Report: Image Classification with CNN using Fashion-MNIST Dataset

Name: Rishabh Chandel

Date: 6 September

Course/Subject: AI/ML Engineer

1. Introduction

In this assignment, we implemented and compared machine learning models for image classification using the **Fashion-MNIST dataset**. The dataset consists of grayscale images (28×28 pixels) of 10 different types of fashion items such as T-shirts, trousers, and sneakers. The main objective was to build models that accurately classify these images into their respective categories.

We first built a baseline model using a **Multilayer Perceptron (MLP)** and then extended it by using a **Convolutional Neural Network (CNN)**, which is specifically designed for image data. Additionally, we improved CNN's performance by incorporating **data augmentation** and **batch normalization** techniques.

2. Dataset Description

The Fashion-MNIST dataset contains:

- **60,000 training images** and **10,000 test images**.
- Each image is a 28×28 grayscale image.
- There are **10 classes**, including items such as T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, and Ankle boot.

The data was normalized to have pixel values between 0 and 1 to improve training efficiency.

---

3. Methodology

Data Visualization

A random selection of images was plotted to understand the dataset better. Each image's label was displayed to get a sense of class distribution and sample quality.

Model Building

MLP Model:

- The input layer flattens the image.
- Two hidden layers with 128 and 64 neurons, using ReLU activation.
- Output layer with softmax activation for classification.

CNN Model:

- Two convolutional layers with filters of sizes 32 and 64.
- Each followed by max-pooling layers to downsample spatial features.
- Dropout layers were added to reduce overfitting.
- Fully connected layers to map features to class probabilities.

Improved CNN Model:

- Added **batch normalization** to stabilize and speed up training.
- Used **data augmentation** to create more diverse samples during training.

Training

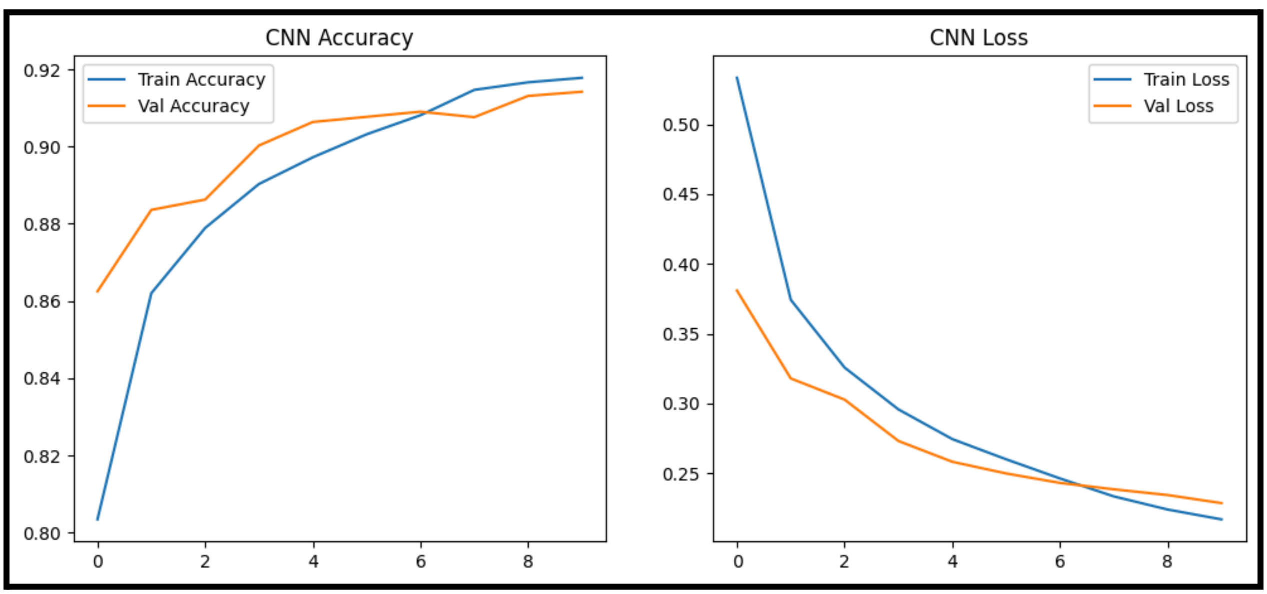
- All models were compiled using the Adam optimizer and sparse categorical crossentropy loss.
- The models were trained for 10 to 15 epochs with a validation split to monitor overfitting.

---

4. Results

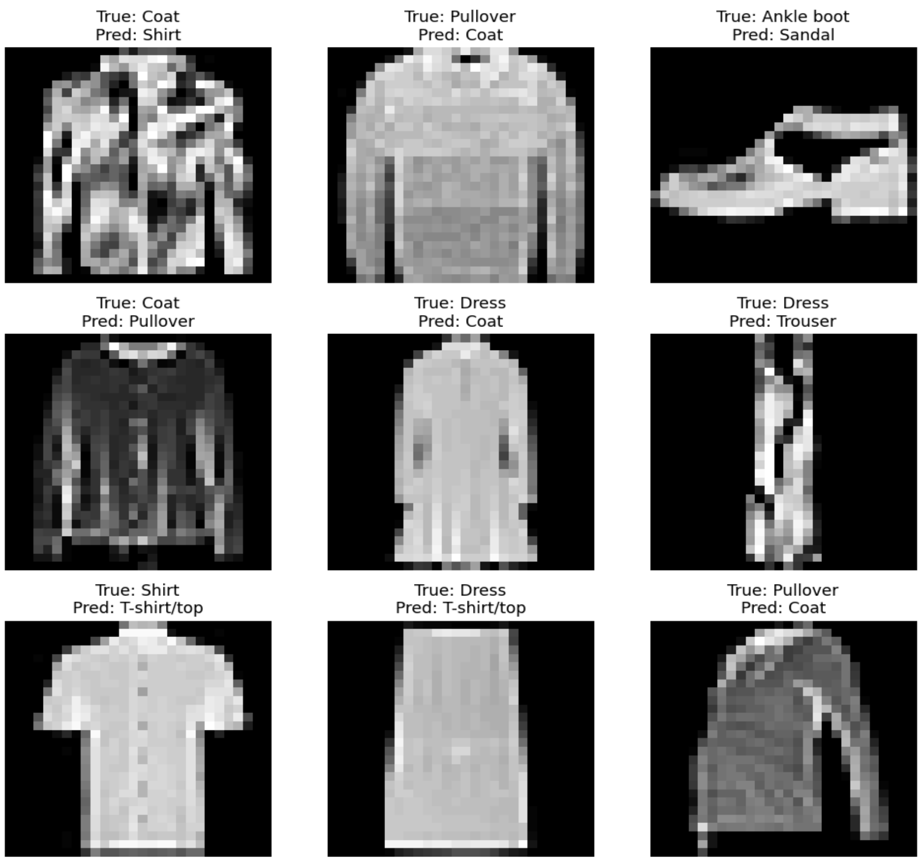
Model Performance

Model	Test Accuracy
MLP	~87.6%
CNN	~91.0%
CNN with Augmentation	~87.0%



Classification Report

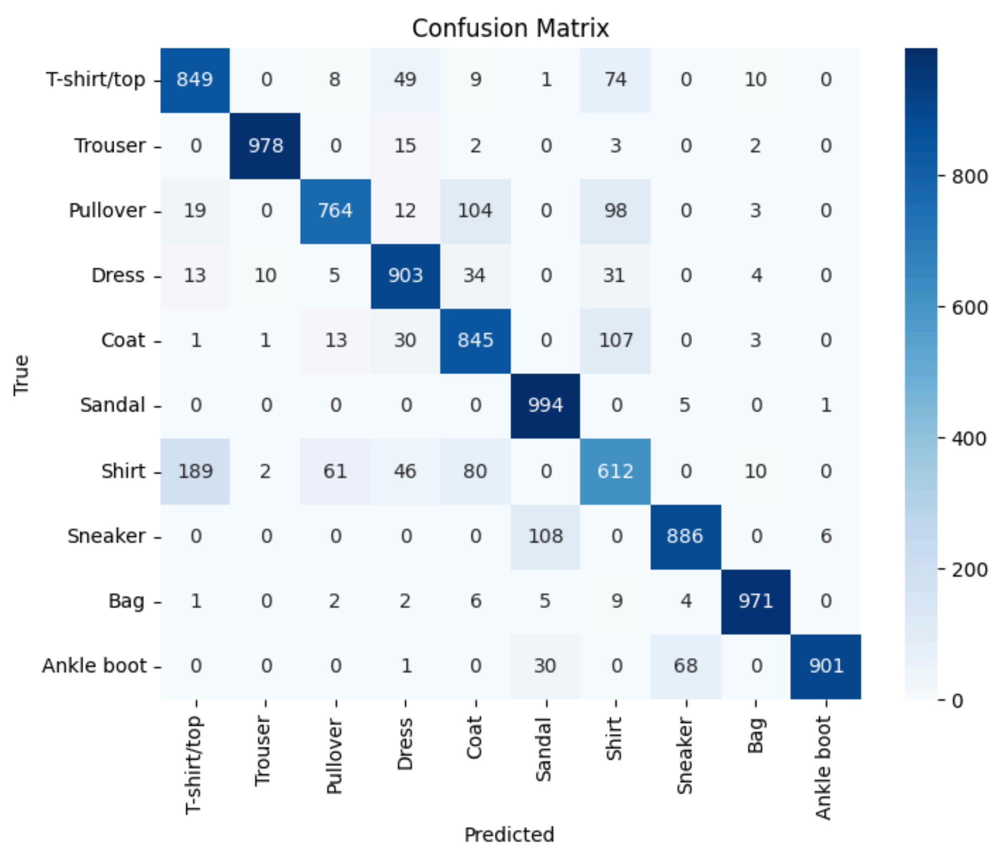
- ❖ Classes like **Trouser, Sneaker, Bag** had high precision and recall (>95%), showing that the models classify these items well.
- ❖ Classes like **Shirt and Pullover** had lower accuracy (~60–76%), likely due to visual similarities.



Some of the Miss Classified Images

Confusion Matrix

The confusion matrix showed that most misclassifications happened between similar categories, confirming that visual resemblance makes certain classes harder to distinguish.



---

5. Discussion

1. The **CNN significantly outperformed the MLP**, highlighting its advantage in extracting spatial features from image data.
2. **Data augmentation** provided more training examples, helping prevent overfitting, though tuning the parameters further could improve accuracy.
3. **Batch normalization** improved model stability and helped training converge faster.

4. Certain classes were more challenging due to their appearance, which suggests that future models could benefit from deeper architectures or ensemble methods.

---

6. Conclusion

This assignment demonstrated that CNNs are more effective than MLPs for image classification tasks due to their ability to learn spatial features. Incorporating data augmentation and batch normalization further improved the performance, although some categories remained difficult to classify accurately.

In future work, experimenting with deeper networks, dropout rates, and other optimization techniques would likely yield better results. The methods used here are widely applicable in real-world scenarios, such as automated image tagging, fashion recommendation systems, and medical image diagnostics.

---

7. References

- Fashion-MNIST Dataset – https://github.com/zalandoresearch/fashion-mnist
- TensorFlow and Keras Documentation – https://www.tensorflow.org/api_docs

---