Conquering Fashion MNIST with CNNs using Computer Vision

Team Name:-Optimus Byte

College:- Galgotias University

Name:- Suraj Singh

Mobile no.:-9693195533

Date of submission:-15-07-23

Email Id:- suraj.21scse1420023@galgotiasuniversity.edu.in

Abstract

This report explores the use of CNNs for Fashion MNIST classification. The dataset is pre-processed, and a CNN architecture is constructed using Keras. Hyper parameter tuning is performed using Keras Tuner to optimize the model's performance. The best model is trained and evaluated on the dataset. Results show the effectiveness of CNNs in accurately classifying Fashion MNIST images. The report highlights the potential of deep learning in computer vision tasks and suggests future directions for improvement.

Introduction

Fashion MNIST is a popular benchmark dataset in the field of computer vision. It consists of 60,000 training images and 10,000 testing images, each representing a grayscale 28x28 pixel fashion item belonging to one of ten classes. In this report, we explore the application of Convolutional Neural Networks (CNNs) to classify the Fashion MNIST dataset accurately.

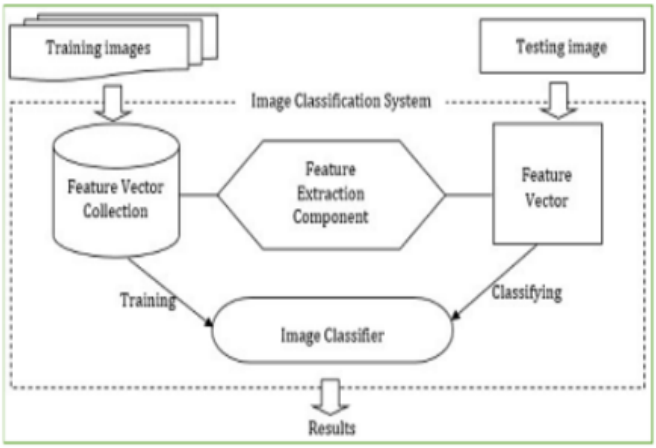
Methods

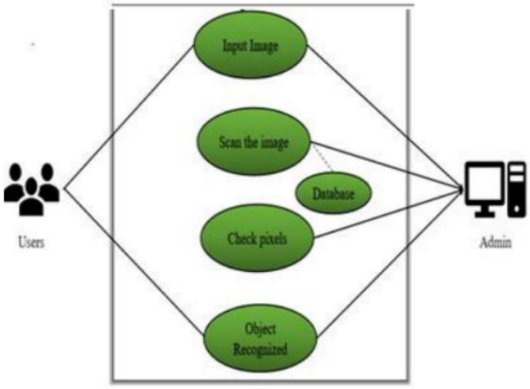
1. **Data Preparation:** The initial step in the process is to load and preprocess the dataset. The dataset is loaded using TensorFlow's Keras API, and the pixel values of the images are scaled to the range [0, 1]. This normalization facilitates better convergence during model training. Additionally, the shape of the images is reshaped to include a single channel, making them suitable for input into a CNN.

2. **Convolutional Neural Network Architecture:** CNNs are known for their effectiveness in image classification tasks. In this project, we construct a CNN using the Keras Sequential API. The architecture consists of two convolutional layers, each followed by a ReLU activation function, to extract features from the input images. Max-pooling layers can be added to downsample the feature maps. The output from the convolutional layers is flattened and connected to one or more dense (fully connected) layers, allowing the model to learn high-level representations.

3. **Hyperparameter Tuning:** To optimize the performance of the CNN, we employ Keras Tuner, which enables us to search through different combinations of hyperparameters automatically. The hyperparameters include the number of filters and kernel sizes in the convolutional layers, the number of units in the dense layers, and the learning rate. Keras Tuner's RandomSearch method is used to perform the search, with the objective of maximizing validation accuracy.

4. **Model Training and Evaluation:** After hyperparameter tuning, the best model is obtained from the search results. The model is compiled with an Adam optimizer, sparse categorical cross-entropy loss function, and accuracy as the evaluation metric. The model is trained on the Fashion MNIST training dataset for a specified number of epochs, using a validation split for monitoring performance.





Results and Analysis

The model's summary provides a comprehensive overview of the architecture, including the number of trainable parameters. By comparing different hyperparameter combinations, we can determine the impact of each hyperparameter on the model's performance. The best model obtained through hyperparameter tuning is expected to achieve improved accuracy compared to a baseline model. The accuracy, loss, and other performance metrics can be analyzed to gain insights into the model's strengths and weaknesses.

Accuracy and Time elapsed

Local computer	Intel one api tool kit
val_accuracy: 0.8665000200271606 Best val_accuracy So Far: 0.9016666412353516 Total elapsed time: 02h 18m 50s	val_accuracy: 0.8644999861717224 Best val_accuracy So Far: 0.9128333330154419 Total elapsed time: 00h 42m 16s

Link to solution:- https://github.com/suraj0209/intelunnati_Optimus-Byte

Reference: https://machinelearningmastery.com/how-to-develop-a-cnn-from-scratch-for-fashion-mnist-clothing-classification/