Steady Model for Classification of Handwritten Digit Recognition

*Anujay Ghosh[1], Aruna Pavate[2], Vidit Gholam[3], Gauri Shenoy[4], Shefali Mahadik[5]

[1,2,3,4,5]Atharva College of Engineering, Mumbai University, Mumbai 400095

{*anujay.ghosh, arunaapavate, viditgholam, gaurishenoy99, shefalimahadik007} @gmail.com

Abstract. Handwritten digit recognition plays important role not only in computer vision but also in pattern recognition. Handwritten Digit Recognition is the competence of a machine to receive, calculate and decipher a human handwritten input from sources such as handwritten manuscripts and other, especially created before the advent of a digital revolution and digital images. This work implements the system to read the handwritten digits with a custom novel method identical to the amalgamation different techniques included Principal Component Analysis, Support Vector Machine and K- Nearest Neighbours to recognize and classify handwritten digits into their respective labels. PCA algorithm finds out the best linear combinations of the original features so that the variance along the new feature is maximum. Recognition of characters done using KNN non-parametric machine learning algorithm and SVM lowers the generalization error of the overall classifier. The proposed work does the analysis on Digits data set having total of 70000 image samples. The performance of the system analyzed using different measurement metrics like precision, recall, f1-score, support and the recognition of the patterns in the images shows the result with classification accuracy of 97%.

Keywords: K- Nearest Neighbour, Support Vector Machine, Principal Component Analysis, classification, handwritten digit

1 Introduction

The human capability to examine and categorize the objects, scenes is very useful skill, researcher tried to implement through the machine learning algorithm in many domains including Education, Sports, Transportation, Oil and Gas, Financial Services, Marketing and Sales, Government, Health-care and in many safety critical applications like finger print recognition, face recognition and many more [1]. Handwriting digit recognition is the one of the major applications in machine learning applied in many wide ranges of real-life applications such as signature identification and verification, zip code recognition in postal mail categorization, form processing, handwritten digit verification in bank, fraud detection etc. Handwritten digit recognition plays crucial role in Optical Character Recognition (OCR), in pattern recognition [2]. There are many devices such as smart-phones, tablets that can take handwriting as an input to a touch screen via a finger or using an electronic stylus. This allows user to quickly transfer the text to the devices which helps especially for the selective individuals who are not well versed with input devices such as keyboards to write text faster rather than typing slowly through input devices. Recognition of such text is very hard even by humans. Thus, a system supports an automatic recognition of text would be very helpful in many applications.

1.1 Need of the System

Handwritten Digit Recognition system is developed to improve on the accuracy of the existing solutions to achieve higher accuracy and reliable performance. Over the last decades , many machine learning algorithms made use of impressive handwritten digit recognition techniques such as Baseline Linear Classifier, Baseline Nearest Neighbour Classifier, pairwise Linear Classifier, Radial basis network, Large Fully Connected Multi-Layer Neural Network, Tangent Distance Classifier, Optimal Margin Classifier[4], Support Vector Machine (SVM)[6][9][13][30][32],CNN[9], Fuzz[10] Neural Network[7],[11],[12],[16][24],[30],PCA[8][13],CNN-SVM classifier [14][23] , KNN[21], recurrent neural network (RNN) [22], DNN[28] classifiers, and many more.

However, still there are some challenges need to be solved. As handwritten characters are different in writing style, stroke thickness, deformation, rotation etc., it’s difficult to recognize [5] [20] [21] [22]. The main challenge in Handwriting recognition system is to classify a handwritten digit based on black and white images. Furthermore, to meet the industry need, accuracy and robustness to the variation in writing style of the individual must be high.

1.2 Scope of the System

The digital world's advent began a mere century or two ago, but scriptures and books after books have been handwritten by human scholars from the beginning of mankind. Accepting the digital world first begins with the task of integrating the scripts that came into existence before the rise of computers and technology. Thus, this conversion and integration must begin with the most common values in the world that transcend different languages as well - numbers.

The problem is to categorize handwritten digits into ten distinct classes with accuracy as high as possible. The digit ranges from zero (0) to nine (9). In this work, we utilized the Support Vector Machines(SVMs), Principle Component Analysis(PCA) and K-Nearest Neighbor (KNN) techniques, by compounding to form a novel method to solve the problem. The experiment applied on Digit data set [34] [35] taken from the well-known MNIST (Modified National Institute of Standards and Technology) data set [35].

2 Related Work

For developing handwritten digit recognition, literature presents a number of researches that have made use of machine learning techniques. Among them, to support few techniques related to work have been presented below.

Matan et. al. developed a neural network architecture for recognizing handwritten digits in a real world. This network has 1% error rate with about 7% reject rate on handwritten zip code digits provided by the U.S portal Service [3]. Jitendra Malik et al. developed simple and an easy approach for finding out the resemblance between shapes and utilize it for object recognition. The proposed approach tested on COIL data set, silhouette, trademarks and Handwritten digits [20].

O. S M Shamim et al. presented an approach to off-line handwritten digit recognition. The main problem is the capability to develop a cost-effective algorithmic program that can acknowledge hand written digits and which is submitted by users by the way of a scanner, tablet, and other digital devices [24].

Caiyun Ma et al. proposed an approach based on specific feature extraction and deep neural network on MNIST database. The proposed work compared with SOM()

[13],P-SVM [18] and result shows the proposed algorithm with accuracy 94.2% with 24 dimension & showed that the deep analysis is more beneficial than traditional in terms of visualization of features[28]. Anuj Dutt et al, compares the results of some of the most widely used Machine Learning Algorithms like SVM, KNN & RFC 4 and with Deep Learning algorithm like multilayer CNN using Keras with Theano and Tensor-flow. Result showed the accuracy of 98.70% using CNN (Keras+Theano) as compared to 97.91% using SVM, 96.67% using KNN, 96.89% using RFC and the lowest error rate 1.28% using Convolution Neural Network [29]. Chayaporn Kaensar presented comparative analysis using three different algorithms like Neural network, support Vector Machine and K- Nearest Neighbour. The analysis of the presented work demonstrates that the SVM is the best classifier with 96.93% accuracy with more time required for training as compared to neural network and K- Nearest Neighbour [30].

Mohd Razif Shamsuddin et al. presented handwritten digit recognition on MNIST data set. In this work four different methods (Logistic regression, Random Forest, Extra Trees classifier and Convolution neural network) applied on normalized MNIST data set and binary data set. The analysis result shows the Convolution neural network gives the system validation with best result 99.4% on normalized data set and 92.4% on binary data set using extra trees algorithm. The analysis shows that the system works better on normalized data set [31]. Saeed AL-Mansoori proposed Multilayer Perceptron (MLP) Neural Network to solve the problem of the handwritten digit recognition. The system performance is observed on MNIST data set by altering the number of hidden layers, the number of iterations and result showed the overall training accuracy of 99.32% and testing accuracy of 100% [32].

Cheng-Lin Liu et al. presented handwritten digit recognition on binary and gray images using eight different classifiers like K-NN, MLP, PC, RBF, LVQ, DLQDF, SVC-poly and SVC-rbf tested on three different data sets CENPARMI, CEDAR, MNIST. The presented work is concluded as SVC-rbf gives the highest accuracy among all the algorithms but this algorithm is extremely expensive in memory space and computation [33]. In addition to the above, other important work include research on local similarity [15] [25], prototype generation techniques [17], handwriting verification [19], trajectory and velocity modelling [26], and feature extraction [27].

3 MATERIALS AND METHODS

The work is implemented and tested in the following system requirements: Intel i-3 or later Processor, Minimum 2 GB RAM Minimum 2 GB Graphics Processing Unit, Operating System (Windows 7 and above), Anaconda Python 3.7. All the algorithms tried using scikit-learn Python library, version 0.17.1.

3.1 **Data Set**

The proposed system implemented and tested using MNIST dataset (Modified National Institute of Standards and Technology database). The MNIST dataset contains handwritten digits having 60,000 examples in the training set and 10,000 examples in the test set. The MNIST dataset associated with NIST dataset which is the super-set of MNIST. The size of the image is 28 x 28 pixel= 748 pixels. There are close to 60000 images in the combined data-set that can be used for training and judge the system. The data set contains the input and likelihood that the image belongs to different classes (i.e. the machine-encoded digits, 0-9) [34] [35].

3.2 Methods

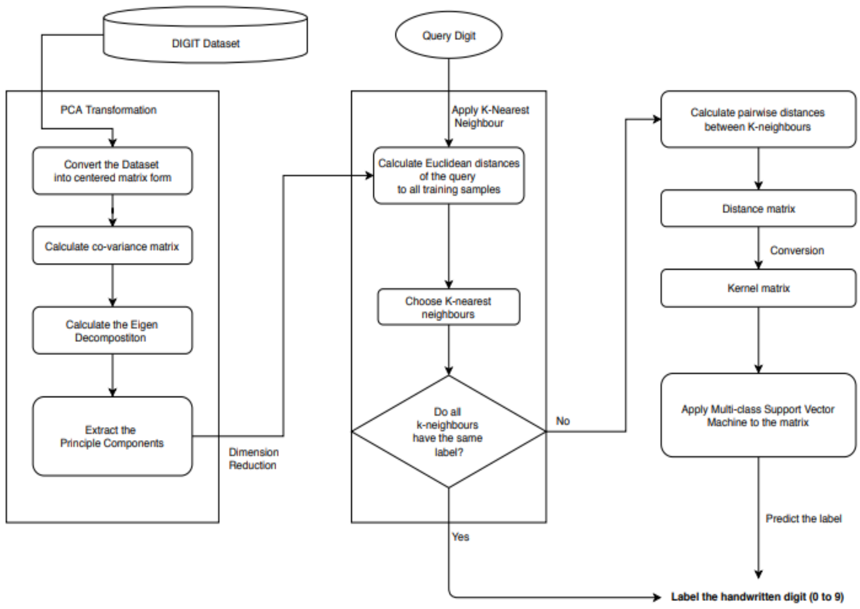


Fig. 1. Working of the proposed system for Handwritten Digit Recognition

Fig.1. shows proposed approach is an association of PCA, KNN and SVM algorithm to improve the classification accuracy. The PCA algorithm helps to reduce the number of attributes which contribute more towards classification. The first step is to load the data-set and abstract the feature columns with target columns. The size of the data-set is rather large (60000 samples with 784 features) thus extraction of features from the original large dimensional features of the data done using PCA in the initial stage.

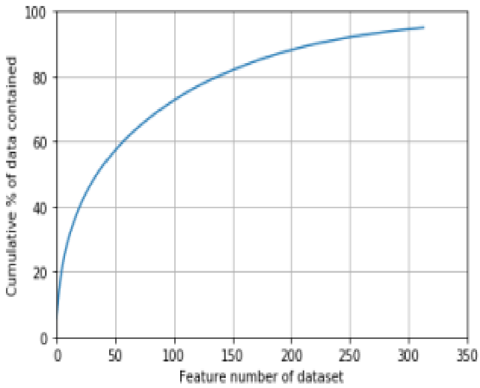


Fig.2. The amount of data v/s component number First 314 principal components as the extracted features using PCA

The first 60 features can explain approximately 97% of total substance (in terms of total variance retained), which fulfil to be typical of the information in the original data-set as shown in Fig.2. Thus, the first 60 principal components are implemented as the extracted features. The data is then split into training and testing sets. The simple implementation of SVM-KNN goes as follows: The KNN model is created and fit to the training set values which trains the KNN classifier. For a query, it is necessary to compute the Euclidean distances of the query to all the training samples and pick the K-nearest neighbours. The general value of Euclidean distance (d) is calculated using equation 1.

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(qi - pi)^2} \tag{1}$$

Where, p is the First data point, q is the second data point and n is the number of dimensions in data point.

Table 1. Initial Test

# of features selected	penalty parameter c	Prediction Time(Test)	Accuracy(Train)	Accuracy(Test)
2	0.5	51.002	1.0	0.964

If the K-neighbours (excluding the query) all have the same class, the query is flagged with the respective class same as its neighbours. Further, it calculates the distance between k neighbour pairwise and convert the distance matrix into kernel matrix. Finally, the multiclass SVM is applied to the kernel matrix to flag the query. In the initial implementation, 314 principal components are extracted and use parameters values of k=2 for KNN and C=0.5 for SVM. This resulted in an accuracy score of 0.964 as shown in Table 1. Then, number of iterations used to tune the k parameter by changing its value while keeping the other parameter values as the same and observing the results. The same steps applied for 20 distinct values of k (number of neighbours), keeping the c (penalty parameter) value constant.

The bold value in the Table 2 highlights the best k value with respect to highest test set accuracy achieved as well as fastest prediction time taken to do so. Fig.3. plots accuracy of test set with respect to changing k values taken from Table 2.

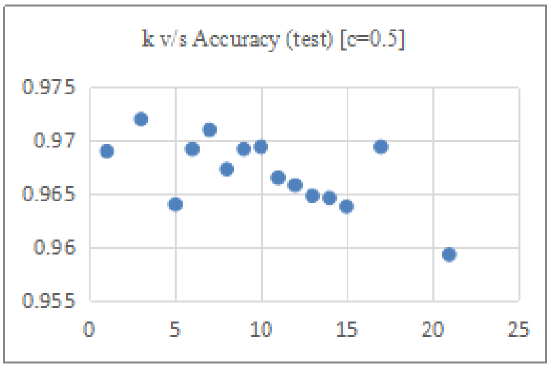


Fig. 3. k v/s Accuracy(test)

The value of k=3 gives the highest accuracy as shown in the Table 2, hence keeping k=3 constant and changing the values of c to understand variation in accuracy with change in c as follows:

Table 2. For different values of k with c=0.5 accuracy observed

# of features selected	c	Prediction Time(Test)	Accuracy(Train)	Accuracy(Test)
5	0.5	42.3s	1	0.964
3	**0.5**	**54.9s**	**1**	**0.972**
1	0.5	55.6s	1	0.969
7	0.5	54.8s	1	0.971
8	0.5	65.5s	1	0.9673
9	0.5	66.5s	1	0.9692
10	0.5	62.2s	1	0.9694
11	0.5	69.9s	1	0.9665
12	0.5	68.8s	1	0.9658
13	0.5	70.1s	1	0.9648
14	0.5	70.6s	1	0.9646
15	0.5	71.6s	1	0.9638
6	0.5	61.8s	1	0.9692
17	0.5	68.6s	1	0.9694
21	0.5	76.9s	1	0.9593

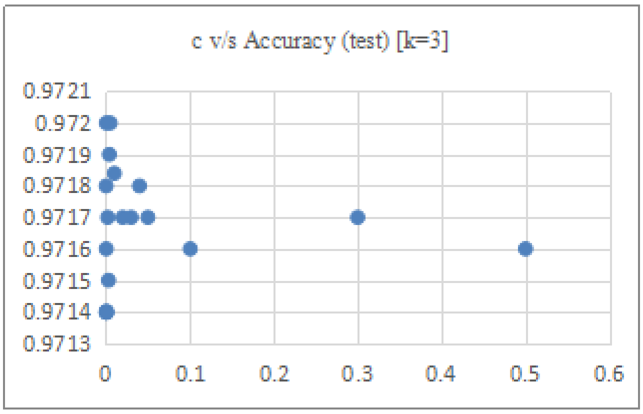


Fig.4. Accuracy(test) with respect to C

Table 3. For k= 3 with c=0.5 accuracy observed

# of features selected	c	Prediction Time(Test)	Accuracy(Train)	Accuracy(Test)
3	0.01	54.9	1	0.97184
3	**0.02**	**55.2**	**1**	**0.9717**
3	0.03	55.1	1	0.9717
3	0.04	55.3	1	0.9718
3	0.05	55.2	1	0.9717
3	0.001	55	1	0.9714
3	0.002	61.9	1	0.9717
3	0.003	54.9	1	0.9715
3	0.004	54.9	1	0.9719
3	0.005	54.58	1	0.972
3	0.1	55.14	1	0.9716
3	0.3	54.89	1	0.9717
3	0.5	55.28	1	0.9716
3	0.0001	55.35	1	0.9714
3	0.0002	55.95	1	0.9718

From the above result concluded that the best value of k is k=3. However, changes in the C value do not impact the final accuracy score. This result is quite unusual because the input space to the SVM is very small (size 3) and the SVM algorithm can classify the dataset pretty quickly hence changing the parameters does not have much effect on the accuracy. The final solution then uses k=3, C=0.005, and yields and accuracy score of 0.9720 as shown in Table 2.

4 RESULTS & DISCUSSIONS

4.1. **Dataset Analysis**

Digits dataset has a total of 70000 image samples (42000 training set and 28000 testing set samples, each with 784 features). Figure 5 represents the number of occurrences of all the digits vs. Labels (i.e. 0-9) present in the training dataset of 42000 samples.

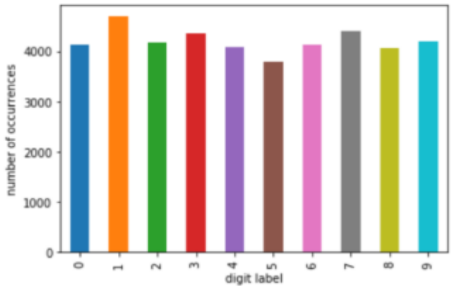


Fig.5. Occurrence of each digit in the training set

4.2. Classification Report

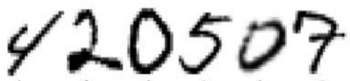
Figure 6. below displays the extensive classification report containing details about the precision of the model, recall, f1-score and support.

	precision	recall	f1-score	support
0	0.99	0.99	0.99	1653
1	0.98	0.99	0.98	1874
2	0.99	0.97	0.98	1671
3	0.97	0.96	0.96	1740
4	0.98	0.97	0.98	1629
5	0.96	0.96	0.96	1518
6	0.98	0.99	0.98	1655
7	0.97	0.97	0.97	1760
8	0.98	0.95	0.96	1625
9	0.93	0.97	0.95	1675
micro avg	0.97	0.97	0.97	16800
macro avg	0.97	0.97	0.97	16800
weighted avg	0.97	0.97	0.97	16800

Fig.6. Classification Report

4. **3 Manual Result Testing**

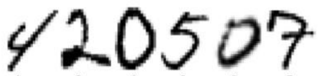
By manually taking out digits from the data set, plotting their 28px by 28px square image using image show function in matplotlib, and comparing the results with the predicted outcome, we get the following:



4 2 0 5 0 7

Fig.7. Actual images with their true labels

The actual images and their labels are as show in Figure 7. The same images were fed to the model and the model’s prediction was as show in Figure 8 below. This model incorrectly labels the fifth image and identifies it as a 9, but the correct label is 0.



4 2 0 5 9 7

Fig.8. Images with their predicted labels

4.4. Confusion Matrix

Figure 9 below visualizes the confusion matrix. It plots the predicted values versus actual values where the actual labels are represented on Y-axis and predicted values are represented on X-axis. This model has been applied to the testing dataset. The model predicted the label to be 0 correctly 1636 times.

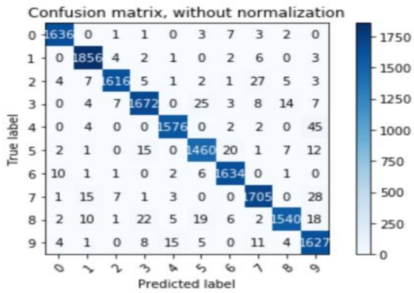


Fig.9. Confusion matrix, without normalization

5 CONCLUSION & FUTURE SCOPE

In this work, the model is acceptable for providing a solution for classifying handwritten digits into their respective labels in the MNIST dataset as it is able to accurately categorize well with accuracy quite close to humans using a combination of two classification techniques such as Support Vector Machine and K-Nearest Neighbours. However, the model is still in its rudimentary stages & useful in a limited domain. To solve large problem for recognizing multiple digits in an image or to recognize arbitrary multi-digit information in unspecified or not constrained natural images several changes need to be done in this work.

References

1. https://www.outsource2india.com/software/articles/machine-learning-applications-how-it-works-who-uses-it.asp last accessed date: Aug 2019
2. Xiao-Xiao Niu, and Ching Y. Suen, "A novel hybrid CNN-SVM classifier for recognizing handwritten digits," in Pattern Recognition, vol. 45, pp. 1318-1325, 2012.
3. O.matan, R.K. Kiang, C.E. Stenard, B. Boser, J.S. Denker, D.Henderson, R.E. Howard, W. Hubbard, L.D. Jackel and Y.Le cun , "Handwritten Character Recognition Using Neural Network Architecture", In Proceedings of the 4th USPS advanced Technology Conference,Washington D.C, pp 1003-1011, November 1990.
4. Y. LeCun, L. Jackel, and L. Bottou, "Comparison of learning algorithms for handwritten digit recognition," Proceedings of the International Conference on Artificial Neural Networks, Nanterre, France, pp. 53-60, 1995

5. Serge Belongie, Jitendra Malik, Jan Puzicha,Shape Matching and Object Recognition Using Shape Contexts,IEEE transctions on patern analysis and Machine Intelligence,Vol 24, No 24, April 2002 pp 509-522
6. Dennis Decoste, Bernhard scholkope,Training Invariant Support Vector Machines,Machine Learning, 46, 161–190, 2002 Kluwer Academic Publishers.
7. Kussul Ernst, and Tatiana Baidyk, "Improved method of handwritten digit recognition tested on MNIST database," in Image and Vision Computing , vol. 22 (12), pp. 971-981, 2004
8. Fabien Lauer, Ching Y. Suen, and Gérard Bloch, "Atrainable feature extractor for handwritten digit recognition," in Pattern Recognition vol. 40, pp. 1816-1824, 2007.
9. Monji Kherallah, Lobna Haddad, Adel M. Alimi, and Amar Mitiche, "On-line handwritten digit recognition based on trajectory and velocity modeling," in Pattern Recognition Letters, vol. 29, pp. 580-594, 2008.
10. Chichang Jou and Hung-Chang Lee, "Handwritten numeral recognition based on simplified structural classification and fuzzy memberships," in Expert Systems with Applications, vol. 36, pp. 11585-11863, 2009
11. Apurva A. Desai, "Gujarati handwritten numeral optical character reorganization through neural network," in Pattern Recognition, vol. 43, pp. 2582-2589, 2010
12. Ueli Meier, Dan Claudiu Ciresan, and Juergen Schmidhuber, "Better digit recognition with a committee of simple Neural Nets," in International Conference on Document Analysis and Recognition, vol.1, pp.1250-1254, 2011
13. Li Rui, and Zhang Shiqing, "Handwritten digit recognition based on principal component analysis and support vector machines," in International Conference on Advances in Computer Science, Environment, Ecoinformatics, and Education, pp. 595-599, 2011
14. Xiao-Xiao Niu, and Ching Y. Suen, "A novel hybrid CNN-SVM classifier for recognizing handwritten digits," in Pattern Recognition, vol. 45, pp. 1318-1325, 2012
15. Qiang Hua, Lijie Bai, and XizhaoWang, "Local similarity and diversity preserving discriminant projection for face and handwriting digits recognition," in Neurocomputing, vol. 86, pp. 150-157, 2012
16. Ciresan Claudiu Dan, Dan Ueli Meier and Luca Maria Gambardella, "Deep Big Simple Neural Nets Excel on Handwritten Digit Recognition," Neural Computation 22(12). 2013
17. S. Impedovo, F.M.Mangini, and D. Barbuzzi, "A novel prototype generation technique for handwriting digit recognition," in Pattern Recognition, vol. 47, pp. 1002-1010, 2014.
18. Wang Yimu, Pan Yun, and Long Yanchen, "Parallel implementation of handwritten digit recognition system using self-organizing map," in Journal of Zhejiang University, vol. 48, pp. 742-747, 2014.
19. Tobias Kutzner, Mario Dietze, Ingrid Bönninger, Carlos M. Travieso, Malay Kishore Dutta, and Anushikha Singh, "Online Handwriting Verification with Safe Password and Incresing Number of Features," 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN).
20. Serge Belongie, Jitendra Malik and Jan Puzicha, "Shape Matching Aand Object Recognition Using Shape Contexts", IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL 24,No 24, April 2016
21. Nurul Ilmi, Tjokorda Agung Budi W and Kurniawan Nur R, "Handwriting Digit Recognation using Local Binary Pattern Varience and K-Nearest Neighbor," 2016 Fourth International Conference on Information and Communication Technologies (ICoICT).
22. Youssouf Chherawala, Partha Pratim Roy and Mohamed Cheriet, "Feature Set Evaluation for Offline Handwriting Recognition Systems: Application to the Recurrent Neural Net-

work," IEEE TRANSACTIONS ON CYBERNETICS, VOL. 46, NO. 12, DECEMBER 2016

23. Shahab Tayeb, Matin Pirouz, Brittany Cozzens, Richard Huang, Maxwell Jay, Kyle Khembunjong, Sahan Paliskara, Felix Zhan, Mark Zhang, Justin Zhan, Shahram Latifi, "Toward data quality analytics in signature verification using a convolutional neural network", Big Data (Big Data) 2017 IEEE International Conference on, pp. 2644-2651, 2017.
24. S M Shamim, Mohammad Badrul Alam Miah, Angona Sarker, Masud Rana & Abdullah Al Jobair," Handwritten Digit Recognition using Machine Learning Algorithms", Global Journal of Computer Science and Technology: D Neural & Artificial Intelligence Volume 18 Issue 1 Version 1.0 Year 2018
25. Qiang Hua, Lijie Bai, and XizhaoWang, "Local similarity and diversity preserving discriminant projection for face and handwriting digits recognition," in Neurocomputing, vol. 86, pp. 150-157, 2012.
26. Monji Kherallah, Lobna Haddad, Adel M. Alimi, and Amar Mitiche, "On-line handwritten digit recognition based on trajectory and velocity modeling," in Pattern Recognition Letters, vol. 29, pp. 580-594, 2008.
27. Fabien Lauer, Ching Y. Suen, and Gérard Bloch, "A trainable feature extractor for handwritten digit recognition," in Pattern Recognition, vol. 40, pp. 1816-1824, 2007.
28. Caiyun Ma,Hong Zhang,Effective Handwritten Digit Recognition Based on Multi-feature Extraction and Deep Analysis,2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD),pp297-301
29. Anuj Dutt, Aashi Dutt,Handwritten Digit Recognition Using Deep Learning,International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 6, Issue 7, July 2017, ISSN: 2278 – 1323 pp 990-997
30. Chayaporn Kaensar,A Comparative Study on Handwriting Digit Recognition Classifier Using Neural Network, Support Vector Machine and K-Nearest Neighbor,P. Meesad et al. (Eds.): IC2 IT2013, AISC 209, pp. 155–163. DOI: 10.1007/978-3-642-37371-8_19 © Springer-Verlag Berlin Heidelberg 2013
31. Mohd Razif Shamsuddin, Shuzlina Abdul-Rahman(&) , and Azlinah Mohamed,Exploratory Analysis of MNIST Handwritten Digit for Machine Learning Modelling ,B. W. Yap et al. (Eds.): SCDS 2018, CCIS 937, pp. 134–145, 2019
32. Saeed AL-Mansoori,Intelligent Handwritten Digit Recognition using Artificial Neural Network,Journal of Engineering Research and Applications www.ijera.com ISSN : 2248-9622, Vol. 5, Issue 5, (Part -3) May 2015, pp.46-51
33. Cheng-Lin Liu∗ , Kazuki Nakashima, Hiroshi Sako, Hiromichi Fujisawa,Handwritten digit recognition: benchmarking of state-of-the-art techniques,2003 Pattern Recognition Society.pp 2271 – 2285
34. http://www.pymvpa.org/datadb/mnist.html
35. http://yann.lecun.om/exdb/mnist/