International Conference on IoT Based Control Networks and Intelligent Systems
(ICICNIS 2020)

Pre-Print: will be available in *Elsevier-SSRN*

# A Comparative Study of Different Deep Learning Model for Recognition of Handwriting Digits

*Pronab Ghosh* [a], *Atqiya Abida Anjum* [b], *Asif Karim* [c], *Masum Shah Junayed* [a, d], *Md. Zahid Hasan* [a], *Khan Md. Hasib* [e], *Al Nahian Bin Emran* [a]

[a] Daffodil Iternational University, *102/1, Sukrabad, Dhaka 1207, Bangladesh*
[b] United International University, United City, Madani Ave, Dhaka 1212, Bangladesh
[c] *Member*, IEEE
[d] Bahcesehir University, Yıldız, Çırağan Cd., 34349 Beşiktaş/İstanbul, Turkey
[e] Ahsanullah University of Science & Technology, 141 & 142, Love Road, Dhaka 1208, Bangladesh

ABSTRACT

With the expansion of Artificial Neural Network (ANN), Deep Learning (DL) has brought interesting turn in the various fields of Artificial Intelligence (AI) by making it smarter and more efficient than what we had even in 10-2 years back. DL has been in use in various fields due to its versatility. Convolutional Neural Network (CNN) is at the major point of advancement that brings together the ANN and innovative DL techniques. In this research paper, we have contrived a multi-layer, fully connected neural network (NN) with 10 and 12 hidden layers for handwritten digits (HD) recognition. The testing is performed on the publicly attainable MNIST handwritten database. We selected 60,000 images from the MNIST database for training, and 10,000 images for testing. Our multi-layers ANN (10), ANN (12) and CNN are able to achieve an overall accuracy of 99.10%, 99. 34% and 99.70% respectively while determining digits using the MNIST handwriting dataset.

*Keywords: Handwriting Digit Recognition, CNN, ANN, Deep Learning*

## 1. Introduction

Machine Learning has increasingly been in use from Crowdsourcing platforms [27] [28] to Healthcare, Crime Detection [35], Defense, Smart Power Distribution [30], Biometrics [36], Transportation and Cybersecurity and Data Privacy [29] [33] to name a few. The latest focus on machine learning (ML) and computer vision has been the discovery of the CNN hierarchical function [1]. Since the value of GPUs for ML was identified in 2005, the CNN sector is immensely increased its visual classification performance in terms of efficiency [2] and has widely been in use in different demanding optical identification tasks in recent years. The structure of the CNN has close proximity to a simpler ANN. Usually the Hidden Layers are placed between input and output layers [3].

In ANN, there are multiple neurons in each layer. The weighted accumulation of all of the neurons in the previous layer is derived by each neuron in a layer, which concludes in a bias added value. Instead of the neuron being completely connected, every neuron has solely a small, locally associated area of the previous layer, known as a local receptive field as the neurons in a CNN layer are three dimensional which is opposite of ANN. Since CNN uses the structure of the space in learning and weights and distinctions in reception are exchanged, CNN works much faster than usual ANN. For training the network, a cost function is produced to compare the result of the network with the result we want. Then the signal dissipates over and over again to the device to change common weight and bias in all reception areas, thus reducing the cost-benefit and the network output [4-5]. The algorithm of backpropagation utilizes a stochastic descent gradient for reducing errors [6] [31]. Moreover, the network needs to have a huge number of cached layers in order to converge in some cases. Thus this was inspired by the observation on HD from the MNIST data collection of the effects of the veneered layers of a CNN.

In this experiment, we proposed a novel deep learning model which is based on Convolutional Neural Network (CNN) that is delivered to perceive manually written digits. Our presented paper shows an effective recognition of handwritten digits based on transition and diagonal properties using the most recognized CNN model. The characters' bitmap calculates the diagonal and transition characteristics of a character based on the point distribution of the image. In this system, the most efficient classifiers such as NN with 10 and 12 layers have been evaluated to show the comparison with our proposed CNN model. The main motive of this paper is to create a remarkable method for the recognition of handwritten digit strings. In order to accomplish the recognition task, first, the digit string is segmented into individual digits. Then, a digit recognition module is employed to

* *Corresponding Author: Asif Karim*
*Email: asif.karim@ieee.org*

classify each segmented digit completing the handwritten digit string recognition task. Our main contribution in this work is that CNN, which took the required number of input layers and Maxpooling2D, was the highest accuracy compared to other previous studies on the same dataset. Furthermore, CNN classifier performed slightly better than the suggested two NN layers (10 and 12).

The rest of this paper arranged as follows: Sect. 2 literature reviews the relevant works, Sect. 3 describes the various DL classifier, Sect. 4 describe research methodology of our proposed solution, Sect. 5 describes the result discussions of our experiment and lastly, Sect. 6 concludes the paper.

## 2. Literature Review

Several studies have been already done based on the MNIST handwritten dataset. Some of them are shown to make this research reliable. Kessab et al., [7] presented an idea about over the MNIST database considering extraction method. Dataset had been taken from the web, LeCun (1998). There have been 70,000 data selected for training and testing phases. The overall context is handled by Multi-layer perception neural networks. They divided each image considering five characteristics to get a superior result. At the end of the result, they successfully were able to gain 80% success rate over the dataset of MNIST. To detect their performance criteria, Chen. et al., [8] selected five ML classifiers such as Decision Tree, [32] NN, K-Nearest Neighbor, Bagging with Gradient Boost, and Random Forest. The experiment has been done considering the dataset of MNIST where each image contains 784 pixels. The method of preprocessing was completed by MATLAB. Afterward, the highest performance was observed through Neural Network which was about to 96.8%. The evaluation process has been addressed under the supervision of 4 known classifiers.

Dutt et al., [9] has shown multi-layer classifier of CNN based on Keras and Theano and evaluated by MNIST dataset. The highest reported result was gathered by multi-layer CNN, providing an accuracy of around 98.70% and the lowest was 96.89%, demonstrated by Random Forest classifier. In a spiking CNN model, Tavanaei et al. [10] implemented a multilayered CNN approach where they picked the MNIT dataset to resolve the opaque images and gained 98% accuracy overall and loss output ranges were 0.1 to 8.5%. A study of Rezoana et al. [11] was suggested multilayer CNN techniques based on handwritten digit recognition (DR) in which they evaluated the effect of CNN's hidden layer output by MNIST dataset. The plotted value of loss curves had been found against the number of epochs just under 0.1. An L-layered feed-forward neural network is observed by Siddique et al., [12] for MNIST handwritten digit recognition to show the variance of ANN accuracies for different epochs. Their best performance for 4 hidden layers was at 50 epochs have been calculated 97.32%. Jemimah et al. [13] built a CNN model to recognize the HD with miscellaneous hidden layers and epochs using TensorFlow, tested on MNIST database. The proposed model gives an accuracy of about 98% on recognizing hand written data with a very low error rate. Shiddineqy et al. [14] has implemented a system integrating a state of the art client server system with hand written digit recognition system. Server service utilizes TensorFlow as platform to predict digit recognition using ANN with backpropagation learning. The system is capable of classifying HD with the accuracy of 93.85% in the test data of MNIST.

## 3. Various Deep Learning Classifiers

### *3.1. Artificial Neural Network*

An artificial neuron network is a predictive model [15] based on the structure and functions human brain networks. Based on that input and output the structure of ANN changes as neural networks changes through learning and sensing ANN are a nonlinear statistical data modeling tool where the patterns are found through the complex relation of inputs and outputs. The equation for a single hidden layer is: [15]

$$Yn = B1 + LW * \tanh(B2 + IW *) * xn \tag{1}$$

Where xn and tanh are normalized to [-1, 1] where weights IW and B1, B2 are directly from the trained net.

### *3.2. Convolution Neural Network*

Convolution is used in CNNs rather than the general matrix multiplication in at least one of their layers. [16]

$$(\mathrm{f} * \mathrm{g})\ (\mathrm{t})\ \mathrm{def} \int_{\infty}^{\infty} f(\mathrm{T}) g(t - T)\ DT \tag{2}$$

A CNN comprises of an input and an output layer with multiple hidden layers that have been shown in Fig. 1.

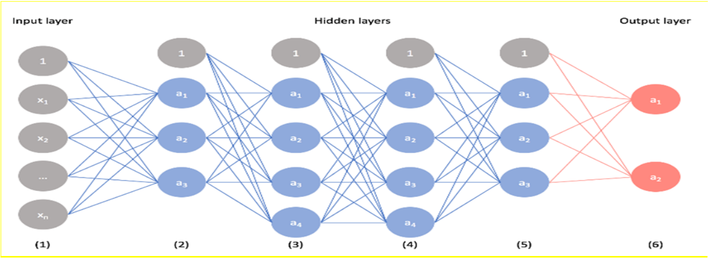

**Fig. 1 - the working flow of CNN model.**

## 4. Research Methodology

The diagram of our proposed model has been made in order to give an overview of our experiment. Dataset is gathered from online as well as preprocessed the data. To fix the size of data regarding our experiment, we have reshaped the overall data. Moreover, we also evaluated one hot encoding [17] method in our desired system because of getting more precise outcomes. The dataset of the image is partitioned into training and testing. Subsequently, 60,000 modified data has been trained followed by validation and the rest of the data are selected for testing. To evaluate our working scenarios of the proposed model, two DL methods, such as CNN and multi-layers ANN, such as 10 and 12, are addressed on the MNIST dataset. Having been completed model evaluation process, the highest result has been recorded through CNN model that was close to 99.70%. A descriptive analysis is illustrated below with the help of Fig. 2 process.

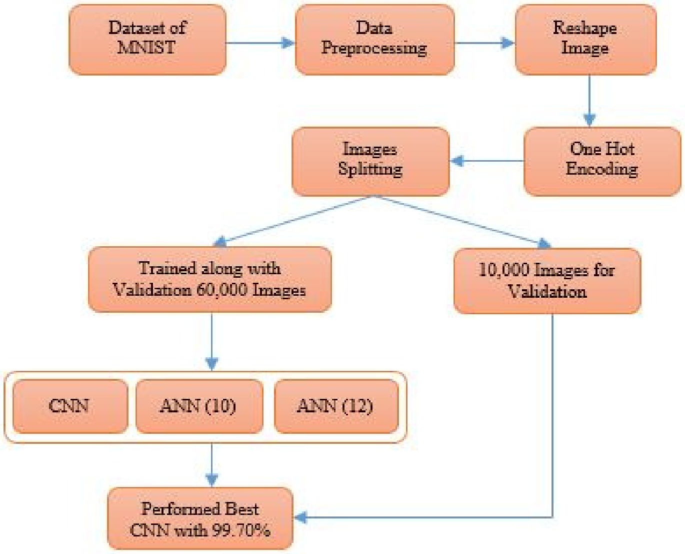
Dataset of MNIST
Data Preprocessing
Reshape Image
One Hot Encoding
Images Splitting
Trained along with Validation 60,000 Images
10,000 Images for Validation
CNN
ANN (10)
ANN (12)
Performed Best CNN with 99.70%

**Fig. 2 - the working flow of our proposed system.**

We have addressed TensorFlow [18] and Keras [19] to build this system. As a consequence, a brief description is added below in detail. TensorFlow [18] is an open-source platform that has been beneficial for numerical computation. It has been manufactured by scientists and specialists of Google. Furthermore, using flow graphs all types of numerical analysis along with mathematical expressions are evaluated. Edges represent multidimensional array data in these graphs, whereas nodes contain mathematical operations.

Keras [19], which was written in Python, is a high-level neural network application of programming interface. It is capable of working with TensorFlow, the Theano, or the CNTK. Various applications of neural building blocks exist for networks such as activation function, layers, optimization, and many other tools that act on ensuring a smooth operation with image and text information.

In order to contrive our own Convolution Neural Networks [20], we included a number of layers that includes of pooling layers, convolution layers, activation function, and fully connected layers.

- According to general consensus, Convolution is recognized as one of the most reasonable layers identifying highlights from meticulously detailed picture. Convolution Input spares relations using tiny squares between pixels to know the highlights of the images. Image Matrix as well as channels are accepted as contributions due to the scientific task.
- On the off chance of images being excessively expansive, segmentation of pooling layers can diminish the number of parameters. Impressive pooling is known as subsampling or down sampling, which decreases each guide's magnitude but preserves the vital data. There may be some types of spatial pooling, namely Average pooling, Max pooling, and Sum pooling. Max pooling always works with the highest element from the transformed component diagram. Taking the largest chunk can take up normal pooling as well. The "Whole Pooling" is the total of all elements in the element map.
- We supplement our matrix-vector with the fully connected (FC) model layer and feed it in a completely connected layer like a neural network [26].
- To make our output non-linear, we use an activation function. The expected result of the feature will be guided to pass through the activation function considering CNN situation [20]. It might be the activation function, such as ReLU (Rectified Linear Unit). Considering result is displayed in (3).

$$F(x) = \max(0, x) \tag{3}$$

### *4.1. Dataset Collection*

A large number of datasets are gathered to train and test various systems in the Modified National Institute of Standards and Technology (MNIST). This was generated from the two special databases of NIST, which includes binary images of handwritten numbers. Various digits of handwritten are taken from 250 individuals from two different groups, such as Census Bureau employees, and high school students [21]. This contains 10 individual classes, such as 0 to 9. To build this estimation, 60,000 training images are allocated for training with cross-validation, whereas other 10,000 images for testing [22].

### *4.2. Data Preprocessing*

To handle overfitting issues, this predictive model is addressed based on a huge number dataset. Since the collected dataset was taken from online, the size of all images was not in a proper format. Regarding this critical issue, we had to reshape our dataset into 28×28 for getting a better accuracy. Fig. 3 is given to show some HD of our working dataset. Moreover, another experiment was also conducted on experimented dataset namely, One hot encoding to convert all test values into categorical forms.

### *4.3. Descriptive Analysis on Proposed Model*

With the completion of preprocessing technique of our proposed model, we addressed only 2 convolution layers with some regularization methods, such as dropout, batch normalization, flatten layers, pooling layers, fully connected layer, and activation functions. The convolutional layer is thought of as fundamental building block of CNN [23, 18]. In order to create a function map, input data with filters or the kernel is major focus on which the convolution layer operates. Every layer consists of 32 and 64 in Fig. 3.

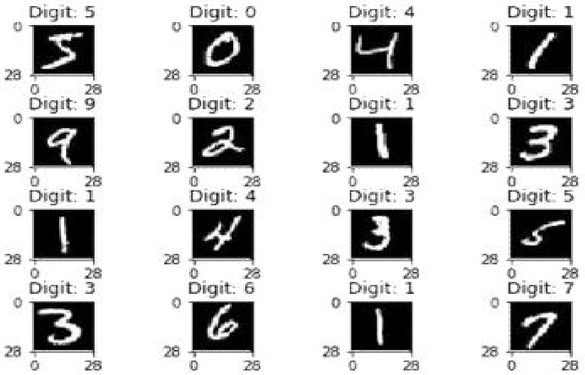

**Fig. 3 - Outputs of MNIST handwritten digit dataset.**

This function map is also referred to as 'Bottleneck Features' that have been generated by the pooling layers. We used RELU activation with the same padding function. The working size of pooling is 2x2. The value of stride, which indicates to input matrix for pixel shift numbers, is 2. Moreover, we can switch channels in two pixels without a second's delay after addressing stride's value. To solve the most common rising issues including overfitting for our model is solved by 0.25 and 0.5 dropouts' values. Later, for translating 3D images into 1D images, we have used flatten layers. The size of ReLU function is 128 has been included. To end with, another activation function is introduced such as 'softmax' as an output layer for this model. A clear overview has been added in the following Fig. 4.

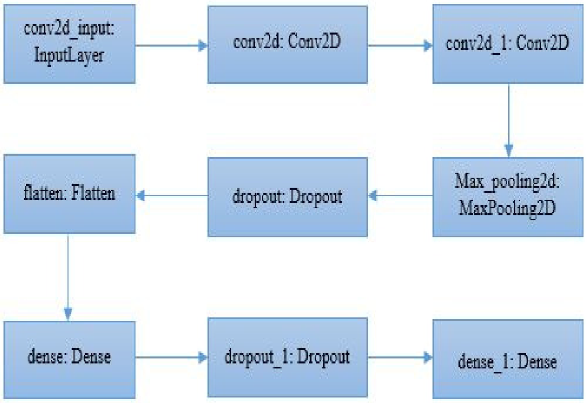

**Fig. 4 - the architecture of our experimental work.**

### *4.4. Trained the model*

In our proposed method, two DL algorithms, such as CNN and ANN have been evaluated due to their computations, facilities, and little memory requirements that have focused to computer vision. In the training part, we have considered 32 batch size and 10 epochs for both cases to fit our model. The closest output is almost received from each epoch. The proposed architecture is conducted based on MNIST dataset and found a very good result over the training, testing data as well. Training sample data has been investigated to provide a neutral evaluation over the test set model. We have divided over 85% for the training with validation and only 15% of the data has been selected for testing. Moreover, we have 70,000 images in total, whereas 60,000 images are used for preparation and 10,000 images for making prediction.

### *4.5. Evaluation matrix*

The number of False Negative (FN), True Positive (TP), True Negative (TN), and False Positive (FP) [24] [34] of two class problems is considered for a confusion matrix that represents a binary classification. For instance, a confusing matrix n = (n > 2) may occur because of numerous classes in excess of a 2-class problem. It generates n rows, n columns and everything in the confusion matrix is n = n entries. The amount of FP, TN, TP, and FN cannot be specifically tabulated from these frameworks. The different classes are identified according to this following technique:

$$\text{TPi} = \text{aii} \quad (4)$$

$$\text{FPi} = \sum_{j=1, j\neq i}^{n} a_{ji} \quad (5)$$

$$\text{FNi} = \sum_{j=1, j\neq i}^{n} a_{ij} \quad (6)$$

$$\text{TNi} = \sum_{j=1, j\neq i}^{n} \sum_{k=1, k\neq i}^{n} a_{jk} \quad (7)$$

To figure out the output of our model, we have evaluated F1-Score, Precision, Accuracy, and Recall. The overall calculation has been made with the help of following equations in detail.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (8)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (9)$$

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad (10)$$

$$\text{F1 –Score} = 2 * \frac{Recall*Precision}{Recall+Precision} \quad (11)$$

## 5. EXPERIMENTAL RESULTS AND DISCUSSION

### *5.1. Different Outcomes of ANN & CNN Based on Accuracy, Sensitivity, Precision and F1-Score*

Our dataset consists of 70000 data, examined based on 10 specific classes including 0 to 9 that have been clearly displayed in Table 1. It also hold the columns of precision, recall and f1-score of the ten classes which exposes the performance of ANN (10) layer on our Dataset. To highlight the comparison, we can see the highest precision 98% which belongs to Class 8, and 9. Along with the precision, 99% recall which is the best rate belongs to Class 1. On the other hand f1-score is 99% in class 8.

**Table 1 - Displayed outcomes of Precision, Recall, and F1-score on NN (10) layers.**

| Class | Precision | Recall | F1-Score |
|---|---|---|---|
| 0 | 97% | 98% | 98% |
| 1 | 98% | 99% | 99% |
| 2 | 97% | 97% | 97% |
| 3 | 95% | 97% | 96% |
| 4 | 96% | 98% | 97% |
| 5 | 97% | 96% | 96% |
| 6 | 98% | 97% | 98% |
| 7 | 98% | 97% | 97% |
| 8 | 99% | 98% | 99% |
| 9 | 99% | 98% | 98% |

Table 2 is illustrated the overall performance of ANN (12) layer on our dataset in terms of precision, recall and f1-score. To draw the clear distinction, we have charted the highest precision 99% which belongs to Class 4, 5, 8 and 9. Along with the precision column, 99% recall which is the highest rate belongs to Class 1, 5, 8, and 9. On the other hand f1-score is 99% in all of the classes except class 1, and 8.

**Table 2 - Displayed outcomes of Precision, Recall, and F1-score on NN (12) layers.**

| Class | Precision | Recall | F1-Score |
|---|---|---|---|
| 0 | 98% | 98% | 98% |
| 1 | 98% | 99% | 99% |
| 2 | 97% | 98% | 97% |
| 3 | 96% | 98% | 98% |
| 4 | 99% | 98% | 97% |
| 5 | 99% | 99% | 98% |
| 6 | 98% | 97% | 98% |
| 7 | 98% | 97% | 97% |
| 8 | 99% | 99% | 99% |
| 9 | 99% | 99% | 98% |

Table 3 is displayed the overall performance of CNN on our dataset in terms of precision, recall and f1-score. To draw the clear distinction, we have charted the highest precision 100% which belongs to Class 1, 4, and 6. Along with the precision column, 99% recall which is the highest rate belongs to Class 1,2,3,5 and 8. On the other hand f1-score is 99% in all of the classes except class 5, 8 and 9.

**Table 3 - Displayed outcomes of Precision, Recall, and F1-score on CNN.**

| Class | Precision | Recall | F1-Score |
|---|---|---|---|
| 0 | 99% | 100% | 99% |
| 1 | 100% | 99% | 99% |
| 2 | 98% | 99% | 99% |
| 3 | 98% | 99% | 99% |
| 4 | 100% | 98% | 99% |
| 5 | 98% | 99% | 98% |
| 6 | 100% | 98% | 99% |
| 7 | 99% | 98% | 99% |
| 8 | 97% | 99% | 98% |
| 9 | 99% | 98% | 98% |

### *5.2. Recorded Findings of ANN (10 and 12 layers) & CNN Based on Various Number of Epochs*

ANN layers (10) model scores 99.10% accuracy on our working dataset. After 10 epochs, ANN has gained the highest training accuracy of 99.79%, on the other hand, and epoch 1 scores only 93.92% on our dataset. After analyzing the different epoch values from Table 4, we can state that our proposed model exhibits a good performance on our dataset in training and also in testing.

**Table 4 - Estimated outcomes of different epochs for NN (10) Layers.**

| Epochs | Training accuracy | Testing accuracy |
|---|---|---|
| 1 | 93.92% | 91.75% |
| 2 | 94.04% | 92.68% |
| 3 | 95.29% | 94.36% |
| 4 | 95.69% | 95.01% |
| 5 | 96.57% | 95.42% |
| 6 | 97.73% | 96.77% |
| 7 | 98.35% | 97% |
| 8 | 98.77% | 97.73% |
| 9 | 99.09% | 98.47% |
| 10 | 99.79% | 99.10% |

ANN layers (12) model scores 99.4% accuracy on our working dataset. After 10 epochs, ANN has gained the highest training accuracy of 99.94%, on the other hand, and epochs from 1 to 5 score just over 97% on our dataset. With every increasing epoch, accuracy has increased and on the final epoch, CNN has achieved the training accuracy of 99.30% and the validation accuracy at 99.57% at a small loss on our dataset. After the illustration of the 10 epochs from Table 5, we can say that our CNN has performed tremendously well with excellent accuracy and minimalistic loss on our dataset.

**Table 5 - Predicted outcomes of different epochs for NN (12) layers.**

| Epoch | Training loss | Training accuracy | Validation loss | Validation accuracy |
|---|---|---|---|---|
| 1 | 40.75% | 90.75% | 7.29% | 97.92% |
| 2 | 15.17% | 95.68% | 6.31% | 98.04% |
| 3 | 12.13% | 96.36% | 4.57% | 98.51% |
| 4 | 9.80% | 97.01% | 5.23% | 98.29% |
| 5 | 8.81% | 97.42% | 4.64% | 98.57% |
| 6 | 7.58% | 97.77% | 4.02% | 98.73% |
| 7 | 6.90% | 97.94% | 3.52% | 98.95% |
| 8 | 6.81% | 98.03% | 3.64% | 98.77% |
| 9 | 5.71% | 98.68% | 3.16% | 99.34% |
| 10 | 5.13% | 99.30% | 2.5% | 99.57% |

Fig. 5 shows the training accuracy and the validation accuracy graph of CNN. Here the green line determines the performance of the training accuracy and the blue line represents the validation accuracy; while X-axis displays each epoch while Y-axis displays the increasing accuracy.

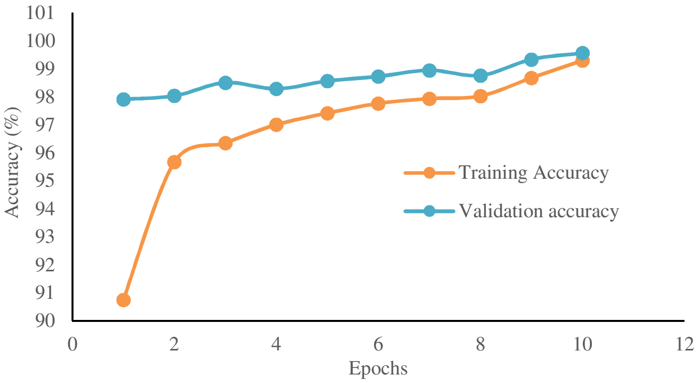

**Fig. 5 - The detailed graph of the training accuracy versus the validation accuracy of CNN.**

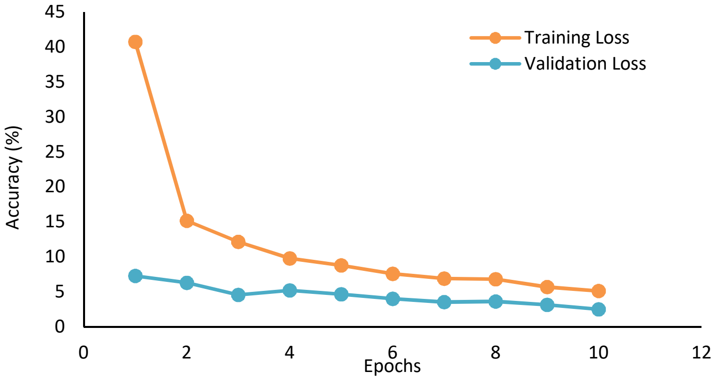

**Fig. 6 - The graph of the training loss vs the validation loss of CNN.**

Fig. 6 shows the training loss and the validation loss [25] graph of CNN. According to graph, the performance of the training loss depicted by the green line and determines the validation loss is represented by the blue line; while X-axis displays each epoch while Y-axis displays the loss against each epoch.

### *5.3. Compared to others related works*

It is clearly seemed in Table 6 that the previous works of CNN was comparatively lower than our proposed CNN model based on pixel. In addition, Dutt et al. and Tavanaei et al. experimented results was 98.7% and 98% respectively, while our proposed system accuracy is achieved 99.70%. Another model ANN has been examined by Chen et al. and Siddique et al. The research findings of our ANN layers 10 and 12 achieve 99.10 % and 99.34% which have outperformed Chen et al. (96.8%) and Siddique et al (97.23%).

**Table 6 - Comparison between our studies and previous studies.**

| References | Approach | Database | Feature | Accuracy |
|---|---|---|---|---|
| Chen et al. | ANN | MNIST | Pixel Based | 96.8% |
| Dutt et al. | CNN | MNIST | Pixel Based | 98.7% |
| Tavanaei et al. | CNN | MNIST | Pixel Based | 98% |
| Siddique et al. | ANN | MNIST | Pixel Based | 97.23% |
| Proposed Model | CNN | MNIST | Pixel Based | 99.70% |
| Proposed Model | ANN (10) | MNIST | Pixel Based | 99.10% |
| Proposed Model | ANN (12) | MNIST | Pixel Based | 99.34% |

## 6. Conclusions

In this paper, we proposed CNN and ANN for handwritten digit recognition that involves automatic feature generation and output prediction. The model coalesces the advantage of CNN and ANN in recognizing HD. This model also brings the focus on the use of automatically generated features over the hand-designed features. The experimental results pointed out that our proposed approach achieved the classification accuracy of 99.70% for the MNIST dataset. We have also drawn the contrast between these two models. In the future, the proposed model can be improved for the recognition of handwritten characters in different languages such as French, English, Hindi, Bengali, etc. Some optimizing techniques can also be designed to boost the classification performance.

## References

[1] LeCun. (2012). Learning invariant feature hierarchies. in European Conference on Computer Vision. Springer (pp. 496–505).

[2] Steinkraus, D. et al. (2005). Using GPUs for machine learning algorithms. Proceedings. Eighth International Conference on Document Analysis and Recognition (ICDAR'05) (pp. 1115-1120), 2005.

[3] Schalkoff, R. J. (1997). Artificial neural networks. McGraw-Hill New York, 1.

[4] LeCun, Y. (2016). LeNet-5, convolutional neural networks (2015), 1.

[5] Russell, S. J. & Norvig, P (2016). Artificial intelligence: a modern approach: Malaysia; Pearson Education Limited.

[6] Amari, S. I. (1993). Backpropagation and stochastic gradient descent method," Neurocomputing (pp. 185-196), 5.

[7] Kessab, B. E. E., Douai, C., Bouikhalene, B., Fakir, M., & Moro. K. (2013). Extraction Method of Handwritten Digit Recognition Tested on the MNIST Database. International Journal of Advanced Science and Technology, 50.

[8] Chen, S., Almamlook, R., Gu, Y., & Wells, L. (2018). Offline Handwritten Digits Recognition Using Machine learning. Proceedings of the International Conference on Industrial Engineering and Operations Management Washington DC, USA.

[9] Dutt, A., & Dutt, A. (2017). Handwritten Digit Recognition Using Deep Learning. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 6.

[10] Tavanaei, A., & Maida, A. S. (2017). Multi-layer unsupervised learning in a spiking convolutional neural network. 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK (pp. 2023-2030).

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

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

[13] Jemimah. K. (2019). Recognition of Handwritten Characters based on Deep Learning with TensorFlow. International Research Journal of Engineering and Technology (IRJET) (pp. 1164—1165).

[14] Shiddieqy, H. A., Adiono. T., & Syafalni, I. (2019). Mobile Client-Server Approach for Handwriting Digit Recognition. 2019 International Symposium on Electronics and Smart Devices (ISESD) (pp. 1-4), Badung-Bali, Indonesia.

[15] Karim, A., Azam, S., Shanmugam, B., Kannoorpatti, K., & Alazab, M. (2019). A Comprehensive Survey for Intelligent Spam Email Detection. IEEE Access (pp. 168261–168295), 7.

[16] A brief explanation of CNN. [online]. Available at: https://medium.com/@jamesdacombe/an-introduction-to-artificial-neural-networks-with-example-ad459bb6941b, (Accessed on: 31-07-2020).

[17] One hot encoding. [online]. Available at: https://medium.com/@oraheem/one-hot-encoding-in-machine-learning-b2d344284d9e, (Accessed on: 31-07-2020).

[18] Junayed, M. S., Jeny, A. A., Atik, S. T., Neehal, N., Karim, A., Azam, S., Shanmugam, B. (2019). AcneNet - A Deep CNN Based Classification Approach for Acne Classes. 2019 12th International Conference on Information & Communication Technology and System (ICTS) (pp. 203-208), Surabaya, Indonesia.

[19] Vidnerova, P., Neruda, R. (2017). Evolving Keras architectures for sensor data analysis. Federated Conference on Computer Science and Information Systems (FedCSIS).

[20] Krizhevsky, A., Sutskever, I., Hinton, G. (2017). ImageNet Classification with Deep Convolutional Neural Networks. Communications of the ACM CACM Homepage archive (pp. 84-90), 60.

[21] Zhang, B., & Srihari, S. N. (2004). Fast k-nearest neighbor classification using cluster-based trees. IEEE Transactions on Pattern analysis and machine intelligence (pp. 525-528), 26.

[22] Kussul. E., & Baidyk, T. (2004). Improved method of handwritten digit recognition tested on MNIST database. Image and Vision Computing (pp. 971-981), 22.

[23] Jeny, A. A., Junayed, M. S., Ahmed, I., Habib, T., Rahman, R. (2019). FoNet - Local Food Recognition Using Deep Residual Neural Networks. 2019 International Conference on Information Technology (ICIT) (pp. 184-189), Bhubaneswar, India.

[24] Shamrat, F. M. J. M., Ghosh, P., Sadek, M. H., Kazi, M. A., & Shultana, S. (2020). Implementation of Machine Learning Algorithms to Detect the Prognosis Rate of Kidney Disease. 2020 IEEE International Conference for Innovation in Technology.

[25] Shamrat, F. M. J. M., Asaduzzaman, M., Ghosh, P., Sultan, D., & Tasnim, T. (2020). A Web Based Application for Agriculture: “Smart Farming System". International Journal of Emerging Trends in Engineering Research.

[26] Hasan, M. A., Noor N. F. M., Rahman, A. B. S. S., Rahman, M. M. (2020). The Transition From Intelligent to Affective Tutoring System: A Review and Open Issues. in IEEE Access (pp. 204612-204638), 8.

[27] Abdullah, N. A. et. al., (2020). A Framework for Optimal Worker Selection in Spatial Crowdsourcing Using Bayesian Network. in IEEE Access (pp. 120218-120233)., 8.

[28] Ghosh, P., et al. (2020). Expert Model of Cancer Disease Using Supervised Algorithms with a LASSO Feature Selection Approach. International Journal of Electrical and Computer Engineering, 11.

[29] Zobaed, S. M., Ahmed, S., Gottumukkala, R., Salehi, M. A. (2019). ClustCrypt: Privacy-Preserving Clustering of Unstructured Big Data in the Cloud. 2019 IEEE 21st International Conference on High Performance Computing and Communications. IEEE 17th International Conference on Smart City. IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS) (pp. 609-616), Zhangjiajie, China.

[30] Siddique, A. H., Al Bloushi, H. K., & Lamont, L. A. (2012). Distributed Generation and smart power grid UAE vision for 2030. 2012 3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe) (pp. 1-6), Berlin.

[31] Ghosh, P., et al. (2020). Optization of Prediction Method of Chronic Kidney Disease with Machine Learning Algorithms. The 15th International Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP 2020) and International Conference on Artificial Intelligence & Internet of things (AIoT 2020).

[32] Shamrat, F. M. J. M., Tasnim, Z., Ghosh, P., Majumder, A., & Hasan, M. Z. (2020). Personalization of Job Circular Advertisement to Candidates Using Decision Tree Classification Algorithm. 2020 IEEE International Conference for Innovation in Technology.

[33] Ghosh, P., Hasan, M. Z., Atik, S. T., & Jabiullah, M. I. (2019). A Variable Length Key Based Cryptographic Approach on Cloud Data. 2019 International Conference on Information Technology (ICIT) (pp. 285-290), Bhubaneswar, India.

[34] Ghosh, P., Hasan, M. Z., & Jabiullah, M. I. (2018). A Comparative Study of Machine Learning Approaches on Dataset to Predicting Cancer Outcome. Journal of the Bangladesh Electronic Society, 18.

[35] Manlangit S., Azam S., Shanmugam B., Kannoorpatti K., Jonkman M., Balasubramaniam A. (2018) An Efficient Method for Detecting Fraudulent Transactions Using Classification Algorithms on an Anonymized Credit Card Data Set. In: Abraham A., Muhuri P., Muda A., Gandhi N. (eds) Intelligent Systems Design and Applications. ISDA 2017. Advances in Intelligent Systems and Computing, vol 736. Springer, Cham. https://doi.org/10.1007/978-3-319-76348-4_41.

[36] Azam S, Gavrilova M (2017) Person"s discriminating visual features for recognising gender: LASSO regression model and feature analysis. International Journal of Biometrics 9:305. doi: 10.1504/ijbm.2017.10009342.