PROJECT REPORT: FASHION-MNIST CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS (CNN)

TABLE OF CONTENTS

EXECUTIVE SUMMARY

This project report outlines the development and assessment of a Convolutional Neural Network (CNN) designed to categorise images from the Fashion-MNIST dataset. Fashion-MNIST is a collection of article images from Zalando, featuring 60,000 samples for training and 10,000 for testing. Every sample is a 28x28 pixel grayscale image, linked to one of 10 different labels.

The objective was to build and train a CNN model to recognise and classify different fashion products from low-resolution images accurately. The report covers data preprocessing, model architecture design, training process, evaluation results, and conclusions drawn from the findings.

1 INTRODUCTION

The advent of deep learning has revolutionised the field of computer vision, enabling computers to outperform humans in tasks such as image recognition, object detection, and classification. At the forefront of this revolution are Convolutional Neural Networks (CNNs), a class of deep neural networks that are particularly powerful for analysing visual imagery.

1.1 Convolutional Neural Networks (CNNs)

CNNs are designed to automatically and adaptively learn spatial hierarchies of features from input images. These networks are composed of multiple layers of convolutional filters that apply learned filters to input data, capture spatial features, and reduce the dimensions of the data, making the network less sensitive to the exact location of features. The convolutional layers are typically followed by pooling layers that further reduce the dimensionality of the data, and fully connected layers that perform classification based on the features extracted by the convolutional and pooling layers.

1.2 Image Classification

Image classification is a pivotal task in computer vision where the objective is to assign an input image one label from a fixed set of categories. This is a challenging task due to the variability in images, including changes in angle, size, and lighting. The success of CNNs in image classification tasks across various domains has been significant, making them the go-to model for many applications that involve understanding content within images.

1.3 Objectives

This project explores the development of a CNN model tailored to classify images from the Fashion-MNIST dataset. The aim is to demonstrate the model's learning ability and classification accuracy, as well as to understand its performance across different categories of fashion items.

Data Source: https://github.com/zalandoresearch/fashion-mnist

2 METHODOLOGY

2.1 Data Acquisition and Preprocessing

The Fashion-MNIST dataset was loaded through TensorFlow/Keras datasets module. Training and test datasets were loaded separately to maintain the integrity of the model evaluation process.

Normalisation

Each image's pixel values, originally ranging from 0 to 255, were normalised to a [0,1] scale. This normalisation process is a crucial step in deep learning as it ensures that the model processes the input data within a standardised range, which helps in speeding up the learning process and achieving better performance.

One-hot Encoding

The categorical labels associated with the images were transformed using one-hot encoding. This technique converts the categorical integer labels into a binary matrix representation that is more suitable for use with categorical classification. This process is essential for the model to interpret the labels correctly during the training phase.

2.2 Model Architecture

A CNN model was architected using the Keras Sequential API. The Sequential API allows stacking layers in a linear manner, which simplifies the model construction.

Convolutional Blocks

The core of the model consisted of two convolutional blocks. The first block contained 64 filters, while the second block had 128 filters. Each filter in the convolutional layers was of size 3x3. The role of these filters is to extract various features from the input images at different levels of abstraction.

Pooling and Dropout Layer

Each convolutional block was followed by a max-pooling layer of size 2x2, which serves to reduce the spatial dimensions of the output from the convolutional layers. Dropout layers with a rate of 0.4 were introduced after each pooling operation to reduce overfitting.

Flattening Layer

Post the convolutional blocks, a flattening layer was included to convert the 2D feature maps into a 1D feature vector. This transformation is necessary to feed the convolutional network's output into the dense layers for further processing.

Dense Layers

The architecture further included two dense layers, each with 1024 neurons. These layers are aimed at enabling the network to learn non-linear combinations of the high-level features extracted from the convolutional layers.

Output Layer

The final layer of the model was a dense layer with 10 units corresponding to the 10 categories of the Fashion-MNIST dataset. The softmax activation function was employed in this layer to obtain the probability distribution over the 10 classes.

2.3 Training

The model compilation was done using categorical crossentropy as the loss function, which is appropriate for multi-class classification problems.

The training was conducted over 5 epochs, and a batch size of 32 was selected to balance the trade-off between computational efficiency and the ability to converge to the global minimum of the loss function. Before each epoch, the training data was shuffled to ensure that the model does not learn anything from the order of the samples.

3 EVALUATION AND RESULTS

3.1 Model Training and Learning Curve

The model was subjected to a training regimen spanning 5 epochs. Here is a detailed analysis of the learning curve based on the provided training output:

1. Initial Training Performance (Epoch 1):
   - The training began with a notable loss of 0.5283 and an accuracy of approximately 80.73%. This initial epoch served as the baseline, from which improvements were measured.
2. Rapid Improvement (Epoch 2):
   - By the second epoch, the model's performance had improved significantly, with accuracy increasing to approximately 85.98% and loss decreasing to 0.4043. This rapid improvement indicated that the model was quickly learning from the training data.
3. Peak and Subsequent Performance:
   - The optimal performance was observed around the fourth epoch, where the accuracy peaked at 86.18%. However, post the fourth epoch, a slight deterioration in model performance was observed. By the final epoch, the accuracy slightly decreased to 85.91%, and the loss varied, ending at approximately 0.4156. This slight decrease in performance towards the end could suggest the beginning of overfitting or the model reaching its learning capacity under the current architecture and parameters.

3.2 Quantitative Metrics

The model achieved an accuracy of 87.04% on the test dataset, which is indicative of a high level of performance in image classification tasks. Other relevant metrics were computed to gain a deeper understanding of the model's performance across all classes, which are summarised in the table 1.

Metric	Value
Accuracy	87.04%
Precision	0.87
Recall	0.87
F1-Score	0.87

Table 1: Classification Model Performance Metrics

Interpretation of Metrics:

- Accuracy (87.04%): This metric indicates the proportion of correct predictions made by the model out of all predictions. An accuracy of approximately 87.04% signifies that the model is generally effective at classifying the Fashion MNIST images.
- Precision (0.87): Precision measures the model's accuracy in labeling an image as belonging to a particular class. A precision of 0.87 indicates that when the model predicts an image as belonging to a certain class, it is correct about 87% of the time. This metric is particularly important in scenarios where the cost of a false positive is high.
- Recall (0.87): Recall measures the model's ability to identify all instances of a particular class. A recall of 0.87 suggests that the model successfully identifies 87% of all actual instances of each class. This is crucial in situations where missing an actual instance (false negative) is costly.
- F1-Score (0.87): The F1-score is the harmonic mean of precision and recall, providing a single measure that balances both the concerns of precision and recall. An F1-score of 0.87 indicates a balanced performance between precision and recall, suggesting the model is robust in terms of both false positives and false negatives.

The precision, recall, and F1-score, each reported as approximately 0.87, indicate that the model maintains a good balance between sensitivity (recall) and specificity (precision). This balance is important for a model's applicability in real-world scenarios, ensuring it does not disproportionately favour avoiding false positives over false negatives or vice versa. The quantitative metrics suggest that the model is quite capable but may still benefit from further optimization or data augmentation to push the performance even higher, especially in classes where it might be underperforming.

3.3 Confusion Matrix Analysis

The confusion matrix offers a comprehensive view of the model's performance across all classes, providing both the number of correct predictions (along the diagonal) and specific misclassification patterns (off-diagonal elements). This analysis allows us to understand not just the overall accuracy but also how the model performs for each individual class.

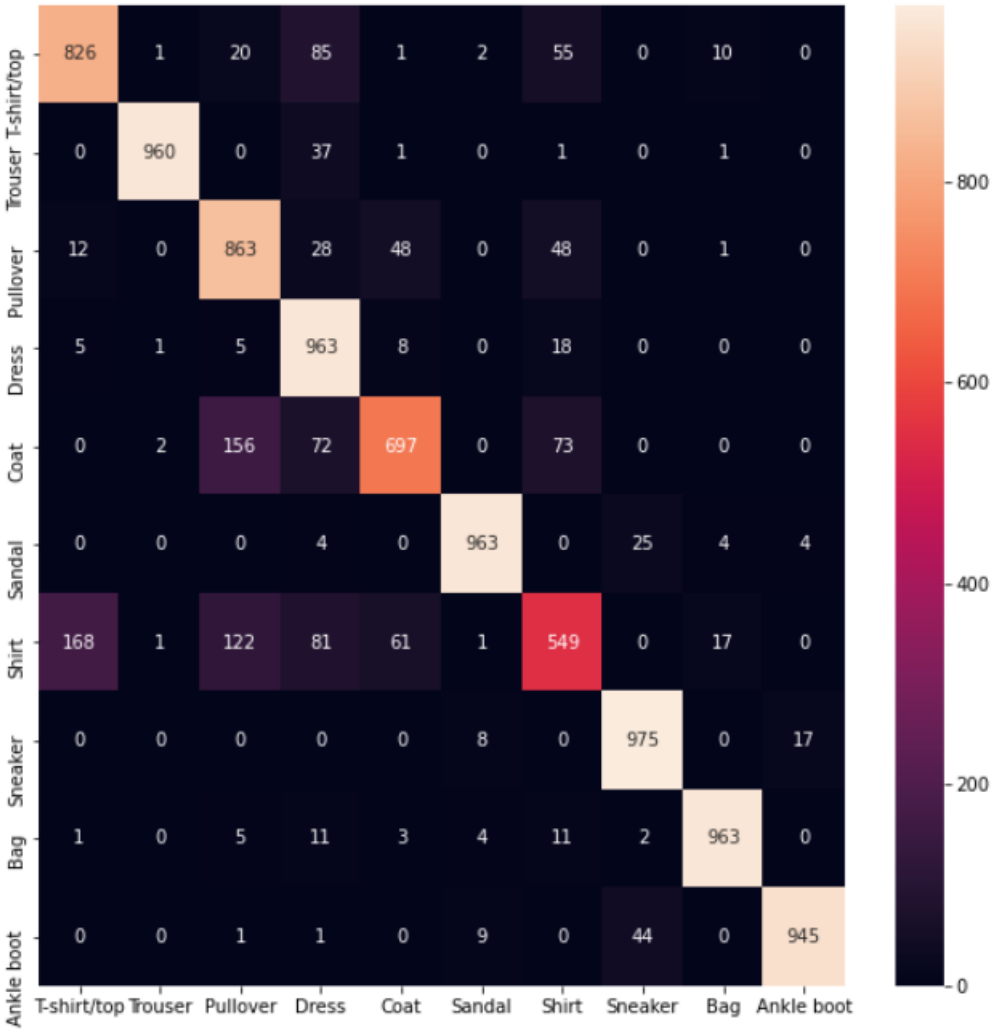

Figure 1: Confusion Matrix

High Performance Classes:

Several classes exhibited high true positive rates, indicating the model's exceptional ability to classify them correctly:

- **Trouser:** Almost exclusively classified correctly, showing the model's strong capability in distinguishing this class from others.
- **Bag:** Another class with high classification accuracy, indicating the model's effectiveness in capturing features unique to bags.
- **Ankle boot:** Showed a high true positive rate, reflecting the model's proficiency in recognizing this footwear class.

These categories had the highest diagonal values in the confusion matrix, signifying a substantial number of correct predictions. The high performance on these classes suggests that the model is very effective in capturing and utilizing distinctive features that set these items apart from other categories.
Classes with Higher Misclassification Rates:

Conversely, certain classes demonstrated higher rates of misclassification, as evidenced by their confusion matrix patterns:

- Shirt: This class had a noticeably higher misclassification rate, particularly with classes like T-shirt/top and Coat. The lower diagonal value and higher off-diagonal values for this class indicate frequent confusion, suggesting that the model struggles to distinguish shirts from other similar top-wear. This might be due to the inherent visual similarities between these garments, such as shape and size.
- Pullover and Coat: These classes also showed some confusion, particularly with each other and with other similar classes, indicating areas where the model's classification capabilities could be enhanced.

Analysis of Misclassifications:

The higher off-diagonal numbers for certain pairs of classes (e.g., Shirt with T-shirt/top and Pullover with Coat) suggest that the model sometimes confuses items with similar characteristics. Understanding these patterns can direct efforts to collect more discriminative data for these classes or to engineer features that help distinguish between them.

4 CONCLUSION

In assessing the performance of a Convolutional Neural Network (CNN) on the Fashion-MNIST dataset, a commendable overall test accuracy of 87.04% was observed. This success rate highlights the CNN's capability to effectively discern and classify complex patterns within grayscale fashion images.

The model's precision was particularly striking in the classification of 'Trouser', 'Bag', and 'Ankle boot', suggesting that features of these categories are well-captured by the network. However, the classification of 'Shirt' items emerged as a challenge, indicating a potential confusion with other similar clothing items.

The outcomes of this project are encouraging for the application of CNNs in image-based classification within the fashion industry.

Future work may involve fine-tuning the model, employing techniques to mitigate overfitting, and experimenting with deeper architectures. By addressing these areas, there is a clear pathway to elevate the performance of CNNs.