NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CZ4042 Neural Networks and Deep Learning

Assignment 2

AY 2023-2024

Team Members / Emails:
Wong Yi Pun / U2021844D / wong1219@e.ntu.edu.sg
Jeremy U Keat / U2021794H / ju001@e.ntu.edu.sg
Ng Yue Jie Alphaeus / U2021469L / ang096@e.ntu.edu.sg

1. Table of Contents

2. Introduction

2.1. Project Background

Neural networks and deep learning have revolutionised the field of Computer Vision. The Fashion MNIST dataset serves as a benchmark for image classification tasks.

2.2. Project Objective

The primary objective of our project is to explore various methods and attempt to improve upon existing methods of image classification for the Fashion MNIST dataset. This will be done by exploring the use of building custom Convolutional Neural Network (CNN) models, transfer learning, and data augmentation.

2.3. Fashion MNIST Dataset

2.3.1. Number of Samples

It contains a total of 70,000 grayscale images, which are divided into 60,000 training samples and 10,000 test samples.

2.3.2. Class Labels

There are ten distinct class labels, each corresponding to a different fashion item or clothing category.

The different fashion item are as follows:

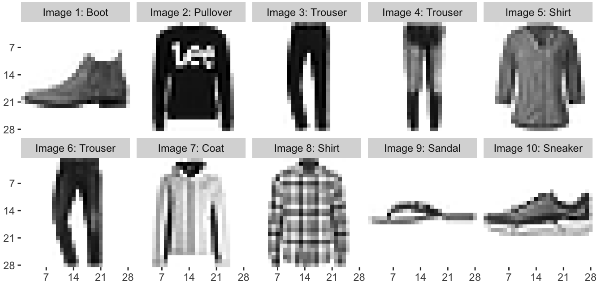

Figure 1: Example of each category images [1]

2.3.3. Image Size

Each image in the Fashion MNIST dataset is 28x28 pixels, making it a relatively low-resolution dataset. This small image size presents a challenge for deep learning models, as it requires them to extract meaningful features from very limited pixel information.

2.3.4. Domain Information

2.3.4.1. Limited Classes of Clothing

The dataset exhibits a limited array of clothing categories, 10 classes in total. Models employed should be suited to classifying a relatively small number of classes efficiently and exhibit a capacity for streamlining classification tasks for a smaller number of classes.

2.3.4.2. Uniform Image Size and Resolution

Fashion MNIST standardised image dimensions at 28x28 pixels in grayscale. The domain limitation of uniform image size necessitates model characteristics that excel at efficiently processing small, low-resolution grayscale images. This calls for model architectures with a focus on efficiency and lightweight design.

2.3.4.3. Lack of Context

The dataset lacks contextual information, which may impact model adaptability to real-world scenarios with diverse backgrounds and lighting conditions. Model characteristics should encompass the ability to recognize clothing items in isolation without undue reliance on contextual cues that are not present in the dataset.

2.3.4.4. Simplified Items

Fashion MNIST simplifies the representation of fashion items. Models must be adept at recognizing these simplified items, and they may not require intricate feature extraction capabilities. Simplicity and efficiency in model characteristics are, therefore, preferred.

2.3.4.5. Intra-Class Variability

The dataset exhibits limited intra-class variability, affecting the model's ability to recognize subtle differences within clothing categories. Models should demonstrate sensitivity to variations in style, texture, and patterns within these categories.

3. Related Work

3.1. Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) have revolutionised image classification, dominating the field through their success in competitions like ImageNet [2]. Their hierarchical feature learning architecture has enabled state-of-the-art accuracy and generalisation. Transfer learning with models such as VGG [3], ResNet [4], and Inception has become commonplace, reducing the need for extensive labelled data and extending the impact of CNNs to various domains. Challenges, including robustness and interpretability, persist, and future directions involve the development of more efficient models and continued exploration of transfer learning across domains, reaffirming CNNs' enduring influence on image classification and beyond [5].

3.2. ShuffleNet and MobileNet

ShuffleNet and MobileNet, lightweight convolutional neural network architectures, have gained prominence due to their efficiency and suitability for resource-constrained applications. ShuffleNet introduces channel shuffling and group convolutions to reduce computational complexity while maintaining competitive performance [6]. MobileNet, on the other hand, leverages depthwise separable convolutions to achieve high efficiency and is well-suited for mobile and embedded devices [7]. These architectures, designed with a focus on computational efficiency, have proven valuable in applications where computational resources are limited, such as mobile devices and edge computing. As a result, they offer promising solutions for image classification tasks in domains with hardware constraints. Their lightweight nature not only enhances inference speed but also minimises resource utilisation, making them particularly appealing for real-time applications.

3.3. Data Augmentation

Data augmentation techniques have emerged as essential tools for enhancing the performance and robustness of neural networks in image classification tasks. Augmentations such as random rotations, flips, and colour variations provide additional training samples, thereby mitigating overfitting and improving model generalisation [8]. Moreover, more advanced techniques, including CutMix and Mixup, have demonstrated their efficacy in creating diverse and informative training data, leading to superior model performance [9]. These strategies are instrumental in overcoming limitations posed by limited training data, a common challenge in various domains. By introducing diversity and variability into the training data, data augmentations enable neural networks to better adapt to complex real-world scenarios and exhibit improved robustness to variations in input data.

4. Approach

Early stopping was systematically applied to each model in our study. The criterion for termination was predicated on test loss rather than test accuracy. While test accuracy may increase, it might not accurately reflect improvements in the model's ability to generalise, as it can obscure potential deteriorations in test loss—a more robust indicator of overall model performance.

4.1. Simple CNN Model

We implemented a simple CNN model with a pooling layer in between two convolutional layers and three fully connected layers. We used Rectified Linear Unit (ReLU) activation functions, and trained the model with the Adam optimizer at a learning rate of 0.001 and a Cross Entropy loss function.

4.2. Transfer Learning with MobileNet and ShuffleNet

4.2.1. Network Architecture Details

ShuffleNet and MobileNet are known for their efficiency and small model size. This is a crucial advantage for the Fashion MNIST dataset as it enables faster training and inference while preventing overfitting.

The small model size is particularly well-suited for Fashion MNIST's domain due to these reasons:

Faster Iteration: Smaller models require less computational resources and time to train. This allows us to iterate through different model configurations and experiments more quickly. This would lead to a more efficient exploration of different techniques and approaches, ultimately accelerating the development process.

Reduced Resource Demands: The lightweight design of ShuffleNet and MobileNet reduces the computational and memory requirements. This is advantageous when working with limited computational resources, which is the case in the scope of this project using the academic settings or on personal machines. Smaller models facilitate experimentation without the need for extensive GPU resources.

Quick Testing: Faster inference times for smaller models make it easier to assess model performance and fine-tune hyperparameters. This speed of testing and evaluation streamlines the optimization process and allows for a more rapid assessment of the impact of different techniques and strategies.

4.2.2. Implementation

We incorporated MobileNet_v3_small and ShuffleNet_v2_x0_5 into our project through the PyTorch library. Our exploration involved the systematic experimentation of training these models, considering an array of factors such as batch size, the status of unfreezing different layers, and the application of data augmentation techniques like CutMix. Our objective was to enhance the performance of these models and unravel potential optimizations.

4.3. Deformable Convolution

While traditional convolution layers use fixed grid-based receptive fields, a deformable convolution layer modifies this by predicting a set of offset values for each location in the input feature map. These offsets learned during training determine how the kernel should be adjusted to capture features outside of the usual receptive field.

We explored the use of deformable convolution layers in our models, aiming to investigate their impact on performance and object localization.

4.4. CutMix Data Augmentation

We applied the CutMix data augmentation technique to our original Fashion MNIST dataset. CutMix combines two data augmentation strategies: Cutout and Mixup to create new augmented samples.

Cutout: Cutout randomly masks out rectangular regions of an image by setting the pixel values within the region to zero or some other predefined value. This helps prevent the model from memorising specific details and encourages it to learn more robust and generalised features.

Mixup: Mixup is a data augmentation technique that combines two images by taking a weighted average of their pixel values, along with their corresponding labels. It encourages the model to learn from a mixture of two different images, making it more resilient to variations and reducing overfitting.

This combination of both data augmentation techniques has shown more effective results than when used individually. [10]

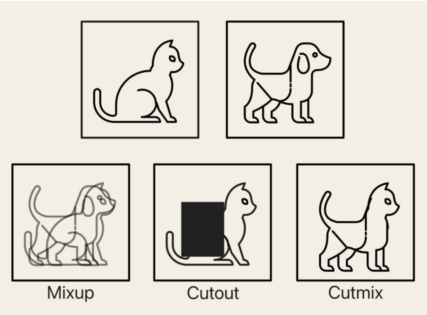

Figure 2: Example of Cutout and Mixup data augmentation being applied to images of a dog and a cat
(Image obtained from https://towardsdatascience.com/cutout-mixup-and-cutmix-implementing-modern-image-augmentations-in-pytorch-a9d7db3074ad)

5. Experiments

5.1. Simple CNN

With a simple CNN, we obtained 89.52% test accuracy, a loss of 0.363 at epoch 27 (early stop triggered). While the accuracy of this simple model is not as high as what could be achieved with more advanced or sophisticated models. The simple CNN still performed well with decent results, given the relatively smaller number of features in the fashion MNIST dataset.

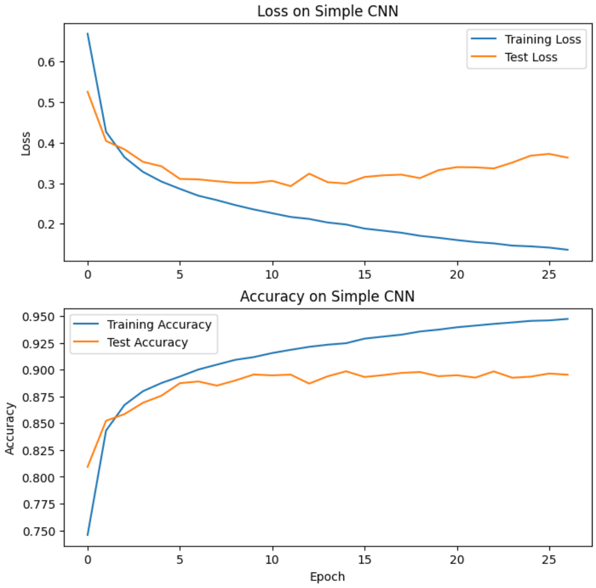

Figure 3: Loss and Accuracy values on Simple CNN Model

5.1.1. CutMix

Comparing the accuracy from both model's epochs with the least test loss, the use of CutMix achieved a slightly worse accuracy, at 89.22% as compared to 89.53% without CutMix.

With the use of CutMix, the Simple CNN model obtained 89.22% test accuracy, with a loss of 0.346, at epoch 40 (no early stopping triggered). It is clear from the graph that both the test loss and test accuracy were far better than the training loss and accuracy, with differences of about 0.6 and 0.2, respectively. As test samples were the original test images and the training samples were the augmented images, the model would have a harder time classifying images. For example, features such as the curve in a sneaker image can be disrupted by CutMix, leading to the model incorrectly classifying it.

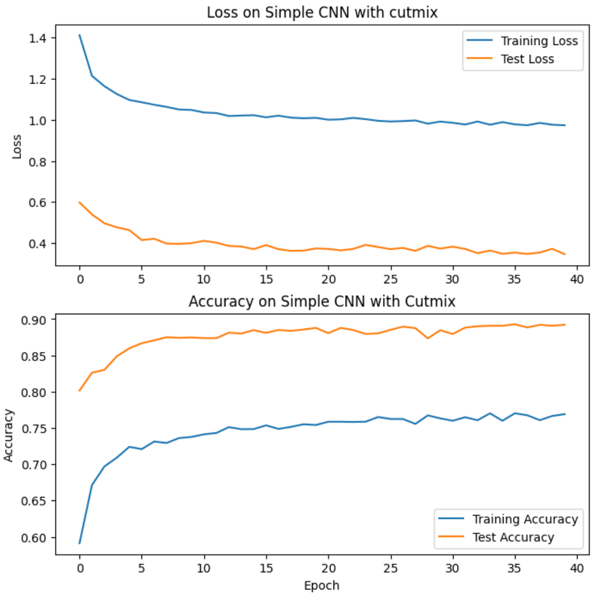

Figure 4: Loss and Accuracy values on Simple CNN Model with Cutmix

5.2. Transfer Learning

5.2.1. ShuffleNet

The selection of the 'shufflenet_v2_x0_5' variant aligns with our pursuit of efficiency and reduced model size for the Fashion MNIST dataset.

5.2.1.1. Unfreeze FC Layer

By keeping most of the pre-trained layers frozen and only modifying the final fully connected layers to match the 10 classes of the Fashion MNIST dataset, we were able to keep the time required for training the model low.

This enabled us to run ShuffleNet with various batch sizes, giving us the following test loss and accuracies (best result is bolded):

Batch size	Best epoch	Test loss	Test accuracy (%)
8	25	1.165	55.65
16	35	1.144	56.89
32	34	1.151	56.86
64	34	1.179	55.71
128	33	1.202	55.47

Table 1: Test loss and accuracies for various batch sizes on ShuffleNet

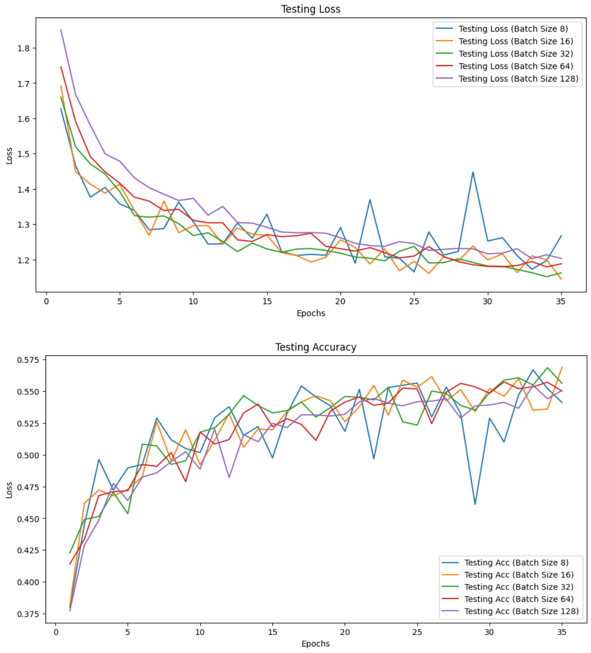

Figure 5: Testing Loss and Accuracy on ShuffleNet on Batch size 8,16,32,64,128.

ShuffleNet with a batch size of 16 appeared to give the best test accuracy, but it did not perform as well as expected. While the model was able to leverage the high-level task-agnostic features it learned during pre-training, training only the final fully connected layer meant that it was unable to adapt well to our specific dataset and did not learn the features of the clothing well.

5.2.1.2. Unfreeze All Layers

Next, by unfreezing all layers, the entire network is now fine-tuned for our target task, including the convolution layers. Retraining the model with the optimal batch size of 16 found from the previous section, ShuffleNet with no frozen layers was more flexible and could learn the intricacies of our clothing classification task.

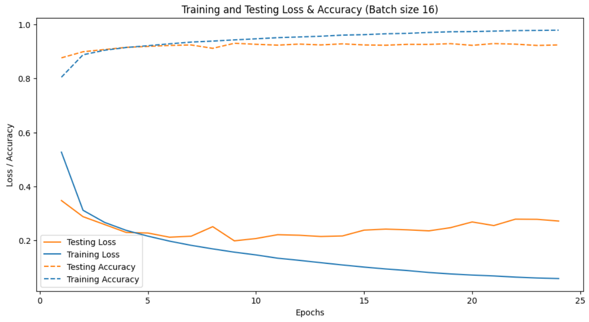

Figure 6: Testing and Training loss and accuracy on a fully unfrozen ShuffleNet (Batch size 16). It achieved 0.198 loss with 93.01% accuracy (Epoch 8)

This significantly increased our test accuracy to 93.01% from 56.89%, a huge leap from ShuffleNet with only 1 unfrozen layer.

5.2.1.3. CutMix

With the use of CutMix for data augmentation, we attempted to improve the robustness of our modified pretrained ShuffleNet, and reduce overfitting by increasing the diversity of our training data. This slightly improved the test accuracy to 93.82%, compared to 93.01% without.

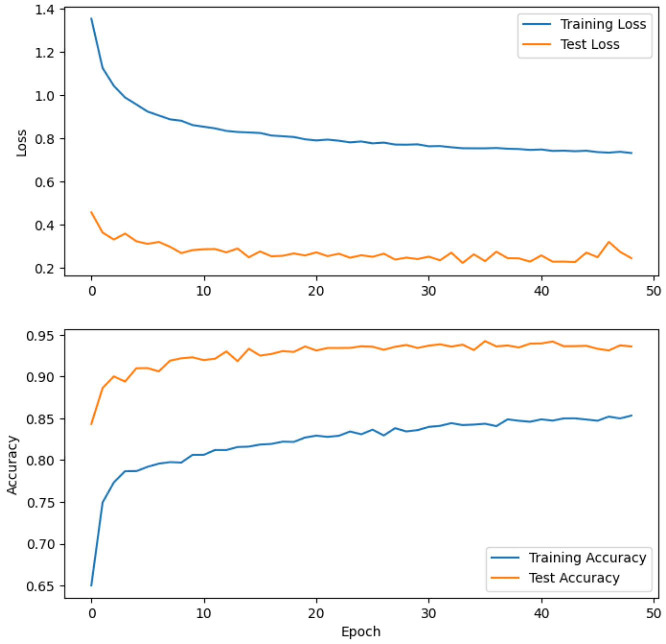

Figure 7: Testing and Training loss and accuracy on a fully unfrozen ShuffleNet (Batch size 16) with CutMix. It achieved 0.223 loss with 93.82% accuracy (Epoch 34)

5.2.2. MobileNet

The selection of the 'mobilenet_v3_small' variant aligns with our pursuit of efficiency and reduced model size for the Fashion MNIST dataset.

5.2.2.1. Unfreeze FC Layer

Similar to ShuffleNet's methodology, we repeated the same for MobileNet.

Batch size	Best epoch	Test loss	Test accuracy (%)
8	33	1.126	58.36
16	40	1.061	60.59
32	33	1.159	56.63
64	39	1.142	57.01
128	37	1.189	56.01

Table 2: ShuffleNet results with only FC layer unfrozen

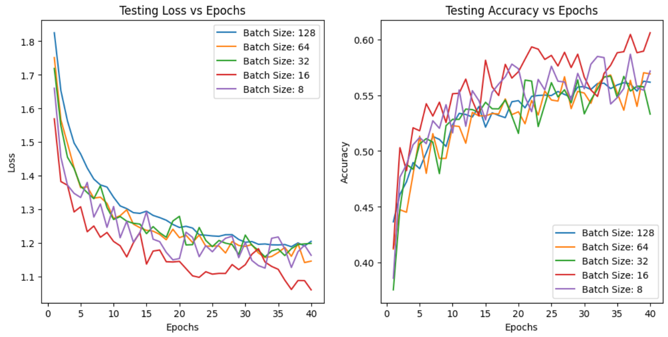

Figure 8: Test loss and accuracies for ShuffleNet on Batch size 8,16,32,64,128.

5.2.2.2. Unfreeze All Layers

Similar to ShuffleNet's methodology, we retrained the model with the optimal batch size of 16. This significantly increased our test accuracy to 92.9% from 60.59%.

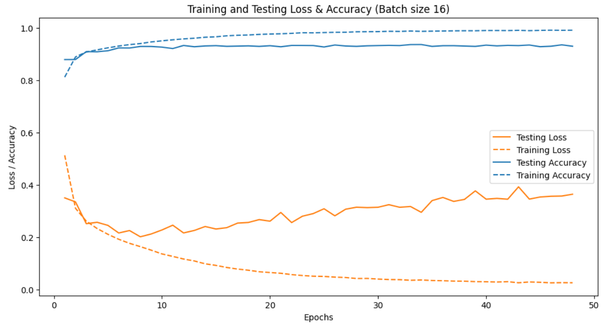

Figure 9: Testing and Training loss and accuracy on a fully unfrozen MobileNet (Batch size 16). It achieved 0.202 loss with 92.9% accuracy (Epoch 7)

5.2.2.3. CutMix

With the use of CutMix, the test accuracy slightly improved to 94.19% from 92.9%, without.

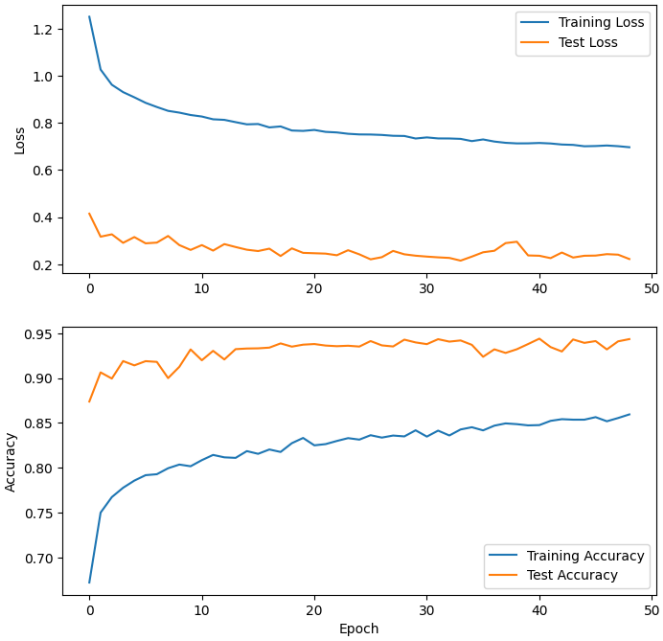

Figure 10: Testing and Training loss and accuracy on a fully unfrozen MobileNet (Batch size 16) with CutMix. It achieved 0.216 loss with 94.19% accuracy (Epoch 34)

5.3. Deformable Convolution

Improving from our simple CNN, we adopted a new model architecture to explore the use of deformable convolution layers. With 3 regular convolution layers, 2 deformable convolution layers, a pooling layer and a fully connected layer, the deformable convolutional neural network showed similar results to our simple CNN, with about 90.92% in test accuracy at its lowest loss epoch.

The ability of deformable convolution layers to be able to capture fine-grained spatial information and object deformation may be of limited usefulness on this dataset.

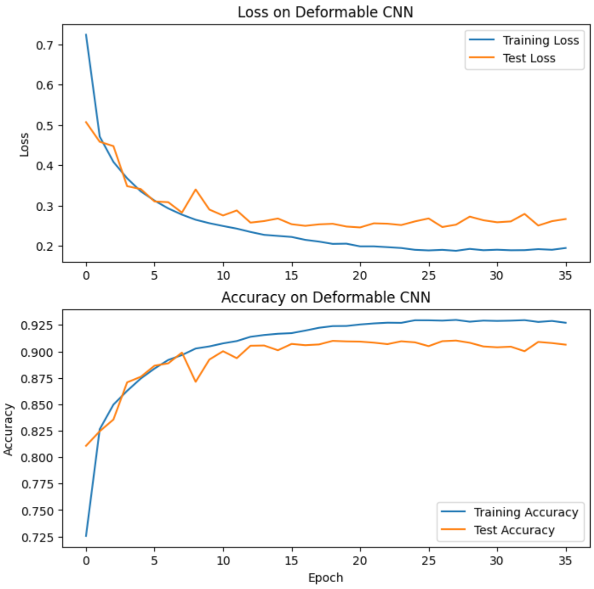

Figure 11: Loss and Accuracy values on Deformable CNN. It achieved 0.246 loss with 90.92% accuracy (Epoch 21)

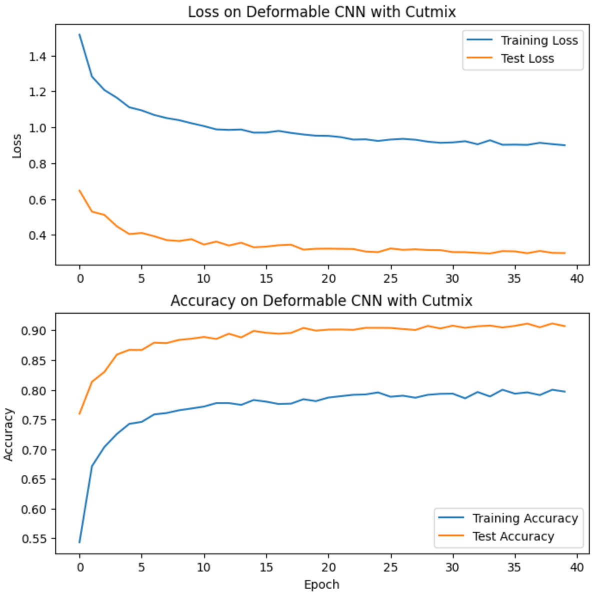

Figure 12: Loss and Accuracy values on Deformable CNN with CutMix. It achieved 0.296 loss with 90.79% accuracy (Epoch 34)

5.4. CutMix Data Augmentation

Model	Without CutMix		CutMix	
	Test loss	Test Accuracy (%)	Test loss	Test Accuracy (%)
Simple CNN	0.293	89.53	0.346	89.22
Deformable CNN	0.246	90.92	0.296	90.79
shufflenet_v2_x0_5 (unfreeze all layers)	0.198	93.01	0.223	93.82
mobile_v3_small (unfreeze all layers)	0.202	92.90	0.216	94.19

Table 3: Comparison of model performance with and without the use of CutMix (The top results are highlighted in green.

The Fashion MNIST dataset is a relatively simple and small-scale dataset, with the objects within it being well-defined, structured, and typically appearing in standard positions and orientations.

It is likely that CutMix was limited in its ability to introduce meaningfully diverse samples. Factors that CutMix improves in models, such as spatial coherence and understanding would not be as relevant given the dataset.

It is interesting to note that the test loss experienced an increase when CutMix was added to each model, with a smaller increase for the larger pretrained models compared to the Simple and Deformable CNN models. A possible explanation could be due to the larger architecture of the pretrained models, resulting in the 'bad' weights being updated slower.

6. Summary and Discussion

Model	Epoch	Test Loss	Test Accuracy (%)
Simple CNN	12	0.293	89.53
Simple CNN (with CutMix)	40	0.346	89.22
Deformable CNN	21	0.246	90.92
Deformable CNN (with CutMix)	34	0.296	90.79
shufflenet_v2_x0_5 (unfreeze FC layer)	35	1.144	56.89
shufflenet_v2_x0_5 (unfreeze all layers)	9	0.198	93.01
shufflenet_v2_x0_5 (unfreeze all layers + CutMix)	34	0.223	93.82
mobile_v3_small (unfreeze FC layer)	40	1.061	60.59
mobile_v3_small (unfreeze all layers)	8	0.202	92.90
mobile_v3_small (unfreeze all layers + CutMix)	34	0.216	94.19

Table 4: Summary of results across all models tested (Top 4 model results are highlighted in green)

Note: shufflenet_v2_x0_5 (unfreeze FC layer) and mobile_v3_small (unfreeze FC layer) used batch size 16's results as it yielded the best result. In the table, test accuracy was taken from the epoch with the lowest test loss.

Transfer Learning with pretrained models had the highest test accuracies. Pretrained models like ShuffleNet or MobileNet have already learnt to capture abstract features from a large dataset, giving them a strong ability to recognise patterns, textures and structures in a range of vision tasks. Following that with specific training on the Fashion MNIST dataset meant that it could also adapt well to our specific classification task. Despite this, simple CNN and deformable convolution showed decent performances with accuracies of around 90%. This shows the potential of tailored architectures for image classification tasks, even without the reliance on pretrained weights.

CutMix did not have a significant impact on test accuracy. The Fashion MNIST dataset is highly regular with objects in similar positions and orientations. Since the baseline model already performed well without showing signs of overfitting, data augmentation techniques introducing diverse samples offered little improvement to the model's accuracy.

In summary, while simple models may be sufficiently performant for small and regular datasets like Fashion MNIST, transfer learning has shown stronger potential with their sophisticated architecture, with a good starting point from their pretrained weights. CutMix, however, showed mixed results, and did not significantly improve the performance of all our models.

7. References

1. Ultralytics, glenn-jocher, and sergiuwaxmann (2023). "Fashion-MNIST, Fashion-MNIST - Ultralytics YOLOv8 Docs." Available at: https://docs.ultralytics.com/datasets/classify/fashion-mnist/#sample-images-and-annotations (Accessed: 08 November 2023).
2. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). "ImageNet Classification with Deep Convolutional Neural Networks." In Advances in Neural Information Processing Systems.
3. Simonyan, K., & Zisserman, A. (2014). "Very Deep Convolutional Networks for Large-Scale Image Recognition." arXiv preprint arXiv:1409.1556.
4. 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.
5. 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 <0.5 MB model size." arXiv preprint arXiv:1602.07360.
6. Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). "ShuffleNet: An extremely efficient convolutional neural network for mobile devices." In Proceedings of the IEEE conference on computer vision and pattern recognition.
7. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). "MobileNets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861.
8. Shorten, C., & Khoshgoftaar, T. M. (2019). "A survey on image data augmentation for deep learning." Journal of Big Data, 6(1), 60.
9. Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2018). "mixup: Beyond empirical risk minimization." In International conference on learning representations.
10. S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features." arXiv, Aug. 07, 2019. doi: 10.48550/arXiv.1905.04899.