LUT
Lappeenranta
University of Technology
LUT School of Engineering

Pattern Recognition

(BM40A0701)

Lasse Lensu

Assignment Pattern Recognition

Handwritten Number Recognition

Chowdhury Ashek Bin Helal - 0524453

Arnob Islam Khan - 0525669

Manna Barua - 0458095

11.12.2017

Table of Contents

Table of Figures and Tables

List of Abbreviations

GMM	Gaussian Mixture Model
kNN	k-Nearest-Neighbour
MLP	Multi-Layer Perceptron
MTC	Map Transformation Cascade
SVM	Support Vector Machines

1 Introduction and Objective

The handwritten digits recognition is an extensive research field (Cardoso & Wichert, 2013, p. 575) . There are different types of recognitions where machines are applied, and handwritten recognition is one of them (Neves, Zanchettin, & Filho, 2012, p. 229). There are lot of fields using the application of handwritten digits recognition, such as in finance, the economy etc. (Dreyfus, 2005, p. 20). Digit recognition and comparable problems such as alphabetic letter recognition are demanding tasks for researchers and their learning models (Alonso-Weber, Sesmero, & Sanchis, 2014, p. 8180; Impedovo & Pirlo, 2014, p. 969).

Many research endeavors are started on machine learning systems, for example, multilayer-perceptrons (MLP), support vector machines (SVM), k-nearest neighbor (kNN) procedures and furthermore classifier groups that combine these approaches (Neves, Zanchettin, and Filho, 2012, p. 229f).

One approach is the MLP since its various leveled structure is proposed to decide fundamental features in the initial layers and all the more high-level structures of the recognition task in the subsequent layers (Alonso-Weber, Sesmero, and Sanchis, 2014, p. 8181). For the outcome a yield neuron for every digit class – so on account of the numbers from '0' to '9' 10 neurons – will be displayed (Haykin, 2009, p. 25). However , the fundamental disadvantage of this method that is may end up in a local minimum for the cost function (Neves, Zanchettin, and Filho, 2012, p. 230).

SVM, regarded as best classifier for binary decision since it finds the largest margin between two classes (Neves, Zanchettin, & Filho, 2012, p. 230). But it can not deal with multiclass problem such as handwritten digit recognition with 10 classes as it mainly focuses on single binary classifications (Neves, Zanchettin, & Filho, 2012, p. 230).

The kNN method is a classifier that is premised on the distance between the training data and chooses the class according to the k nearest patterns of the training data (Neves, Zanchettin, & Filho, 2012, p. 230). The performance of this classifier but also the required processing time is positively correlated with the size of the training set (Neves, Zanchettin, & Filho, 2012, p. 230). In the research conducted by Neves, Zanchettin, & Filho (2012) the use of kNN, especially in a classifier ensemble, shows comparably high accuracy but also long processing times.

But despite those more commonly deployed approaches there are also other methods that are introduced in research. Classical classifiers are commonly applied since they offer high accuracy of the recognition with small processing times, while other, newer approaches come at the expense of computational capacity and processing time (Neves, Zanchettin, & Filho, 2012, p. 235f). Two examples of different approaches are, first, a combination of a Map Transformation Cascade (MTC) with a linear SVM and, second, a discriminative Gaussian Mixture Model (GMM) structure selection method with embedded discriminative learning criterion into a Bayesian model structure selection framework. The MTC-SVM approach is capable to achieve competitive accuracies on the two commonly used MNIST and USPS datasets (Cardoso & Wichert, 2013, p. 579). The discriminative GMM method with discriminative learning criterion is also a very accurate classification technique for handwritten digits, achieving highly competitive results on both the MNIST and the CENPARMI dataset (Chen, Liu, & Jia, 2011, p. 960).

The objective of this paper is to design a classifier in MATLAB that classifies the information of a handwritten digit based on a 3D time series of the stroke. The classification has been done using personally programmed classifier based on KNN method with low-level MATLAB function.

2 Data and Methods

2.1 Sample Data and Pre-processing

In this section the dataset provided for the assignment will be analysed. The stroke data of handwritten digits that is provided was recorded by a LeapMotion sensor. The trial was conducted by human in the air close to the sensor. The person was asked to write each digit 100 times by using the index finger in different ways. The stroke data is then available in 3D-timeseries in a separate Matlab-file. In the data set, for each digit there are 100 Matlab-files and a total of 1000 Matlab-files are supplied for the assignment. For each digit the data set form a uniform distribution over the class which is illustrated in the following histogram in Figure-1.

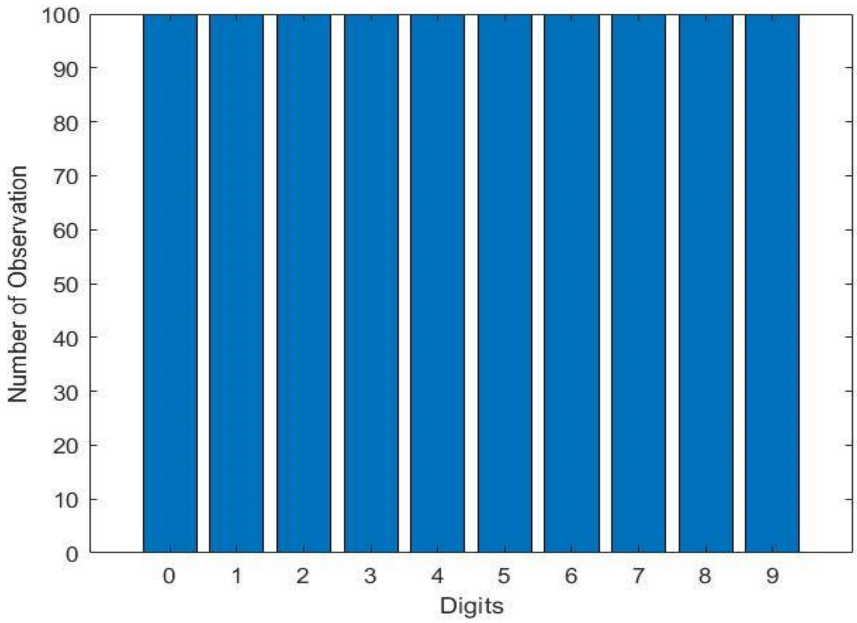

Figure 1: Histogram of observations

In the pre-processing step, the dataset is supplied with transformation in order to supply the classifier a larger dataset in different 3D-timeseries to enhance the classification abilities and accuracy. The dataset is then normalized by using Max-min Normalization and scaled the dataset within the interval [0, 1]. This step of normalization does not have impact on the classification of the digit (Bishop, 2006, p. 261) and the classification itself.

Besides transformation and normalization of the data, an approach of rotating the data can also be found in scientific literature (Alonso-Weber, Sesmero, Gutierrez, Ledezma, & Sanchis, 2015, p. 436). In this step, the rotation of the dataset is conducted along with X, Y and Z axis with a step size of 5°, from 0° rotation to 45°.

By applying rotation to the data set, it is possible to enlarge the dataset from 1000 observation to 28,000 data points to fit in the classifier.

The rotation of the data is performed on a single axis for the different rotation angle. The largest rotation angle of 45° for each axis is displayed in 2D and 3D in **Error! Reference source not found.**, **Error! Reference source not found.** and **Error! Reference source not found.**.

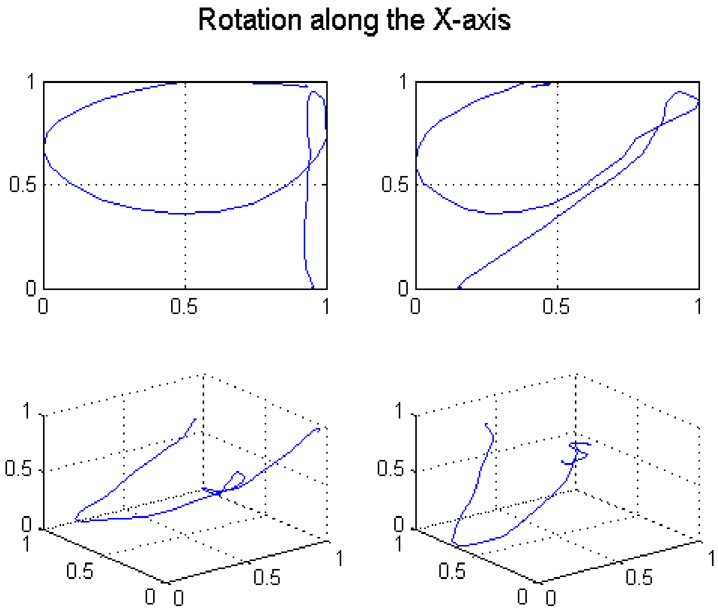

Figure 2: Rotation of Training data along X-axis

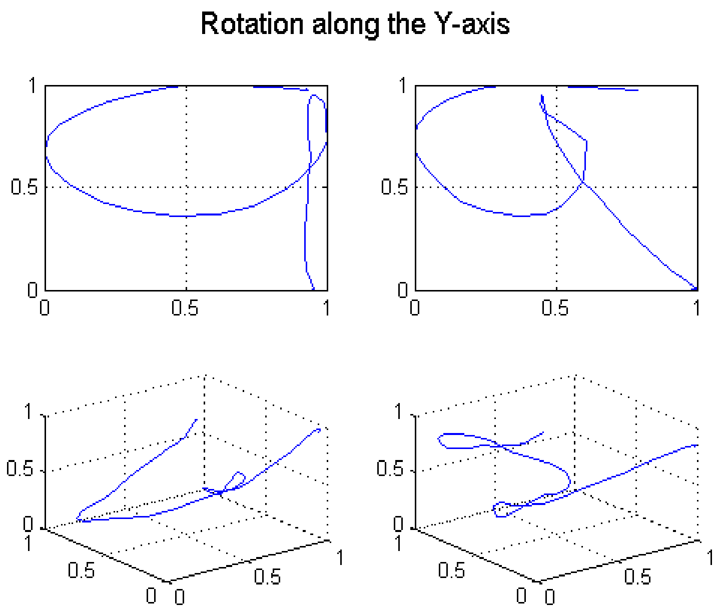

Figure 3: Rotation of Training data along Y-axis

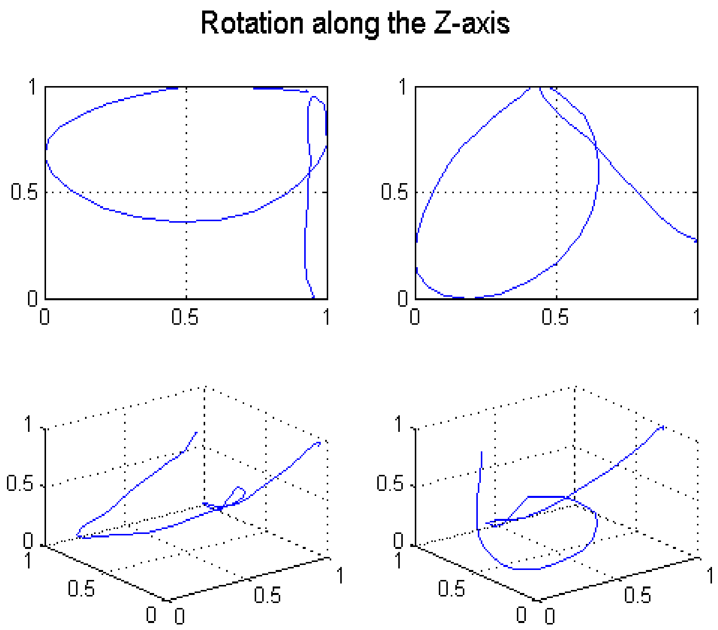

Figure 4: Rotation of Training data along Z-axis

In the following, features extraction and the classifier will be discussed, which are used for a classifier to classify a 3D handwritten digit input.

2.2 Feature Extraction and Model

One of the major challenges of the feature extraction with the existing dataset is the variation in the size of the stroke for the numbers which range between 19 points to 106 points for all digits. It is recommended to focus on the characteristics of the stroke and neglect the amount of point of an observation.

Different approaches can be applied for similar types of problems. For instances imaged based handwritten number recognition, common methods are geometric transformations, projection, from image to binary value transformation, graphs or momentum based features and so on (Impedovo & Pirlo, 2014, p. 969).

In this paper, only projections along the x and y-axis is used for the observation and the normalized range of [0, 1] is used with a step-size of 0.05. It means that for x projection and for the one on y 21 values are calculated. To determine each pair of corresponding points (first to second, second to the third and so on) the equation that defines the line between them. The equation of the straight line as follows

$$y = ax + b$$

After finalizing the feature selection for the data, the next step is to choose an appropriate classifier that fits wells in the large dataset. In this paper, k-nearest neighbours (kNN) classifier is used. The reason behind choosing kNN approach is kNN is a simple classifier and performs accurately when the training dataset is large and has distinct features. A 10-fold cross-validation method is used to find the amount of k-nearest neighbours. For this purpose, the observations are divided randomly in 10 equal subsamples. Nine subsamples are used in the training process and the remaining subsample is used for the test set to test the performance of the classifier. It also prevents over fitting of the data in the classifiers. This process is repeated until each of the 10 subsamples was once used as a test set and each time performance is calculated. These 10 performance data is then averaged to determine the accuracy of the classifier. The value of k is kept in the range from 1 to 21 with an interval of 2. The cross validation process is performed and the average performance is calculated for each k-value. The highest average value of the performance indicates the final accuracy of the classifier.

3 Results

With the given dataset the accuracy of the classifier depicts that the combination of the presented feature selection and the kNN classifier leads to average 90% accuracy in classification. This suggests that the selected features represent the given digit patterns well and enable a reasonable generalization and classification.

It illustrates that out of 1000 samples, the classifier is able to identify almost for 900 the correct digit. The maximum classification accuracy in the cross-validation process for 3-nearest-neigbors even reaches 96.45%.

4 Conclusion and Implications

This paper is able to describe a 3-NN classifier performance with a larger amount of dataset.

In this experiment, a 3-NN classifier based on projections and the changes in x- and y-coordinates work well for 3D-handwritten digits. The third axis is neglected for the purpose of classification since the variation in it is not intended by the human test subject but it is kept for the rotations in order to avoid distorted observation. However, the third axis is regarded as useless and discarded in the feature selection.

The presented methodology and classifier leads to better classification results. However, there are also limitations for the given classifier and feature selection process. Clearly, there are further adaptations to the data that can be conducted to obtain further representative features. This included further pattern distortion techniques and displacements (Alonso-Weber, Sesmero, Gutierrez, Ledezma, & Sanchis, 2015, p. 436) such as creating a more / less narrow and more / less high digit. Also techniques such as zoning could be used where sub-images (partitions) of the overall image are utilized to gather more local information of the overall digit for the classification of it (Impedovo & Pirlo, 2014, p. 969ff; Suen, Guo, & Li, 1994; Suen, Guo, & Li, 1992). Apparently, there are further methods that can potentially be used to enhance the extraction of representative features.

In conclusion, the paper introduced a very accurate 3- nearest neighbour classifier for 3D handwritten digit numbers recorded by a LeapMotion sensor with classification rate of more than 90% in every cross-validation step. The normal execution of 95.26% that the creator could reach is commenced on a fruitful element determination process and in addition a vast improvement of the observational information.

Appendix: Literature

1. Alonso-Weber, J. M., Sesmero, M. P., & Sanchis, A. (2014). Combining additive input noise annealing and pattern transformations for improved handwritten character recognition. *Expert Systems with Applications 41* , 8180–8188.
2. Alonso-Weber, J. M., Sesmero, M. P., Gutierrez, G., Ledezma, A., & Sanchis, A. (2015). Input Transformation and Output Combination for Improved Handwritten Digit Recognition. In *Artificial Neural Networks - Methods and Applications in Bio-/Neuroinformatics* (pp. 435-443). Springer International Publishing.
3. Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* New York: Springer Science+Business Media.
4. Cardoso, A., & Wichert, A. (2013). Handwritten digit recognition using biologically inspired features. *Neurocomputing 99* , 575-580.
5. Chen, X., Liu, X., & Jia, Y. (2011). Discriminative structure selection method of Gaussian Mixture Models with its application to handwritten digit recognition. *Neurocomputing 74* , 954-961.
6. Dreyfus, G. (2005). *Neural Networks. Methodology and Application.* Berlin - Heidelberg: Springer.
7. Haykin, S. (2009). *Neural Networks and Learning Machines (3rd Edition).* New Jersey: Pearson Education.
8. Impedovo, D., & Pirlo, G. (2014). Zoning methods for handwritten character recognition:Asurvey. *Pattern Recogntion 47* , 969–981.
9. Neves, R. F., Zanchettin, C., & Filho, A. N. (2012). An Efficient Way of Combining SVMs for Handwritten Digit Recognition. In A. E. Villa, W. Duch, P. Érdi, F. Masulli, & G. Palm, *Artificial Neural Networks and Machine Learning - ICANN 2012* (pp. 229-237). Lausanne: Springer.
10. Suen, C. Y., Guo, J., & Li, Z. C. (1994). Analysis and recognition of alphanumeric handprints by parts. *IEEE Transactions on SMC 24 (4)* , 614-630.
11. Suen, C., Guo, J., & Li, Z. (1992). Computer and human recognition of hand-printed characters by parts. In S. Impedovo, & J. Simon, *From Pixels to Features III: Frontiers in Handwritten Recognition* (pp. 223-236). Elsevier Science.