Study on the Effects of Increase in the Depth of the Feature Extractor for Recognizing Hand Written Digits in Convolutional Neural Network

Noor - I - Tasnim Monir[1], Umme Nusrat Tabassum[2], Mahabuba Yesmin[3], Ahmed Salman Tariq[4]

[1,2,3]B.Sc. in Engineering, Dept of Computer Science & Engineering, Bangladesh Army University of Engineering & Technology, Qadirabad, Natore-6431
[4]Lecturer, Dept. of Computer Science & Engineering, Bangladesh Army University of Engineering & Technology, Qadirabad, Natore-6431

Abstract - *Recent year's statistics shows that, the dramatic improvements of modern Convolutional Neural Network (CNN) has made a great impact in the field of Machine Learning (ML) as well as in Artificial Intelligence (AI). Deep learning and Artificial Neural Network (ANN) have changed the traditional view of machines perceptions by making them more artificially intelligent and accurate. The applications of Convolutional Neural Network (CNN) can be easily observed in some specific fields like robotics, health care, surveillance, sports and so on. Now the Convolutional Neural Network (CNN) has been using for recognizing patterns, sentences, documentations, hand written digits and signatures as well. So, it has opened a new era of research for the scientists and researchers. The main purpose of this paper is to analyze the performance of CNN for recognizing the hand written digits in Python based machine learning platform TensorFlow, with the variations of the numbers of filters in the proposed model. The analytical process has been performed on Modified National Institute of Standards and Technology (MNIST) data set. Farther, some common algorithms like Stochastic Gradient Descent (SGD) and back propagation process has been used for proper training of the proposed network.*

Key Words: **Convolutional Neural Network (CNN), MNIST dataset, Filters, Machine Learning (ML), Stochastic Gradient Decent (SGD), Artificial Neural Network (ANN), Back propagation.**

1. INTRODUCTION

The machines those we are using right now in our daily life are much smarter than they were used in couple of years ago. Today's machines are now often defined by their problem-solving capability and computational capacity. Artificial Intelligence (AI), Emotional Intelligence (EI), Pattern Recognition, Predictive Analytics, Data Mining capability are some common features of today's Machines. All this kind of modern features has become possible for the dramatic improvements of some well-known fields like Convolutional Neural Network (CNN), Artificial Neural Network (ANN), Machine Learning (ML) etc. One of the common applications of this smart devices is the recognition process of hand written digits. Bank checks, Postal ZIP codes on letters, Medical records, Everyday receipts are some fields where hand written digits are now prevailing. Recognition process of these digits is a complex issue as different people has different style and type of hand writing. So, from the point of view of a machine, it is not an easy task to recognize all types of hand written digits. Therefore, the accuracy rate of hand written digit recognition is an important factor for today's researchers dealing with Machine Learning (ML).

The modern computer science has developed different types of smart models for the proper evaluation and development of this field. Convolutional Neural Network (CNN) based advanced model is one of them. The model performs the convolution operations between its different layers and generates the appropriate output. The output can be affected or biased using the weights and bias for the correspondent nodes. The applications of Deep Convolutional Neural Networks (CNN) for predicting and recognizing the hand written digits have reached the perfection of human level. In 1998, the first initial framework of Convolutional Neural Network (CNN) has designed by LeCun et al [1]. The proposed model has seven layers of CNN for adapting the hand written digits classification directly form the pixel value of different images [2]. Back propagation and the Stochastic Gradient Decent (SGC) algorithms are used for the performance evaluation of the given model [3]. For the training of the network a cost function is generated which compares the original output of the network with the desired output for that network. The signal propagates back to the system again and again for updating the shared biases and weights for the respective fields to minimize the value of cost function. The process is known as back propagation process which increases the performances of the network [4][5]. The main aim of this paper is to observe the performance of a proposed CNN model with the variations of the number of filters used in the depth of the model for recognizing the handwritten digits.

The performance of the proposed CNN model has been observed on Modified National Institute of Standards and Technology (MNIST) dataset using the Python based neural network library and Machine Learning (ML) platform TensorFlow. The main purpose of this paper is to analyze the variation of accuracy and outcome results for using different combination of the number of filters applied in the proposed Convolutional Neural Network model. The feed forward algorithm, Stochastic Gradient Decent and backpropagation algorithm are applied for better training and testing accuracy.

2. LITERATURE REVIEW

Digit recognition is a impotent sector for all the modern sectors of computer science like Machine Learning, Deep Learning, Deep convolutional neural network etc. Hand written digit recognition is more complex and tough sector of research as the variations of hand writing styles. Some special sectors like image processing, pattern recognition is mainly affected by the evaluation of CNN model.

The theory of back propagation model for neural network has been described in [9]. CNN is used for the fault detection and classification in nano technology of manufacturing silicon chips and semiconductors [13]. The deep learning algorithms of multilayer CNN using Keras and Theano and TensorFlow generates the highest accuracy compared to other deep learning and machine learning algorithms like RFC, KNN, SVM etc. As CNN provides the highest accuracy, it is used for various applications like image classification, video analysis etc. CNN has been applied in recognizing the sentences, natural language processing with the variations of different internal parameters [14]. A comparison of performance has been made between MNIST, NORB dataset also with CIFAR10 dataset and shown that Deep Neural Network performs better than other methods [15]. Applying the MNIST dataset an error rate of 1.19% has achieved using the 3-NN [22]. The Coherence recurrent convolutional network (CRCN) as a multidimensional has been applied for recognizing the sentences in the image of hand writing [17]. The error rate has been observed with the MNIST and CIFAR dataset for better performance evaluation [19]. An improved method of hand written digit recognition has been proposed in [27]. The performance of MNIST dataset for recognizing hand written digits has shown in [24]. The epoch size has a great impact of detecting the overall performance in CNN model. Therefore, an overall performance comparison of CNN layers corresponding to the epoch size for detecting hand written digits has shown in [29].

After reviewing the related works, it has been observed that different researchers has implemented various models of CNN for measuring the accuracy, but there is a lack of study on the behavior of CNN models for recognizing hand written digits corresponding to the increase in the depth of feature extractor as well as increase in the number of filters.

In this paper a details performance evaluation of hand written digit recognition has been presented for the proposed CNN model corresponding to the effects of increasing number of filters.

3. PROPOSED MODEL FOR MEASURING PERFORMECE

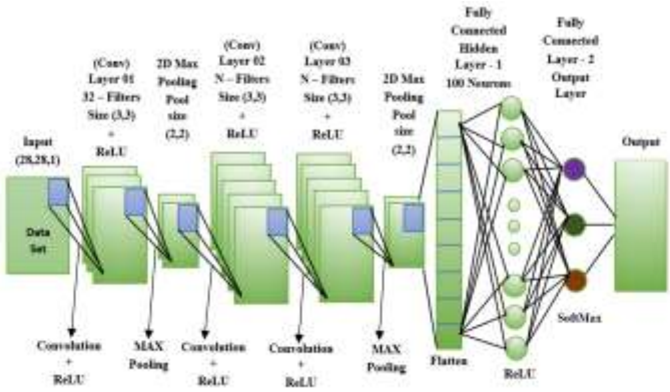

Fig – 01: Proposed CNN Model

The proposed model consists of different combinations of CNN layers. Fig-01 above shows the proposed model. It has been used for MNIST dataset classification. The input layer consists consist of (28,28,1) input matrix which indicates that the network contains 784 neurons as the input data. 0 represents the white and 1 represents the black pixels for the processing of datasets. The convolution operation is performed initially with 32 (fixed) number of filters followed by the activation function ReLU (Rectified Liner Unit) for extracting features form the input data. Here the kernel size determines the locality of the filters. The kernel size applied in this section is (3,3). In the next session, 2D Max pooling operation is performed with a kernel size of (2,2) for subsampling the dimension of feature map. The main motive of using the activation function ReLU after every convolutional operation is to enhance the performance of the model. The applications of pooling layer in the proposed model, reduces the output information as well as the number of parameters and the computational complexity for the convolutional layers.

The main experimental layers are the convolution layer 02 and convolution layer 03. Here the total number of filters is a variable digit and denoted by “N”. The value of N has been changed for individual approaches while performing the experiment. Both of the convolution layer 02 and convolutional layer 03 has the same size of kernel that is (3,3) and each of them is followed by a ReLU layer. Farther, the 2D Max pooling operation is applied for better performance before entering the fully connected layer in the flatten section. The flatten layer after the pooling layer actually converts the 2D featured map matrix into a 1D feature vector for allowing the output to be handled properly by the fully connected layer. The fully connected layer consists of 100 neurons and later on it is followed by another ReLU layer. The dropout regularization method is used at the fully connected layer to reduce the overfitting of the model. The method randomly turns off some neurons while training for the improvement of performance of the network. It helps for better generalization and less compiling to overfit the training dataset. The SoftMax function is applied to classify the output data before the output layer. It enhances the performance of the model by classifying the output digits form 0 through 9 containing the highest activation value.

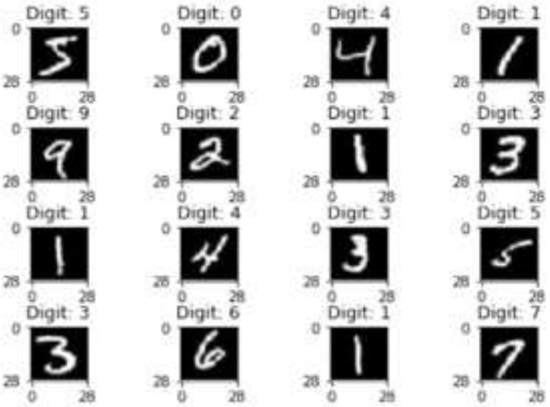

Fig – 02: MNIST handwritten dataset sample Image

The Modified National Institute of Standard and Technology [MNIST] handwritten digits [24] data base has been used for the evaluation of the proposed model. There are 70,000 scanned images of hand written digits in the data base of MNIST data set. All the datasets are classified under two categories. Total 60,000 thousand datasets are selected as the training dataset and rest of the 10,000 thousand are selected as the testing data set. The images are gray scale image having a size of 28×28 pixels. Here the input image is a 784-dimentional vector and the output is 10-dimentional vector. The performance of the proposed model can be modified with the help of cost function. The function is expressed with the following equation below [25].

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

Where w is the cumulation of weights in the network and b is the bias. The total number of training input is denoted by n and the actual output is denoted by a. The actual output a, depends on w, x, and b. The function C(w, b) is a non-negative term. C(w, b) = 0, when the desired output y(x) is equal to the actual output, a, for all the training inputs that is n. The cost function is designed in such a way that the cost is needed to be as minimum as possible with a self-regulation of biases and weights in that function. The total process is performed using the algorithm called Stochastic Gradient Descent (SGD) as the data size is very large. Through this algorithm, a small number of iterations will find the most cost-effective solutions for the problem regarding to the proper optimization. The algorithm effectively utilizes the following equations [25] given below.

$$w^{new}=w^{old}-\frac{\eta}{m}\frac{\partial C_{xj}}{\partial w^{old}} \qquad (2)$$

$$b^{new}=b^{old}-\frac{\eta}{m}\frac{\partial C_{xj}}{\partial w^{old}} \qquad (3)$$

With the use of the equation (2), (3), the output of the network can be expressed like as follows:

$$a=f(z)=f(wa+b) \qquad (4)$$

For calculation of the effective weight that contributes in the total error of the network, the Backpropagation method has applied. It changes the weights for the neural network continuously, until the effective and efficient result in not obtained.

4. RESULT ANALYSIS & DISCUSSION

To evaluate the performance of the proposed model, the k-fold cross validation technique has been applied. The iterative process has been followed both for training and testing the datasets to measure the performance of proposed model. The dataset has been classified into ten sections therefor, the value of "k" becomes10 for all the observation based on the number of filters (N). The values of "N" have been taken as 16, 32, 64, 128 which indicates the number of filters applied in Convolution layer-2 and Convolution layer-3. For any distinct value of "N" we have total ten accuracies as the value of "K" has been selected as ten. Later on, the mean accuracy for all the iteration has been calculated. To measure the variations of accuracies, the standard deviation (SD) of all accuracies has been calculated. For estimating the performance of a model for a particular training run, we have split all training dataset into a train and validation dataset. Performance on the training and validation dataset over each run has plotted to provide the learning curves which indicates how well a model is learning the problem respect to its accuracy.

The total experiment has been classified under four cases. Initially, for test case-1, total 16 numbers of filters have been applied in convolution layer-2 and convolution layer-3 (as N=16) and total 10 different accuracy rates are gained as k=10. Then the same process has been repeated for test case-2, test case-3 and test case-4 where the values of "N" was 32, 64 and 128. For all the test cases, the filter size in the convolution layer-2 and convolution layer-3 was (3,3). During the experiment, the epoch size has been selected as 10 and batch size has been selected as 32 for each of the test cases.

Table-01: Overall performance of proposed CNN model with the increase in depth of feature extractor (N)

Test case-1 (N=16), (epoch=10) (Total "k"=10)			Test case-2 (N=32), (epoch=10) (Total "k"=10)			Test case-3 (N=64), (epoch=10) (Total "k"=10)			Test case-4 (N=128), (epoch=10) (Total "k"=10)		
Batch Size	Fold No: (K)	Accuracy (%)	Batch Size	Fold No: (K)	Accuracy (%)	Batch Size	Fold No: (K)	Accuracy (%)	Batch Size	Fold No: (K)	Accuracy (%)
32	1	98.850	32	1	98.800	32	1	98.983	32	1	99.050
32	2	98.767	32	2	98.833	32	2	99.150	32	2	99.050
32	3	98.983	32	3	99.000	32	3	99.300	32	3	99.217
32	4	98.833	32	4	98.933	32	4	98.933	32	4	99.067
32	5	98.750	32	5	99.133	32	5	98.767	32	5	99.167
32	6	98.717	32	6	99.133	32	6	98.933	32	6	98.733
32	7	99.233	32	7	99.167	32	7	99.300	32	7	99.083
32	8	98.867	32	8	99.150	32	8	99.233	32	8	99.350
32	9	98.700	32	9	98.833	32	9	99.100	32	9	99.233
32	10	98.800	32	10	99.133	32	10	98.917	32	10	99.117
Mean Accuracy (%) = 98.850 Standard Deviation = 0.150			Mean Accuracy (%) = 99.012 Standard Deviation = 0.142			Mean Accuracy (%) = 99.067 Standard Deviation = 0.173			Mean Accuracy (%) = 99.107 Standard Deviation = 0.155		

The overall performance of all four test cases are shown in the Table-1. The table shows the performance of proposed CNN model with the variation of the number of filters as well as the feature extractor used in convolution layer 2 and convolution layer 3 for CNN model. As we have four different test cases, therefore the following graphs can be plotted for individua test cases.

4.1 GRAPH FOR TEST CASE-1 (N=16)

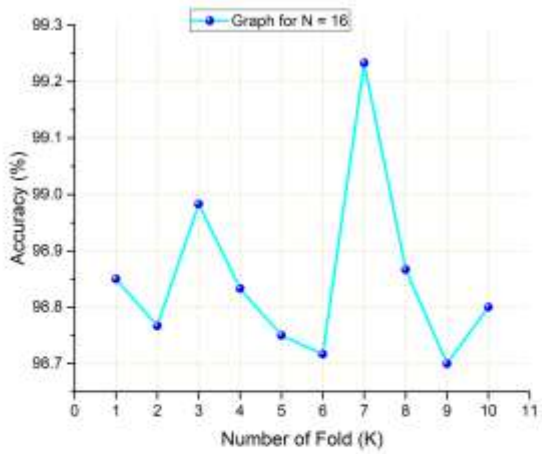

Fig-03: Generated Graph for Number of filters (N)=16

The figure-3 represents the characteristics curve or graph for applying N = 16 numbers of filters used in convolution layer-2 and convolution layer-3. The X axis represents the value of k fold cross validation and Y axis represents the accuracy in percentage. Here the maximum accuracy (99.233) gained at k = 7 and minimum accuracy (98.700) gained at k=9.

4.2 GRAPH FOR TEST CASE-2 (N=32)

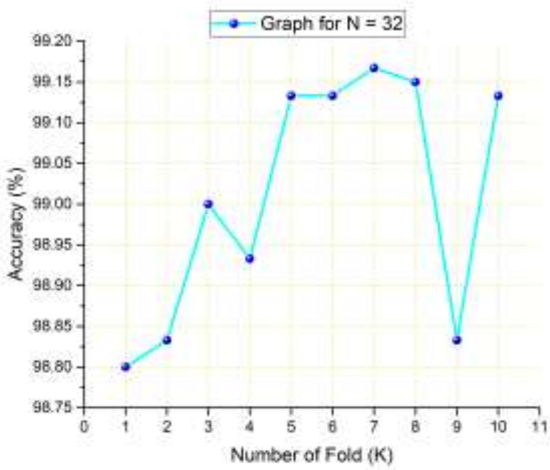

Fig-04: Generated Graph for Number of filters (N)=32

The figure-4 represents the characteristics curve or graph for applying N = 32 numbers of filters used in convolution layer-2 and convolution layer-3. The X axis represents the value of k fold cross validation and Y axis represents the accuracy in percentage. Here the maximum accuracy (99.167) gained at k = 7 and minimum accuracy (98.800) gained at k=1.

4.3 GRAPH FOR TEST CASE-3 (N=64)

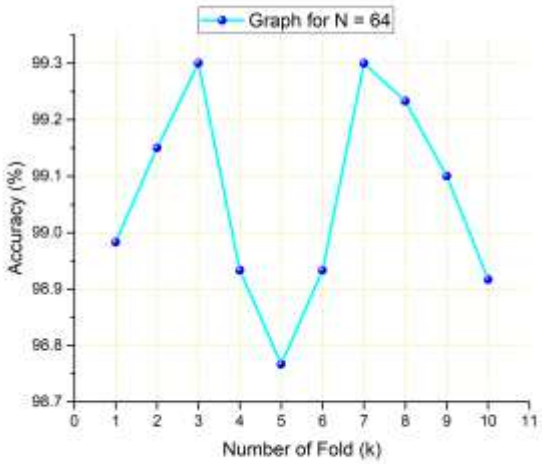

Fig-05: Generated Graph for Number of filters (N)=64

The figure-5 represents the characteristics curve or graph for applying N = 64 numbers of filters used in convolution layer-2 and convolution layer-3. The X axis represents the value of k fold cross validation and Y axis represents the accuracy in percentage. Here the maximum accuracy (99.300) gained at k = 3, and 7. In this case, the minimum accuracy (98.767) gained at k=5.

4.4 GRAPH FOR TEST CASE-4 (N=128)

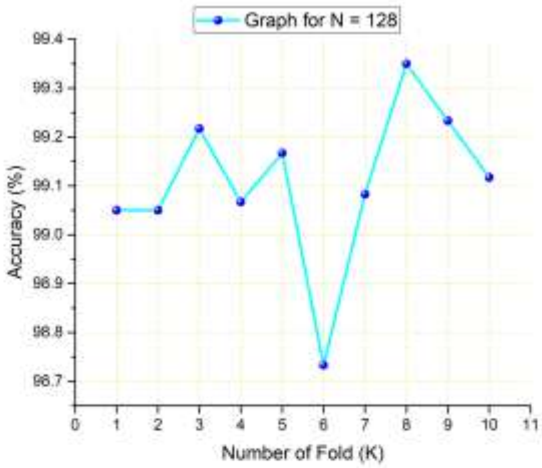

Fig-06: Generated Graph for Number of filters (N)=128

The figure-6 represents the characteristics curve or graph for applying N = 128 numbers of filters used in convolution layer-2 and convolution layer-3. The X axis represents the value of k fold cross validation and Y axis represents the accuracy in percentage. Here the maximum accuracy (99.350) gained at k = 8 and minimum accuracy (98.733) gained at k=6.

4.5 GRAPH FOR All TEST CASES (N=16,32,64,128)

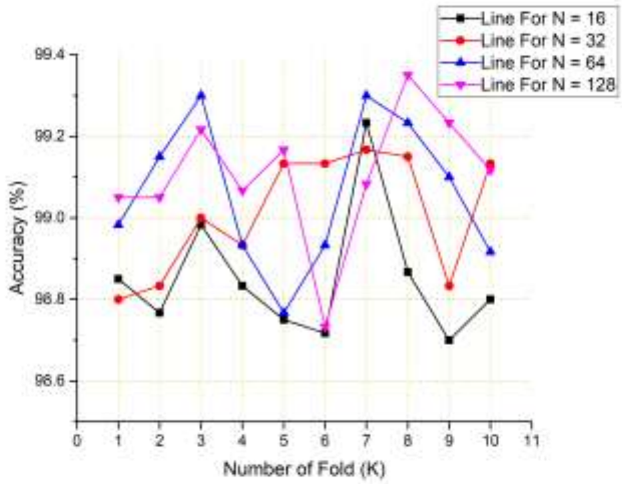

Fig-07: Overall graph for all test cases (N=16,32,64,128)

The figure-7 represents the characteristics curve for all test cases with the variation of filters (N=16,32,64.128) used in convolution layer-2 and convolution layer-3. The X axis represents the value of k fold cross validation and Y axis represents the accuracy in percentage. From this graph we can observe that the maximum accuracy (99.350) at K=8 has been achieved for N=128. The line is represented with Pink color. On the other hand, the minimum accuracy (98.700) at K=9 has been achieved for N=16. This indicates that the performance of CNN Model can be increased with the increase in number of filters used in convolution layers.

4.6 GRAPH RESPECT TO THE STANDARD DEVIATION OF ACCURACY

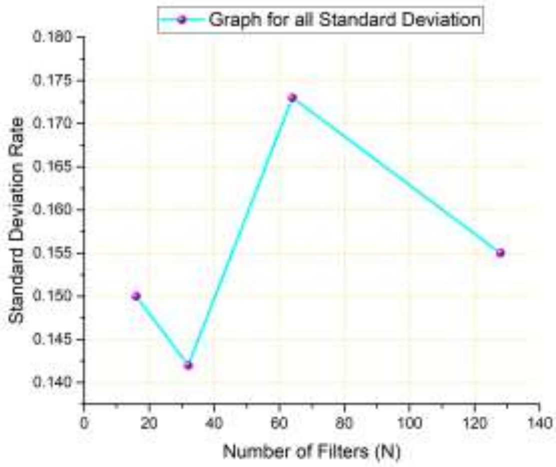

Fig-08: Graph for the standard deviation of accuracies respect to different test cases.

The figure-8 represents standard deviation rate of individual test cases in the experiment. Here the X axis represents the number of filters and Y axis represents the standard deviation rate. The highest standard deviation (0.173) has been achieved for applying 64 numbers of filters (Test case-

4) and the minimum standard deviation (0.142) has been achieved for applying 32 numbers of filters (Test Case-2).

4.7 GRAPH FOR MEAN ACCURACY IN ALL TEST CASES:

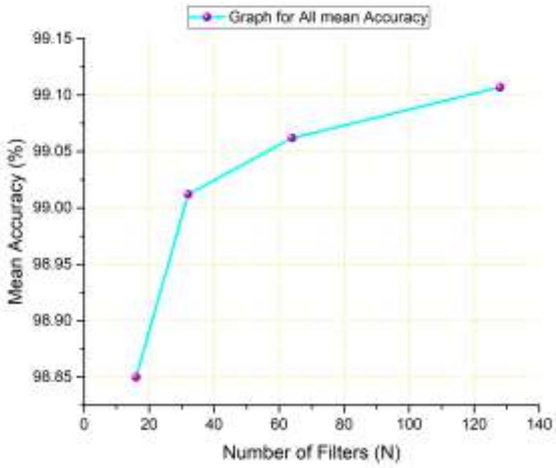

Fig-08: Generated Graph for mean Accuracy in all Test Cases.

The figure-8 represents the characteristics curve of mean accuracy corresponding to all test cases. The X axis represents the number of filters and Y axis represents the accuracy in percentage. Here the maximum accuracy (99.107) gained for the use of 128 numbers of filters at test case-4 and minimum accuracy (98.850) gained for the use of 16 number of filters at Test case-1.

From the discussion above, it can be easily observed that, the performance and accuracy of CNN model can be increased by increasing the number of feature extractor or number of filters applied in convolutional layers. The accuracy rate increases linearly with the increase in number of filters applied in CNN model. But an increase in number of filters also increases the computational time for the machines. So it is necessary to use the appropriate number of filters for better computational efficiency and better performance.

5. CONCLUSION AND FUTURE WORK

In this paper, the overall performance of CNN model has been studied respect to the number of filters applied in its convolutional layer. The study shows that, the performance of CNN model for recognizing handwritten digits can be increased with an increase in feature extractor or increase in number of filters applied in its different layers. The performance had been evaluated based on the accuracy and standard deviation. Four operational test cases have been performed individually to measure the performance with the variation of filter numbers applied in the proposed model. The future work consists of more deep analytical process for the proposed CNN model to ensure better accuracy with respect to the computational time, sensitivity and specificity.

REFERENCES

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

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, 1998.

[3] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in Neural networks for perception: Elsevier, 1992, pp. 65-93.

[4] S. J. Russell and P. Norvig, Artificial intelligence: a modern approach. Malaysia; Pearson Education Limited, 2016.

[5] S. Haykin, "Neural networks: A comprehensive foundation:MacMillan College," New York, 1994.

[6] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in Competition and cooperation in neural nets: Springer, 1982, pp. 267-285.

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

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradientbased learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, 1998.

[9] R. Hecht-Nielsen, "Theory of the back propagation neural network," in Neural networks for perception: Elsevier, 1992, pp. 65-93.

[10] Y. LeCun, "LeNet-5, convolution neural networks," URL: http://yann.lecun.com/exdb/lenet, vol. 20, 2015.

[11] S. J. Russell and P. Norvig, Artificial intelligence: a modern approach. Malaysia;Pearson Education Limited, 2016.

[12] S. Haykin, "Neural networks: A comprehensive foundation: MacMillan College," New York, 1994.

[13] K. B. Lee, S. Cheon, and C. O. Kim, "A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes," IEEE Transactions on Semiconductor Manufacturing, vol. 30, no. 2, pp. 135-142, 2017.

[14] K. G. Pasi and S. R. Naik, "Effect of parameter variations on accuracy of Convolutional Neural Network," in 2016 International Conference on Computing, Analytics and Security Trends (CAST), 2016, pp. 398-403: IEEE.

[15] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in Twenty-Second International Joint Conference on Artificial Intelligence, 2011.

[16] K. Isogawa, T. Ida, T. Shiodera, and T. Takeguchi, "Deep shrinkage convolutional neural network for adaptive

noise reduction," IEEE Signal Processing Letters, vol. 25, no. 2, pp. 224-228, 2018.

[17] C. C. Park, Y. Kim, and G. Kim, "Retrieval of sentence sequences for an image stream via coherence recurrent convolutional networks," IEEE transactions on pattern analysis and machine intelligence, vol. 40, no. 4, pp. 945-957, 2018.

[18] Y. Yin, J. Wu, and H. Zheng, "Ncfm: Accurate handwritten digits recognition using convolutional neural networks," in 2016 International Joint Conference on Neural Networks (IJCNN), 2016, pp. 525-531: IEEE.

[19] L. Xie, J. Wang, Z. Wei, M. Wang, and Q. Tian, "Disturblabel: Regularizing cnn on the loss layer," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4753-4762.

[20] A. Tavanaei and A. S. Maida, "Multi-layer unsupervised learning in a spiking convolutional neural network," in 2017 International Joint Conference on Neural Networks (IJCNN),2017, pp. 2023-2030: IEEE.

[21] J. Jin, K. Fu, and C. Zhang, "Traffic sign recognition with hinge loss trained convolutional neural networks," IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 5, pp. 1991-2000, 2014.

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

[23] Y. Liu and Q. Liu, "Convolutional neural networks with large margin softmax loss function for cognitive load recognition," in 2017 36th Chinese Control Conference (CCC), 2017, pp. 4045-4049: IEEE.

[24] Y. LeCun, "The MNIST database of handwritten digits ,"http://yann. lecun.com/exdb/mnist/, 1998.

[25] [25] M. A. Nielsen, Neural networks and deep learning. Determination press USA, 2015.

[26] B. Zhang and S. N. Srihari, "Fast k-nearest neighbor classification using cluster-based trees," IEEE Transactions on Pattern analysis and machine intelligence, vol. 26, no. 4, pp.525-528, 2004.

[27] E.Kussul and T. Baidyk, "Improved method of handwritten digit recognition tested on MNIST database," Image and Vision Computing, vol. 22, no. 12, pp. 971-981, 2004.

[28] R. B. Arif, M. A. B. Siddique, M. M. R. Khan, and M. R. Oishe, "Study and Observation of the Variations of Accuracies for Handwritten Digits Recognition with Various Hidden Layers and Epochs using Convolutional Neural Network," in 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT), 2018,pp. 112-117: IEEE.

[29] A. B. Siddique, M. M. R. Khan, R. B. Arif, and Z. Ashrafi, "Study and Observation of the Variations of Accuracies for Handwritten Digits Recognition with Various Hidden Layers and Epochs using Neural Network Algorithm," in 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT), 2018, pp. 118-123: IEEE.