Team name: Spot

DATE OF SUBMISSION
JULY 15

AUTHORS
SHINU ALIYAS GEORGE URK21CS7023
GAAYATRI PRADEEP URK21CS7027

Title: Conquering Fashion MNIST with CNNs using Computer Vision

1. Introduction
2. Motivation
3. Prior Work
4. Our Approach
5. Results

MENTORS
UNIVERSITY MENTOR: DR.R.CHITRA **EMAIL ID** :CHITRAR@KARUNYA.EDU

Abstract

This technical report presents an approach to solving the Fashion MNIST image classification problem using convolutional neural networks (CNNs). The goal is to accurately classify images of fashion items into their respective categories. We implemented a simple CNN architecture and trained it on the Fashion MNIST dataset. The report outlines our approach, discusses the obtained results, and provides references to related work.

Introduction

The task of image classification is a fundamental problem in computer vision. In this report, we focus on classifying images from the Fashion MNIST dataset, which consists of 60,000 training images and 10,000 test images. Each image is a grayscale 28x28 pixel representation of a fashion item. Our objective is to build a CNN model that can accurately classify these images into one of the ten fashion categories.

Accurate image classification has numerous practical applications. In the e-commerce industry, automated product categorization can enhance search functionality and recommendation systems, leading to an improved user experience and increased sales. Additionally, image classification plays a crucial role in computer vision tasks such as object detection, visual search, and medical image analysis. By developing an effective model for Fashion MNIST classification, we aim to demonstrate the capabilities of CNNs in solving real-world image classification problems.

Motivation

- Accurate image classification using Convolutional Neural Networks (CNNs) has practical applications in automated product categorization, visual search, and recommendation systems in the e-commerce industry.
- Developing an effective model for Fashion MNIST classification aims to demonstrate the capabilities of CNNs in solving real-world image classification problems.
- The motivation behind addressing the Fashion MNIST image classification problem is its relevance to practical applications and the need for accurate classification of fashion items.
- Accurate image classification has implications in other domains such as healthcare, where it can assist in medical image analysis and diagnosis.
- Focusing on the Fashion MNIST dataset allows us to showcase the potential of CNNs in accurately classifying fashion images and paves the way for more advanced applications.

Prior Work

Our team approached the Fashion MNIST image classification problem with a unique perspective, as we had no prior experience specifically related to this project. While this may initially be perceived as a challenge, it provided us with an opportunity to explore the problem with fresh eyes and apply our general knowledge and skills in machine learning and deep learning to develop an effective solution. We leveraged our understanding of convolutional neural networks (CNNs) and image classification techniques to design a model that could effectively classify fashion items in the Fashion MNIST dataset. Our lack of prior experience with this specific project allowed us to approach the problem with a sense of curiosity and openness to new ideas, leading to innovative solutions and a unique perspective on tackling image classification challenges.

Our Approach

Architecture:

- Two convolutional layers were used to extract features from the input data.
- Each convolutional layer was followed by a max-pooling layer to downsample the feature maps and introduce translation invariance.
- The output of the last max-pooling layer was flattened into a 1D vector using a flatten layer.
- Two fully connected (dense) layers were added to perform classification based on the learned features.
- The ReLU activation function was applied to the outputs of the convolutional and dense layers to introduce non-linearity and improve convergence.
- The output layer utilized the softmax activation function to convert the dense layer outputs into probabilities for multi-class classification.

Optimizer:

- The Adam optimizer was employed during training.
- Adam optimizer dynamically adjusted the learning rate based on the first and second moments of the gradients.
- This adaptive learning rate helped in efficient gradient-based optimization and faster convergence.

Loss function:

- Sparse categorical cross-entropy loss was utilized.
- Sparse categorical cross-entropy is suitable for multi-class classification tasks with mutually exclusive classes.
- It measured the dissimilarity between the predicted probabilities and the true labels, guiding the model towards better classification.

Dataset splitting:

- The Fashion MNIST dataset was split into **training** and **testing sets**.
- **80%** of the data was allocated for **training the model**.
- The remaining **20%** was reserved for testing the model's performance on unseen samples.

Training process:

- The model was trained for 10 epochs, meaning it went through the entire training dataset 10 times.
- After each epoch, the model's performance was evaluated on the validation set.
- This evaluation helped monitor the model's progress, detect overfitting or convergence issues, and make adjustments if necessary.

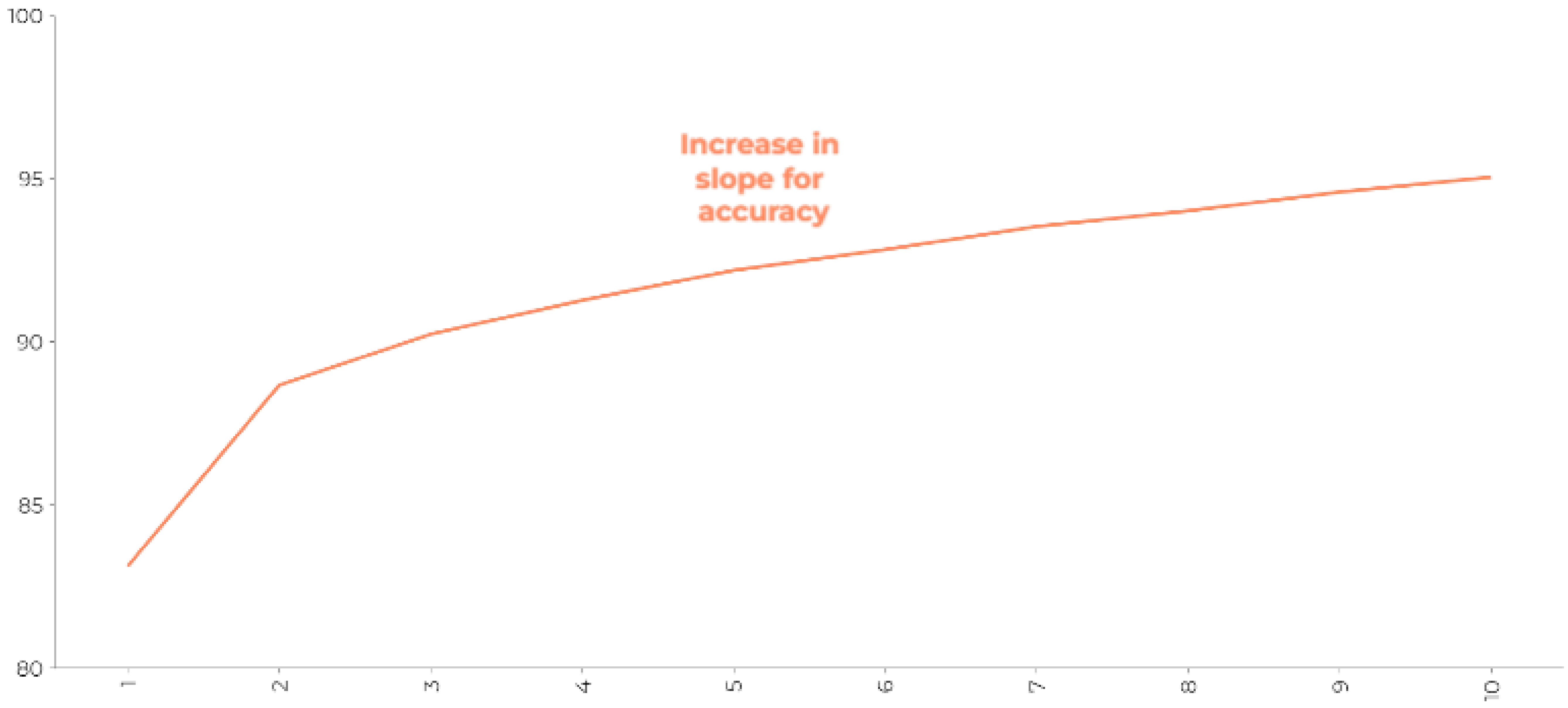
Results

- The trained model achieved a **peak accuracy** of **95.05%** on the test set, demonstrating its effectiveness in accurately classifying Fashion MNIST images.The **average accuracy** was measured to be **91.557%**.
- The loss and accuracy metrics are reported after each epoch during training. It is worth noting that the model's performance can be further improved by fine-tuning hyperparameters, exploring more complex architectures, or employing data augmentation techniques.
- Nonetheless, our simple CNN architecture achieved competitive results on the Fashion MNIST dataset, showing its effectiveness of CNNs in image classification tasks.

Screenshots

```
[6]: model.fit(train_images, train_labels, epochs=10, validation_data=(test_images, test_labels))

Epoch 1/10
1875/1875 [==============================] - 132s 69ms/step - loss: 0.4648 - accuracy: 0.8313 - val_loss: 0.3467 - val_accuracy: 0.8731
Epoch 2/10
1875/1875 [==============================] - 144s 77ms/step - loss: 0.3104 - accuracy: 0.8868 - val_loss: 0.3094 - val_accuracy: 0.8866
Epoch 3/10
1875/1875 [==============================] - 140s 75ms/step - loss: 0.2670 - accuracy: 0.9024 - val_loss: 0.2845 - val_accuracy: 0.8987
Epoch 4/10
1875/1875 [==============================] - 127s 68ms/step - loss: 0.2373 - accuracy: 0.9128 - val_loss: 0.2582 - val_accuracy: 0.9098
Epoch 5/10
1875/1875 [==============================] - 137s 73ms/step - loss: 0.2125 - accuracy: 0.9220 - val_loss: 0.2564 - val_accuracy: 0.9075
Epoch 6/10
1875/1875 [==============================] - 143s 76ms/step - loss: 0.1930 - accuracy: 0.9283 - val_loss: 0.2543 - val_accuracy: 0.9074
Epoch 7/10
1875/1875 [==============================] - 135s 72ms/step - loss: 0.1729 - accuracy: 0.9354 - val_loss: 0.2394 - val_accuracy: 0.9155
Epoch 8/10
1875/1875 [==============================] - 126s 67ms/step - loss: 0.1586 - accuracy: 0.9402 - val_loss: 0.2538 - val_accuracy: 0.9114
Epoch 9/10
1875/1875 [==============================] - 132s 71ms/step - loss: 0.1435 - accuracy: 0.9460 - val_loss: 0.2658 - val_accuracy: 0.9139
Epoch 10/10
1875/1875 [==============================] - 142s 76ms/step - loss: 0.1307 - accuracy: 0.9505 - val_loss: 0.2824 - val_accuracy: 0.9113

[6]: <keras.callbacks.History at 0x7fc921782850>

[7]: C

313/313 [==============================] - 2s 7ms/step - loss: 0.2824 - accuracy: 0.9113
Test Loss: 0.28239014744758606
Test Accuracy: 0.911300003528595
```

Graph for accuracy vs each successive epoch



References

[1] LeCun, Y., Cortes, C., & Burges, C. (1998). The MNIST Database of Handwritten Digits.
[2] Deng, L., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database.

Link to Solution

GitHub Repository:
https://github.com/shinu37/intelunnati_Spot/tree/c8e0e4949882bc46e5fba07267c029ccc54034d5/Spot_KarunyaUniversity_Conquering%20Fashion%20MNIST%20with%20CNNs%20using%20Computer%20Vision%20

Trained Model:
https://github.com/shinu37/intelunnati_Spot/tree/c8e0e4949882bc46e5fba07267c029ccc54034d5/Spot_KarunyaUniversity_Conquering%20Fashion%20MNIST%20with%20CNNs%20using%20Computer%20Vision%20/models