An Image Classification Implementation on the Clothes MNIST Dataset

Filippin Giacomo
giacomo.filippin@studenti.unipd.it

1. Introduction

1.1. Goals

The objective of this project is to develop and fine-tune the best possible image classification model on the MNIST Fashion Dataset.

1.2. Properties and Evaluation

The selection of the final model is based on the overall accuracy, the type of errors made and the computational costs of each method, since accuracy is not the only important factor in judging a model's effectiveness.

1.3. Result Overview

While still obtaining good results with the KNN and the Dense Neural Network, our best results are achieved with a fine tuned Convolutional Neural Network, which is a ML method known for its effectiveness for image classification problems.

2. Dataset

The MNIST Fashion dataset, utilized in this study, consists of 70,000 images across 10 evenly distributed classes. These images, in 28x28 pixels and black and white format, are encoded into linear arrays totaling 784 values per image. Pixel values range from 0 to 255, with 0 representing white and 255 black. The data underwent normalization using a MinMaxScaler, transforming values from the [0, 255] range to [0, 1]. While other normalization methods like the StandardScaler were considered, they were ultimately not utilized due to the non-normal distribution of the data, which could have altered its inherent structure. The dataset was partitioned into three segments: a training set (50000 pictures), a validation set (10000 pictures), and a test set (10000 pictures).

3. Methods and Models

In order to find the best solution for the task, different methods are used and compared to evaluate their performance in image classification. It's important to note that these are not the only methods we could have tried. For instance, models prioritizing interpretability over effectiveness and efficiency, like regression models and decision trees, would lose focus on our need of a "black box", efficient classifier for item categorization. The models compared in this project are, in order:

- **K-Nearest Neighbors (KNN):** A non-parametric method used for classification and regression. It classifies data based on the majority vote of its nearest neighbors, with the object being assigned to the class most common among its k nearest neighbors. This method can become computationally expensive as test dataset size increases, but if the situation requires to classify only a small batch of 'new' objects, KNN could be the right approach (if accurate enough).

- **Support Vector Classifiers (SVC):** Part of the Support Vector Machines (SVM) family, SVCs are effective for high-dimensional spaces and are versatile in handling both linear and non-linear data through the use of kernel functions. SVCs are known for their accuracy and efficiency in classification tasks, but also for the need of a careful tuning process. Considerations must be made for both effectiveness and efficiency, which can vary significantly between cases.

- **Dense Neural Networks:** Also known as fully connected neural networks. DNN are powerful for capturing complex patterns in data, making them suitable for a wide range of classification tasks. Despite their potential for high accuracy, they require substantial data and computational resources to train effectively.

To enhance each method's accuracy, reduce overfitting, and lower computational expenses, PCA was employed to diminish feature space from 784 to 188. Subsequently, each method was optimized to its finest iteration, and finally assessed against a Convolutional Neural Network (CNN), acknowledged for its proficiency in image classification, ensuring spatial data was preserved by not applying PCA prior to CNN training and evaluation.

4. Horse Race

In this section we go through each method's training process and classification capabilities, analyzing the effectiveness of each one of them also in relation with the others.

4.1. K-Nearest Neighbors (KNN):

As shown in fig. 1, the choice of the hyperparameter K is a perfect example of trade-off between bias and variance. While using a low K shows clear signs of overfitting, a higher one increases the generalization capabilities of the KNN but also lowers the accuracy, since the KNN loses sensitivity to small variations and tends to underfit. A good compromise seems to be at K = 8, where KNN starts to generalize better without compromising too much the accuracy on the validation set. For this K, accuracy on test set is 86.09%, which is good enough, but computationally heavy predictions must be taken into consideration. The confusion matrix (fig. 4) shows another important aspect: the inductive bias of the KNN. Given that label assignment is based on the similarity between points, this method struggles particularly in distinguishing classes with similar appearances, such as Shirts, T-Shirts, pullovers and coats, which are critical classes with similar features.

4.2. Support Vector Classifiers (SVC):

We conduct a grid search to optimize the model's hyperparameters, selecting across a set of values. However, the extensive computation time required for training, even after feature space reduction, deems this method impractical despite potential gains in accuracy. This limitation is dictated by the available computational resources, and it's aggravated by the need of a careful tuning. In environments equipped with more advanced computational infrastructure, SVC models could still be considered a viable strategy.

4.3. Dense Neural Networks (DNN):

The DNN that scores the highest accuracy on the test set is structured with three fully connected, funnel-structured layers (256, 128, 64 neurons) followed by a softmax output layer for multi-class classification. Each hidden layer utilizes ReLU activation for non-linearity and includes dropout layers to mitigate overfitting. The model is compiled with the Adam optimizer and categorical crossentropy loss. An EarlyStopping callback is employed to prevent overfitting by halting training if the validation loss does not improve, restoring the weights of the best performing epoch. The accuracy on the test set for the resulting DNN is 89.22%, which is 3% higher than that of KNN. The confusion matrix (fig. 5) shows an improvement in differentiating critical classes. The reason for this could be that the DNN learns to recognise patterns and features in the pictures and does not solely search for the most similar ones. Despite this improvement, shirts and t-shirts remain the most difficult classes to distinguish, and the Dense Neural Network tends to make mistakes classifying both shirts as t-shirts and vice versa.

4.4. Convolutional Neural Networks (CNN):

The CNN we compare the other methods to consists of two convolutional blocks with 64 filters, ReLU activation, and max pooling, followed by a flattening layer, a 128-neuron dense layer with dropout, and a softmax output layer for probability prediction. It uses the Adam optimizer, categorical crossentropy loss, and EarlyStopping based on validation loss. The accuracy score of this CNN on the test set is 91.27%, 2% better than that of our DNN. On top of this improvement, we can see (fig. 6) that the CNN reduces errors on critical classes, with shirts and t-shirts still being the most confusing class, but with fewer errors than the ones committed by the other methods tested. Additionally, the CNN outperforms the DNN in terms of its substantially higher convergence speed, which allows us to spare computational costs.

5. Conclusions

Table 1. Test accuracy and 'shirt' class metrics for each Model

Model	Test Accuracy	Shirt Recall	Shirt Precision
KNN	86.09%	56.30%	67.67%
SVC	-	-	-
DNN	89.22%	70.90%	70.20%
CNN	91.27%	72.00%	75.08%

Concluding our study on the MNIST Fashion dataset, it is clear that each method comes with its own set of advantages and drawbacks. The K-Nearest Neighbors, while straightforward and intuitive, suffers from high computational costs during the classification phase and lower accuracy, especially in distinguishing similar classes like shirts and t-shirts. The Support Vector Classifier, despite its potential, was not explored due to its prohibitive computational cost during the training phase. On the other hand, the Dense Neural Networks showed a significant improvement in accuracy, making it a strong contender for image classification tasks, but it is the Convolutional Neural Network that ultimately stands out as the most effective model in this study. With the highest accuracy, particularly in differentiating between the most challenging classes, and its rapid convergence, the CNN demonstrates its superiority for image classification tasks. It combines efficiency and accuracy, thus becoming the preferred choice, despite its architectural complexity compared to simpler models like KNN. Therefore, for tasks requiring high accuracy in image classification with a focus on computational efficiency, the CNN emerges as the best model among those tested.

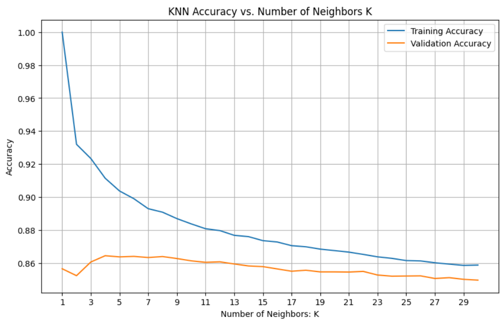

Figure 1. KNN's accuracy by K (balance chosen at K = 8)

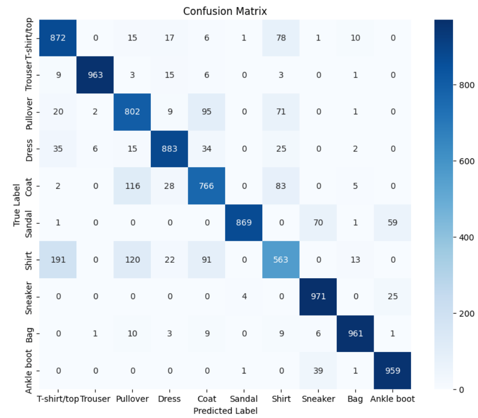

Figure 4. KNN's confusion matrix

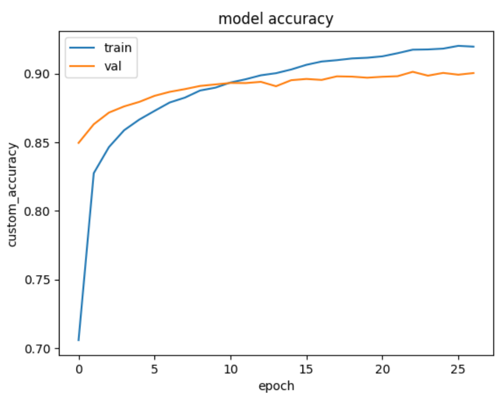

Figure 2. DNN's accuracy (early stoppage at epoch 17)

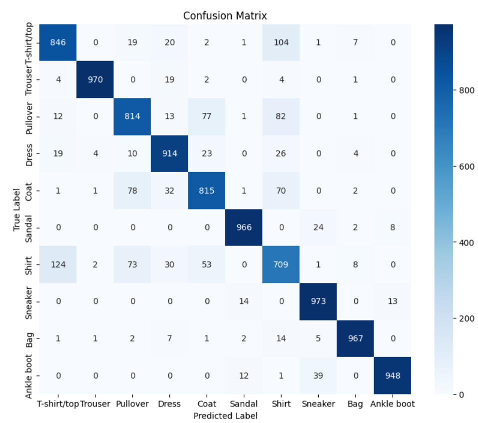

Figure 5. DNN's confusion matrix

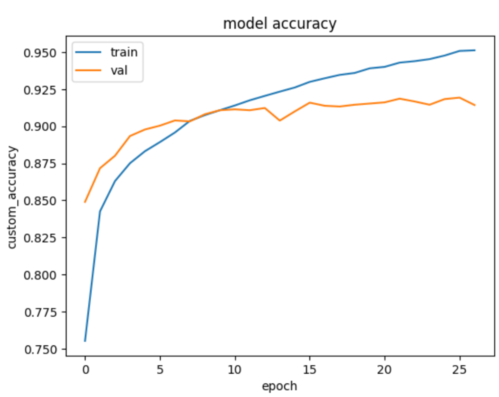

Figure 3. CNN's accuracy (early stoppage at epoch 13)

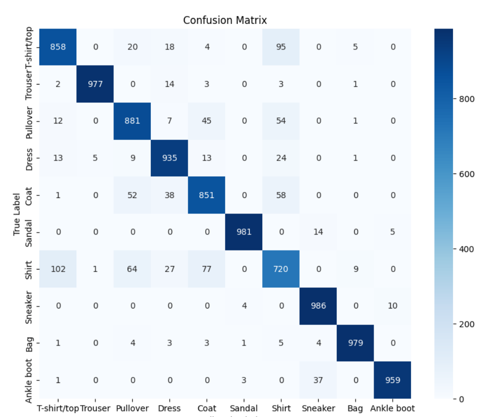

Figure 6. CNN's confusion matrix