Feature Selection for Handwriting Digit Recognition Using Convolutional Neural Network

Tarun Patel[1], Shivansh Tiwari[2], Vaibhav Dubey[3], Vaibhav Singh[4], Dr. Harvendra Kumar[5]

[1, 2, 3, 4]Department of Computer Science and Engineering, ITM, Gida Gorakhpur, U.P, India

[5]Associate Professor, Department of Computer Science and Engineering, ITM Gorakhpur, U.P, India
harvendra.patel81[at]gmail.com

Abstract: ***Handwriting digit recognition (HDR) and machine learning have both been spurred by Digit Recognition. Both OCR and HDR have their own domain in which they can be used in an HDR system for digit recognition. Numerous alternative strategies have been offered in numerous studies and papers on how to transform text from the paper documented into a machine-readable format. Digit recognition systems may play a critical part in the eventual building of a paperless society by digitally converting and processing the remaining paper documents. Deep learning has newly taken a fundamental turn in the domain of machine learning (ML) credit to the discovery of artificial neural networks (ANN), which has made it more artificially intelligent (AI). Due to its vast variety of applications, deep learning is in use in a variety of fields, including surveillance, health, medicine, sports, robots, and drones. At the heart of deep learning's incredible achievements is the convolutional neural network, which combines ANN with cutting-edge deep learning algorithms. Pattern recognition, phrase classification, voice recognition, face recognition, text classification, document analysis, scene identification, and HDR are just a few of the many uses. The goal of the particular study is to see how the number of hidden layers and epochs affect the correctness of the CNN in recognising handwritten numbers.***

Keywords: Handwritten Digit Recognition (HDR), Artificial Neural Network (ANN), Convolutional Neural Network (CNN) MNIST Dataset.

1. Introduction

Handwritten Digit Recognition is the most important, but also the most difficult, of the Digit Recognition disciplines, with a wide range of applications. Since the commencement of computer science, it has been a growing study field. It's the most natural way for humans and computers to communicate. Digit recognition, in another sense, is the process of extracting a digit from a picture and converting it to ASCII or another machine-editable format. The computer is able to distinguish between numbers and symbols thanks to this technology. In its natural language, HDR refers to a photograph taken by a human hand. Handwriting recognition is further sub-divided into two categories: offline and online HDR. Offline printing is possible if the paper is scanned before printing. When a computer recognises a person's handwriting without the need for a computer, this is known as HDR. Online handwritten recognition occurs when handwriting is recognised while being typed on a touchpad with a stylus pen.

In terms of classifiers, there are two types of digit recognition systems: segmentation free (global) and categorization based (local) (analytic). The divided partial, also known as a holistic strategy for identifying digits without breaking them down into subgroups or digits, is a method for recognising digits without breaking them down into subgroups or digits. Each word appears as a feature set asset, such as ascender loops and so on. In contrast, the streak plate method separates each word into uniform or non-uniform subgroups, or subunits that are regarded as independent. Because the HDR system is domain and application specific, it is hard to plan a general system that can practise all types of handwritten scripts and languages. The European languages, as well as Arabic (Urdu), have attracted a lot of attention. Hindi, Punjabi, Bangla, Tamil, Gujarati, and other regional languages are spoken by a limited number of people. We looked into them because they hadn't been used in a long time. The core notion of HDR is discussed in section II of this paper's literature review.

2. Methodology

The comparison of algorithms (SVM, MLRNN, and CNN) is based on an attribute of each method on general basis such as dataset, epoch, complexity, and accuracy. Under ideal conditions, the device specification (Bunt 20.04 LTS, i5 7th gen processor) was utilised to run the programme and algorithm.

a) MINST Dataset

The MNIST (Modified National Institute of Standards and Technology) database includes a huge amount of handwritten digits. It consists of a training set of 60,000 examples and a testing set of 10,000 examples. It's a division of the massive NIST Special Database 3 (numbers in print by US Census Bureau employees) and Special Database 1 (digits created by matrix students) databases, which both have impartial images of handwritten digits. In a fixed-size image, the digits have been size-normalized and entered. The original NIST B/W (bi-level) photos were framed to fit in a 20x20 pixel box while minding the aspect ratio. As the result of the normalising algorithm's by de-interlacing approach, the resultant images have grey levels. By identifying that the images were in a 28x28 image, with the centre of mass (COM) of the pixels, and the image translated to position this point at the centre of the 28x28 field.

Licensed Under Creative Commons Attribution CC BY

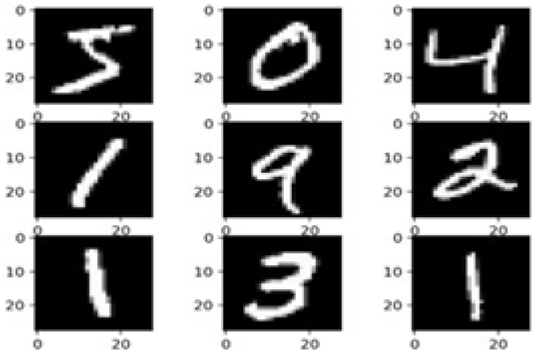

Figure 1: MINST Dataset

b) Convolution Neural Network

Let's talk about CNNs, which have recently gained prominence. CNNs are a type of deep, feed-forward ANN that can execute a diversity of tasks quicker and more precisely than other classifiers in applications such as image and video identification, recommender systems, and natural language processing (NLP). Among other things, Facebook use neural networks for auto tagging algorithms, Google for photo search, Amazon uses for product recommendations, and Interest for personalised home feeds, and Integra for search infrastructure. Image classification, or object recognition, is the process of using an image as a parameter and predicting whether or not a set of criteria is met (cat or not, dot or not). A seven-layered CNN with having one input layer, five hidden layers, and one output layer has been developed. This diagram is shown in Figure 2 to recognise handwritten numbers.

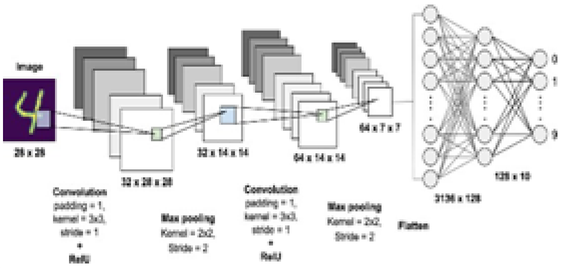

Figure 2: A seven-layered convolutional neural network for digit recognition

Input layer is made up of 28 × 28 pixel pictures, representing a total of 784 neurons in the system. Grayscale pixels are used as input, with zero signifying a white pixel and 1 signifying a black pixel. There are five hidden layers in this CNN model. Convolution layer 1, the first hidden layer, is accountable for extracting features from the input data. This layer convolved a filter with the preceding layer to perform convolution on small localised areas. There are also many feature maps with learnable kernels and adjusted linear units (Rely). The position of the filters is determined by the kernel size. Rely is brought in to assist the model in performing better. Each convolution layer has an activation function as well as a completely linked layer at the end. The next hidden layer is the pooling layer 1. The no. of limitations and computational complexity of the model are reduced by reducing the output information to the convolution layer. Max pooling, min pooling, average pooling, and L2 pooling are the most common poolings. Max pooling is used for subsampling each dimension of the feature map. Convolution layer 2 and pooling layer 2, which serve the same function as convolution layer 2.

Layer 1 pooling & Convolution Layer 1 work the same way, with the difference that their feature maps and kernel widths are different. After the pooling layer, a Flatten layer is applied to decrease the 2D feature. At the conclusion of each layer, a fully linked layer as an activation function pooling layer 1 is the next hidden layer. It helps to reduce the amount of limit and computational intricacy of the convolution layer by lowering the output information. The several types of pooling include maximum, minimum, average, and L2 pooling. Maximal pooling is subsample each feature map's dimension. Layer 2 of convolution and layer 2 of pooling (which serves the same purpose as convolution layer 2). With the exception of feature maps and kernel sizes, Convolution Layer 1 and Layer 1 pooling work in the same way. The 2D characteristic is reduced using a Flatten layer after the pooling layer. The output layer, which improves the model's performance by using an activation function such as soft axe, classifies the output digit from 0 to 9, with 9 being the highest activation value.

MNIST database has been used in the project. 60,000 survey photos of HDR are used to train the network, and 10,000 survey images of digits are used to test the network, from a total of 70,000 digits in the MNIST database collection. All the images are used to training and testing the network are grayscale images with a resolution of 2828 pixels. The character x stand for a training input, where x is 784-dimensional vector with a 2828-pixel input. y(x), where y is a 10-dimensional vector, is the equivalent desired result. The network's goal is to identify the most practical weights and biases such that the network's output is as close as possible to the original y(x) for all training inputs x, because weight and bias values ++are totally dependent on them. A cost function, described by equation 1, is defined to compute network performance.

$$C(w,b) = \frac{1}{2n}\sum_{x}\left[y(x) - a^2\right]^2 \qquad (1)$$

Where w represents the entire number of weights in the network, b represents all biases, n represents the total number of training inputs, and a represents the actual output. The actual output an is determined by the variables x, w, and b. Because all of the terms in the total are nonnegative, C (web) is nonnegative. Furthermore, C (web) =0 for all training inputs, n, when the desired output y(x) is substantially identical to the actual output, a.

In order to reduce the cost C (web) to a decreased degree as a function of weight and biases, the training algorithm must select a set of weights and biases that cause the cost to become as minimal as possible as a function of weight and biases. Ascend the incline the algorithm that was used to do this In other terms, gradient descent is an optimization approach that twists a parameter iteratively to reduce a cost function to a local minimum. The following equations are used through the gradient descent technique to set the weights and biases.

Licensed Under Creative Commons Attribution CC BY

$$w^{new} = w^{old} - \eta \frac{\partial C}{\partial w^{old}} \quad (2)$$

$$b^{new} = b^{old} - \eta \frac{\partial C}{\partial b^{old}} \quad (3)$$

To achieve the global minimum of the cost C (web) shown in figure 3.

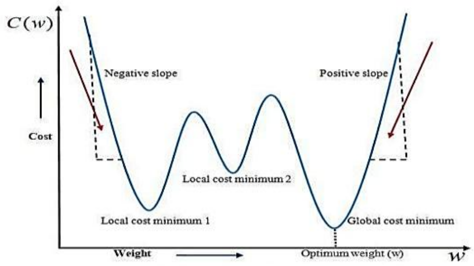

Figure 3: Graphical Representation of Cost vs. Weight

When the training data is too extensive, however, the gradient descent approach may be ineffective. As a result, a stochastic variant of the technique is utilized to improve the network's performance. A less number of iterations are required in SDG to develop efficient solutions to optimization issues. Furthermore, in SDG, a limited number of iterations is all that is required to arrive at a suitable solution. The following equations are used in the Stochastic Gradient Descent (SDG) algorithm.

$$w^{new} = w^{old} - \frac{\eta}{m} \frac{\partial C_{xj}}{\partial w^{old}} \quad (4)$$

$$b^{new} = b^{old} - \frac{\eta}{m} \frac{\partial C_{xj}}{\partial w^{old}} \quad (5)$$

The outcome of the network can be expressed by:

A = f (z) = f (we + b)

So, to find the total contribution of the weights to find the total error network the method used is BackPropagation . The back propagation network is shown by the following equations:

$$\delta^L = \frac{\partial C}{\partial a^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} = \frac{1}{n}(a^{(L)} - 1) f'(z^{(L)}) \quad (7)$$

$$\delta^l = \frac{\partial C}{\partial z^{(l)}} = \frac{\partial C}{\partial z^{(l)}} \frac{\partial z^{(l+1)}}{\partial z^{(l)}} = \frac{\partial z^{(l+1)}}{\partial z^{(l)}} \delta^{l+1} = w^{l+1} \delta^{l+1} f'(z^l) \quad (8)$$

$$\frac{\partial C}{\partial b^{(l)}} = \delta^l \quad (9)$$

$$\frac{\partial C}{\partial w^{(L)}} = a^{l-1} \delta^l \quad (10)$$

c) Support Vector Machine

The Support Vector Machine (SVM) is a algorithm of machine learning that is supervised. In this case, we mark data items in the n-d space, here n is number of features and a given position indicates the value of a feature. We then classify the data by locating the hyper plane that separates the two classes. It will select the hyper lane which will separate the classes. SVM pick the extreme vectors who assist in the formation of the hyper plane. Support vectors represent the extreme examples, which is why the technique is called Support Vector Machine. Linear and non-linear SVMs are the two primary types of SVMs. We used Linear SVM to recognize handwritten digits in this work.

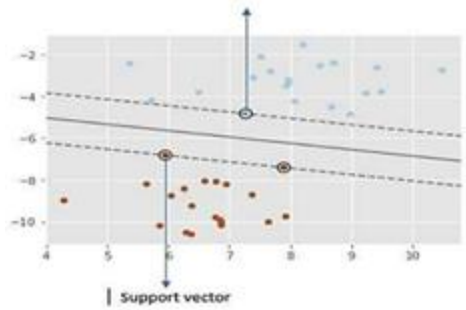

Figure 4: This image describes the working mechanism of SVM Classification with supporting vectors and hyperplanes

d) Multilayered Perceptron

A feed forward ANN with multilayer perception (MLP) is type of feed forward artificial neural network. It has three layers: an input layer, a concealed layer, and a final output layer. Each layer made up of numerous nodes, which also known as neurons, and each node is connected to the nodes of the following layer. There are three layers in a basic MLP; however the number of hidden layers may be increased to any number depending on the problem, with no limit on the nodes. The input and output layers' nodes are determined through the number of attributes and a parent classes in the dataset, respectively. Due to the difficulty in determining the exact number of hidden layers or nodes in the hidden layer, Due of the model's unpredictable character, it was chosen experimentally. For processing, each hidden layer of the model might have a different activation function. It employs back propagation, a supervised learning approach, for learning objectives. In the MLP, each node's connection is made up of a weight that is altered to synchronize with each other during the mode's training phase.

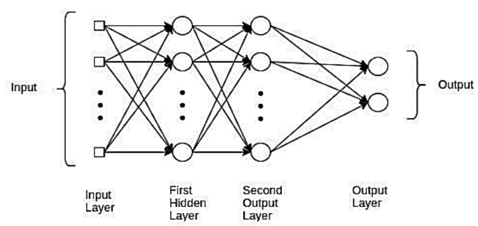

Figure 5: Multilayer perceptron with variable specification of the network.

e) Visualization

Using the MNIST dataset (i.e. handwritten digit dataset), we compared many level deep and machine learning methods (i.e. Support Vector Machine (SVM) Artificial Neural Network (ANN)-Multi layer Perceptron (MLP) Convolutional Neural Network (NLP) based on execution duration, complexity, accuracy rate, number of epochs, and hidden layer number (in the case of deep learning algorithms). The data was shown using bar graphs and tabular format charts made using the matplotlib module, which provides the most precise visualisations of the algorithms' step-by-step advancements in detecting the digit.

Licensed Under Creative Commons Attribution CC BY
Paper ID: SR22515114611 DOI: 10.21275/SR22515114611

The graphs are strategically positioned throughout the programme to provide visual representations of each component and to support the goal.

3. Literature Review

This study by K. Gaurav, P. K. Bhatia [1] et al. discusses the various pre-processing approaches used in Handwritten Digit Recognition (HDR) with various types of pictures, ranging from simple handwritten form papers with colored and complicated backgrounds and varying intensities to complex handwritten form papers with colored and complicated backgrounds and varying intensities. Skew detection and correction, contrast stretching, binarization, noise removal techniques, normalization and segmentation, and morphological processing techniques are among the preprocessing techniques mentioned. We discovered that employing only one pre-processing strategy would not allow us to properly load the image. Even with all of the aforementioned solutions in place, it's likely that achieving 100% accuracy in a preprocessing system is unattainable.

G. Pirlo and D. Impedovo [2] developed a novel family of the member functions is called Fuzzy membership functions (FMFs) because of zoning-based classification in their article (FMFs). These FMF can be modified easily to particular characteristics of a classification task to improve the performance. Genetic algorithms were all coded. Is suggested for technique and features, which include normalized distances. The member function is composed of two structural elements that are calculated using the entropy in order to obtain unity in the membership function. Overall, 95 percent of Hindi numbers are recognized, compared to 98.4% of English numerals.

The approach presented by Nafiz Arica and colleagues [3] avoids most pre-processing steps, resulting in the loss of essential data. One of the method's significant accomplishments is the development of a complex segmentation algorithm. Upper and lower baselines, stroke height and width, character boundaries, local maxima and minima, slant angle, upper and lower baselines ascenders and descanters are all used to optimize the search strategy for the best segmentation route on a grayscale image. Using this strategy, over-segmentation is reduced to a minimum. Some other contributions is the usage of Hidden Markov Models training to estimate numerous global and feature space parameters in addition to model parameters. Probabilities are utilised to rank candidate characters and quantify shape information in addition to HMM. The HMM of the shape recognizer

Model	*Training rate*	*Testing rate*	*Execution time*
SVM	99.96%	94.003%	1:23 min
MLP	99.94%	98.75%	2:4 min
CNN	99.57%	99.34%	43:02 min

4. Problem Statement

HDR is a computer's ability to recognize and classify mortal handwritten integers from a variety of sources, including pictures, documents, and touch defenses (0-9). This has been a content of bottomless-exploration in the subject of deep literacy. Bike number plate recognition, postal correspondence sorting, and bank cheque processing are just a few of the operations that include number recognition.

5. Implementation

For comparing the algo on their accuracy of working, executing time, time complexity, the no. of epochs we use classifier of three different types:

- Vector Machine Classifier (Support Vector Machine Classifier)
- ANN (Analysis of Neural Networks) - Multilayer Perception Classifier
- Classifier based on CNN

We've gone over each algorithm's implementation in great detail below in order to provide a fluid and accurate flow for this analysis.

6. Result and Analysis

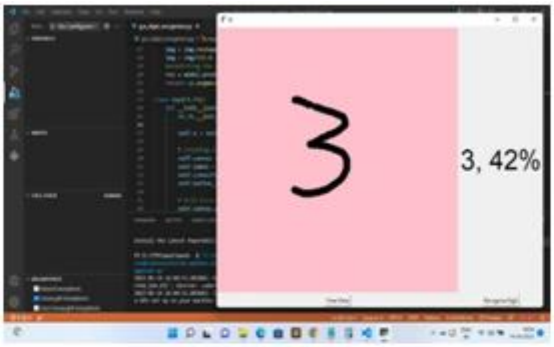

Figure 6: Recognition system for the number from Zero to nine

After building all three algorithms, SVM, MLP, and CNN, the execution time and the accuracies are compared by using experimental graphs to gain a better understanding. All of the models listed above have had their Training and Testing Accuracy. After testing all of the replicas, we discovered that SVM has the accuracy much more on training data, whereas CNN has accuracy highest on testing data. We also compared the execution times to acquire a better understanding of how the algorithms work. In general, an algorithm's execution time is proportional to the number of operations it has completed. So, to acquire the best results, we trained our deep learning model for 30 epochs and SVM models according to norms. Outcome. The SVM took the shortest amount of time to execute, whereas CNN took the longest. Figure eight. The overall performance of each model is represented in this table. The table has five attributes: the second column represents model name, the 3rd and 4th columns indicate model training and testing accuracy, and the fifth column represents model execution time.

SVM: 1.58 min., MLP: 2.53 min., CNN: 44.05 min. We also visualised how deep learning models (DL) improved the accuracy and reduced their error as the number of epochs increased. The sketching the graph is important to determine where we have to apply an stop early to avoid the problem of

Licensed Under Creative Commons Attribution CC BY

overfitting, which occurs when the change in accuracy becomes constant after a certain no echoes.

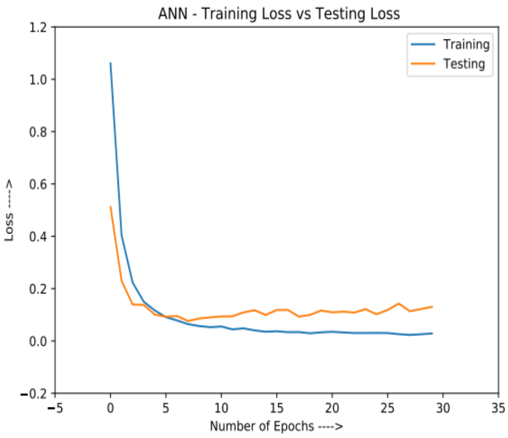

Figure 6: In Multilayer Perceptron, a graph depicting the transition of training loss as the number of epochs increases. (Loss rate v/s Number of epochs)

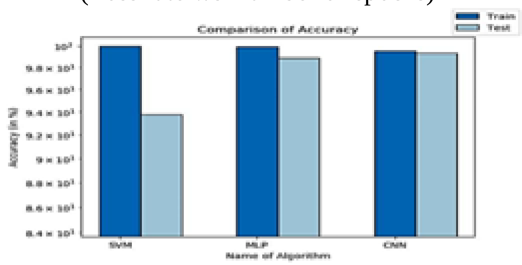

Figure 7: Bar graph showing execution time comparison of SVM, MLP and CNN

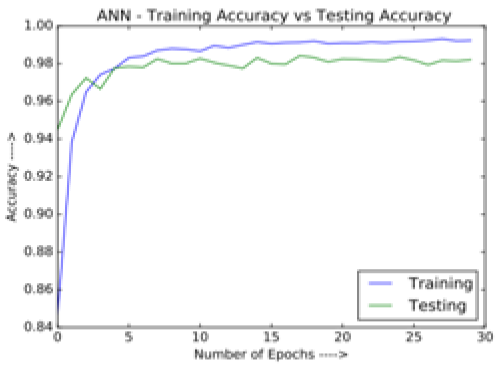

Figure 8: Multilayer Perceptron training accuracy transitions as the number of epochs increases. (Accuracy v/s Number of epochs).

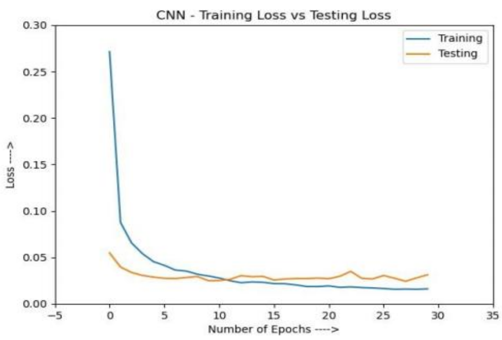

Figure 9: Graph depicting the transition in CNN training accuracy as the number of epochs increases. (Accuracy v/s Number of epochs)

7. Conclusion

In the given paper, we investigate several Handwriting Digit Recognition (HDR) and Artificial Neural Network (ANN)-based recognition ways in order to determine the best approach in relations of the accuracy and performance. Many models were offered by various writers, and some factors, such as implementation time, were taken into account. The algorithms are calculated using both datasets, both random and standard of handwritten digits. In relations of accuracy and performance, the results reveal that DNN is the best algorithm. In terms of accuracy, the Convolutional Neural Network (CNN) algorithm and the DNN algorithm are nearly comparable. In terms of execution speed, however, the DNN algorithm outperformed CNN and DBN. By identifying the proper digits, the margin of error may be reduced due to digit similarities.

References

[1] K. Gaurav and Bhatia P. K., "Analytical Review of Preprocessing Techniques for Offline Handwritten Character Recognition", 2nd International Conference on Emerging Trends in Engineering & Management, ICETEM, 2013.

[2] Nafiz Arica, and Fatos T. Yarman-Vural, Optical Character Recognition for Cursive Handwriting IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.24, no.6, pp. 801-113, June 2002.

[3] G. Pirlo and D. Impedovo, "Fuzzy Zoning Based Classification for Handwritten Characters", IEEE Transaction on pattern Recognition and Machine Intelligence, vol.19, no. 04, pp.780-785, August 2011.

[4] Liana M. Lorigo and Venu Govindaraju, "Offline Arabic Handwriting Recognition: A Survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 28 Issue 5, May 2006.

[5] Salvador España-Boquera, Maria J. C. B., Jorge G. M. and Francisco Z. M., "Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, No. 4, April 2011.

[6] M. Hanmandlu, O.V. Ramana Murthy, "Fuzzy model based recognition of handwritten numerals", pattern recognition, vol.40, pp.1840-1854, 2007.

[7] Mohammed Z. Khedher, Gheith A. Abandah, and Ahmed M. AlKhawaldeh, "Optimizing Feature Selection for Recognizing Handwritten Arabic Characters", proceedings of World Academy of Science Engineering and Technology, vol. 4, February 2005 ISSN 1307-6884.

[8] Ayush Purohit and Shardul Singh Chauhan., "Literature Survey on Handwritten Digit Recognition", International Journal of Computer Science and Information Technologies, Vol. 7 (1), pp 1-5, Jan 2016.

[9] A. Brakensiek, J. Rottland, A. Kosmala and J. Rigoll, "Offline Handwriting Recognition using various Hybrid Modeling Techniques & Character N-Grams".

[10] Reena Bajaj, Lipika Dey, and S. Chaudhary, "Devanagari numeral recognition by combining decision of multiple connectionist classifiers", Sadhana, Vol.27, part. 1, pp.-59-72, 2002.

Licensed Under Creative Commons Attribution CC BY

[11] Sandhya Arora, “Combining Multiple Feature Extraction Techniques for Handwritten Devnagari Character Recognition”, IEEE Region 10 Colloquium and the Third ICIIS, Kharagpur, INDIA, December 2008.
[12] K. H. Aparna,, Vidhya Subramanian, M. Kasirajan, G. Vijay Prakash, V. S. Chakravarthy, Sriganesh Madhvanath, "Online Handwriting Recognition for Tamil", IWFHR, 2004, Proceedings. Ninth International Workshop on Frontiers in Handwriting Recognition, Proceedings. Ninth International Workshop on Frontiers in Handwriting Recognition 2004, pp. 438-443, doi:10.1109/IWFHR.2004.
[13] Sushree Sangita Patnaik and Anup Kumar Panda, “Particle Swarm Optimization and Bacterial Foraging Optimization Techniques for Optimal Current Harmonic Mitigation by Employing Active Power Filter Applied Computational Intelligence and Soft Computing”, Volume 2012, Article ID 897127.
[14] T. Som, Sumit Saha, "Handwritten Character Recognition Using Fuzzy Membership Function", International Journal of Emerging Technologies in Sciences and Engineering, Vol.5, No.2, pp. 11-15, Dec 2011.
[15] Renata F. P. Neves, Alberto N. G. Lopes Filho, Carlos A. B. Mello, Cleber Zanchettin, “A SVM Based Off-Line Handwritten Digit Recognizer”, International conference on Systems, Man and Cybernetics, IEEE Xplore, pp. 510-515, 9- 12 Oct, 2011, Brazil.
[16] Yoshimasa Kimura, "Feature Selection for Character Recognition Using Genetic Algorithm", Fourth International Conference on Innovative Computing, Information and Control 978-0-7695-3873-0/09© 2009 IEEE.

Licensed Under Creative Commons Attribution CC BY