IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

HANDWRITTEN DIGIT RECOGNITION USING OPENCV AND CNN

[1]K. Swetha, [2]Y. Hithaishi, [3]N.L. Tejaswini, [4]P. Parthasaradhi, [5]P.V. Venkateswara Rao

[1,2,3,4]Student, [5]Associate Professor

Computer Science and Engineering,

Madanapalle Institute of Technology and Science, Madanapalle, India

Abstract: Handwritten Digit Recognition (HDR) is the process of converting images of handwritten digit into digital format. A lot of money is wasted on converting the information that is in paper to digital format. This problem can be solved by using HDR. The heart of our project lies within the ability to develop an efficient algorithm that can recognize the handwritten digits which are scanned and sent as input by the user. The goal of this paper is to observe the variation of different algorithms that can classify the handwritten digits using different hidden layers, various number of epochs and to make a comparison based on the accuracy. This experiment is performed using the Modified National Institute of Standards and Technology (MNIST) dataset.

Keywords **– Handwritten Digit Recognition (HDR), Epochs, Hidden Layers, MNIST dataset**

I. INTRODUCTION

Developers are using different machine learning and deep learning techniques to make machines more intelligent. In deep learning, Convolutional Neural Networking (CNN) is being used in many fields like object detection, face recognition, spam detection, image classification. Handwritten digit recognition has not only professional and commercial applications, but also has practical application in our daily life and can be of great help to the visually impaired. It also helps us to solve complex problems easily thus making our lives easier [5]. Many algorithms have been developed for hand written digit recognition. But due to infinite variation in writing styles they are still not up to mark. Poor contrast, image text vagueness, disrupted text stroke, unwanted objects, deformation, disoriented patterns and also interclass and intraclass similarity also cause misclassification in handwritten numeral recognition system [6]. An illustration of Handwritten digit recognition is shown below.

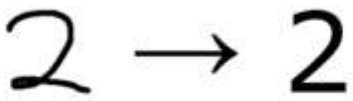

Fig 1: Illustration of HDR

Handwritten Digit Recognition (HDR) is the ability of a machine to recognize human handwritten digits. It is a hard task for a machine because handwritten digits are not perfect. So, the solution to this problem is our project that uses the image of a digit and recognizes the digit present in the image. The goal of this project is to observe the influence of various algorithms for handwritten digits. In our project, we have applied CNN algorithm for training on the Modified National Institute of Standards and Technology (MNIST) dataset using OpenCV, a machine learning library written in python. A CNN [5] is a deep learning method that has been used in developing applications for computer vision, data mining, computer games and handwritten recognition [7]. LeNet5 is the base architecture of CNN [8].

II. DATASET

Modified National Institute of Standards and Technology (MNIST) is a database which is freely available for handwritten digits and is standard for machine learning algorithms. It is similar to TIDigit which is a database of speech created by Texas Instruments, which tasks in speech recognition [9]. For our project, MNIST dataset is used. In this dataset, the images of digits were taken from a variety of scanned documents in which each image is Greyscaled and of 28*28 pixels. It uses 60,000 images to train the network and 10,000 images to evaluate hoe accurately the network learned to classify the images. Some of the sample images of the MNIST dataset are shown below.

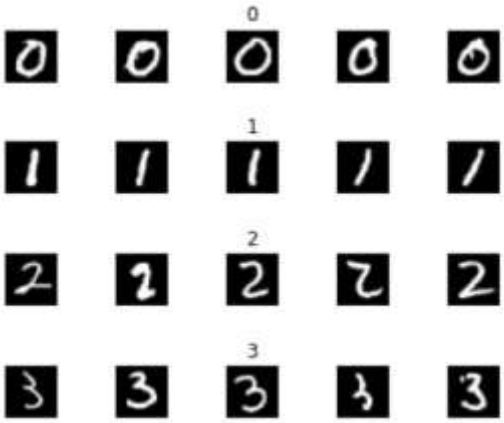

Fig 2: Sample images of MNIST dataset

To use the MNIST dataset in Keras, an API is provided to download and extract images, labels automatically. The task is to classify a given input image of a handwritten digit into one of the 10 classes representing the integer values from 0 to 9 inclusively. The distribution of training data in MNIST is shown below.

Fig 3: Bar diagram showing the distribution of training dataset

III. LITERATURE SURVEY

Handwriting recognition of digits has started around the 1980s. The task of handwritten digit recognition, using a classifier, has great significance and it has many applications. Handwritten digits are not similar as they differ in size, thickness, position relative to the margin. These are some of the difficulties we faced while trying to solve this problem. Y. LeCun et al. [1] presented an application of back-propagation networks to handwritten digit recognition

A. Existing System

X. Han et al. [2] summarizes the latest development of CNN and expounds the relative research of image recognition technology and elaborates on the application of CNN in handwritten numeral recognition. However, every neural network has some error rate due to parallel in digit shape. R. Sudhakar et al. [16] developed a hybrid model by integrating a non-linear regression model and optimization-driven deep learner for video super resolution. Initially, the low-resolution frames are subjected to framing, and each frame is provided to both Fractional-Group Search Optimizer-based Deep Belief Network (FrGSO-DBN) classifier and the non-linear regression model.

Caiyun Ma et al. [3] normalized the images of various sizes and stroke thickness in preprocessing to eliminate negative information and keep relevant features. The architecture of CNN for the existing system is shown below.

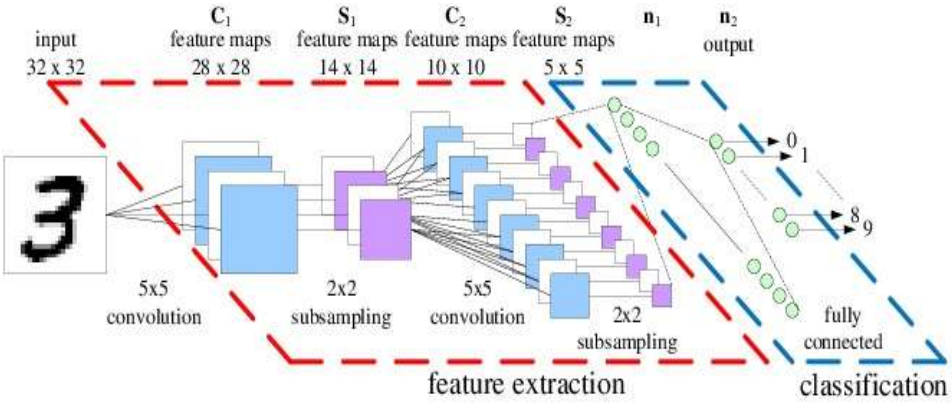

Fig 4: Existing system of HDR

V.C. Bharathi et al. [15] recognized the query words using the Euclidean distance of the keyword and the keyword search is acquired from the index position by retrieving the appropriate words from the document. The general problem was the similarity between the digits like 1 and 7, 5 and 6, 3 and 8, 9 and 8, etc. Also, people write the same digit in many different ways. Analysis of Digit Recognition by various methods are shown below in a tabular format.

Table 1: Survey on different Research Analysis

S.no	Title	Author	Algorithm used	Drawbacks	Accuracy(%)
1	The MNIST Database of Handwritten Digits images for Machine Learning Research [4]	Li Deng et al.	Neural network (after distortion)	lack of accuracy due to absence of convolution networks	99.2
2	Deep big simple neural Nets Excel on Handwritten Digit Recognition [11]	Dan ClaudiuCiresan et al.	Simple Neural network and back propagation	Higher processor required, High cost, Time consuming	99.1
3	Digit's recognition using single layer neural Network with principal component analysis [12]	Vineet Singh et al.	PCA Principal component analysis.	Consumes more training time	98.39
4	Handwritten digits recognition using ensemble neural networks and ensemble decision tree [13]	Retno Larasati et al.	Ensemble neural networks that combined with ensemble decision tree	Less accuracy	84
5	Comparison of Classifier methods a case study in handwritten digit recognition [14]	L. Bottou et al.	Baseline Linear Classifier, LeNet 1, Le Net 4, Large fully connected multi network	Much complex networks with high computation time.	1)92.2% 2)98.3% 3)98.9% 4)98.4%

IV. ARCHITECTURE

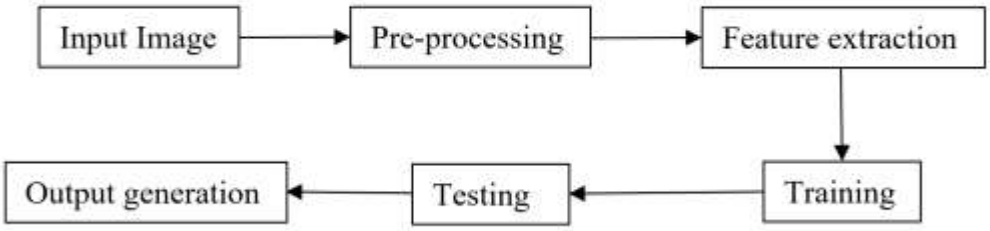

Fig 5: Architecture

The proposed model contains four stages to classify and detect the digits:

A. Pre-Processing:

Pre-processing is a part of HDR. If there are some rules like a box for each digit then, it will be much easier to detect the boundaries. The fundamental motivation behind pre-processing is to take off noise filtering, smoothing, and standardization. Binarization converts a Greyscaled image into a binary image.

B. Feature Extraction:

Different type of algorithms used for feature extraction have different types of error rate. The errors made by each separate algorithm does not overlap, so combining all these methods lead to a perfect recognition rate and also helps to reject the ambiguous digits recognition and improve the recognition rate of misclassified digits that can be recognized by humans [10].

C. Classification and Recognition:

In the classification and recognition step, the extracted feature vectors are given as single input values to each classifier. CNN Convolution layer and the subsampling layer can have various different layers [17]. The down sampling layer is also known as pooling layer [19]. The image is divided into small segments of small areas, and a value is calculated for each area. Then the calculated values are rearranged in sequence to form a new image [7]. This process is similar to fuzzy filter, which can increase the robustness of image feature withdrawal [20]. Extracted features are combined and are defined using the following four classifiers:

i. K-Nearest Neighbor:

It is instance-based learning. There are mainly two advantages of using the KNN algorithm, they are, it is robust to noisy training data and it is very efficient if the data is very large in size. In Machine Learning, one of the best ways to learn more about data is by classifying it with what you already know. It is a supervised learning algorithm for multiclass classification.

ii. Random Forest Classifier:

A Random Forest merges a collection of independent decision trees to get a more accurate and stable prediction. It says that there is an immediate connection between the total number of trees and the result it gets. This classifier can be used for regression and also classification. It is a type of ensemble method.

iii. Support Vector Machine:

It finds the hyperplane by performing classifications between the two classes. The most significant advantage of this algorithm is that it provides a regularization parameter which helps in avoiding overfitting problems. It is a classifier that finds an optimal hyperplane that maximizes the margin between two classes.

iv. Logistic Regression:

It is a form of regression where the target variable is binary. What Logistic Regression is great for is an initial benchmark model on a binary classification problem with fairly well-behaved data. It is one of the more transparent algorithms and doesn't work well with really messy data

D. Training and Testing:

Using the fit() method, a model can be trained. In order to see the skill of the trained model, test data is used as a validation dataset. Finally, to evaluate a model, the test dataset is used. Training is less complex because each module is designed to handle a specific subproblem. It is expected that each module can tackle the specific problem more efficiently and accurately because each module is trained independently which is easy to add and delete modules.

V. METHODOLOGY

The goal is to create a model to predict the digit in an image. Steps involved in this project are as follows:

- Gathering, examining, and cleaning the data (data exploration)
- Choose a model and measure the evaluation (Neural network)
- Training
- Evaluate the models on a metric and compare them to the benchmark.
- Compare different Machine Learning algorithms based on their accuracy of predicting the digit.

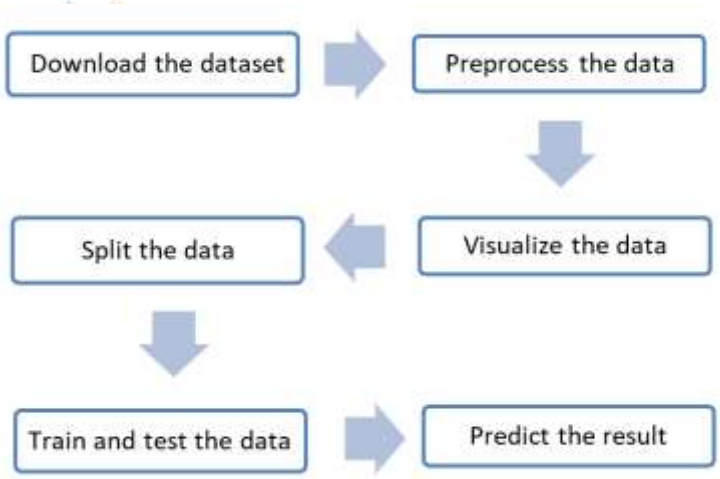

Fig 6: Flow of Training module

A. Download the dataset:

Import MNIST dataset for handwritten digits from Keras. It consists of 60,000 training images and 10,000 test images, which are 28×28 pixel grayscale images of handwritten single digits between 0 and 9. The initial step to be carried out is loading the dataset.

B. Preprocess the data:

Before training, the pre-processing model is needed whose output images will be the inputs for the training model. The MNIST handwritten images have been size-normalized, centered, and stored sequentially as 28 × 28 pixel images in the gray-level bitmap. Pre-processing is mainly done to remove noise, resize, crop, detect edges.

C. Visualize the data:

Data visualization is a process of representing data or information in the form of a graph, chart, or another visual format. this makes the data simpler and hence makes the data easier to identify trends, patterns, and outliers within large data sets.

D. Split the data

The Complete dataset is divided into train and test datasets. The training dataset is used to fit and tune your models. The Test dataset is used to evaluate your models. Before performing anything, data split should be split. It is the best way to get efficient estimates of models' performance.

E. Train and Test the data

In any dataset, a training dataset is used to build a model, and a test dataset (or validation) is used to validate the model. So, we use the training data for fitting the model and testing data for testing it.

F. Predict the result

The models produced here are then used to predict the results which are unknown, and this is named the test dataset. To test the model, some of the images are used from the test dataset.

VI. IMPLEMENTATION

The data which is already collected can be used for extracting the features of each digit. The availability of more powerful machine learning algorithms introduces an efficient and better approach to solve this problem. The project is divided two modules as shown below.

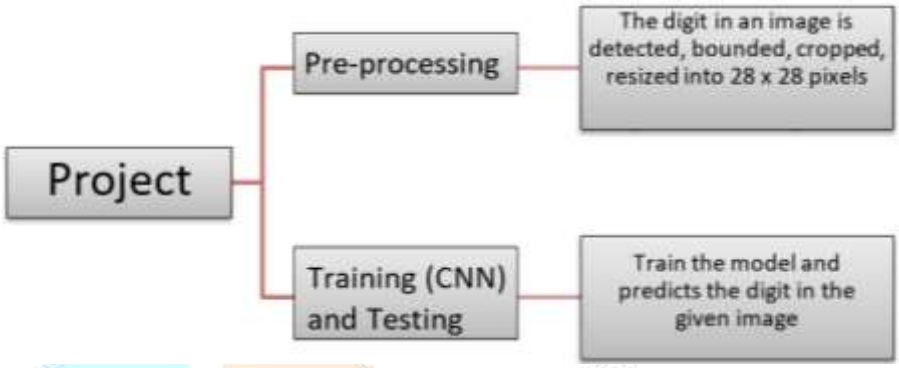

Fig 7: Modules of HDR

A. Pre-processing module:

i. Read the image

OpenCV is a Machine Learning library that is used to read and manipulate images. The image is read and then stored in multiple copies for performing different operations. After reading the image is plotted in its shape to make sure it is read perfectly.

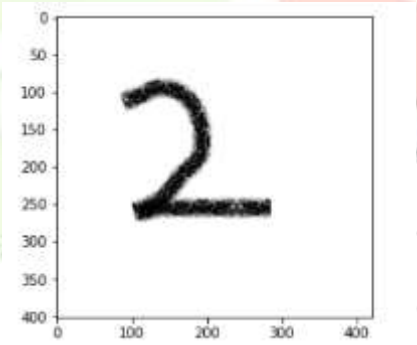

Fig 8: Handwritten image

ii. Converting an RGB image to a Greyscale image

An RGB image that is three-dimensional is converted to a Greyscale image that is one-dimensional. A BGR image is a three-dimensional image (w, h, c)

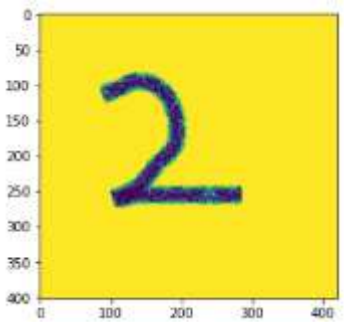

Fig 9: Greyscale image

iii. Remove noise

Gaussian blur is applied to the greyscale image to remove noise in the image.

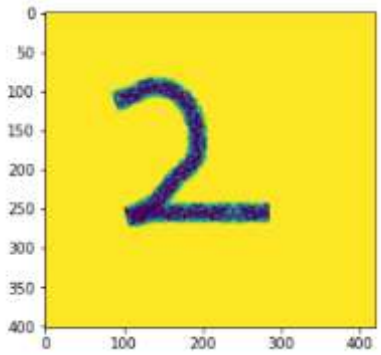

Fig 10: Image after applying Gaussian blur

iv. Object Detection

The standard step for object detection is Otsu thresholding.

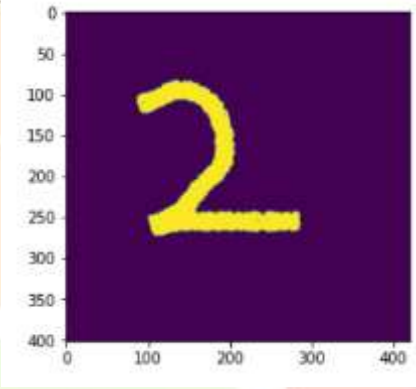

Fig 11: Image after applying Otsu thresholding

v. Finding and drawing contours

The findContours() and drawContours methods() are used for finding and drawing boundaries of the detected object in an image.

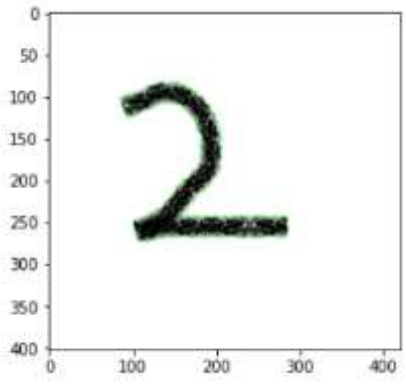

Fig 12: Image with Contours

B. Training & Testing module:

Any model learns by using past data and Machine Learning algorithms. It learns from the past data by feature extractions. For training the model, a sequence of hidden layers is created with some nodes in each layer.

Then, we compile the model by ‘categorical_crossentropy’as loss function, ‘adam’ as an optimizer , and ‘accuracy’ as metrics. Then, the model is trained using CNN and tested by giving new handwritten digit images that are not present in the dataset.

VII. Results and Analysis

In this section, we compare the results of five classification algorithms namely K-Nearest Neighbors, Logistic Regression, Convolutional Neural Network, Random Forest Classifier and Support Vector Machine on the MNIST database.

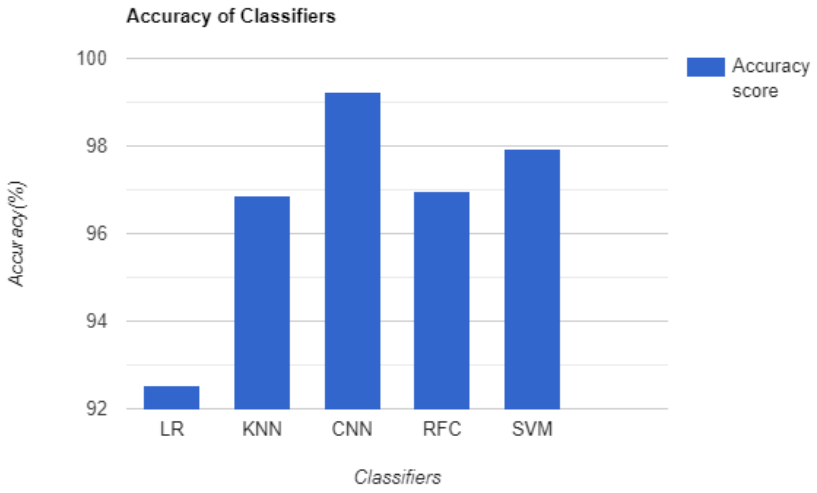

Fig 13: Comparison of accuracy

As CNN got high accuracy, it is used for training the model. The accuracies of the training and the testing dataset are plotted below.

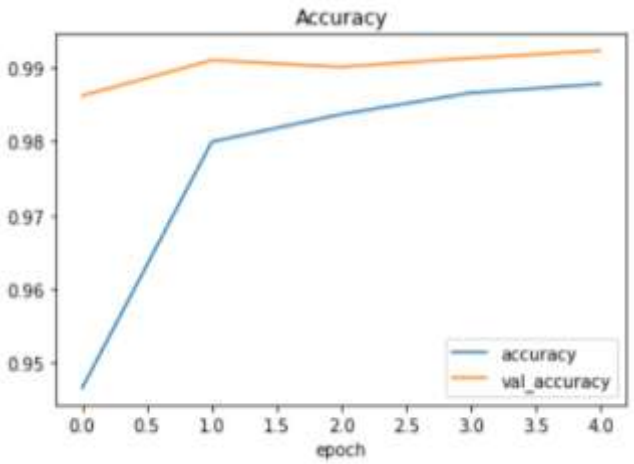

Fig 14: Training and Testing accuracy

So, the training accuracy of CNN is 99.23% and test accuracy is 99.53%.

Model	Loss	Accuracy
CNN	2.67	99.63

VIII. Conclusion

Convolutional Neural Network gets trained from the real-time data and makes the model very simple by reducing the number of variables and gives relevant accuracy. In our project, we used CNN with some libraries like Keras, Matplotlib, CV2, Tensorflow to get the maximum accuracy.

A comparison on different Machine Learning algorithms like Random Forest Classifier, Convolutional Neural Network, Linear Regression, K-Nearest Neighbors, Support vector machine is done, in which the accuracy for CNN is 99.63%. This project can be taken to next level by extending its scope to different writing styles. We can increase the accuracy by

- Taking huge datasets
- Adopting many suitable algorithms
- Hyper-parameter tuning
- Compile the model with a greater number of epochs

ACKNOWLEDGMENT

We would like to thank Dr. P.V. Venkateswara Rao (Senior Asst Prof, Dept of CSE, MITS, Madanapalle) for his valuable suggestions and advice that helped us improve this works, and his support is greatly appreciated. We also would like to acknowledge the help of MITS, Madanapalle for the kind support provided by our faculty and friends.

REFERENCES

[1] Y. LeCun et al., "Handwritten digit recognition with a backpropagation network," in Advances in neural information processing systems, 1990, pp. 396-404

[2] Xiaofeng Han and Yan Li (2015), "The Application of Convolution Neural Networks in Handwritten Numeral Recognition" in International Journal of Database Theory and Application, Vol. 8, No. 3, pp. 367-376.

[3] Caiyun Ma, Hong Zhang (2015), "Effective Handwritten Digit Recognition Based on Multi-feature Extraction and Deep Analysis", 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), pp. 297-301.

[4] Dan ClaudiuCiresan, Ueli Meier, Luca Maria Gambardella, JurgenSchmidhuber, "Deep big simple neural Nets Excel On Handwritten Digit Recognition", MIT Press, 1st March 2010.

[5] Xiaofeng Han and Yan Li (2015), "The Application of Convolution Neural Networks in Handwritten Numeral Recognition" in International Journal of Database Theory and Application, Vol. 8, No. 3, pp. 367-376.

[6] Yoshihiro Shima, Meisei, Yumi Nakashima, Michio Yasuda, Meisei (2017), "Pattern Augmentation for Handwritten Digit Classification based on Combination of Pre-trained CNN and SVM", 6th international Conference on informatics, Electronics and vision (ICIEV) and 7th International Symposium n Computational medical and health technology (ISCMHT).

[7] Matthew Y.W. Teow Artificial Intelligence Lab (21 October 2017), "Understanding Convolutional Neural Networks Using A Minimal Model for Handwritten Digit Recognition", 2017 IEEE 2nd International Conference on Automatic Control and Intelligent Systems (I2CACIS 2017), Kota Kinabalu, Sabah, Malaysia, pp. 167-172.

[8] Dan ClaudiuCires¸an, Ueli Meier, Luca Maria Gambardella, Jurgen Schmidhuber (March 2010), ¨Deep, Big, Simple Neural Nets for Handwritten Digit Recognition", arXiv, pp. 1-14.

[9] Li Deng (November 2012), "The MNIST Database of Handwritten Digit Images for Machine Learning Research", Best of the web series, IEEE signal processing magazine, pp. 141-142.

[10] Y. Le Cun, L. D. Jackel, B. Boser, J. S. Denker, H. P. Graf, I. Guyon, D. Henderson, R. E. Howard, W. Hub, "Handwritten Digit Recognition : Applications of Neural Network Chips and Automatic Learning" NATO ASI series F: Computer and system sciences, Vol. 68, pp. 41-46.

[11] Li Deng, "The MNIST Database of Handwritten Digits images for Machine Learning Research", MIT press, November 2012.

[12] Vineet Singh, Sunil Pranit Lal, "Digits recognition using single layer neural Network with principal component analysis", Computer Science and Engineering (APWC on CSE), 2014 Asia-Pacific World Congress IEEE, 4-5 Nov 2014.

[13] Retno Larasati, Hak Keung Lam, "Handwritten digits recognition using ensemble neural networks and ensemble decision tree", Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS), 2017 International Conference on, 8-10 Nov. 2017.

[14] L. Bottou, C. Cortes, "Comparison of Classifier methods a case study in handwritten digit recognition", Pattern Recognition, 1994. Vol. 2 Conference B; Image Processing, Proceedings of the 12th IAPR International. Conference IEEE, 06 August 2002.

[15] V.C.Bharati, K.Veningston, and P.V.Venkateswara Rao, "Query -Based Word Spotting in Handwritten Documents Using HMM", Springer Nature Singapore pte Ltd.2020,Data Engineering and communication Technology, Advances in Intelligent systems and computing 10789.

[16] R.Sudhakar,P.V.Venkateswara Rao(2019), "Video Super Resolution Using non - Linear regression and deep Learning ", The Imaging Science Journal,DOI:10.1080/13682199.2019.1652445A.

[17] Hongjian Zhan, ShujingLyu, Yue Lu Shanghai (August 2018), "Handwritten Digit String Recognition using Convolutional Neural Network", 24th International Conference on Pattern Recognition (ICPR), pp. 3729-3734.

[18] Seiichi Uchida, Shota Ide, Brian Kenji Iwana, Anna Zhu (2016), "A Further Step to Perfect Accuracy by Training CNN with Larger Data", 15th International Conference on Frontiers in Handwriting Recognition, 405-410

[19] Alessandro Giusti, Dan C. Cires□an, Jonathan Masci, Luca M. Gambardella, Jurgen Schmidhuber (2010), "Fast image scanning with deep max-pooling convolutional neural networks" Published in 17th IEEE conference on image processing, pp. 4034-4038.

[20] Vladimir Golovko, MikhnoEgor, AliaksandrBrich, and AnatoliySachenko (October 2016), "A Shallow Convolutional Neural Network for Accurate Handwritten Digits Classification" 13th international conference, PRIP, Minsk, Belarus, pp. 77-85

[21] M. Wu and Z. Zhang, "Handwritten digit classification using the mnist data set," Course project CSE802: Pattern Classification & Analysis, 2010.

[22] Yoshiki Mizukami, Taiji Sato, Kanya Tanaka (2000), "Handwritten digit recognition by hierarchical displacement extraction with gradual prototype elimination" Published in Proceedings 15th International Conference on Pattern Recognition (ICPR), pp. 339-342.

[23] A. Calderon, S. Roa and J. Victorino Grupo Simbiosis Bogota, Colombi (2003), "Handwritten Digit Recognition using Convolutional Neural Networks and Gabor filters" Proceedings of the International Congress on Computational Intelligence (CIIC), pp. 1-8.

[24] Seiichi Uchida, Shota Ide, Brian Kenji Iwana, Anna Zhu (2016), "A Further Step to Perfect Accuracy by Training CNN with Larger Data", 15th International Conference on Frontiers in Handwriting Recognition, 405-410.

[25] T Siva Ajay (July 2017), "Handwritten Digit Recognition Using Convolutional Neural Networks" International Research Journal of Engineering and Technology (IRJET), Vol. 04, Issue 07, pp. 2971-2976.