Fashion-MNIST Image Classification using CNN

Faizan Wali Bhutto
fbhutto@depaul.edu

March 15, 2025

Abstract

Deep learning has greatly advanced image classification, enabling models to recognize complex patterns with high accuracy. This project implemented a **Convolutional Neural Network (CNN)** to classify images from the **Fashion-MNIST dataset**, which contains grayscale images of clothing items in 10 categories. The goal was to build a model that accurately distinguishes different clothing types while generalizing well to unseen data.

My approach involved **data preprocessing (normalization and reshaping), model design, training, and evaluation**. The CNN architecture included **two convolutional layers, max pooling layers, and fully connected dense layers** optimized for classification. The model was trained using the **Adam optimizer** with **sparse categorical crossentropy loss** and evaluated on a separate test set.

The final model achieved a **test accuracy of 90.73%**, closely matching the validation accuracy, confirming strong generalization. Training and validation loss curves indicated **minor overfitting beyond epoch 6**, but overall, the model effectively classified the dataset. Analyzing misclassified examples revealed that **visually similar categories (e.g., shirts vs. coats, sneakers vs. sandals)** were the most challenging to differentiate.

The findings highlight the effectiveness of CNNs in image classification tasks and suggest potential improvements, including data augmentation, dropout layers, and fine-tuning the model architecture. The results confirm that CNNs can achieve high accuracy on Fashion-MNIST with a relatively simple architecture, making them a powerful tool for real-world applications in image recognition.

1 Introduction

Image classification is a fundamental task in computer vision, with applications in fields like autonomous driving (for object and pedestrian recognition), medical imaging (detecting anomalies in scans), and retail (product image categorization). Recent advances in deep learning, especially Convolutional Neural Networks (CNNs), have dramatically improved image recognition performance. Traditionally, image classification relied on manual feature extraction and classical machine learning algorithms; however, those methods often struggle to capture complex, high-level patterns in images.

This project explores the use of **Convolutional Neural Networks (CNNs)** to classify images from the **Fashion-MNIST dataset**, a dataset specifically designed to serve as a more challenging alternative to the traditional MNIST handwritten digit dataset. Unlike MNIST, Fashion-MNIST includes grayscale images of fashion items such as shirts, dresses, sneakers, and boots, which share visually similar patterns that present unique challenges for classification algorithms. As such, this dataset provides an excellent benchmark for evaluating the performance and generalization capabilities of deep learning models.

My approach was to design and train a CNN inspired by the **LeNet-5** architecture, known for its effectiveness on image tasks. We systematically preprocessed the data, trained the model, and evaluated its performance on unseen test data. By examining both correct and incorrect predictions, we identified where the model struggles (particularly with visually similar categories) and considered strategies for improvement.

The main contributions of this work include:

- Implementing and evaluating a basic CNN architecture specifically tailored for fashion image classification.

- Providing clear visualizations and analysis of the training process to highlight model behavior, including possible signs of overfitting.
- Investigating common misclassification patterns and discussing practical strategies to enhance performance.

  Through this project, I tried to demonstrate the practicality and limitations of CNNs in image classification and highlight areas where further research or fine-tuning can lead to even better results.

2 Related Work

Convolutional Neural Networks have become the dominant approach for image classification due to their ability to automatically learn hierarchical feature representations. The foundational LeNet-5 CNN by LeCun et al. (1998) was one of the earliest successes, achieving an error rate around 0.8% on the MNIST digit dataset, matching or beating classical methods like Support Vector Machines and k-Nearest Neighbors, while also reducing the need for manual feature engineering. This early work established CNNs as a powerful alternative to traditional algorithms for vision tasks.

The field progressed rapidly with deeper networks. AlexNet (Krizhevsky et al., 2012) revolutionized large-scale image classification by winning the ImageNet competition: it lowered the top-5 error from 26% (achieved by traditional approaches) to 15%. AlexNet introduced techniques like ReLU activations for faster training and dropout layers for regularization, which have since become standard in deep learning. Likewise, Cireșan et al. (2012) demonstrated that an ensemble of deep CNNs (a multi-column network) could reach near-human performance on MNIST (¿99.7% accuracy). These breakthroughs solidified CNNs' superiority for vision tasks and influenced my choice to use a CNN for Fashion-MNIST.

Fashion-MNIST, introduced by Xiao et al. (2017), was proposed to benchmark machine learning algorithms on a problem more difficult than MNIST. They showed that traditional classifiers (like logistic regression, decision trees, SVMs) only achieve about 80–85% accuracy on Fashion-MNIST, whereas CNN-based methods can surpass 90%. More recent work by Nocentini et al. (2022) used multiple CNNs in an ensemble, achieving about 94% accuracy on Fashion-MNIST – highlighting that even relatively simple CNN architectures can yield high performance on this dataset.

Studies comparing CNNs with traditional machine learning methods further motivated my choice of methodology. Hoang (2018) compared CNNs with traditional classifiers (SVM, k-NN, Random Forest, Decision Trees) on Fashion-MNIST, reporting that CNNs consistently achieved higher accuracy (approximately 89-90%) compared to the best-performing traditional method (SVM with HOG features at around 86.5%). In another study, Agarap (2018) attempted to combine CNNs with an SVM classifier, but found minimal improvement, concluding that CNNs' learned features are the primary reason for their superior performance rather than the choice of classifier. These related works guided my project by underscoring that a stand-alone CNN is a sensible and effective choice for Fashion-MNIST. I build on this prior research by using a straightforward CNN architecture and focusing on analyzing its performance and failure cases, rather than integrating complex ensemble or hybrid methods.

3 Preliminary/Background

A **Convolutional Neural Network (CNN)** is a specialized neural network architecture designed for image data. CNNs typically consist of three main types of layers:

- **Convolutional layers:** Extract spatial patterns from images by applying small, learnable filters to detect features such as edges, textures, and shapes.
- **Pooling layers:** Reduce the dimensions of feature maps, helping the network learn invariant features and reduce computational complexity. A common pooling method is max pooling, which selects the strongest activation from a local region.
- **Fully connected (dense) layers:** Use the extracted features to classify images. These layers connect every neuron from the previous layer to every neuron in the next, producing final predictions. They often utilize activation functions like ReLU for non-linear transformations and softmax for classification.

Fashion-MNIST dataset: This dataset, introduced by Xiao et al. (2017), consists of 70,000 grayscale images of clothing items, each 28×28 pixels. It contains 10 distinct categories: T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, and Ankle boot. Fashion-MNIST serves as a more challenging alternative to the classic handwritten digit MNIST dataset due to visual similarities between categories.

Optimization and Training Concepts: CNN training involves minimizing a **loss function** to measure prediction errors. A common loss function for classification tasks is sparse categorical crossentropy, which compares predicted labels against true labels. Training typically uses an optimization algorithm like Adam, known for dynamically adjusting learning rates for effective training. Important training parameters include the number of epochs (full passes through the data) and batch size (the number of examples per weight update), both of which significantly influence model performance.

Assumptions: In this project, I assumed that the provided dataset is properly labeled and representative of each class. I also assumed that **normalizing** pixel values (scaling from 0–255 to 0–1) would improve training stability and convergence speed. Additionally, a simple CNN architecture was intentionally selected to balance accuracy and computational efficiency, allowing clear insights into model performance and generalization. These considerations guided the choice of the architecture, training strategy, and evaluation methods.

4 Methodology

Dataset and Preprocessing:

The dataset used in this project is the Fashion-MNIST, consisting of 70,000 grayscale images (28x28 pixels) divided into ten distinct classes of clothing. The dataset was split into 60,000 training and 10,000 testing examples provided by TensorFlow's built-in datasets. As a preprocessing step, all images were **normalized**. That is, each pixel intensity (originally 0 to 255) was scaled to a 0 to 1 range. This normalization helps stabilize and speed up network training. Additionally, each image, originally of shape 28×28 (single channel), was **reshaped** to 28×28×1 to explicitly include the channel dimension, as required by CNN input layers.

CNN architecture:

A simple and computationally efficient CNN architecture inspired by the classic **LeNet-5 model** was implemented. This architecture includes the following layers:

- **Input Layer:** Accepts images of size (28×28×1), representing grayscale fashion items.
- **First Convolutional Layer:** Applies 32 convolutional filters (3×3 kernel) with ReLU activation, capturing basic visual features such as edges and textures.
- **First Max Pooling Layer:** A 2×2 max-pooling operation reduces spatial dimensions, keeping essential features while reducing computation.
- **Second Convolutional Layer:** Uses 64 convolutional filters to extract more complex patterns from the feature maps obtained from the first convolution.
- **Second Max-Pooling Layer:** Further reduces the dimensionality of feature maps.
- **Flatten Layer:** Converts the 2D feature maps into a 1D vector, preparing data for fully connected layers.
- **Dense Layer (Fully Connected Layer):** A fully connected layer with 128 neurons and ReLU activation helps the model learn intricate patterns within the extracted features.
- **Output Layer:** A dense layer with 10 neurons (one for each category), using a **softmax activation function** to produce class probabilities for classification.

The CNN model was implemented using the Keras API within the TensorFlow framework.

Training Procedure:

The model was trained using the following parameters:

- **Optimizer:** Adam optimizer was chosen due to its effectiveness in adaptive learning rate management, accelerating convergence.
- **Loss Function:** Sparse categorical crossentropy was selected as it suits multi-class classification tasks with integer labels.
- **Hyperparameters: Batch size** of 64 was selected as a balance between training speed and stable convergence. **For Epochs,** the model was trained for 10 epochs, sufficient to achieve good accuracy while also demonstrating clear trends regarding potential overfitting.

Evaluation and Analysis:

Upon completing training, the model's performance was assessed on the test dataset. Metrics such as accuracy and loss were evaluated to determine how well the model generalized to unseen data. Additionally, predictions were visually analyzed, and misclassified examples were investigated to identify challenging cases, such as visually similar clothing items. This analysis provided insights into areas of potential improvement, including more sophisticated regularization techniques, data augmentation, or further fine-tuning.

The methodology I used was intentionally straightforward, maintaining simplicity while ensuring that all key deep learning concepts and requirements were clearly demonstrated.

5 Numerical Experiments

The experiments involved classifying images from the dataset, consisting of 60,000 training and 10,000 testing images. I obtained the data directly from Tensorflow and did normalization and reshaping to fit the CNN input requirements.

The CNN model was trained for 10 epochs, with a batch size of 64, using the Adam optimizer and sparse categorical crossentropy loss function. The final results achieved were:

- Training Accuracy: 95.54%, Training Loss: 0.1216
- Validation Accuracy: 90.73%, Validation Loss: 0.3153
- Test Accuracy: 90.73%, indicating strong generalization

Training and validation accuracy steadily increased initially, but validation accuracy plateaued around epoch 6. Validation loss slightly increased after epoch 6, suggesting minor overfitting.

I further analyzed predictions and identified common misclassification patterns. Items such as shirts, coats, and pullovers were frequently confused due to visual similarity, along with sneakers and sandals. Despite these minor challenges, the model performed significantly better than traditional machine learning methods like SVMs or decision trees, which typically achieve lower accuracy (around 80-85%) on the Fashion-MNIST dataset, as demonstrated by previous research (Xiao et al., 2017; Hoang, 2018). Although advanced models and ensembles have achieved higher accuracy (around 94%), the implemented model provides a solid baseline, balancing performance and computational efficiency.

These experiments confirm the effectiveness of CNNs for fashion image classification and highlight potential improvements, including data augmentation and regularization, to address identified weaknesses.

6 Conclusion

In this project, I successfully implemented a Convolutional Neural Network (CNN) to classify images from the Fashion-MNIST dataset, achieving a test accuracy of **90.73%**. This result confirms the effectiveness of CNNs in image classification tasks, highlighting their ability to generalize well to unseen data with relatively simple architectures.

Analysis of model predictions revealed specific challenges, particularly in differentiating visually similar items, such as shirts versus coats or sneakers versus sandals. These misclassifications demonstrate inherent limitations due to overlapping visual features within the dataset.

While the model showed strong overall performance, there is room for improvement. Future work could explore techniques such as **data augmentation**, incorporating **dropout or batch normalization** layers, or experimenting with **deeper architectures or transfer learning** to improve accuracy further. These enhancements could help the model better differentiate between visually similar categories and further reduce misclassification rates.

Overall, this project successfully demonstrates the practical application and effectiveness of CNNs for image classification, providing insights and a solid foundation for future improvements.

7 References

Agarap, A. F. (2018). An architecture combining convolutional neural network (CNN) and support vector machine (SVM) for image classification. *arXiv preprint arXiv:1712.03541*.

Cireșan, D. C., Meier, U., & Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 3642-3649.

Hoang, Q. (2018). Image Classification with Fashion-MNIST and CIFAR-10. *Technical Report, California State University, Sacramento.*

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems, 25*, 1097-1105.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278-2324.

Nocentini, G., Cianfruglia, L., & Strollo, A. G. M. (2022). Image Classification Using Multiple Convolutional Neural Networks on the Fashion-MNIST Dataset. *IEEE Transactions on Emerging Topics in Computing.*

Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv preprint arXiv:1708.07747*.

A Appendix

A.1 Model Summary

A.2 Accuracy and Loss Curves

A.3 Misclassification Examples

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d (MaxPooling2D)	(None, 13, 13, 32)	0
conv2d_1 (Conv2D)	(None, 11, 11, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 5, 5, 64)	0
flatten (Flatten)	(None, 1600)	0
dense (Dense)	(None, 128)	204,928
dense_1 (Dense)	(None, 10)	1,290

Total params: 225,034 (879.04 KB)
Trainable params: 225,034 (879.04 KB)
Non-trainable params: 0 (0.00 B)

Figure 1: CNN Model Summary

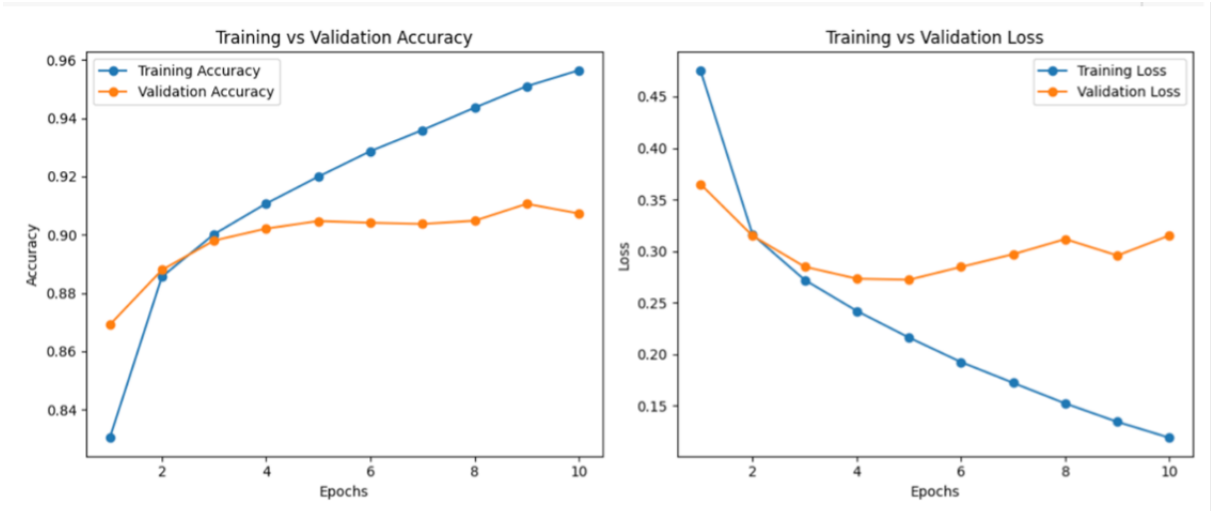

Figure 2: Training and Validation Accuracy and Loss Curves

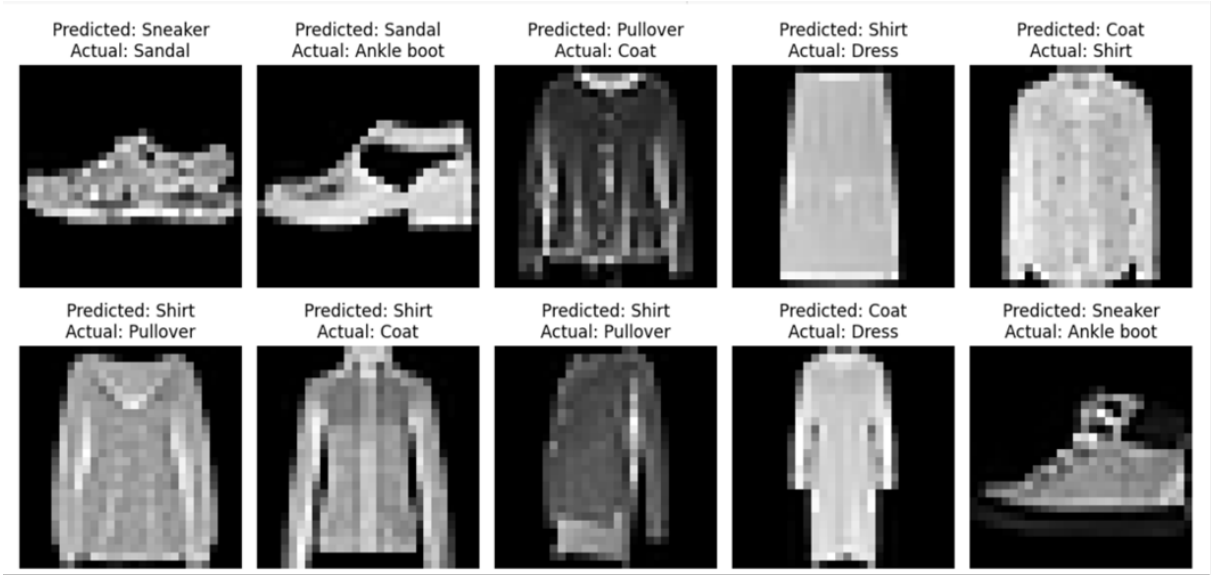

Figure 3: Examples of Misclassified Images