GALGOTIAS UNIVERSITY
(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

CLASSIFICATION OF IMAGES REPRESENTING THE NUMERICAL DIGITS WITH THE HELP OF CONVOLUTIONAL NEURAL NETWOK

A Report for the Evaluation 3 of Project 2

Submitted by

ASHUTOSH GUPTA

(1613107016 / 16SCSE107053)

in partial fulfillment for the award of the degree

of

Bachelor of Technology

IN

Computer Science and Engineering With Specialization of

Business Analytics and Optimization

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

Under the Supervision of

MS. VARSHA SISAUDIA

Assistant Professor

MAY- 2020

TABLE OF CONTENTS

Abstract

In recent years, with an increase in the amount of digital contents, identification and classification of images has become an exciting task in the fields of computer science. Automatic understanding and analyzing of images by intelligent devices can uncover so much information which might be difficult for humans to interpret . A lot of research has been done to overcome problems in existing classification system, but the output had been narrowed only to low level image primitive. Further, those approaches lack precise classification of image. In this paper, our system uses deep learning algorithm to attain the supposed outcome in the area like computer visions. Our system presents Convolutional Neural Network (CNN), a machine learning algorithm being used for automatic classification of images. It uses the Digit of MNIST data set as a bench mark for classification of grayscale images. The grayscale images in the data set are used for training which requires more computational power for classification of images. By training the images using CNN network we obtain the 97% accuracy result in the experimental part. It shows that our model achieves high accuracy in classification of grayscale digit images.

1. Introduction

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. 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 . Handwriting styles even differ when the same person writes a digit twice at two different places. In numeral recognition, the digits are written on a paper and are converted into digital images using a classifier, SVM is the most famous classification technique. To solve the stated problem there are mainly three approaches statistical, multilayer neural network and deformable ones. This paper focuses on digit recognition using multilayer neural network. It mainly comprises of three phases that is pre-processing phase, feature extraction and then the final phase of classification. The pre-processing phases comprises of capturing the image in a spatio-luminance representation and then reducing the noise etc. Basically in the feature extraction phase the image is divided into multiple parts and even the finest details and features are extracted from the image. Then the system is trained on extracted features using different techniques like multilayer perceptron and SVM. From this, a trained model is designed that is used for classifying the test image. For this, details features are withdrawn from the test image and these features are fed to the trained model. Then applying many techniques and procedures such as Multilayer Perceptron(MLP) and Support Vector Machines(SVM) etc. This phase mainly focusses on training the system by extracting even the finest details from the image. Then comes the classifying stage in which all the details extracted from the image is compared by the trained data set and the image with which most of its characteristics matches is finally given the tag. A CNN is a deep learning method that has been extensively used in developing applications for computer vision, natural language processing, data mining, computer games and handwritten recognition. LeNet5 is the base architecture of CNN. LeNet-5 was one of the first multilayer neural network where the concept of Convolutional neural network (CNN) was used in 1990. As the topic was researched further, more layers were added making the system more accurate but also started to increase the computational time. The computational time can be reduced to some extent by using better GPU's. But even then achieving 100% accuracy rate is practically impossible. Although in the past few years CNN has performed well but still human beings have better ability. Mainly, the database for western roman numerals classification using deep learning is Modified National Institute of Standards and Technology database (MNIST) which comprises of about 60,000 images for training the system and about 14,000 images for testing it Then, finally the trained system can be used for practical purposes such as recognizing numerals written on bank cheques, sorting of mails etc.

In this we have used MNSIT(Modified National Institute of Standards and Technology) 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. MNIST database is considered as modified form of NIST Database. In MNIST there are 60,000 images which are used for training the system, and for validation purpose. These black and white digits are distributed uniformly and centered in a image which is of fixed size with 28*28 pixels. The dimension of every sample image vector is 28*28=784.

2. Existing System

Wu et al. have applied deep learning to the real-word handwritten character recognition, and obtained good performance for image recognition. They analyzed the different between CNN and DBN by comparing the experiment results. Deep learning can approximate the complex function through deep nonlinear network model. It does not only avoid the large workload of manually extract features, but also it is better to describe potential information of the data. However, they did not consider the evaluation factors as execution time. “Kaensar et al. have concluded that different classifier affects the recognition rate for handwritten digit recognition. Accordingly, they applied three classification techniques by using the open source Weka tool kit for training and testing the dataset which was obtained from the UCI repository. The presented results show that SVM is the best classifier to recognize handwritten digits. However, the main problem of the SVM classifier is the time consuming of the training process. Conversely, other methods like neural networks give insignificantly worse results, but their training is much quicker . Saabni et al. have presented an algorithm that trains ksparse auto encoders and used their hidden layers to be stacked as retrained hidden layers into a deep neural network. Their proposed system is a part of a more complex system which aims to analyze images of checks in order to extract and recognize important information such as amounts and texts from checks images. To avoid training of deep layer with the back propagation algorithm directly on randomized weights, the first two layers have been trained a side using sparse auto encoders to extract important features in hierarchical manner .

Table 1. Side-by-side comparison of the most competitive (error rate < 1%) results found in the state of the art for the MNIST database with data augmentation or preprocessing. Best achieved performances are boldfaced.

Technique	**Error rate**
CNN (2 conv, 1 dense, relu) with DropConnect	0.21%
Committee of 25 CNNs	0.23%
CNN with APAC	0.23%
CNN (2 conv, 1 relu, relu) with dropout	0.27%
Committee of 7 CNNs	0.27%
Deep CNN	0.35%
CNN (2 conv, 1 dense), unsup pretraining	0.39%
CNN, XE loss	0.40%
Scattering convolution networks + SVM	0.43%
Feature Extractor + SVM	0.54%
CNN Boosted LeNet-4	0.70%
CNN LeNet-5	0.80%

Table 2. Side-by-side comparison of the most competitive (error rate < 1%) results found in the state of the art for the MNIST database without data augmentation or preprocessing. Best achieved performances are boldfaced.

Technique	**Error rate**
Convolutional highway networks	0.45%
CNN (5 conv, 3 dense) with retraining	0.46%
Network-in-network	0.47%
CNN (3 conv, 1 dense), stochastic pooling	0.49%
CNN (2 conv, 1 dense, relu) with dropout	0.52%
CNN, unsup pretraining	0.53%
CNN (2 conv, 1 dense, relu) with DropConnect	0.57%
SparseNet + SVM	0.59%
CNN (2 conv, 1 dense), unsup pretraining	0.60%
CNN LeNet-5	0.95%
CNN (2 conv, 2 dense)	0.62%
Boosted Gabor CNN	0.68%

3. Proposed Method

A popular demonstration of the capability of deep learning techniques is object recognition in image data. In this, we will develop a deep learning model to achieve near state of the art performance on the MNIST handwritten digit recognition task in Python using the Keras deep learning library. we will mainly use the Convolutional neural network.

Convolutional neural network

CNN is a category deep neural networks which is used to analyzing the visual images.CNN are regularised type of multilayer perceptron. Multilayer perceptron is fully connected neural network where each neuron is connected to all neurons in the next layer.CNN are insipired by biological neural network

A CNN comprises of an input and output layer as well as multiple hidden layers.these hidden layers of CNN basically comprises set of convloutional layers that perform convolve operation using multiplication or dot product. RELU layer mainly performs as activation function along with it there are hidden layers like pooling layers , fully connected layer , normalization layer whose input and output hide by activation function,

CNN consists of input layer as first layer where the size of input image is 28 X 28.Convolutional layer and pooling layer are the second and third layer respectively. convolutional layer acquire four distinct feature maps by convolutional with the images. pooling layer performs the calculation of local mean or largest of feature maps.

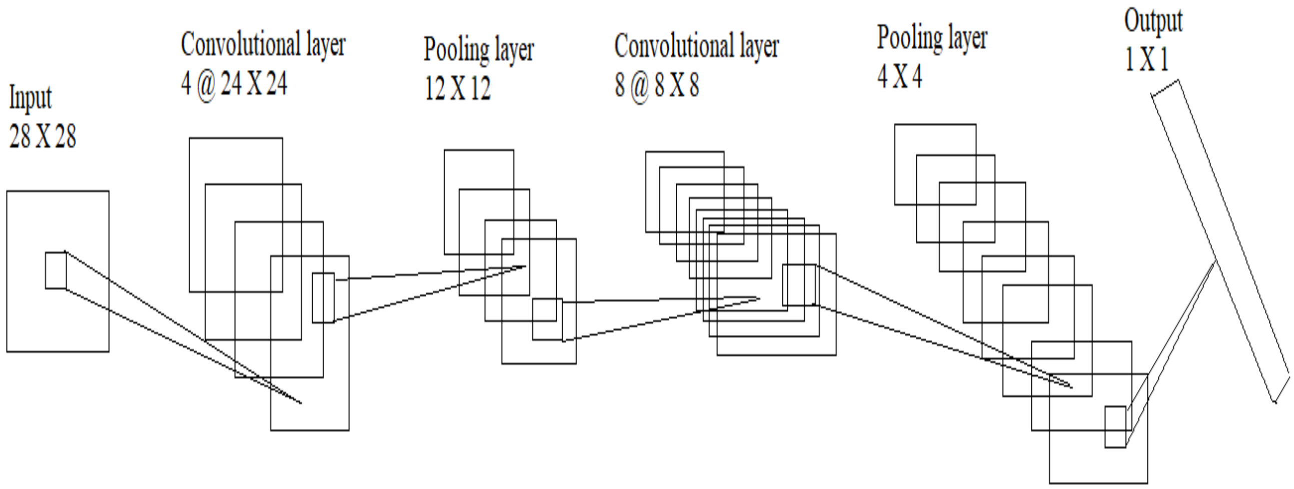

Figure 1 – Basic architecture of CNN

Table 3- Shows the steps in Proposed Method

Steps in Proposed Method
Data Preprocessing
Prepare Pixel data
Define Model
Feature Extraction
Evaluate the Model

4. Implementation or architecture diagram

4.1 Data Preprocessing

Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. Pre-processing refers to the transformations applied to our data before feeding it to the algorithm.

For achieving better results from the applied model in Machine Learning projects the format of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format, for example, Random Forest algorithm does not support null values, therefore to execute random forest algorithm null values have to be managed from the original raw data set.

Another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in one data set, and best out of them is chosen.

4.2 Prepare Pixel Data

We know that the pixel values for each image in the dataset are unsigned integers in the range between black and white, or 0 and 255. We do not know the best way to scale the pixel values for modeling, but we know that some scaling will be required. A good starting point is to normalize the pixel values of grayscale images, e.g. rescale them to the range [0,1]. This involves first converting the data type from unsigned integers to floats, then dividing the pixel values by the maximum value. The prep_pixels() function implements these behaviors and is provided with the pixel values for both the train and test datasets that will need to be scaled.

4.3 Define Model

We need to define a baseline convolutional neural network model for the problem.The model has two main aspects: the feature extraction front end comprised of convolutional and pooling layers, and the classifier backend that will make a prediction.

For the convolutional front-end, we can start with a single convolutional layer with a small filter size (3,3) and a modest number of filters (32) followed by a max pooling layer. The filter maps can then be flattened to provide features to the classifier.

Given that the problem is a multi-class classification task, we know that we will require an output layer with 10 nodes in order to predict the probability distribution of an image belonging to each of the 10 classes. This will also require the use of a softmax activation function. Between the feature extractor and the output layer, we can add a dense layer to interpret the features, in this case with 100 nodes.

All layers will use the ReLU activation function and the weight initialization scheme, both best practices.

We will use a conservative configuration for the stochastic gradient descent optimizer with a learning rate of 0.01 and a momentum of 0.9. The categorical cross-entropy loss function will be optimized, suitable for multi-class classification, and we will monitor the classification accuracy metric, which is appropriate given we have the same number of examples in each of the 10 classes.

4.4 Feature Extraction

There are different types of features or characteristics are extracted for the classification of handwritten numeral digits. Different algorithms used for feature extraction have different types of error rate. Errors made by each separate algorithm does not overlap, therefore we can combine all types of features extraction methods which will lead to a perfect recognition rate. This method helps to reject the ambiguous digits recognition and improve the recognition rate of misclassified digits that can be recognized by humans. Here different types of features extracted are given below

Structural Characteristics - In this, the algorithm extracts histograms profiles and then convert it to a single feature vector. The input image is resized in a 32x32 matrix. Radial histograms are calculated by counting the number of pixels in 72 directions at 5 degree intervals.

2. Modified Edge maps - The input image is divided to small parts of 25x25 each. Sobel operators are used to obtain mainly four edge maps; horizontal, vertical and the two diagonals. These four maps are divided to 25 further images of 5x5 pixels each. The features obtained are then merged to make a single feature vector having 125 features. 3. Image projections- It mainly extracts diagonal and radial projections. For this, the image is divided into four parts like quadrants; top, bottom, left and right. This is done to remove rotational uniformity which is entirely not required. Radial projections are attained by centering the image pixels by grouping it with its radius. These normalized features are then converted to a single vector which has 128 features.

Multi zoning - In this the percentage of black pixels is used as a feature from several images subdivided. To obtain better results many different configurations are used. In this 13 different configuration features are used (3 by 1, 1 by 3, 2 by 3, 3 by 2, 3 by 3, 1 by 4, 4 by 1, 4 by 4, 6 by 1, 1 by 6, 6 by 2, 2 by 6 and 6 by 6).

Concavities Measurement- First the image is converted to a 18x15 size matrix, then it is divided to six zones, each having 13-d feature vector. For any white pixel the algorithm searches for any black pixel in all the directions. The feature vector of each zone are formed into a single vector having 78 features.

Gradient Features - It calculates the gradient elements in a grayscale image. The reason for the use of grayscale image is that it has more information. First, the input image is modified to a pseudo-grayscale using Medial Axial Transformation(MAT). To generate amplitude and phases Sobel operators are used. The image is branched to 16 sub-images and for each sub image, the number of pixels in all the eight directions is counted as a feature.

4.5 Evaluate the Model

After the model is defined, we need to evaluate it.the model will be evaluated using five-fold cross-validation. the value of *k=5* was chosen to provide a baseline for both repeated evaluation and to not be so large as to require a long running time. Each test set will be 20% of the training dataset, or about 12,000 examples, close to the size of the actual test set for this problem.The training dataset is shuffled prior to being split, and the sample shuffling is performed each time, so that any model we evaluate will have the same train and test datasets in each fold, providing an apples-to-apples comparison between models.We will train the baseline model for a modest 10 training epochs with a default batch size of 32 examples. The test set for each fold will be used to evaluate the model both during each epoch of the training run, so that we can later create learning curves, and at the end of the run, so that we can estimate the performance of the model.

As such, we will keep track of the resulting history from each run, as well as the classification accuracy of the fold.

5. Output

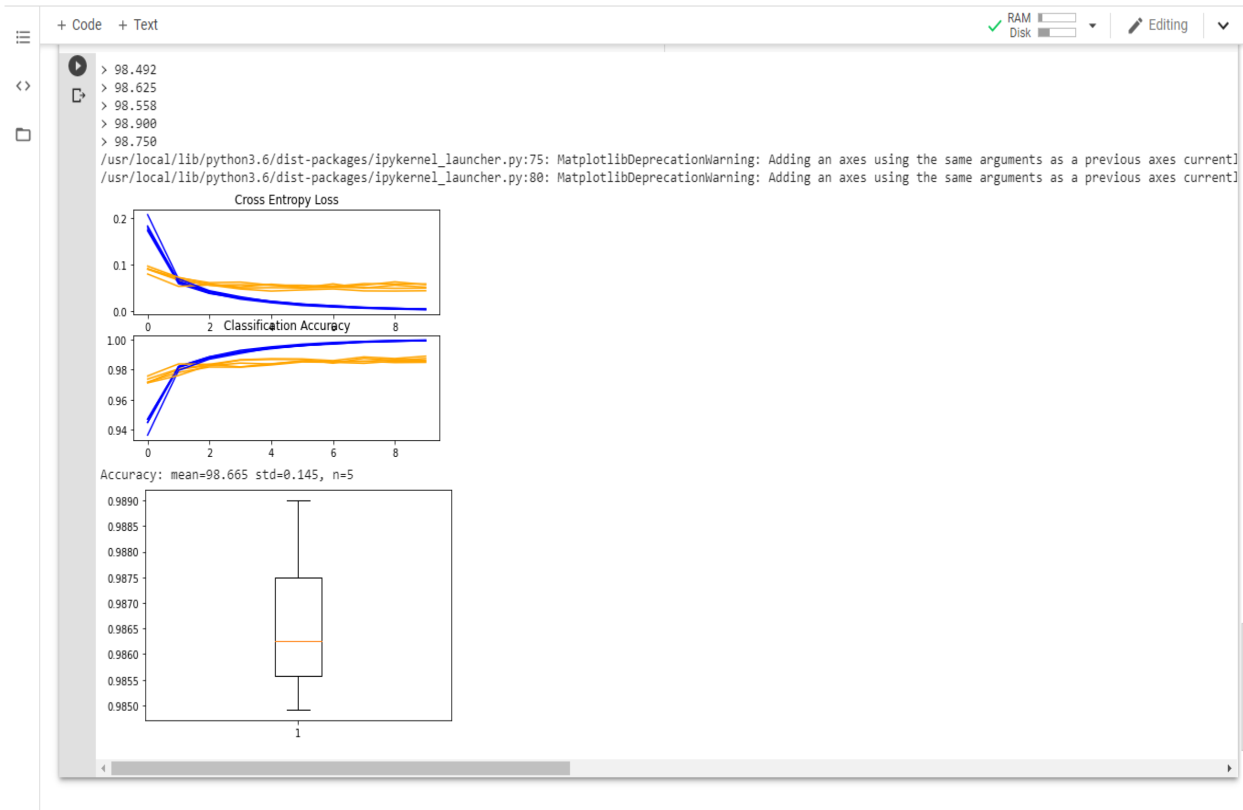

Figure 2- Shows the mean accuracy of baseline model, Loss and Accuracy learning curves for the baseline model during K-fold Cross Validation

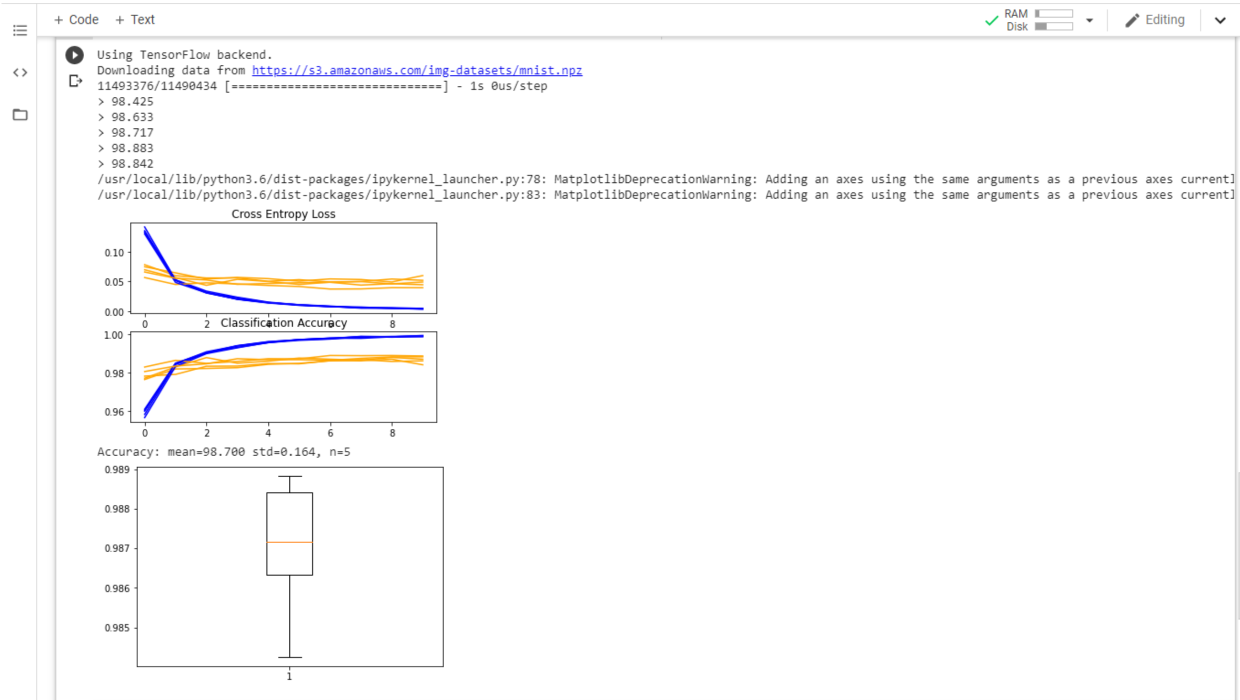

Figure 3 – Shows the mean accuracy of baseline model , Loss and Accuracy learning curves for the baseline model during K-fold Cross Validation after the Batch Normalization

Figure 4 - Shows the mean accuracy of deeper model, Loss and Accuracy learning curves for the deeper model during K-fold Cross Validation

6. Conclusion/Future Enhancement

In this a CNN model has been proposed to solve the handwritten digit recognition problem. This model took the CNN as an automatic feature extractor The efficiency and feasibility of the proposed model were evaluated in two aspects: the recognition accuracy and the reliability performance. Experimental results on the MNIST digit database showed the great benefit of the proposed CNN model. It achieved an error rate of 0.90 % with no rejections, and 100% reliability with a rejection rate of 0.56%. Both are the best results up to date compared with other research works. Our results indicate that the proposed CNN model is quite a promising classification method in the handwriting recognition field due to property that the salient features can be automatically extracted by the CNN model, while the success of most other traditional classifiers relies largely on the retrieval of good hand-designed features which is a laborious and time consuming task.. Research on the CNN learning model is continue.The performance of the CNN model can be further improved through the fine tuning of its structure and its parameters. For example, improvements might be made based on the size of the input layer, the number of feature layer maps in layers 2 to 4, the kernel functions used in the model, etc. Extending the proposed CNN model to other applications is a task worth investigating. It is very easy to apply our work on the isolated special symbols, such as ‘‘,’’, ‘‘.’’, ‘‘?’’, ‘‘!’’ etc. Without being limited to the handwritten digit recognition.

7. References

[1]. 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.

[2]. 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).

[3]. Xiao-Xiao Niu n , Ching Y. Suen (April 2012), “A novel hybrid CNN–SVM classifier for recognizing handwritten digits”, Elsevier, Vol. 45, Issue 4, Pages 1318-1325.

[4]. 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.

[5]. 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.

[6]. 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.

[7]. 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.

[8]. 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.

[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]. Rafael M. O. Cruz, George D. C. Cavalcanti and Tsang Ing Ren (2010), “Handwritten Digit Recognition Using Multiple Feature Extraction Techniques and Classifier Ensemble.” 17th Internatonal conference on systems, signals and image processing (IWSSIP), pp. 215-218.

[11]. 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.

[12]. 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. [13]. 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.

[14]. 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.

[15]. 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).

[16]. 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.

[17]. 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.