



Handwritten Digit Recognition Using Machine Learning Algorithms

S. M. Shamim, Mohammad Badrul Alam Miah, Angona Sarker, Masud Rana, Abdullah Al Jobair

Department of Information and Communication Technology
Mawlana Bhashani Science and Technology University Santosh, Tangail-1902, Bangladesh
Correspondence: E-mail: ictshamim@yahoo.com

ABSTRACT

Handwritten character recognition is one of the practically important issues in pattern recognition applications. The applications of digit recognition include in postal mail sorting, bank check processing, form data entry, etc. The main problem lies within the ability on developing an efficient algorithm that can recognize hand written digits, which is submitted by users by the way of a scanner, tablet, and other digital devices. This paper presents an approach to off-line handwritten digit recognition based on different machine learning techniques. The main objective of this paper is to ensure the effectiveness and reliability of the approached recognition of handwritten digits. Several machines learning algorithms (i.e. Multilayer Perceptron, Support Vector Machine, Naïve Bayes, Bayes Net, Random Forest, J48, and Random Tree) have been used for the recognition of digits using Waikato Environment for Knowledge Analysis (WEKA). The experimental results showed that the highest accuracy was obtained by Multilayer Perceptron with the value of 90.37%.

© 2018 Tim Pengembang Jurnal UPI

ARTICLE INFO

Article History:

Submitted/ Received 03 Aug 2017

First Revised 05 Jan 2018

Accepted 05 Mar 2018

First available online 09 Mar 2018

Publication Date 01 Apr 2018

Keyword:

*Pattern recognition,
Handwritten recognition,
Digit recognition,
Machine learning,
Off-line handwritten recognition,
Machine learning algorithm.*

1. INTRODUCTION

Intelligent image analysis is one of the most attractive research area in artificial intelligence. One of the subjects in the intelligent image analysis is handwritten digits recognition. Handwritten digits recognition is a well-researched subarea within the field that is concerned with learning models to distinguish pre-segmented handwritten digits. This

has been used widely in data mining, machine learning, and pattern recognition along with many other disciplines of artificial intelligence (Watada & Pedrycz, 2008). The main application has determined efficacious in conforming decisive systems, which compete to human performance and accomplish far for the improvement of classical artificial intelligence systems (Seewald, 2011). However, not all

features of those specific models have been previously inspected.

A great attempt of research worker in machine learning and data mining has been contrived to achieve efficient approaches for approximation of recognition from data. In the twenty first century, handwritten digit communication has its own standard. Most of the times in daily life for using this communication have been known in the conversation and the recording for the information to be shared with individuals. One of the challenges in handwritten characters recognition wholly lies in the variation and distortion of handwritten character set. This is because distinct community may use diverse style of handwriting and control to draw the similar pattern of the characters of their recognized script.

Identification of digit from where the best discriminating features can be extracted is one of the major tasks in the area of digit recognition system. To locate such regions, different kinds of region sampling techniques are used in pattern recognition (Das *et al.*, 2012). The challenge in handwritten character recognition is mainly caused by the large variation of individual writing styles (Plamondon & Srihari, 2000). Hence, robust feature extraction is very important to improve the performance of a handwritten character recognition system. Nowadays, handwritten digit recognition has obtained lot of concentration in the area of pattern recognition system sowing to its application in diverse fields. In the next days, character recognition system might serve as a cornerstone to initiate paperless surroundings by digitizing and processing existing paper documents.

Handwritten digit dataset is vague in nature because there may not always be sharp and perfectly straight lines. The main goal in digit recognition is feature extraction is to remove the redundancy from the data and gain a more effective embodiment of the word image through a set of numerical attributes. It deals with extracting most of the essential

information from image raw data (AlKhateeb *et al.*, 2011). In addition, the curves are not necessarily smooth like the printed characters. Furthermore, characters dataset can be drawn in different sizes and the orientation, which are always supposed to be written on a guideline in an upright or downright point. Accordingly, an efficient handwritten recognition system can be developed by considering these limitations. It is quiet exhausting that sometimes to identify hand written characters as it can be seen that most of the human beings can not even recognize their own written scripts. Hence, there exists constraint for a writer to write apparently for recognition of handwritten documents.

Before revealing the method used in conducting this research, software engineering module is firstly presented. Pattern recognition along with image processing plays compelling role in the area of handwritten character recognition. The previous study (Tokas & Bhadu, 2012), describes numerous types of classification of feature extraction techniques, such as structural feature-based methods, statistical feature-based methods, and global transformation techniques. Statistical approaches are established for planning on how data are selected. It utilizes the information of the statistical distribution of pixels in the image. (Neves, 2011) provided method based offline handwritten digit recognition system. Authors claim that this method outperforms in the experiment. Experiment is carried out on NIST SD19 standard dataset. Other previous study provide the conversion of handwritten data into electronic data, nature of handwritten characters and the neural network approach to form machine that competent for recognizing hand-written characters (Perwej, 2012). The other study addresses a comprehensive criterion of handwritten digit recognition with various state of the art approaches, feature representations, and datasets (Liu, *et al.*, 2003). However, the relationship in the training set size versus accuracy/error and the dataset-independence

of the trained models are analysed (Firdaus, *et al.*, 2017). Either paper presents convolution neural networks into the handwritten digit recognition research and describes a system which can still be considered state of the art (LeCun, *et al.*, 1998).

This paper presents an approach to offline handwritten digit recognition based on different machine learning techniques. The main objective of this paper is to ensure the effectiveness and reliability of the approached recognition of handwritten digits. Several machine learning algorithms (i.e. Multilayer Perceptron, Support Vector Machine, Naïve Bayes, Bayes Net, Random Forest, J48, and Random Tree) have been used for the recognition of digits using Waikato Environment for Knowledge Analysis (WEKA).

2. THEORY

2.1. Multilayer Perceptions

A neural network-based classifier, called Multi-Layer perception (MLP), is used to classify the handwritten digits. MLP consists of three different layers: input layer, hidden layer, and output layer. Each of the layers can have certain number of nodes also called neurons, and each node in a layer is connected to all other nodes to the next layer (Bhowmik, *et al.*, 2014). For this reason, it is also known as feed forward network. The number of nodes in the input layer depends upon the number of attributes present in the dataset. The number of nodes in the output layer relies on the number of apparent classes exist in the dataset. The convenient number of hidden layers or the convenient number of nodes in a hidden layer for a specific problem is hard to determine. But in general, these numbers are selected experimentally. In multilayer perceptron, the connection between two nodes consists of a weight. During training process, it basically learns the accurate weight adjustment, which corresponds to each connection (Kruse, *et al.*, 2013). For the learning

purpose, it uses a supervised learning technique named as Back propagation algorithm.

2.2. Support Vector Machine

Support Vector Machine (SVM) is a specific type of supervised ML method that intends to classify the data points by maximizing the margin among classes in a high-dimensional space (Pereira, 2009). SVM is a representation of examples as points in space, mapped due to the examples of the separate classes are divided by a fair gap that is as extensive as possible. After that new examples are mapped into that same space and anticipated to reside to a category based on which side of the gap they fall. The optimum algorithm is developed through a “training” phase in which training data are adopted to develop an algorithm capable to discriminate between groups earlier defined by the operator (e.g. patients vs. controls), and the “testing” phase in which the algorithm is adopted to blind-predict the group to which a new perception belongs (Orru, *et al.*, 2012). It also provides a very accurate classification performance over the training records and produces enough search space for the accurate classification of future data parameters. Hence, it always ensures a series of parameter combinations no less than on a sensible subset of the data. In SVM, it is better to scale the data always because it will extremely improve the results. Therefore, to be cautious with big dataset, it may lead to the increase in the training time.

2.3. J48

The J48 algorithm is developed for the MONK project along with WEKA (Cooper & Herskovits, 1991). The algorithm is an extension for C4.5 decision tree algorithm (Salzberg, 1994). There are many options for tree pruning in case of J48 algorithm. The classification algorithms convenient in WEKA try to clarify the results or prune. This method will help us to produce more generic results and also can be used to correct potential over-fitting issues. J48 helps to recursively classify

until each of the leaf is pruned, that is to classify as close knit to the data. Hence, this will help to ensure the accuracy although excessive rules will be produced. However, pruning will cause to less accuracy of a model on training data. This is due to pruning employs various means to relax the specificity of the decision tree, hopefully improving its performance on the test data. The complete concept is to increasingly generalize a decision tree until it gains a balance of accuracy together with flexibility. The J48 applies two pruning methods. The first method is known as a subtree replacement. This concludes that nodes in the decision tree can be replaced with a leaf -- which reduces the number of tests along a particular path. This process begins from the leaves of the completely formed tree and attempts backwards toward the root. The second method category known as pruning adopted in J48 is termed subtree rising. In this respect, a node can be moved upwards towards the root of the tree, replacing other nodes another way. Subtree rising repeatedly has an insignificant effect on decision tree models.

There is generally no clear way to anticipate the utility of the option, though it may be desirable to try turning it off if the induction process is catching a long time. This is because of the fact that subtree rising may be somewhat computationally complicated. Error rates are needed to make actual conclusions about which parts of the tree to rise or replace. There exist multiple ways to perform this. The straight forward way is to reserve a portion of the training data in order to test on decision tree. Reserved portion may then be adopted as test data for the decision tree, aiding to reduce potential over fitting. This method is recognized as reduced-error pruning. Though the approach is straightforward, it also decreases the overall volume of data available for training the model. For specifically small datasets, it may be advisable to avert using reduced error pruning.

2.4. Random Forest Algorithm

Random forest as is an ensemble of unpruned regression or classification trees, activated from bootstrap samples of the training data, adopting random feature selection in the tree imitation process. The prediction is made by accumulating the predictions of the ensemble by superiority voting for classification. It returns generalization error rate and is more potent to noise. Still, similar to most classifiers, RF may also suffer from the curse of learning from an intensely imbalanced training data set. Since it is constructed to mitigate the overall error rate, it will tend to focus more on the prediction efficiency of the majority class, which repeatedly results in poor accuracy for the minority class.

2.5. Naive Bayes

The Naive Bayes classifier contributes a simple method, representing and learning probabilistic knowledge with clear semantics (Nguyen & Choi, 2008). It is termed naive due to it relies on two important simplifying assumes that predictive attributes are conditionally self-reliant given the class, and it considers that no hidden attributes influence the prediction method. It is a probabilistic classifier which relies upon Bayes theorem with robust and naive independence assumptions. It is one of the best basic text classification approaches with numerous applications in personal email sorting, email spam detection, sexually explicit content detection, document categorization, sentiment detection, language detection (bin Othman & Yau, 2007). Although the naive design and oversimplified assumptions that this approach uses, Naive Bayes accomplishes well in many complicated real-world problems. All though it is often out performed by other approaches such as boosted trees, Max Entropy, Support Vector Machines, random forests etc, Naive Bayes classifier is very potential as it is less computationally intensive (in both memory and CPU) and it needs a small extent of training data. Moreover, the training time with Naive Bayes

is considerably smaller as opposed to alternative approaches.

2.6. Bayes Net

Bayesian networks are a powerful probabilistic representation, and their use for classification has received considerable attention (Bouckaert, 1994). It reflects the states of some part of a world that is being modeled and it describes how those states are related by probabilities. Bayesian network is a graphical model that encodes probabilistic relationships among variables of interest. When used in conjunction with statistical techniques, the graphical model has several advantages for data analysis. The first advantages are because the model encodes dependencies among all variables, it readily handles situations where some data entries are missing. Two, a Bayesian network can be used to learn causal relationships, and hence can be used to gain understanding about a problem domain and to predict the consequences of intervention. This classifier learns from training data the conditional probability of each attribute given the class label (Buntine, 1991; online available at "Random Tree", 2015).

2.7. Random Tree

The algorithm may deal with both regression and classification problems. Random trees are an ensemble of tree predictors which is called *forest*. The classification performs as follows: random trees classifier takes the input feature vector, categories it with individual tree in the forest, outputs the class label which received the most of "votes". In the event of a regression, the classifier response is the average of the responses over all the trees in the forest (Trier, et al., 1996). In random tree algorithm, all the trees are trained with the same parameters but on different training sets. These sets are created from the original training set adopting the bootstrap procedure and for each training set, randomly choose the same number of vectors as in the initial set. The vectors are

chosen with replacement. That is, some vectors occur more than once and some will be absent. In random trees, there is no need for any accuracy estimation techniques, like cross-validation or bootstrap, or a separate test set to obtain an estimate of the training error. The error is estimated internally during the training.

2.8. Dataset Description

The handwritten digit recognition is an extensive research topic which gives a comprehensive survey of the area including major feature sets, learning datasets, and algorithms (Koerich, et al., 2003). Contrary to optical character recognition which focuses on recognition of machine-printed output, special fonts can be used, the variability between characters along with the same size, font, and font attributes is fairly small.

The feature extraction and the classification technique play an important role in offline character recognition system performance. Various feature extraction approaches have been proposed for character recognition system (Rahman, et al., 2002). The problems faced in handwritten numeral recognition has been studied while using the techniques like Dynamic programming, HMM, neural network, Knowledge system and combinations of above techniques (Chandrasekaran, et al., 1984). Wider ranging work has been carried out for digit recognition in so many languages like English, Chinese, Japanese, and Arabic. In Indian mainly worked in Devanagari, Tamil, Telugu and Bengali numeral recognition (Seewald, 2005).

In our experiment, we used digit dataset provided by Austrian Research Institute for Artificial Intelligence, Austria (see **Figure 1**). This data set indicate that arbitrary scaling and a blur setting of 2.5 for the Mitchell down-sampling filter should perform well and used down-sample to 16x16 pixels.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 5 | 9 | 7 | 6 | 3 | 5 | 0 | 8 |
| 4 | 5 | 8 | 6 | 9 | 3 | 2 | 9 | 7 | 2 |
| 3 | 3 | 3 | 9 | 5 | 0 | 3 | 2 | 3 | 0 |
| 1 | 1 | 4 | 0 | 2 | 1 | 5 | 3 | 3 | 6 |
| 8 | 6 | 2 | 0 | 4 | 0 | 4 | 5 | 3 | 9 |
| 8 | 5 | 4 | 2 | 2 | 7 | 1 | 6 | 0 | 9 |
| 1 | 7 | 0 | 3 | 9 | 7 | 2 | 0 | 7 | 7 |
| 2 | 6 | 5 | 7 | 6 | 4 | 2 | 2 | 2 | 9 |
| 4 | 4 | 4 | 2 | 0 | 6 | 9 | 4 | 8 | 3 |
| 1 | 5 | 0 | 3 | 4 | 6 | 8 | 2 | 5 | 1 |

Figure 1. A small portion of handwritten dataset example

This dataset is divided in two parts training set and testing set. Training set has 1893 samples and test set has 1796 samples. The detail of the dataset is provided in ([online available at "Weka Software Documentation", accessed, August 2015](#)).

3. EXPERIMENTAL TOOLS

WEKA is a prominent suite of machine learning which is written in Java and developed at the University of Waikato. It is free software accessible under the GNU General Public License. It contains a collection of algorithms and visualization tools for predictive modelling, data analysis, along with graphical user interfaces for smooth access to this functionality. It supports various standard data mining tasks, more particularly, data pre-processing, classification, visualization, clustering, feature selection, regression. All of Weka's approaches are predicated on the assumption that the data is convenient as a single flat file or relation, where each data point is characterized through a fixed number of attributes (Le, 2013).

WEKA has numerous user interfaces. Its main user interface is the Explorer, however essentially the same functionality can be accessed by the component-based Knowledge Flow interface and from the command line.

The Experimenter allows the systematic comparison of the predictive performance of the Weka's machine learning algorithms on an accumulation of datasets.

4. RESULTS AND DISCUSSION

WEKA has several graphical user interfaces that enable easy access to the underlying functionality. To gauge and investigate the performance on the selected methods or algorithms namely Support Vector Machine, Multilayer Perceptron, Random Forest Algorithm, Random Tree, Naïve Bayes, Bayes Net and j48 Decision tree algorithms are used. We used the same experiment procedure as suggested by WEKA.

In WEKA, all dataset is considered as instances and features in the data are also known as attributes. The experiment results are partitioned into several sub division for easier analysis and evaluation. In the first part, correctly and incorrectly classified instances will be divided in numeric and percentage value and subsequently Kappa statistic, mean absolute error and root mean squared error will be in numeric value. Experiment shows the relative absolute error and root relative squared error in percentage (%) for references and in evaluation process. Our simulation results are shown in **Table 1** and **2**. In Table 1 mainly summarizes the result based on accuracy and time taken for each simulation in our experiment. Moreover, **Table 2** shows the result based on error during the simulation in WEKA.

Based on the above **Table 1**, the highest accuracy is 90.37 % and the lowest is 75.06%. The other algorithm yields an average accuracy of around 83.89%. In fact, the highest accuracy belongs to the Multilayer Perceptron classifier, followed by Support Vector Machine with a percentage of 87.97% and subsequently Random Forest Algorithm (85.75%), Bayes Net (84.35%), Naïve Bayes (81.85%),

Table 1. Simulation result based on accuracy and time consumption.

| Name of Algorithms | Correctly Classified Instances % (value) | Incorrectly Classified Instances % (Value) | Time Taken (seconds) | Kappa Statistic |
|------------------------|--|--|----------------------|-----------------|
| Multilayer Perceptron | 90.37 | 9.63 | 3.15 | 0.893 |
| Support Vector Machine | 87.97 | 12.03 | 0.56 | 0.8664 |
| Random Forest | 85.75 | 14.25 | 0.44 | 0.8416 |
| Bayes Net | 84.35 | 15.65 | 0.86 | 0.8262 |
| Naïve Bayes | 81.85 | 18.15 | 3.45 | 0.7983 |
| J48 | 79.51 | 20.49 | 0.53 | 0.7722 |
| Random Tree | 85.6 | 24.94 | 0.55 | 0.7228 |

Table 2. Simulation result based on different error

| Name of Algorithms | Mean Absolute Error | Root Mean Squared Error | Relative Absolute Error (%) | Root Relative Squared Error (%) |
|------------------------|---------------------|-------------------------|-----------------------------|---------------------------------|
| Multilayer Perceptron | 0.023 | 0.1231 | 12.78 | 41.04 |
| Support Vector Machine | 0.1611 | 0.2734 | 89.49 | 91.15 |
| Random Forest | 0.0593 | 0.1532 | 32.97 | 51.06 |
| Bayes Net | 0.0312 | 0.1745 | 17.36 | 58.15 |
| Naïve Bayes | 0.0361 | 0.1879 | 20.06 | 62.65 |
| J48 | 0.0444 | 0.1957 | 24.66 | 65.25 |
| Random Tree | 0.0499 | 0.2234 | 27.72 | 74.45 |

j48 (79.51%), and Random Tree (75.06%). Kappa statistics value ranges from 0 to 1. Value 0 means totally disagreement and 1 means full agreement. It checks the reliability of Classifying algorithm on dataset. The total time, mean absolute error, root mean absolute error, relative absolute error, and root relative absolute error is also a crucial parameter to build the model in comparing the different classification algorithm. Mean absolute error is the mean of overall error made by classification algorithm and least the error will be best classifier. In the **Table 2** Multilayer Perceptron has least 0.023 mean absolute errors among all seven algorithms.

Based on (Ciregan, 2012), experimental results reveal that it is possible to train a face detector without having to label images as containing a face or not. Their experiment is

only sensitive to high-level concepts such as cat faces and human bodies. Multi-column deep neural networks for image classification have been presented in (Schmidhuber, 2015). They only improve the state-of-the-art on a plethora of common image classification benchmarks. Supervised learning unsupervised learning, reinforcement learning & evolutionary computation, and indirect search for short programs encoding deep and large networks has been presented in (Das et al., 2014). They proposed only how different technique can be used for pattern recognition. In (Das et al., 2014) recognition of handwritten bangla basic characters and digits using convex hull-based feature set has been proposed (Liu, et al., 2013). Their experiment result shows that with a database of 10000 samples, the maximum recognition rate of

76.86% is observed for handwritten Bangla characters. Online and offline handwritten Chinese character recognition has been proposed in literature (Zhang, *et al.*, 2018). Their experiment result reported that the highest test accuracies of 89.55% for offline. In our experiment, different machine learning algorithm has been used for handwrite digit recognition we obtained the highest value of 90.37% for accuracy obtained for Multilayer Perceptron.

5. CONCLUSION

This study has investigated a representation of isolated handwritten digits for allowing their effective recognition. This paper used different machine learning algorithm for recognition of handwritten numerals. In any recognition process, the important problem is to address the feature extraction and correct classification approaches. The proposed algorithm tries to address both the factors and

well in terms of accuracy and time complexity. The overall highest accuracy of 90.37% is achieved in the recognition process by Multilayer Perceptron. This work is carried out as an initial attempt, and the aim of the paper is to facilitate for recognition of handwritten numeral without using any standard classification techniques.

5. ACKNOWLEDGEMENTS

The authors wish thank Department of Information and Communication Technology Mawlana Bhashani Science and Technology University Santosh, Tangail-1902, Bangladesh.

6. AUTHORS' NOTE

The author(s) declare(s) that there is no conflict of interest regarding the publication of this article. Authors confirmed that the data and the paper are free of plagiarism.

7. REFERENCES

- AlKhateeb, J. H., Pauplin, O., Ren, J., and Jiang, J. (2011). Performance of hidden Markov model and dynamic Bayesian network classifiers on handwritten Arabic word recognition. *Knowledge-based systems*, 24(5), 680-688.
- Bhowmik, S., Roushan, M. G., Sarkar, R., Nasipuri, M., Polley, S., and Malakar, S. (2014). Handwritten bangla word recognition using hog descriptor. *Proceedings of emerging Applications of Information Technology (EAIT), 2014*, 193-197.
- bin Othman, M. F., and Yau, T. M. S. (2007). Comparison of different classification techniques using WEKA for breast cancer. *Proceeding of 3rd Kuala Lumpur International Conference on Biomedical Engineering, 2006*, 520-523.
- Bouckaert, R. R. (1994). Properties of Bayesian belief network learning algorithms. *Uncertainty Proceedings, 1994*, 102-109.
- Buntine, W. (1991). Theory refinement on Bayesian networks. *Uncertainty Proceedings, 1991*, 52-60.

- Chandrasekaran, R., Chandrasekaran, M., and Siromoney, G. (1984). Computer recognition of Tamil, Malayalam and Devanagari characters. *IETE Journal of Research*, 30(6), 150-154b.
- Ciregan, D., Meier, U., and Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. *Proceedings of Computer vision and pattern recognition (CVPR), 2012*, 3642-3649.
- Cooper, G. F., and Herskovits, E. (1991). A Bayesian method for constructing Bayesian belief networks from databases. *Uncertainty Proceedings, 1991*, 86-94.
- Das, N., Pramanik, S., Basu, S., Saha, P. K., Sarkar, R., Kundu, M., and Nasipuri, M. (2014). Recognition of handwritten Bangla basic characters and digits using convex hull-based feature set. *arXiv preprint arXiv:1410.0478*.
- Das, N., Sarkar, R., Basu, S., Kundu, M., Nasipuri, M., and Basu, D. K. (2012). A genetic algorithm-based region sampling for selection of local features in handwritten digit recognition application. *Applied Soft Computing*, 12(5), 1592-1606.
- Firdaus, C., Wahyudin, W., & Nugroho, E. P. (2017). Monitoring System with Two Central Facilities Protocol. *Indonesian Journal of Science and Technology*, 2(1), 8-25.
- John, G. H., and Langley, P. (1995). Estimating continuous distributions in Bayesian classifiers. *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence, 1995*, 338-345.
- Koerich, A. L., Sabourin, R., and Suen, C. Y. (2003). Large vocabulary off-line handwriting recognition: A survey. *Pattern Analysis and Applications*, 6(2), 97-121.
- Kruse, R., Borgelt, C., Klawonn, F., Moewes, C., Steinbrecher, M., & Held, P. (2013) Multi-Layer Perceptrons. *Computational Intelligence, 2013*, 47-81.
- Le, Q. V. (2013). Building high-level features using large scale unsupervised learning. *Proceedings of Acoustics, Speech and Signal Processing (ICASSP), 2013*, 8595-8598.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- Liu, C. L., Nakashima, K., Sako, H., and Fujisawa, H. (2003). Handwritten digit recognition: benchmarking of state-of-the-art techniques. *Pattern recognition*, 36(10), 2271-2285.
- Liu, C. L., Yin, F., Wang, D. H., and Wang, Q. F. (2013). Online and offline handwritten Chinese character recognition: benchmarking on new databases. *Pattern Recognition*, 46(1), 155-162.

- Neves, R. F., Lopes Filho, A. N., Mello, C. A., and Zanchettin, C. (2011). A SVM based off-line handwritten digit recognizer. *Proceedings of Systems, Man, and Cybernetics (SMC), 2011*, 510-515).
- Nguyen, H. A., and Choi, D. (2008). Application of data mining to network intrusion detection: classifier selection model. *Proceedings of Asia-Pacific Network Operations and Management Symposium, 2008*, 399-408.
- Orru, G., Pettersson-Yeo, W., Marquand, A. F., Sartori, G., and Mechelli, A. (2012). Using support vector machine to identify imaging biomarkers of neurological and psychiatric disease: a critical review. *Neuroscience and Biobehavioral Reviews*, 36(4), 1140-1152.
- Pereira, F., Mitchell, T., & Botvinick, M. (2009). Machine learning classifiers and fMRI: a tutorial overview. *Neuroimage*, 45(1), S199-S209.
- Perwej, Y., & Chaturvedi, A. (2012). Machine recognition of Hand written Characters using neural networks. *arXiv preprint arXiv:1205.3964*.
- Plamondon, R., & Srihari, S. N. (2000). Online and off-line handwriting recognition: a comprehensive survey. *IEEE Transactions on pattern analysis and machine intelligence*, 22(1), 63-84.
- Rahman, A. F. R., Rahman, R., and Fairhurst, M. C. (2002). Recognition of handwritten Bengali characters: a novel multistage approach. *Pattern Recognition*, 35(5), 997-1006.
- Random trees, online: http://docs.opencv.org/modules/ml/doc/random_trees.html, Accessed on August 2015.
- Salzberg, S. L. (1994). C4. 5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. *Machine Learning*, 16(3), 235-240.
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61, 85-117.
- Seewald, A. K. (2005). Digits-a dataset for handwritten digit recognition. *Austrian Research Institut for Artificial Intelligence Technical Report, Vienna (Austria)*.
- Seewald, A. K. (2011). On the brittleness of handwritten digit recognition models. *ISRN Machine Vision*, 2012.
- Tokas, R., & Bhadu, A. (2012). A comparative analysis of feature extraction techniques for handwritten character recognition. *International Journal of Advanced Technology & Engineering Research*, 2(4), 215-219.

Trier, Ø. D., Jain, A. K., & Taxt, T. (1996). Feature extraction methods for character recognition- a survey. *Pattern recognition*, 29(4), 641-662.

Watada, J., & Pedrycz, W. (2008). A fuzzy regression approach to acquisition of linguistic rules. *Handbook of Granular Computing*, 719-732.

Weka Software Documentation, online: <http://www.cs.waikato.ac.nz/ml/weka/>, accessed on August 2015.

Zhang, X. Y., Yin, F., Zhang, Y. M., Liu, C. L., & Bengio, Y. (2018). Drawing and recognizing Chinese characters with recurrent neural network. *IEEE transactions on pattern analysis and machine intelligence*, 40(4), 849-862.