

Steady Model for Classification of Handwritten Digit Recognition



Anujay Ghosh, Aruna Pavate, Vidit Gholam, Gauri Shenoy and Shefali Mahadik

Abstract Handwritten digit recognition plays an 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, 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 of different techniques, including 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 is done using KNN nonparametric machine learning algorithm, and SVM lowers the generalization error of the overall classifier. The proposed work does the analysis on digit data set having a total of 70,000 image samples. The performance of the system is analysed using different measurement metrics like precision, recall, f1 score and 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 objects and scenes is a very useful skill, researchers have tried to implement this through 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 fingerprint recognition, facial recognition and many more [1]. Handwriting digit recognition is one of the major applications in machine learning applied

A. Ghosh (✉) · A. Pavate · V. Gholam · G. Shenoy · S. Mahadik
Atharva College of Engineering, Mumbai University, Mumbai 400095, India

© Springer Nature Singapore Pte Ltd. 2020
R. Sharma et al. (eds.), *Innovation in Electrical Power Engineering, Communication, and Computing Technology*, Lecture Notes in Electrical Engineering 630,
https://doi.org/10.1007/978-981-15-2305-2_32

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 a crucial role in optical character recognition (OCR) and in pattern recognition [2]. There are many devices such as smart phones and 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 that 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 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 Multilayer neural network, tangent distance classifier, optimal margin classifier [3], support vector machine (SVM) [4–8], CNN [5], fuzzy [9] neural network [7, 10–14], PCA [6, 15], CNN-SVM classifier [2, 16], KNN [17], recurrent neural network (RNN) [18] and DNN classifiers [19].

However, there are still some challenges that need to be solved. As handwritten characters are different in writing style, stroke thickness, deformation, rotation, etc., it is difficult to recognize [17, 18, 20, 21]. 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), principal component analysis

(PCA) and K -nearest neighbour (KNN) techniques, by compounding to form a novel method to solve the problem. The experiment applied on digit data set [22, 23] is taken from the well-known Modified National Institute of Standards and Technology (MNIST) data set [23].

2 Related Work

For developing handwritten digit recognition, the literature presents a number of researches that have made use of machine learning techniques. Among them, a few techniques related to the 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 US portal service [24]. Jitendra Malik et al. developed simple and an easy approach for finding out the resemblance between shapes and utilized it for object recognition. The proposed approach was tested on COIL data set, silhouette, trademarks and handwritten digits [21].

S. M. Shamim et al. presented an approach to offline handwritten digit recognition. The main problem is the capability to develop a cost-effective algorithmic program that can acknowledge handwritten digits and which is submitted by users by the way of a scanner, tablet and other digital devices [14].

Caiyun Ma et al. proposed an approach based on specific feature extraction and deep neural network on MNIST database. The proposed work is compared with SOM() [6] and P-SVM [25], and the result shows the proposed algorithm with accuracy 94.2% with 24 dimensions and showed that the deep analysis is more beneficial than traditional in terms of visualization of features [19]. Anuj Dutt et al. compared the results of some of the most widely used machine learning algorithms like SVM, KNN and RFC 4 and with deep learning algorithms like multilayer CNN using Keras with Theano and TensorFlow. The 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 [26]. 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 [7].

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) were applied on normalized MNIST data set and binary data set. The analysis result shows that the convolution neural network gives the system validation with the 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 [27]. 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 and the number of iterations, and the result showed the overall training accuracy of 99.32% and testing accuracy of 100% [8].

Cheng-Lin Liu et al. presented handwritten digit recognition on binary and grey images using eight different classifiers like KNN, MLP, PC, RBF, LVQ, DLQDF, SVC-poly and SVC-RBF tested on three different data sets CENPARMI, CEDAR and 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 [28]. In addition to the above, other important works include research on local similarity [29], prototype generation techniques [30], handwriting verification [31], trajectory and velocity modelling [5] and feature extraction [15].

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 was implemented and tested using MNIST data set (Modified National Institute of Standards and Technology database). The MNIST data set contains handwritten digits having 60,000 examples in the training set and 10,000 examples in the test set. The MNIST data set was associated with MNIST data set which is the super-set of MNIST. The size of the image is 28×28 pixels = 748 pixels. There are close to 60,000 images in the combined data set that can be used for training and judging 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) [22, 23].

3.2 Methods

Figure 1 shows that proposed approach is an association of PCA, KNN and SVM algorithms 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 (60,000 samples with 784 features); thus, extraction of features from the original large dimensional features of the data is done using PCA in the initial stage.

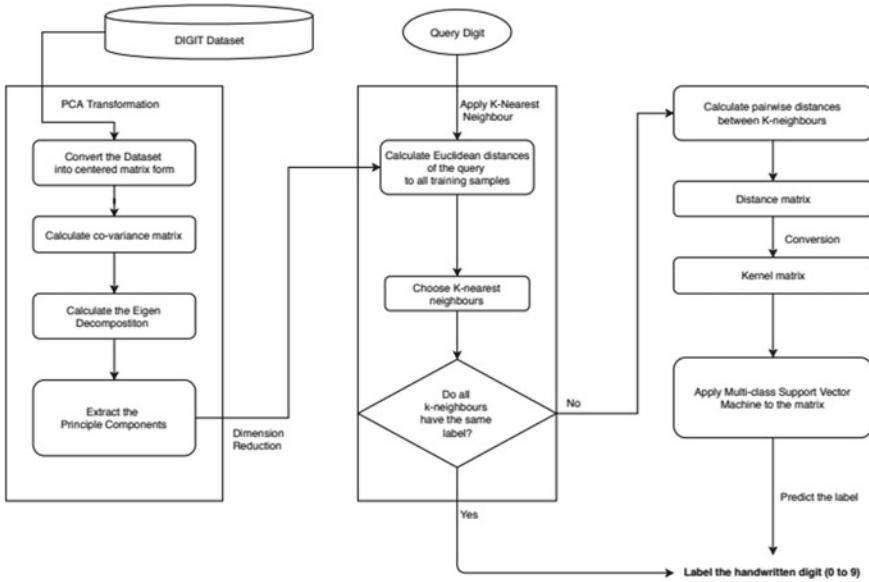
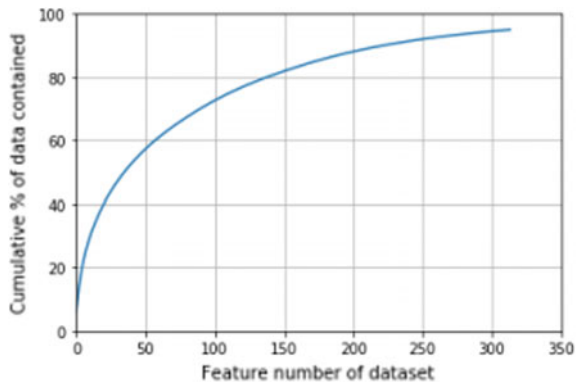


Fig. 1 Working of the proposed system for handwritten digit recognition

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 Eq. 1.

Fig. 2 Amount of data versus component number first 314 principal components as the extracted features using PCA



$$d(p, q) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (1)$$

where p is the first data point, q is the second data point and n is the number of dimensions in data point.

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 converts 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, the number of iterations is 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 are applied for 20 distinct values of k (number of neighbours), keeping the c (penalty parameter) value constant.

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

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

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

Fig. 3 k versus accuracy (test)

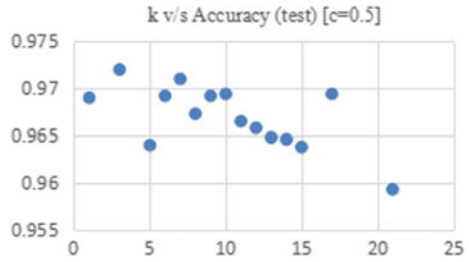
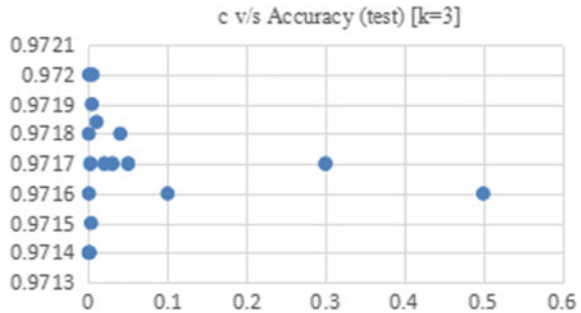


Fig. 4 Accuracy (test) with respect to C



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

The value of $k = 3$ gives the highest accuracy as shown in Table 2, hence keeping $k = 3$ constant and changing the values of c to understand variation in accuracy with change in c as follows (Fig. 4; Table 3):

From the above result, it is 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 data set 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 an accuracy score of 0.9720 as shown in Table 2.

4 Results and Discussions

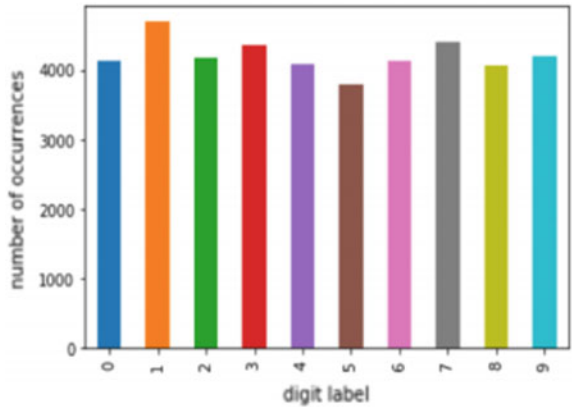
4.1 Data set Analysis

Digit data set has a total of 70,000 image samples (42,000 training set and 28,000 testing set samples, each with 784 features). Figure 5 represents the number of occurrences of all the digits versus labels (i.e. 0–9) present in the training data set of 42,000 samples.

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

Fig. 5 Occurrence of each digit in the training set



4.2 Classification Report

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

Fig. 6 Classification report

	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. 7 Images with their predicted labels



Fig. 8 Actual images with their true labels



4.3 Manual Result Testing

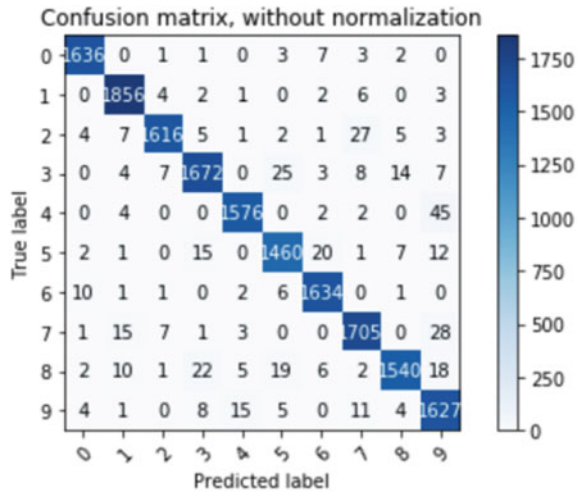
By manually taking out digits from the data set, plotting their 28 px by 28 px square image using image show function in matplotlib and comparing the results with the predicted outcome, we get the following:

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

4.4 Confusion Matrix

Figure 9 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 data set. The model predicted the label to be 0 correctly 1636 times.

Fig. 9 Confusion matrix, without normalization



5 Conclusion and Future Scope

In this work, the model is acceptable for providing a solution for classifying handwritten digits into their respective labels in the MNIST data set 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 and useful in a limited domain. To solve large problem for recognizing multiple digits in an image or to recognize arbitrary multidigit 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. Niu X-X, Suen CY (2012) A novel hybrid CNN-SVM classifier for recognizing handwritten digits. *Pattern Recogn* 45:1318–1325
3. LeCun Y, Jackel L, Bottou L (1995) Comparison of learning algorithms for handwritten digit recognition. In: *Proceedings of the international conference on artificial neural networks*, Nanterre, France, pp 53–60
4. Decoste D, Scholkope B (2002) Training invariant support vector machines. In: *Machine learning*, vol 46. Kluwer Academic Publishers, Amsterdam, pp 161–190
5. Kherallah M, Haddad L, Alimi AM, Mitiche A (2008) On-line handwritten digit recognition based on trajectory and velocity modeling. *Pattern Recogn Lett* 29:580–594
6. Li R, Zhang S (2011) 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

7. Kaensar C (2013) A comparative study on handwriting digit recognition classifier using neural network, support vector machine and K-nearest neighbor. In: Meesad P et al (eds) IC2 IT2013, AISC 209. Springer, Berlin, pp 155–163. https://doi.org/10.1007/978-3-642-37371-8_19
8. Saeed AL-M (2015) Intelligent handwritten digit recognition using artificial neural network. *J Eng Res Appl* 5(5) (Part-3):46–51. www.ijera.com. ISSN: 2248-9622
9. Jou C, Lee H-C (2009) Handwritten numeral recognition based on simplified structural classification and fuzzy memberships. *Expert Syst Appl* 36:11585–11863
10. Kussul E, Baidyk T (2004) Improved method of handwritten digit recognition tested on MNIST database. *Image Vis Comput* 22(12):971–981
11. Desai AA (2010) Gujarati handwritten numeral optical character reorganization through neural network. *Pattern Recogn* 43:2582–2589
12. Meier U, Cireşan DC, Schmidhuber J (2011) Better digit recognition with a committee of simple neural nets. In: International conference on document analysis and recognition, vol 1, pp 1250–1254
13. Claudiu Cireşan D, Meier U, Gambardella LM (2013) Deep big simple neural nets excel on handwritten digit recognition. *Neural Comput* 22(12)
14. Shamim SM, Miah MBA, Angona Sarker MR, Al Jobair A (2018) Handwritten digit recognition using machine learning algorithms. *Glob J Comput Sci Technol D Neural Artif Intell* 18(1) (Version 1.0)
15. Lauer F, Suen CY, Bloch G (2007) A trainable feature extractor for handwritten digit recognition. *Pattern Recogn* 40:1816–1824
16. Tayeb S, Pirouz M, Cozzens B, Huang R, Jay M, Khembunjong K, Paliskara S, Zhan F, Zhang M, Zhan J, Latifi S (2017) Toward data quality analytics in signature verification using a convolutional neural network. In: 2017 IEEE international conference on big data (Big Data), pp 2644–2651
17. Ilmi N, Budi WTA, Nur RK (2016) Handwriting digit recognition using local binary pattern variance and K-nearest neighbor. In: 2016 Fourth international conference on information and communication technologies (ICOICT)
18. Chherawala Y, Roy PP, Cheriet M (2016) Feature set evaluation for offline handwriting recognition systems: application to the recurrent neural network. *IEEE Trans Cybern* 46(12)
19. Ma C, Zhang H (2015) Effective handwritten digit recognition based on multi-feature extraction and deep analysis. In: 2015 12th international conference on fuzzy systems and knowledge discovery (FSKD), pp 297–301
20. Belongie S, Malik J, Puzicha J (2002) Shape matching and object recognition using shape contexts. *IEEE Trans Pattern Anal Mach Intell* 24(24):509–522
21. Belongie S, Malik J, Puzicha J (2016) Shape matching and object recognition using shape contexts. *IEEE Trans Pattern Anal Mach Intell* 24(24)
22. <http://www.pymvpa.org/datadb/mnist.html>
23. <http://yann.lecun.com/exdb/mnist/>
24. Matan O, Kiang RK, Stenard CE, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD, Lecun Y (1990) Handwritten character recognition using neural network architecture. In: Proceedings of the 4th USPS advanced technology conference, Washington D.C., pp 1003–1011, Nov 1990
25. Yimu W, Yun P, Yanchen L (2014) Parallel implementation of handwritten digit recognition system using self-organizing map. *J Zhejiang Univ* 48:742–747
26. Dutt A, Dutt A (2017) Handwritten digit recognition using deep learning. *Int J Adv Res Comput Eng Technol (IJARCET)* 6(7):990–997. ISSN: 2278-1323
27. Shamsuddin MR, Abdul-Rahman S, Mohamed A (2019) Exploratory analysis of MNIST handwritten digit for machine learning modelling. In: Yap BW et al (eds) SCDS 2018, CCIS 937, pp 134–145
28. Liu C-L, Nakashima K, Sako H, Fujisawa H (2003) Handwritten digit recognition: benchmarking of state-of-the-art techniques. *Pattern Recogn* 2271–2285
29. Hua Q, Bai L, Wang X (2012) Local similarity and diversity preserving discriminant projection for face and handwriting digits recognition. *Neurocomputing* 86:150–157

30. Impedovo S, Mangini FM, BarbuZZi D (2014) A novel prototype generation technique for handwriting digit recognition. *Pattern Recogn* 47:1002–1010
31. Kutzner T, Dietze M, Bönninger I, Travieso CM, Dutta MK, Singh A (2016) Online handwriting verification with safe password and increasing number of features. In: 2016 3rd international conference on signal processing and integrated networks (SPIN)