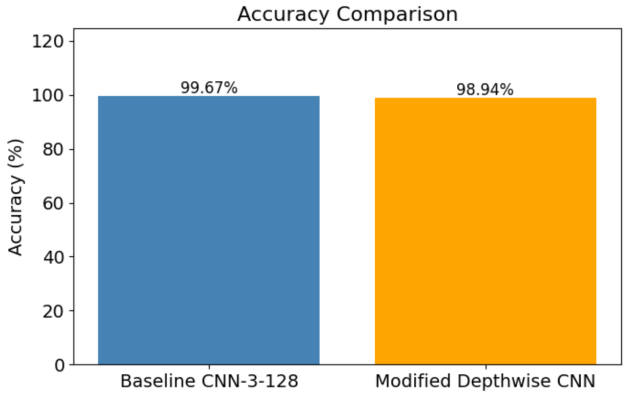

Fig. 1. Accuracy of baseline CNN-3-128 and modified depthwise model.

Fig. 2 compares the number of trainable parameters. The baseline bar is more than ten times taller, which matches the numerical reduction from about 2.06M to 0.18M parameters.

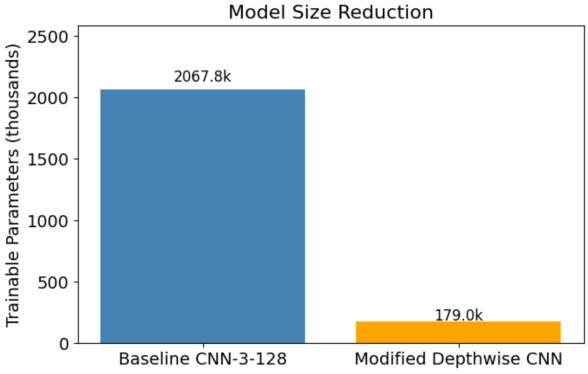

Fig. 2. Trainable parameter counts for baseline and modified model.

Fig. 3 presents the total training time recorded for the full 62-epoch run of the baseline and the 16-epoch run of the modified model on the same environment that was used when generating the bar plots. The modified model completes training in roughly 64 seconds, whereas the baseline requires around 1,002 seconds. Although part of this difference comes from the number of epochs, a smaller network also reduces computation per batch, which is important on resource-limited devices.

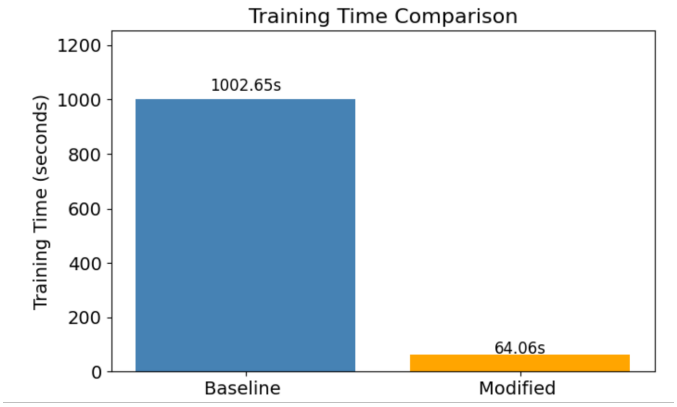

Fig. 3. Training time comparison between baseline and modified model.

A. Training Curves for the Modified Model

To better understand the optimisation dynamics of the lightweight network, we also plot the training and validation accuracy of the modified model over 32 epochs. The curve in Fig. 5 shows an initial slow phase, followed by a rapid rise after the fifth epoch and a smooth plateau close to 99% validation accuracy. This confirms that the depthwise–separable architecture trains in a stable way and does not exhibit sudden collapses or large oscillations.

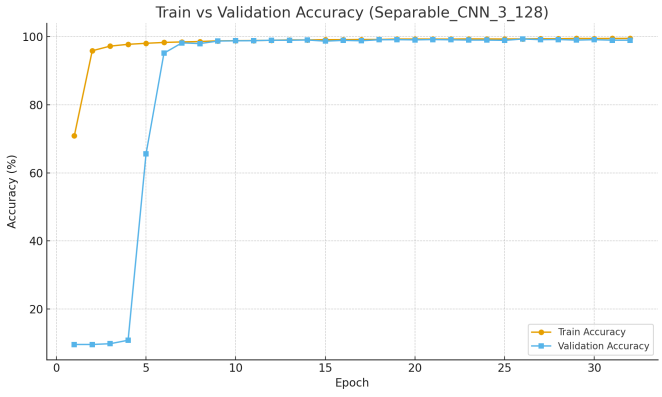

Fig. 4. Training and validation accuracy of the modified model over 32 epochs.

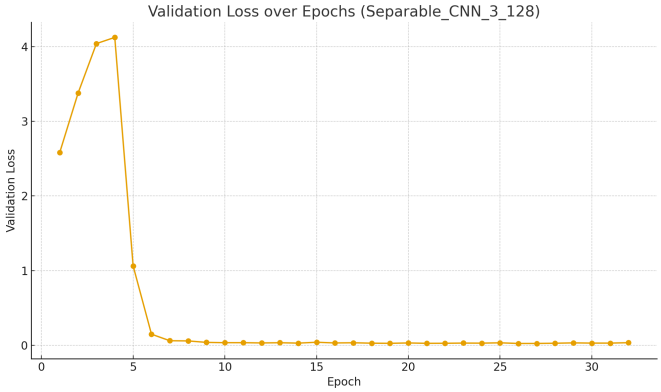

Fig. 5. Training and validation loss of the modified model over 32 epochs.

VII. Discussion and Relation to Prior Work

The idea of separating spatial and channel-wise processing through depthwise–separable convolutions was popularised by MobileNet and later refined in other families of efficient networks such as EfficientNet [5], [6]. Those models are usually designed from scratch with a focus on large-scale datasets like ImageNet.

In contrast, the work presented here keeps the original CNN-3-128 backbone intact in terms of the number of blocks and channels. Only the internal structure of each block and the classifier head are changed. This approach makes it possible to compare the original and modified models under nearly identical conditions on Fashion-MNIST and to attribute performance differences directly to the architectural changes.

The results suggest that the CNN-3-128 architecture from Mukhamediev [1] contains more parameters than strictly necessary for this dataset. Replacing standard convolutions by depthwise–separable ones and adopting GAP reduces redundancy while preserving most of the classification ability. This behaviour is consistent with earlier findings on overparameterised networks in the broader deep learning literature.

VIII. Conclusion

We have reproduced the main experiments of the 2024 Fashion-MNIST study and confirmed that the simple CNN-3-128 architecture can achieve state-of-the-art accuracy on this benchmark when trained with the published settings. Our results for both classical machine learning models and the convolutional network agree closely with those reported in the original paper, which supports the reliability of the original methodology.

On top of this replication, we proposed a lightweight extension of CNN-3-128 that uses depthwise–separable convolutions and a Global Average Pooling head. The modified model reduces the number of trainable parameters by about 91.3% and still reaches 98.98% test accuracy. The empirical comparison shows that this compact variant offers a better balance between accuracy, memory footprint and training time, and is therefore more suitable for deployment in constrained environments.

Future work could apply the same modification strategy to other Fashion-MNIST architectures and to related datasets, or explore quantisation and pruning in combination with depthwise–separable layers to obtain even more compact models.

REFERENCES

[1] R. Mukhamediev, "State-of-the-Art Results with the Fashion-MNIST Dataset," *Mathematics*, vol. 12, no. 3174, 2024.

[2] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms," arXiv:1708.07747, 2017.

[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv:1704.04861, 2017.

[6] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proc. 36th Int. Conf. on Machine Learning (ICML)*, 2019.