A Comprehensive Data Analysis on Handwritten Digit Recognition using Machine Learning Approach

Meer Zohra, D.Rajeswara Rao

Abstract: ***Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or characters automatically. Because of the progress in the field of science and technology, everything is being digitalized to reduce human effort. Hence, there comes a need for handwritten digit recognition in many real-time applications. MNIST data set is widely used for this recognition process and it has 70000 handwritten digits. Many Machine Learning and Deep Learning Algorithms are developed which can be used for this digit classification. This paper performs the analysis of accuracies and performance measures of algorithms Support Vector Machine (SVM), K-Nearest Neighbor (KNN) and Convolutional Neural Networks (CNN).***

Keywords: Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Convolutional Neural Networks (CNN)

I. INTRODUCTION

Handwriting recognition is one of the compelling and fascinating works because every individual in this world has their own style of writing. It is the capability of the computer to identify and interpret handwritten digits or characters automatically. In real-time applications like the conversion of handwritten information into digital format, postal code recognition, bank check processing, verification of signatures, number plate recognition, this recognition is required.

This recognition was implemented using many Machine Learning techniques like Random Forest, Naive Bayes and Support Vector Machine etc. Yet 100% accuracy is something that is to be achieved and the research is still actively going on in order to reduce the error rate. The accuracy and correctness are very crucial in handwritten digit recognition applications. Even 1% error may lead to inappropriate results in real-time applications.

The MNIST data set is widely used for this recognition process. The MNIST data set has 70,000 handwritten digits [13]. Among these 60,000 are training examples and 10,000 are testing examples [20]. All the digits are already normalized to a normalized size and also centered. Each image in this data set is represented as an array of 28x28. The array has 784 pixels having values ranging from 0 to 255. If the pixel value is ‘0’ it indicates that the background is black and if it is ‘1’ the background is white [14].

Revised Manuscript Received on April 07, 2019.

Meer Zohra, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India.

D.Rajeswara Rao, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India.

II. LITERATURE SURVEY

In 1959, Grimsdale made an effort in the area of character recognition. Later in 1968 Eden suggested an approach termed as analysis-by-synthesis method to carry on the research work. Eden showed that all handwritten characters have some schematic features.

Parveen Kumar, Nitin Sharma and Arun Rana [8] made an attempt to recognize a handwritten character using SVM classifier and MLP Neural Network. Different kernel-based SVM like the linear kernel, polynomial kernel, and quadratic kernel-based SVM classifiers are used. In the SVM classifier model, there are two phases of training and testing. From each character, about 25 features are extracted with the help of which SVM is trained. Amongst the three kernels used the linear kernel gives an accuracy of 94.8%.

Reena Bajaj, Lipika Dey and Santanu Chaudhury [7] in 2002 decided to present well built and systematic methodology for handwritten numeral recognition. In this paper, connectionist networks are used for building recognition architecture.

Patrice Y. Simard, Dave Steinkraus, John C. Platt [1] narrated that the convolutional neural networks are better for the visual document analysis like handwriting recognition task. The main essential practices for this task are the expansion of the data sets using elastic distortions and the use of convolutional neural networks.

T.Siva Ajay [2] also proposed that the higher rate of accuracy in handwritten digit recognition task can be achieved by the use of convolutional neural networks. The implementation of CNN is made easy and simple by the use of LeNet engineering. As a result of this accuracy greater than 98% is obtained in this paper.

Ming Wu and Zhen Zhang [3] in 2010 made a comparison between different classifiers to conclude which gives better performance in the recognition task. The comparison is done between the six classifiers namely LDA (Linear Discriminant Analysis), GMM (Gaussian Mixture Models), QDA(Quadratic Discriminant Analysis), SVML (SVM with linear kernel function), SVMR (SVM with radial basis kernel function) and k-NN. Out of all the classifiers,k-NN (k=3) gives the lowest error rate.

Haider A.Alwzwazy, Haider M.Albehadili, Younes S.Alwan, and NazE. Islam [4] started a challenging task of recognition task of Arabic handwritten digits. For this, they decided to carry on the research using the Deep Convolutional Neural Networks. The accuracy of 95.7% is achieved as a result of this work.

In 2015, Saeed AL Mansoori [5] applied the Multi-Layer Perceptron model to identify handwritten digits. The samples from the data set are trained by employing gradient descent backpropagation algorithm and later feedforward algorithm. From the obtained results it can be observed that the digit 5 has the highest accuracy of 99.8% whereas digit 2 has the lowest accuracy of 99.04%. And the proposed system achieved an overall accuracy of 99.32%.

Xuan Yang and Jing Up [6] focused on the handwritten multi-digit recognition on mobile using Convolutional Neural Network. In this paper, a mobile application MDig is explained which is used to identify multiple digits using CNN. A narrow CNN is designed on mobile which gives a 99.07% accuracy using MNIST data set.

Shobhit Srivastava, SanjanaKalani, Umme Hani and SayakChakraborty [9] illustrated the handwritten recognition using Template Matching, Support Vector Machine and Artificial Neural Networks. Among the methods used, Artificial Neural Networks turned out to give more accurate results.

Shashank Mishra, D.Malathi and K.Senthil Kumar [10] attempted the handwritten recognition using Deep Learning. They used Convolutional Neural Network as a result of which they concluded that accuracy is increased and there is a reduction in the computation time. The accuracy of 99.2% is obtained.

III. IMPLEMENTATION

A. Support Vector Machine (SVM) Classifier for digit recognition

Support Vector Machine is a supervised machine learning technique which is applied for classification and regression. It is nothing but the representation of the input data into points in the space and mapped thus classifying them into classes. The SVM classifies or separates the classes using the hyper-plane concept. The separation margin should be equidistant from the classes.

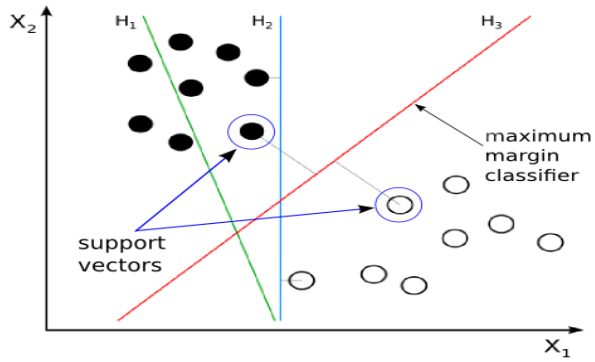

Figure 1: Support Vector Machine Classifier

It is basically used for two class classification problems. But it can be used for multi-class problems by one-against-rest approach [15]. SVM is well known because it offers attractive features to handle the classification problem [17].

For digit classification, MNIST dataset is loaded into our program initially. After that training set data is fed as input to the SVM classifier so that the classifier gets trained. SVM polynomial kernel is used. The guessed label is matched with the original to get the accuracy of the trained classifier.

Once the training is done, the testing data is given to the classifier to predict the labels and testing accuracy is obtained. The confusion matrix is generated which gives the probability between the actual data and the predicted data. Using the confusion matrix, the performance measures like precision, recall and f1 score can be calculated.

Using Polynomial SVM classifier, an accuracy of 97.9% is obtained on training data set and accuracy about 97.6% is obtained on test data.

$$\text{Precision} = \frac{TP}{TP+FP} \text{---------- (i)}$$

where TP = True Positive, FP = False Positive

$$\text{Recall} = \frac{TP}{TP+FN} \text{---------- (ii)}$$ where FN = False Negative

$$\text{F1 score} = \frac{2*Precision*Recall}{Precision+Recall} \text{---------- (iii)}$$

The confusion matrix obtained here is in the form of matrix M 10x10 because it is a multi-class classification (0-9) and as a result the formulae for the precision and recall are as follows

$$\text{Precision}_i = \frac{M_{ii}}{\sum_j M_{ji}} \text{---------- (iv)}$$

$$\text{Recall}_i = \frac{M_{ii}}{\sum_j M_{ij}} \text{---------- (v)}$$

The below Table 1 is the confusion matrix obtained for the trained data set using SVM classifier.

Table 1: Confusion matrix for Trained Data Set

Digit	0	1	2	3	4	5	6	7	8	9
0	570	0	0	0	1	1	1	0	1	0
1	0	690	1	1	1	0	1	0	2	0
2	5	1	595	1	3	0	0	3	2	0
3	1	0	8	577	0	5	0	1	3	1
4	0	2	3	0	644	1	2	0	1	5
5	0	0	1	1	1	514	2	0	4	0
6	4	0	1	0	1	3	558	0	0	0
7	0	4	2	1	3	1	0	591	0	1
8	1	2	2	6	0	3	2	0	584	2
9	2	2	1	4	4	1	0	2	2	553

The precision, recall and f1 score values for trained data set are calculated using the above confusion matrix. These values are as mentioned in table 2.

Table 2: Precision, Recall and F1 score using SVM classifier

Digit	Precision	Recall	F1 score
0	0.98	0.99	0.98
1	0.98	0.98	0.98
2	0.97	0.98	0.97
3	0.98	0.96	0.97
4	0.96	0.97	0.97
5	0.97	0.97	0.97
6	0.98	0.98	0.98
7	0.97	0.97	0.97
8	0.97	0.96	0.97
9	0.97	0.96	0.96
Average	**0.97**	**0.97**	**0.97**

In a similar way, using the confusion matrix obtained for the test data, precision, recall, and f1 score values are calculated.

B. K-Nearest Neighbor (KNN) Classifier for digit recognition

KNN is also one of the machine learning techniques which can be used for both classification and regression. Depending on the weight of the nearest neighbors among the classes the new data is classified. It estimates the value using Euclidian or Hamming distance between the neighbors.

For the digit recognition process, the MNIST data set is first loaded into our program. Now KNN (k=5) classifier is to be trained using the training data. The classifier calculates the distance between the original points and the points given to classifying the digit image. As soon as the digit label is identified, it is compared with the original labels to get the training accuracy. In the same way, test accuracy is also obtained.

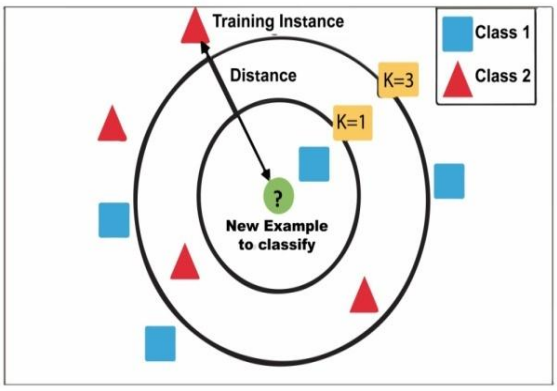

Figure 2: Example of KNN Classifier

Here also, the confusion matrix is obtained and precision and recall values are computed using the formulas (iv) and (v).

Table 3: Confusion matrix for Trained Data set using KNN classifier

Digit	0	1	2	3	4	5	6	7	8	9
0	613	0	0	0	0	1	2	0	0	1
1	0	701	0	0	1	0	1	1	0	0
2	5	9	538	1	1	0	0	7	0	1
3	0	1	4	597	0	6	0	1	1	1
4	0	4	0	0	576	0	0	0	0	13
5	1	1	1	4	3	521	6	0	0	5
6	3	1	0	0	1	1	588	0	0	0
7	0	7	2	0	2	1	0	575	0	3
8	1	10	1	5	0	8	1	2	554	7
9	2	0	0	5	4	4	1	8	0	574

The above Table 3 shows the confusion matrix obtained for the trained data set using the KNN classifier. The precision, recall and f1 score values are calculated using this confusion matrix.

The below table 4 shows the precision, recall and f1 score values obtained for the trained data set using the KNN classifier. In the same way, test data set accuracy and precision, recall and f1 score are obtained.

Table 4: Precision, Recall and F1 score for KNN on trained data set

Digit	Precision	Recall	F1 score
0	0.98	0.99	0.98
1	0.95	0.99	0.97
2	0.98	0.95	0.97
3	0.97	0.97	0.97
4	0.97	0.97	0.97
5	0.96	0.96	0.96
6	0.98	0.98	0.98
7	0.96	0.97	0.97
8	0.99	0.94	0.96
9	0.94	0.95	0.95
Average	**0.96**	**0.96**	**0.96**

The Trained Data Set obtained accuracy of 97.1% and the Test Data Set obtained accuracy of 96.7% using the KNN classifier on MNIST data set.

C. Convolutional Neural Networks (CNN) for digit recognition

Neural networks are often mostly used in pattern recognition and image proceesing problems [18]. CNN is supposed to be one of the most widely used for this digit recognition problem [11]. CNN is a deep learning technique which is basically inspired by the neuron connectivity pattern in animal visual cortex. CNN is predominantly used in image recognition, object recognition, face recognition etc.CNN is like Feed-Forward Artificial Neural Networks and the neurons have learnable weights and biases. In comparison with the other algorithms, CNN involves minimum preprocessing. In neural networks, the input is always a vector but in CNN the input is a multi-channeled image.

CNN is composed of an input layer, hidden layers, and an output layer. The hidden layers include a convolutional layer, ReLU layer i.e. activation function, pooling layers, fully connected layers, and normalization layers.

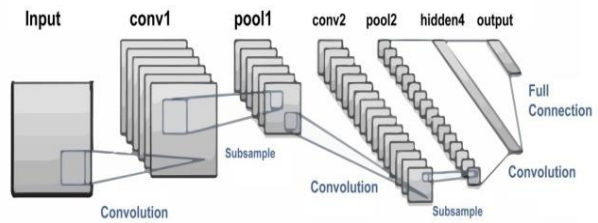

Figure 3: CNN Layers Architecture

The Input layer contains the actual pixel values of the image. The Convolutional layer is next to the input layer and it receives the output from the input layer. This layer executes a convolution operation on the input. This layer is comprised of independent filters. The operation performed in this layer helps in reducing the number of free parameters. And design each filter as 5x5 window such that it slides above the input data to acquire the pixels having the utmost intensity.

The Rectified Linear Unit (ReLU) layer has a rectifier function which is an activation function. It is very quick to evaluate and it does not saturate i.e. if the previous values are to be retained then this function is used. The Pooling layer helps in combining the outputs of the neurons at one layer into a single neuron of the next layer. The Fully connected layer computes the scores of the input digits i.e. the maximum score of a corresponding input image is obtained. A Softmax Classifier is applied at the end which returns the probabilities of all the output classes. From these probabilities, the class with the largest value is selected as the final classification.

For the array to be worked on Keras API, it is converted into 4 Dimensional numpy arrays. And also normalization is to be done by dividing RGB codes by 255. The Adam optimizer is used and it is an extension to Stochastic Gradient Descent (SGD) optimizer. It is used to update the neuron weights and it requires little memory.

From the accuracy obtained, the cross-entropy loss can be calculated using the formula:

Loss = (1- Accuracy)

The trained data accuracy of about 99.4% is achieved and the test accuracy 98.4% is achieved on MNIST data set using CNN.

IV. RESULTS AND DISCUSSION

The accuracies of the algorithms SVM, KNN and CNN are tabulated below in table 5.

Table 5: Accuracy Percentage of Algorithms

Algorithm	Trained Data Accuracy	Test Data Accuracy
SVM	97.9%	97.6%
KNN	97.1%	96.8%
CNN	99.4%	98.4%

It can be clearly observed that CNN has more accuracy compared to SVM and KNN Classifiers for both the trained data set and test data. Moreover, KNN is giving less accuracy in comparison with the remaining.

The below image figure 4 shows the comparison of the accuracies of the algorithms on both trained and test data set.

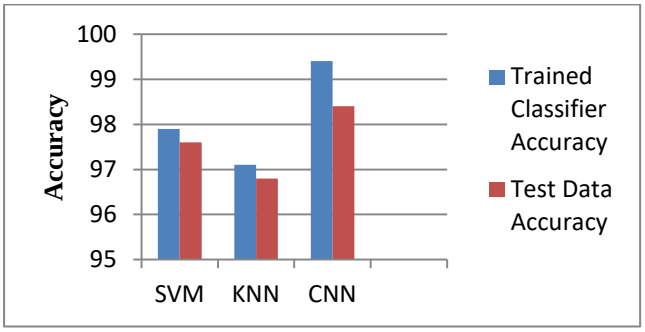

Figure 4: Comparison of Accuracies

The error rate for KNN classifier is 3.2% on the test data which is higher when compared to both SVM classifier and CNN. It is clear that CNN has given the least error rate of about 0.6% on the trained data set, yet it's not optimal and this error has to be reduced in the future.

The below figure 5 shows the comparison of the error rates of the algorithms used on trained and test data.

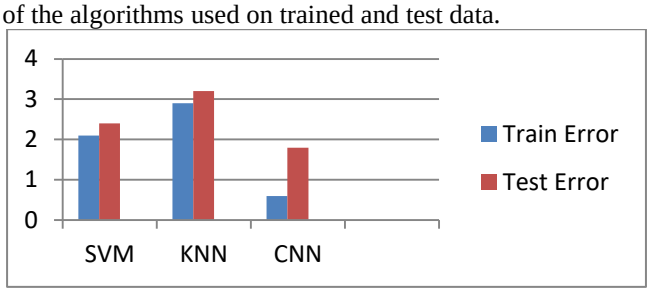

Figure 5: Error Rates of Algorithms

An epoch is one complete forward and backward passage of data in the neural network. With the change in the number of epochs, the difference in the accuracies can be observed in the below table 6.

Table 6: Accuracies of CNN depending on the number of epochs

Number of epochs	Trained Data Accuracy (%)	Test Data Accuracy (%)
1	93.6	97.7
2	97.4	98.0
3	98.1	97.9
4	98.6	98.0
5	98.9	98.4
6	99.1	98.3
7	99.1	98.3
8	99.2	98.5
9	99.3	98.3
10	99.4	98.4

From the below figure 6, it can be clearly observed and stated that the increase in the number of epochs increases the Trained Data Accuracy. It can be noticed that for epoch=1, the accuracy is about 93.6% and as the number of the epochs is increased to 10 the accuracy is also increased to 99.4%.

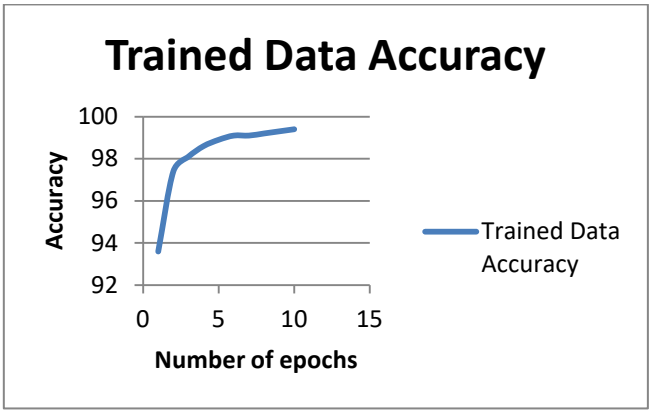

Figure 6: Epochs Vs Trained Data Accuracy

The below figure 7 clearly depicts that variation in the Test Data Accuracy with an increase in the number of epochs. It can be observed that there is an increment in the accuracy by increasing the number of epochs.

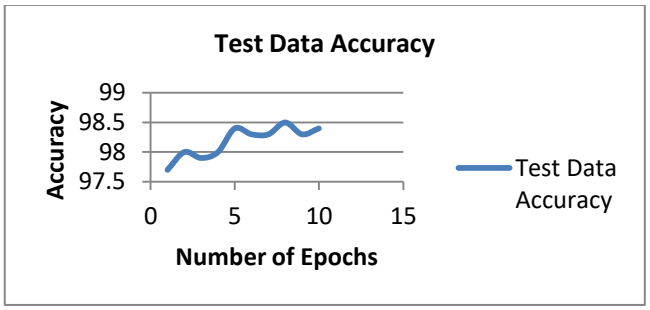

Figure 7: Number of epochs Vs Test Data Accuracy

In CNN, the change in the number of epochs also impacts the cross-entropy loss and it can be seen in the below figure 8. As the number of epochs is being increased there is a gradual reduction in the cross-entropy loss. Initially, for epoch=1 that is the input data is passed only once then there is a loss of about 0.2 and for epoch=10 it can be seen that the loss is reduced to 0.01.

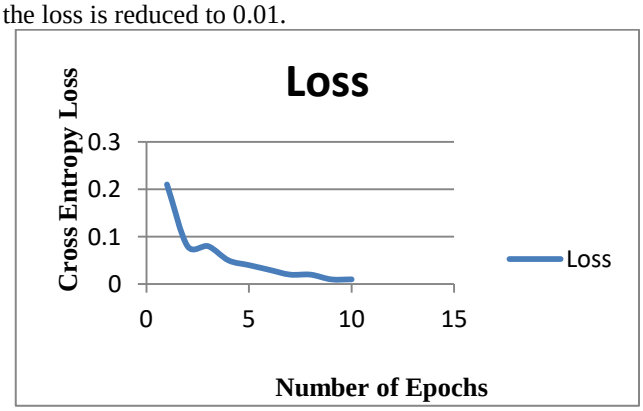

Figure 8: Epochs Vs Cross entropy loss

V. CONCLUSION

In this paper, the performances of the algorithms like SVM, KNN, and CNN are analyzed and compared. Using Tensorflow, CNN has achieved an accuracy of 99.4% on trained set whereas 98.4% on test data. Similarly, it is observed that KNN has yielded the least accuracy of about 97.1% on the trained data and 96.7% on the test data. The SVM classifier has achieved 97.9% trained data accuracy and 97.6% test data accuracy. Therefore, it can be concluded that CNN tends to give better performance for this handwritten digit recognition process.

REFERENCES

1. P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, Edinburgh, UK, 2003, pp. 958-963. T Siva Ajay, "Handwritten Digit Recognition Using Convolutional Neural Networks",
2. International Research Journal of Engineering and Technology (IRJET) Volume: 04 Issue: 07 | July -2017.
3. Wu, Ming & Zhang, Zhen. (2019). Handwritten Digit Classification using the MNIST Data Set.
4. Al-Wzwazy, Haider& M Albehadili, Hayder&Alwan, Younes& Islam, Naz& E Student, M &, Usa. (2016). Handwritten Digit Recognition Using Convolutional Neural Networks. International Journal of Innovative Research in Computer and Communication Engineering. 4. 1101-1106.
5. AL-Mansoori, Saeed. (2015). Intelligent Handwritten Digit Recognition using Artificial Neural Network. 10.13140/RG.2.1.2466.0649.
6. Yang, Xuan Jiang. "MDig : Multi-digit Recognition using Convolutional Neural Network on Mobile." (2015).
7. Reena Bajaj, LipikaDey, and S. Chaudhury, "Devnagari numerals recognition by combining decision of multiple connectionist classifiers", Sadhana, col.27, part I pp-59-72, 2002.
8. Parveen Kumar, Nitin Sharma, and Arun Rana. Article: Handwritten Character Recognition using Different Kernel-based SVM Classifier and MLP Neural Network (A COMPARISON). *International Journal of Computer Applications* 53(11):25-31, September 2012.
9. Srivastava, Shobhit&Kalani, Sanjana& Hani, Umme& Chakraborty, Sayak. (2017). Recognition of Handwritten Digits using Machine Learning Techniques. International Journal of Engineering Research and. V6. 10.17577/IJERTV6IS050456.
10. Mishra, Shashank&Malathi, D &Senthilkumar, K. (2018). DIGIT RECOGNITION USING DEEP LEARNING.
11. Hayder M. Albeahdili, Haider A. Alwzwazy and Naz E. Islam, "Robust Convolutional Neural Networks for Image Recognition" International Journal of Advanced Computer Science and Applications(IJACSA), 6(11), 2015.
12. Y.LeCun, B.Baser, J.S.Denker, D.Henderson, R.E.Howard, R.Hubbard and L.D.Jackel, Handwritten digit recognition with a back-propagation network in D.Tourezky, Advances in Neural Information Processing Systems 2, Morgan Kaufman (1990).
13. Li Deng, "The MNIST Database of Handwritten Digits images for Machine Learning Research", MIT Press, November 2012.
14. AnujDutt, AashiDutt, "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.
15. Arora, S. Bhattacharjee, D. Nasipuri, M. Malik, L. Kundu, M and Basu, D. K.2010. "Performance Comparison of SVM and ANN for Handwritten Devnagari Character Recognition" IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, pp-18-26.
16. Pradeep, J. Srinivasan, E.Himavathi, "Neural Network based handwritten character recognition system without feature extraction" International Conference on Computer, Communication and Electrical Technology, ICCCET, pp-40-44.
17. Nasien, D. Haron, H and SophiayatiYuhaniz, S.2010. "Support Vector Machine (SVM) for English Handwritten Character Recognition," Computer Engineering and Applications (ICCEA), 2010 Second International Conference on, vol.1, no., pp.249-252.
18. Mangal, M and Pratap Singh, M. 2006 "Handwritten English vowels recognition using a hybrid evolutionary feed-forward neural network, Malaysian Journal of Computer Science, Vol. 19(2), pp.No.169-187.
19. Yann LeCun, Leon Bottou, YoshuaBengio, and Patrick Haffner. "Gradient-based learning applied to document recognition". Proceedings of the IEEE, 86(11):2278–2324, 1998.
20. Y. LeCun, "The MNIST database of handwritten digits," http://yann.lecun.com/exdb/mnist.