Assignment 5

Bendik Nordeng (Student ID: 478128, user name: bendikno)
Peder Møyner Lund (Student ID: 478109, user name: pederml)

November 2019

1 Introduction

This paper explores and evaluates the use of an artificial neural network (ANN) and a Support Vector Machine (SVM) to classify characters. Furthermore, it builds an OCR (Optical Character Recognition) system using the implemented ANN.

The complete system consist of the files described below. We use Python with scikit-learn to build the model, as well as numpy and matplotlib to process and visualize the data and results. The files run with a simple python3 command in terminal, for example "python3 OCR.py". The files containing the models (ANN.py and SVM.py) include a grid search, which takes some time to run. If you would just like to see the output, they can be found in jupyter notebook format here: `https://github.com/bendiknordeng/Machine-learning/tree/master/Assignment-5`

Files:

- tools.py: functions for preprocessing the data, sliding window algorithm and plotting.
- ANN.py: defining the neural network model, finding optimal hyperparameters and accuracy score.
- SVM.py: defining the SVM model, finding optimal hyperparameters and accuracy score.
- OCR.py: execution of character recognition using sliding window and the trained ANN.

2 Feature Engineering

2.1 Explanation of selected feature engineering techniques

We decided to us Principal Component Analysis (PCA) and standardization as our two feature engineering techniques. PCA reduces the dimensionality of the dataset by looking at linearly uncorrelated features (principal components) of the data. The technique saves a lot of processing time for our models without losing too much information, as well as improving the accuracy score on the test set. Before reducing the data with PCA, each input consisted of 400 features. We found that reducing the images to 40 features lead to the best results.

Moreover, we decided to use standardization to make our data more manageable while preserving the significance of the data. The standard scaler subtracts the mean and divides by the standard deviation of the training set, and furthermore saves these values so that we can scale the data

we classify later. Also, some of the classifiers in scikit-learn requires the data to be standardized to be able to perform well.

2.2 Future feature technique exploration

If we had more time, we would have liked to explore scale-invariant feature transform (SIFT) method. This is because of SIFTs ability to detect and describe local features in images. By identifying local features in the image, SIFT is supposed to be indifferent to various image transformations. Thus, we might have further increased the accuracy of our character identifier as some of the characters in the data have been exposed to different transformations.

3 Character Classification

When we were given the data set for the assignment, our initial thoughts were that artificial neural networks would be suitable for the task. ANN have the ability to take in a wide range of inputs and process them to infer hidden as well as complex, non-linear relationships. This makes ANN very suitable for image and character recognition. Moreover, we were also interested in evaluating how SVM performed classifying characters. SVM works well with even unstructured and semi structured data like images, and usually performs well with high dimensional data. SVMs are usually outclassed by deep neural networks and thus our initial hypothesis was that the ANN would outperform the SVM.

3.1 SVM

A Support Vector Machine (SVM) is a non-parametric, supervised classifier that classify based on a separating hyperplane. The algorithm outputs an optimal hyperplane which categorizes new examples. SVM works well for classification in higher dimensions and when there is a clear margin of separation between the cases. It can be applied on problems related to text and image categorization. SVM does not perform well when there is noise and overlap in the training data.

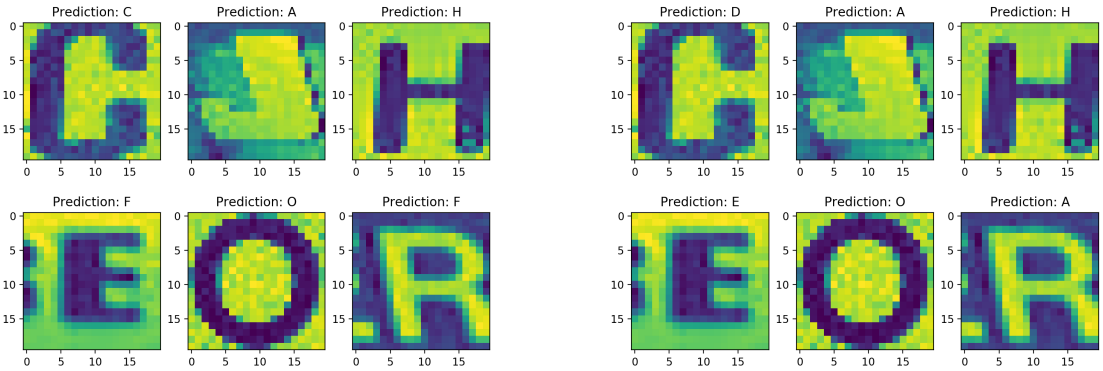
3.2 ANN

Artificial Neural Networks is a specific machine learning method within the field of deep learning. The basic structure of an ANN consists of artificial neurons (similar to biological neurons in the human brain) that are grouped into layers. The most common ANN structure consists of an input layer, one or more hidden layers and an output layer. ANNs usually takes longer time to train, increasing with amount of data and number of layers in the network. Deep models are also able to extract better features than shallow models.

4 Performance

As the task specified, we used a train-test split of respectively 80% and 20% of the data set. After training both models on the same data set, we used the test set to evaluate the models. The results show that the ANN achieved an accuracy of 83.7% while the SVM achieved an accuracy of 79.48%. Thus, the ANN slightly outperformed the SVM. However, the ANN showed signs of overfitting with 100% accuracy on the training set. The SVM scored 93.49% on the training set and hence is maybe less prone to overfitting on this data set.

Example images classified with the different models are shown below. As we can see, both of the models have trouble classifying the letter J (this is further substantiated later when we look at the confusion matrix for the models). Furthermore, SVM wrongly classifies some of the example images that the ANN are correct about.



(a) ANN (b) SVM

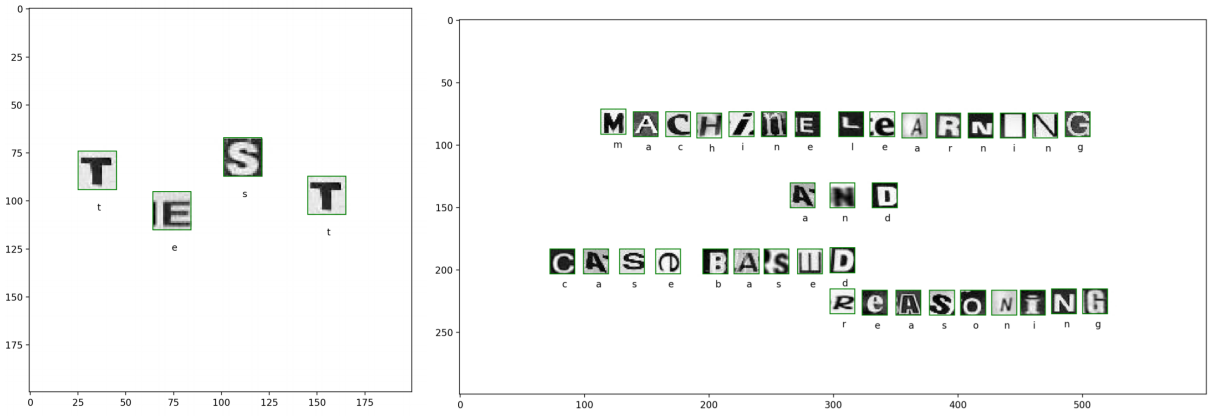
Figure 1: Examples of classification predictions

4.1 Future machine learning technique exploration

If we were to try any additional machine learning techniques we would like to test a Convolutional Neural Networks (CNN). CNNs are known for its good image-recognition qualities and it would be very interesting to see if a CNN would outperform our implemented methods.

5 Character Detection

Figure 2a and 2b show the performance of our character detector on detection-1.jpg and detection-2.jpg. The green boxes represents the windows in which the OCR-system detected a character and the small characters below the respective boxes are the predictions.



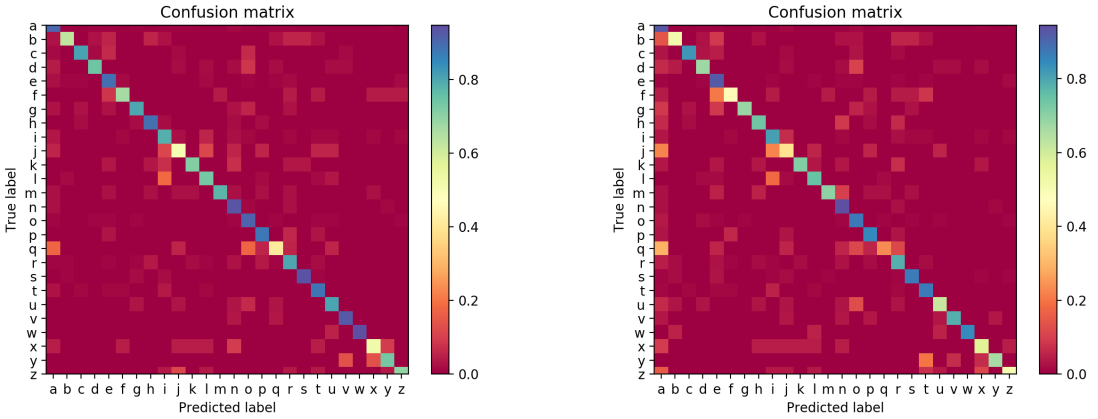
(a) Detection-1.jpg (b) Detection-2.jpg

Figure 2: Test images for OCR system

5.1 Results

As seen from figure 2a and 2b, the ANN performed exceptionally well achieving 100% accuracy on both images and precisely detected where the characters were located in the image.

Despite the high accuracy, we wanted to understand what kind of characters our detector identified well and where it could be improved. In order to get a good overview of the performance, we created two confusion matrices to the respective methods as seen in figure 3a 3b. If one studies the matrices closely, we can see that the ANN performed somewhat better than the SVM.



(a) ANN (b) SVM

Figure 3: Confusion matrices

The classifiers performed better on some characters than others. The characters on which the ANN-classifier performed worst were the characters Q, J and X, while it performed best on the characters V, W, N and A. The SVM-classifier had the same classification patterns as the ANN, but it performed a bit worse in general.

Some characters were in general easier to classify than others given the specific features in some characters. For example, the character S has very distinct features and does not share many similarities with other letters. This is reflected in the results as both ANN and SVM achieve high accuracy classifying this character. Moreover, some of the characters are often confused with each other. An example of this are the letters X and Y. As seen from the matrices, both classifiers confuse these letters. On this particular area, the SVM performs slightly better than the ANN.

5.2 Improvements

In order to improve the performance of our OCR-system we identified two areas of improvement. Firstly, we could have taken other feature engineering techniques into use. For example, we could include techniques to deal with rotation of characters. As discussed earlier, another improvement could have been to look at the use of CNNs and TensorFlow to maybe achieve higher accuracy scores.

6 Conclusion

To conclude, our OCR-system performed well. The ANN performed slightly better than the SVM. In our view, the weakest component of the OCR-system we created is the feature engineering, using only two techniques to process the data. The strongest component of the system was maybe the character detection using sliding window, which seemed to perform perfectly.