An Efficient CNN Model for Automated Digital Handwritten Digit Classification

Article *in* Journal of Information Systems Engineering and Business Intelligence · April 2021

DOI: 10.20473/jisebi.7.1.42-55

Angona Biswas
Chittagong University of Engineering & Technology
24 PUBLICATIONS **126** CITATIONS
SEE PROFILE

Md. Saiful Islam
260 PUBLICATIONS **3,584** CITATIONS
SEE PROFILE

Journal of
Information Systems Engineering
and Business Intelligence

Vol.7, No.1, April 2021
Available online at: http://e-journal.unair.ac.id/index.php/JISEBI

An Efficient CNN Model for Automated Digital Handwritten Digit Classification

Angona Biswas[1)] [*], **Md. Saiful Islam**[2)]

[1)2)]Department of Electronics and Telecommunication Engineering, Chittagong University of Engineering & Technology, Bangladesh

Kaptai, Highway Raozan Pahartali Rd, Chittagong 4349, Chittagong

[1)]angonabiswas28@gmail.com, [2)] saiful05eee@gmail.com

Abstract

Background: Handwriting recognition becomes an appreciable research area because of its important practical applications, but varieties of writing patterns make automatic classification a challenging task. Classifying handwritten digits with a higher accuracy is needed to improve the limitations from past research, which mostly used deep learning approaches.
Objective: Two most noteworthy limitations are low accuracy and slow computational speed. The current study is to model a Convolutional Neural Network (CNN), which is simple yet more accurate in classifying English handwritten digits for different datasets. Novelty of this paper is to explore an efficient CNN architecture that can classify digits of different datasets accurately.
Methods: The author proposed five different CNN architectures for training and validation tasks with two datasets. Dataset-1 consists of 12,000 MNIST data and Dataset-2 consists of 29,400-digit data of Kaggle. The proposed CNN models extract the features first and then performs the classification tasks. For the performance optimization, the models utilized stochastic gradient descent with momentum optimizer.
Results: Among the five models, one was found to be the best performer, with 99.53% and 98.93% of validation accuracy for Dataset-1 and Dataset-2 respectively. Compared to Adam and RMSProp optimizers, stochastic gradient descent with momentum yielded the highest accuracy.
Conclusion: The proposed best CNN model has the simplest architecture. It provides a higher accuracy for different datasets and takes less computational time. The validation accuracy of the proposed model is also higher than those of in past works.

Keywords: Convolutional Layer, Convolutional Neural Network, Handwritten digit classification, MNIST dataset, SGD optimizer

Article history: Received 2 January 2021, first decision 9 February 2021, accepted 8 March 2021, available online 28 April 2021

I. INTRODUCTION

Handwriting plays an important role in everyday life, not only as a medium of communication, but also for legal documentation. Each person has inherent writing patterns. This diversity often makes it difficult to recognize and read [1]. Different computer-based recognition systems are thus applied to identify the correct handwritten information. Several deep learning algorithms were already applied in the handwriting recognition research area. They provide better performance than conventional ways and take less time to recognize, especially when a large amount of dataset is involved. Some significant, practical applications of this research include signature verification of bank check, vehicle plate detection, digit classification from images, information extraction from historical documents [2] and so on. This recognition process uses handwritten images as a dataset. Firstly, features from the image are extracted and then classification is performed. Characters, numbers, cursive texts are mainly used in this recognition process [3]. Automatic classification technique is challenging to be applied in this area due to the high variety of individual writing patterns.

Recognition or classification process is done from text images and, thus, is known as optical character recognition [4]. This plays a crucial role in many applications, especially for commercial purposes. Various datasets are used for recognition in this research paradigm [4]. Different authors have used different methods to classify various digits and characters. Authors in [5] used convolutional neural networks to recognize Kannada, a south Indian character. 497 classes were included in the experiment dataset. AlexNet was used for training and the accuracy obtained was 92% from handwritten text. Another CNN model was used for Devanagari numerical recognition by authors in [6] and their

[*] Corresponding author

ISSN 2443-2555 (online) 2598-6333 (print) © 2021 The Authors. Published by Universitas Airlangga.

doi: http://dx.doi.org/10.20473/jisebi.7.1.42-55

proposed model combined Genetic Algorithm (GA) with CNN. Data pre-processing was used, and so was feature extraction before training. The obtained accuracy was 96.06%. Authors in [7] used GA and support vector machines for the classification of Bangla numeric digits. Features were extracted using Genetic Algorithms and the dataset was trained by support vector machine (SVM) classifier. The proposed model yielded 97.70% accuracy using a total of 6,000 data. A combination of Restricted Boltzmann Machine (RBM) and CNN was implemented by the author in [8] for handwriting recognition of Arabic digit. Two phases were proposed: first, meaningful features were extracted from the dataset and then Arabic digits were classified by CNN classifier. The accuracy reached 98.59%.

Another DNN based Bangla handwritten digit classification work had been done by authors in [9]. They worked with more than 85,000 image data and emphasized on the pre-processing steps. Five pre-processing steps were used before CNN training. Their proposed model consisted of six convolutional layers and two dense layers. The validation accuracy obtained was 98.57%. Another magnificent work was done by authors in [10] as they worked with five different types of language data to test the accuracy of their model. Raw data were firstly normalised and then five CNN layers were used, which was similar to the pre-trained DNN LeNet-5. They found that the experimental accuracy was 98.38% for Bangla digits, 97.2% for Oriya digits, 98.54% for Devanagari digits, 96.5% for Telugu digits and 99.1% for English digits. Authors in [11] presented a Korean Hangul recognition using deep CNN. Their architecture was composed of three main layers: convolutional layer, max-pooling layer and classification layer and they used two datasets. They observed the accuracy of 95.96% for SERI95a dataset and 92.92% for PE92 dataset. In this way, authors of different papers found different numerical classification accuracy by implementing various methods. In the case of English digit recognition, some false predictions seem to have lowered the accuracy.

The novelty of this work is to evaluate an efficient model for English handwritten digit classification by improving the classification accuracy and minimizing the computational time of the computer vision recognition system. A shallow five CNN architectures are proposed to obtain the best model. The ideal model is the simple one but with fast computation. Traditional classification methods undergo pre-processing [12] and image segmentation time before the classification process. Again, pre-trained networks of transfer learning contain a sizable number of layers and thus take more time to compute vast layers [13]. There is a trade-off between computational time and accuracy [14], but simplex CNN models are considered capable in classifying more accurately with less consuming time.

The remaining parts of this paper are as follows. Section II presents concise discussion of related work and background study of CNN. Section III describes the proposed methodology. Section IV presents the obtained results. Section V presents the analysis and the comparison. Finally, Section VI concludes the overall work.

II. LITERATURE REVIEW

In the present research area, Deep Neural Network (DNN) is a well-recognized application for image recognition, object classification, speech recognition and others [8]. It has the ability to provide higher classification accuracy. Authors in [15] proposed a CNN-based model for handwritten digit recognition from MNIST dataset and their model consisted of eight layers. They claimed that their proposed architecture is able to provide an improved accuracy of 98.85% within 8569 seconds. Another CNN model was presented for MNIST dataset in the same manner by authors in [16] in which CNN models consisted of seven layers. This model included one input and one output layer with five hidden layers in the middle. The best accuracy was evaluated by varying the number of hidden layers from the model and number of epochs and finally the best-obtained accuracy was 99.21% within 15 numbers of epochs.

A minimal model for handwritten digit recognition based on CNN was presented by authors in [17]. They maintained that their minimal model reduced the mathematical computation for recognition from MNIST dataset. Their model is similar to LetNet5 and obtained 99.5% accuracy. A CNN model of simplified version was proposed for MNIST database by authors in [18] and they compared their simplified model with LeNet-1, LeNet-5. They showed that their model was able to reduce the error rate by 0.7% which is lower than other networks. Three different approaches: DNN, DBN, CNN were proposed by authors in [1] for digit recognition. Before neural network training, pre-processing, segmentation, and feature extraction were performed. They found the best accuracy of 98.08% from the DNN approach.

Seven layers CNN model was proposed by authors in [19] for MNIST dataset. They found the 95.7% test accuracy within 500 epochs from their proposed CNN model. The combined architecture of CNN along with Deeplearning4j (DL4J) was presented for MNIST dataset recognition by authors in [4]. They avoided any kind of pre-processing steps and obtained 99.21% accuracy from their model. Another recognition system with Multilayer Perceptron (MLP) was proposed for MNIST dataset by authors in [20]. For training purposes, they used back-propagation and for validation purposes, they used feed forward networks. They evaluated the best result by varying the number of iterations. They used 5,000 data and obtained 99.32% accuracy within 250 iterations. For automatic handwritten digit detection from MNIST dataset, authors in [21] proposed a four layer CNN model. This simplified model was trained for different epochs and finally 98% accuracy was obtained. Another effective classifier is Support Vector Machine (SVM) which was used for MNIST dataset recognition [22] [23]. A composition of CNN and SVM system was proposed [20] for MNIST dataset recognition. Features from the digit images were extracted using CNN; and SVM was utilized for image classification at the output layer. Their proposed method applied a pre-processing step before CNN application and achieved accuracy of 99.28%.

Past deep learning research shows that it is an excellent performance provider for image classification, as well as for handwriting classification [9]. CNN is a part of deep learning technique. Basic CNN structure consists of one input layer, manifold hidden layers and one output layer. Image classification using CNN includes two main stages: extraction of image features and classification [17]. CNN building blocks accommodate some important layers, and the descriptions are given as follows.

A. Input Layer

This layer is the first layer of the CNN blocks. Input layer looks through the input images for the next blocks of CNN [17]. It represents the image data as a three-dimensional matrix and defines the input layer neurons from this information.

B. Convolution Layer

In this layer, filters are applied to extract the features from images [24] [25]. Various sizes and numbers of filters can be applied in this layer as necessary. These filters are also known as convolution kernels. Convolution operation performs the between input images and filters [17] [25]. It creates a feature map. The mathematical operation [16] is seen on (1).

$$Conv(m,n) = I(x,y) \otimes F(x,y) \tag{1}$$

where, $\otimes$ = convolution operation, $I(x,y)$= expressing input image matrix, $F(x,y)$ = filter or kernel function.

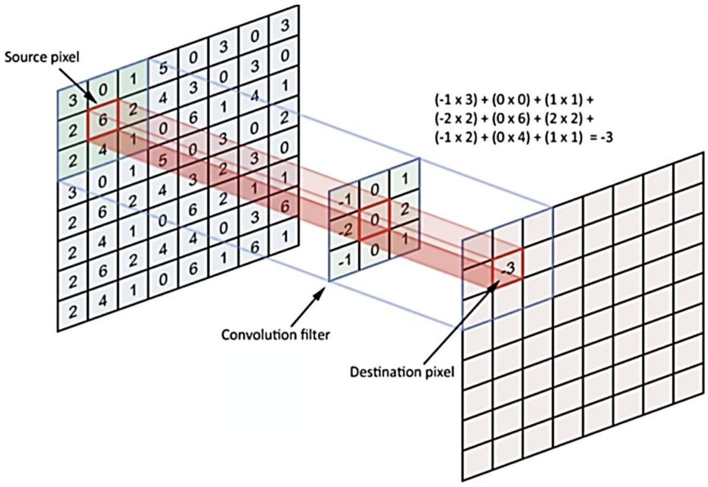

Source pixel
(-1 x 3) + (0 x 0) + (1 x 1) +
(-2 x 2) + (0 x 6) + (2 x 2) +
(-1 x 2) + (0 x 4) + (1 x 1) = -3
Convolution filter
Destination pixel

Fig. 1 Visualization of convolution layer [26]

Fig. 1 demonstrates how the filter or kernel moves across the input image and performs the convolution operation in this layer. This operation produces an integer value, and this process is continued for the full image. This result is passed to the next layer.

C. Batch Normalization Layer

In order to adjust, scale of the input layer, a purpose batch normalization layer has been used. This layer speeds up the training process. It helps the network to gain stability. It is relatively similar to the dropout layer, which is applied before the activation layer.

D. ReLU Activation Layer

Different types of activation functions are employed in deep learning. Sigmoid activation function is responsible for overlooking the image information and for that, most of the network uses the ReLU layer. ReLU or Rectified

Linear Unit is a non-linear activation function that is simple to use and performs faster. The mathematical expression [19] is seen on (2). Equation (2) is representing the ReLU function, which is represented graphically in Fig. 2.

$$y = f(x) = \max(0, \lambda x) \qquad (2)$$

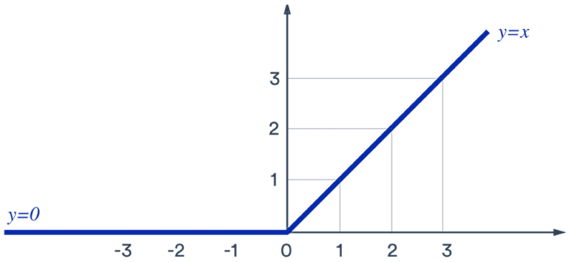


Fig. 2 Graphical representation of ReLU function which is expressed in equation [27]

E. Pooling Layer

Pooling layer is the process of shrinking data volume of the previous network layer [15] and accelerated the network computation. It is employed within two convoluted layers to reduce the dimensional [28]. Generally, two types of pooling are added to the network: max-pooling and average pooling [15] [28]. The max-pooling layer takes the highest value from convoluted output and diminishes the size. This process is also known as down-sampling [29]. It can be shown from the following Fig. 3. In Fig. 3, the filter size, and the stride size both are kept the same 2×2, taking the highest value from the sub-region [29].

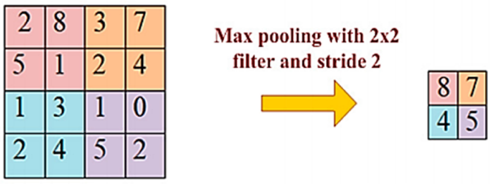


Fig. 3 Visualization of max-pooling layer [28]

F. Softmax Layer or Activation Function

Softmax is a logical approach which can be used for a multi-class classifier [17]. This function is mainly used in the final layers for classification and provides the probabilistic output for the final layer. The mathematical expression [30] is seen on (3).

$$\sigma(z)_i = \frac{e^{zi}}{\sum_{j=1}^{K} e^{zj}} \qquad (3)$$

where, z = input vector, zi = elements of input vector, e^{zi} = standard exponential, K= number of classes.

G. Fully Connected Layers

This layer is the arrangement of a normal neural network or feed-forward network [28]. In this layer, each node is directly connected to the past and next layers [29] [31]. A fully connected layer leans the feature from its previous layer. One or more fully connected layers can exist. The last FC layer is commonly known as the output layer, which predicts the desired classes. This part is included in the CNN classification that works after the feature extraction.

The proposed CNN architecture is slightly different and novel from earlier approaches for the handwritten English digit recognition with different datasets. The proposed CNN model is better than others and simpler, as well as less time-consuming. To manifest the effectiveness, two datasets were used. The improved accuracy shows that the automated CNN classifier of the proposed method is an effective approach.

III. Methods

The proposed model uses CNN for classification of handwritten digit recognition. Five architectures are presented to explore the best performance. In this section of data specification, architecture details and model parameters are described in more details.

A. Data Details

For this handwritten classification experiment, MNIST dataset [32] (Dataset-1) and a handwritten dataset from 'Kaggle' [33] (Dataset-2) were used. The datasets consist of 10 English digits (0-9) of different writers. From these datasets, 12,000 image data of MNIST and 29,400 image data of Dataset-2 were used for the experiment. The digit images of MNIST dataset are at the size of 28 × 28 pixels and the digit images of Dataset-2 are at the size of 60× 60 pixels. A portion of the experimental datasets are shown below in Fig. 4.

(a) (b)

Fig. 4 Exhibition of a part of experimental (a) MNIST dataset (b) Handwritten digit dataset or Dataset-2

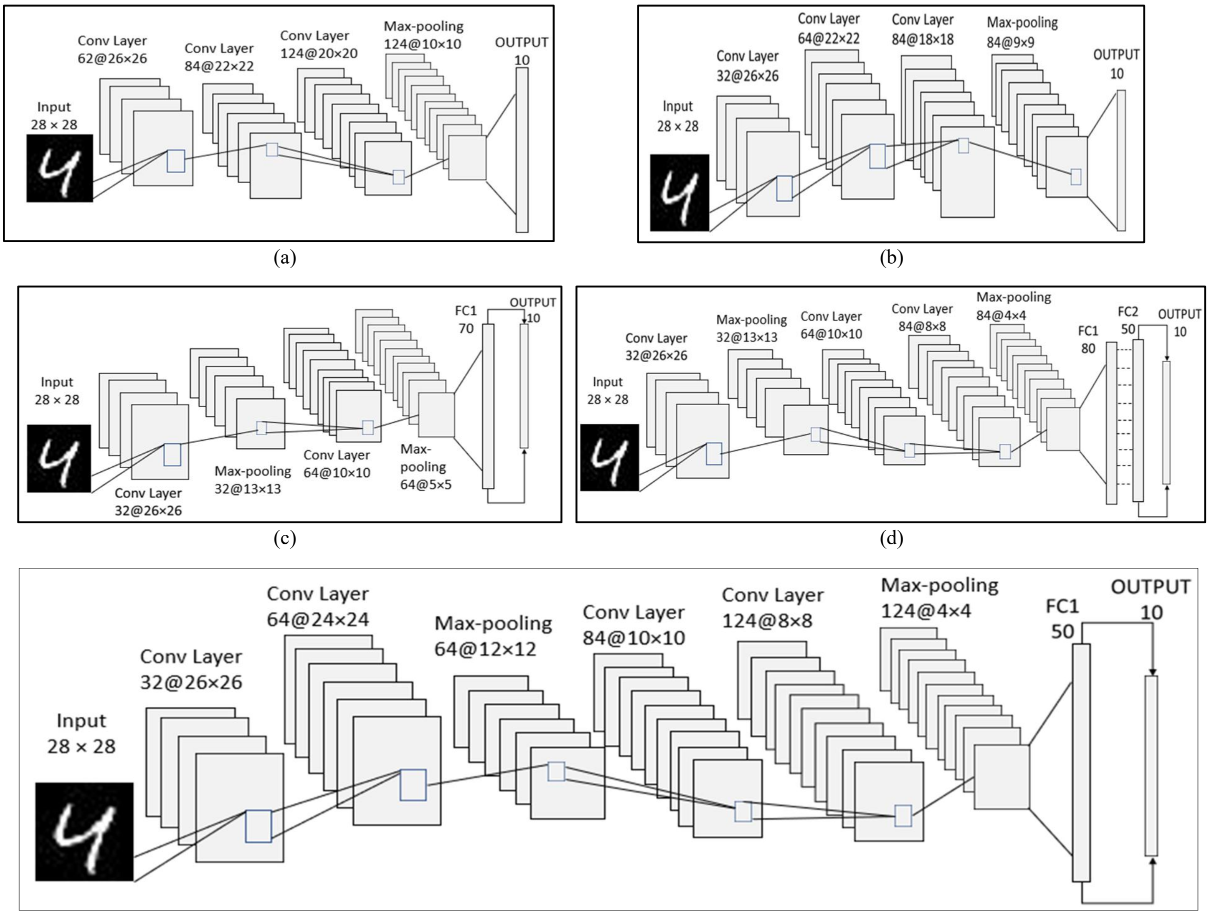
TABLE 1
Data Arrangement of Both Datasets For Training and Validation for CNN architectures

Image of English Digits (Dataset-1)	Number of data per digit (Dataset-1)	Taking Data for training (Dataset-1)	Taking Data for Validation (Dataset-1)	Image of English Digits (Dataset-2)	Number of data per digit (Dataset-2)	Taking Data For training (Dataset-2)	Taking Data for Validation (Dataset-2)
0	1200	900	300	0	2940	2352	588
1	1200	900	300	1	2940	2352	588
2	1200	900	300	2	2940	2352	588
3	1200	900	300	3	2940	2352	588
4	1200	900	300	4	2940	2352	588
5	1200	900	300	5	2940	2352	588
6	1200	900	300	6	2940	2352	588
7	1200	900	300	7	2940	2352	588
8	1200	900	300	8	2940	2352	588
9	1200	900	300	9	2940	2352	588
	Total Used Data = 12,000	Total training data = 9,000	Total validation data = 3,000		Total Used Data = 29400	Total training data = 23520	Total validation data = 5880

Table 1 shows how training and validation data are split for the CNN architectures. From the total 12,000 MNIST data, 9,000 data were used for training of the network and 3,000 data were used for validation. Again, from total used of 29,400 data from Dataset-2, 23,520 data were used for training of the network and 5,880 data were used for validation.

B. Proposed Five CNN Architectures

Five different CNN architectures are proposed for handwritten digit classification to explore the best accuracy which is shown in Fig. 5. These proposed models are simpler than the pre-trained deep network and less time-consuming. Pre-trained network architecture refers to the saved model of a network, which is already trained with an immense amount of image data. It requires fine-tuning before being used. Pre-trained networks are built-in models and users do not need to establish network models for their own problem. GoogleNet, AlexNet, VGG16, LetNet 5, ResNet 50 are the most popular pre-trained networks. At first, MNIST data were inputted at the networks for experiment. Then, the experiment was done for the Kaggle digit dataset. SGD is a superior optimizer, which shows faster performance. Network's parameters are kept the same for all CNN architectures, which are presented in Table 2.



(a) (b) (c) (d) (e)

Fig. 5 The proposed five CNN architectures for handwritten digit classification: (a) Architecture 1 (b) Architecture 2 (c) Architecture 3 (d) Architecture 4 (e) Architecture 5

Here, Fig. 5(a) is architecture 1, where the network consists of three consecutive convolutional layers, one pooling layer and finally output layer. Fig. 5(b) has the same structure as Fig. 5(a) but the main difference is the filter size and the number of each layer. Fig. 5(c) consists of two convolutional layers, two pooling layers and two fully connected layers, where the final FC layer is the output layer. Fig. 5(d) consists of three convolutional layers, two pooling layers and three fully connected layers. The last CNN architecture of Fig. 5(e) consists of four convolutional layers, two pooling layers and two fully connected layers. The 1st convolutional layer used 32 filters, the 2nd convolutional layer used 64 filters, the 3rd convolutional layer employed 84 filters and the last convolutional layer used 124 filters. Different numbers of FC layers and max-pooling layers were used in the models to observe the effect.

TABLE 2
PROPOSED CONVOLUTION NEURAL NETWORK PARAMETERS

Parameters	Values
Optimizer	Stochastic Gradient Descent with Momentum 'sgdm'
Learning rate	0.01
Epochs	10
Maximum Iterations	700
Validation Frequency	30

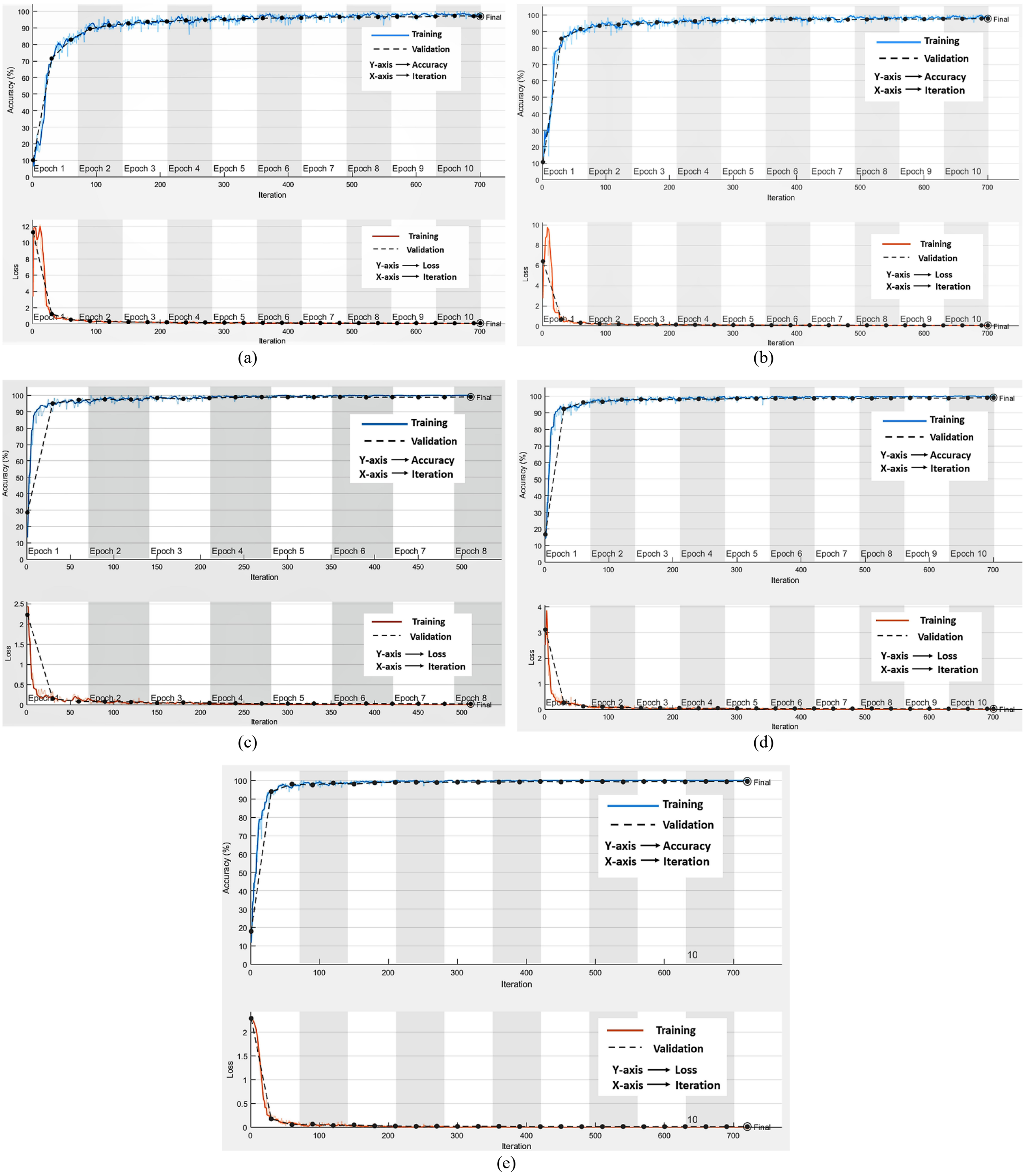


Fig. 6 Accuracy and loss graph of training progress in case of MNIST dataset: (a) Architecture 1 (b) Architecture 2 (c) Architecture 3 (d) Architecture 4 (e) Architecture 5

C. *Algorithm*

General algorithm for this proposes classification process is described below dataset step by step:
Step 1: Epoch number= 10, total number of classes= 10
Step 2: Input image dimension 28×28 (For MNIST Dataset)
Step 3: Input digit images loading from MNIST dataset
Step 4: Data splitting: training data = (9000,28,28,1) and validation data = (3000,28,28,1) (For MNIST Dataset)

Step 5: Creation of CNN model
Step 6: Network training
Step 7: Observation of validation accuracy and loss.

IV. RESULTS

MATLAB 2018a was used for this experiment environment, which was done in Core i5-7200U processor laptop. The proposed five different architectures were run in MATLAB for both datasets separately with respectable parameters (Table 2). After the CNN architectures are successfully implemented, the simulation provides worthwhile values of the validation graph. The simulation took some time for training. The validation graphs and loss graphs are provided in Fig. 6, where all the parameters are kept the same for each architecture. Then, the accuracies were observed for different datasets.

Fig. 6 presents the training, the validation accuracy, and the loss contingency of the proposed five CNN models, where MNIST dataset was used. Fig. 6 (a), Fig. 6(b), Fig. 6(c), Fig. 6(d), Fig. 6(e) indicate the training, the validation accuracy, and the loss contingency of the proposed architecture 1, architecture 2, architecture 3, architecture 4 and architecture 5, respectively. The blue line in the graph indicates the training progression, the black dot line indicates the validation accuracy, and the red line indicates the loss of data in progression of the inputted MNIST dataset. Fig. 6(e) demonstrates less data loss than other architecture graphs and so, it provides the best performance for the classification.

The validation accuracy and loss of the proposed CNN models for handwritten digit recognition are recorded in the following Table 3. This table is presented to help explore the best result and to show the best model among all proposed models.

TABLE 3
VALIDATION ACCURACY OF PROPOSED FIVE CNN MODELS

Proposed CNN models	Validation Accuracy (Dataset-1)	Error Rate (Dataset-1)	No. of iterations (Dataset-1)	Epochs (Dataset-1)	Validation Accuracy (Dataset-2)	Error Rate (Dataset-2)	No. of iterations (Dataset-2)	Epochs (Dataset -2)
Architecture 1	96.92%	3.0%	700	10	96.0%	4.0%	915	10
Architecture 2	97.7%	2.3%	700	10	96.8%	3.2%	1140	10
Architecture 3	99.0%	1.0%	510	10	98.2%	1.8%	630	10
Architecture 4	99.1%	0.9%	700	10	98.3%	1.7%	990	10
Architecture 5	**99.53%**	**0.47%**	720	10	**98.93%**	**1.07%**	870	10

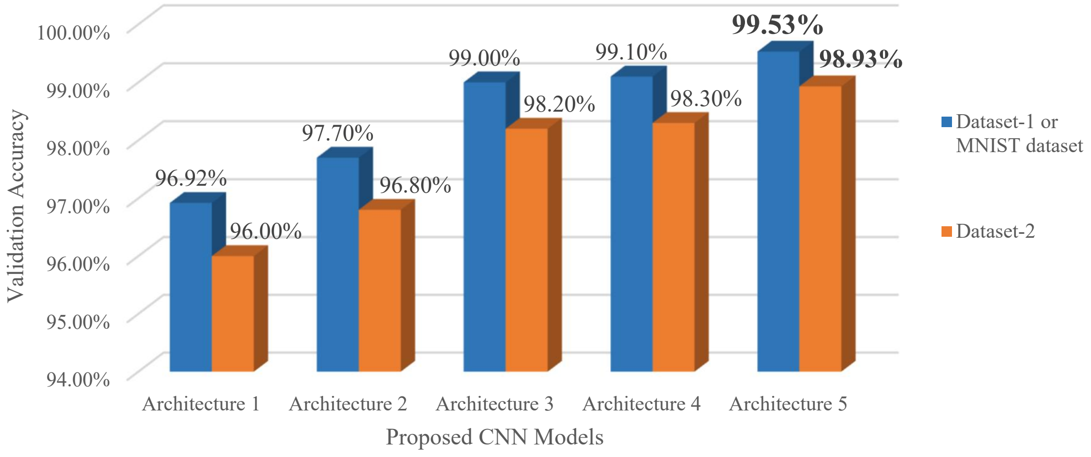


Fig. 7 Graphical representation of the validation accuracy of the proposed CNN models

Table 3 displays clearly that among the five proposed simplified CNN models, architecture 5 yielded the best validation accuracy of 99.53% (in case of MNIST dataset) and 98.93% (in case of Dataset-2), which are high enough for handwritten digit classification. This architecture 5 CNN model reduced the error rate up to 0.4% (in case of MNIST dataset), which means this CNN model can classify the handwritten English digits efficiently. Architecture 5 performed better than others and the MNIST dataset provided the best result through architecture 5. The network took

15 minutes 41 seconds to obtain the validation accuracy to work with MNIST dataset and 65 minutes 47 seconds for the Kaggle dataset. Increment of the network layer is responsible for the higher computational time to obtain the accuracy [4] [11] [15]. Again, the traditional recognition processes are also time-consuming for several processing stages [8][37].

Fig. 7 shows the validation performance of proposed five CNN architectures and among them, architecture 5 resulted in the highest accuracy. Again, the MNIST dataset provided a higher accuracy than Dataset-2. Fig. 8 presents the graphical representation of error rates of the proposed CNN models for both datasets. It shows that the error is reduced gradually up to 0.47% in case of MNIST dataset or Dataset-1. Architecture 5 resulted in less error rate, which means it can classify the handwritten digits more accurately.

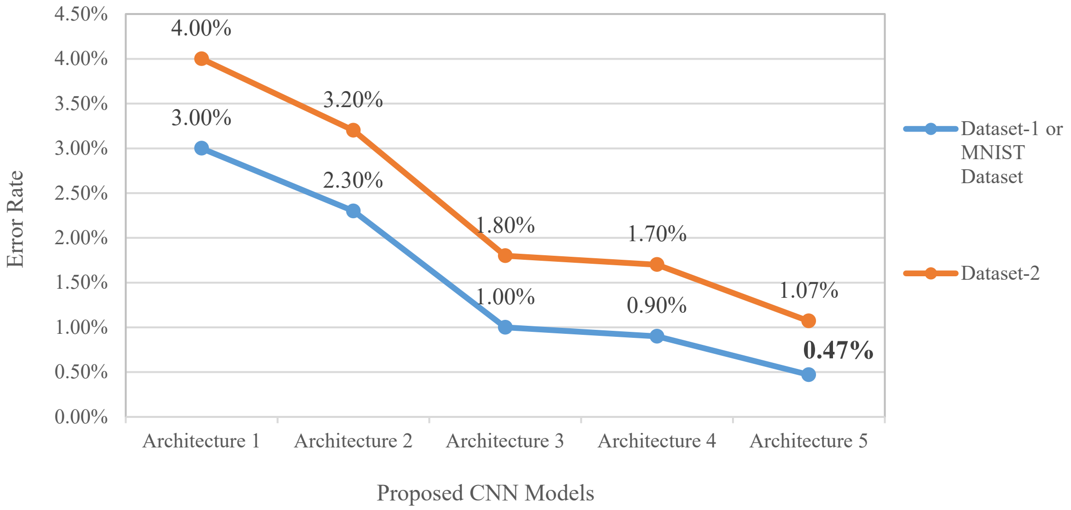


Fig. 8 Graphical representation of the error rates of the proposed CNN models

Fig. 9 is the confusion matrix of architecture 5 for MNIST dataset, which shows that this model can achieve the highest accuracy of 99.5% (i.e., round off of 99.53%) for ten-class classification. The diagonal values represent the individual class accuracy.

TABLE 4
FOR DIFFERENT OPTIMIZERS VALIDATION ACCURACY OF BEST CNN MODEL (USING MNIST DATASET OR DATASET-1)

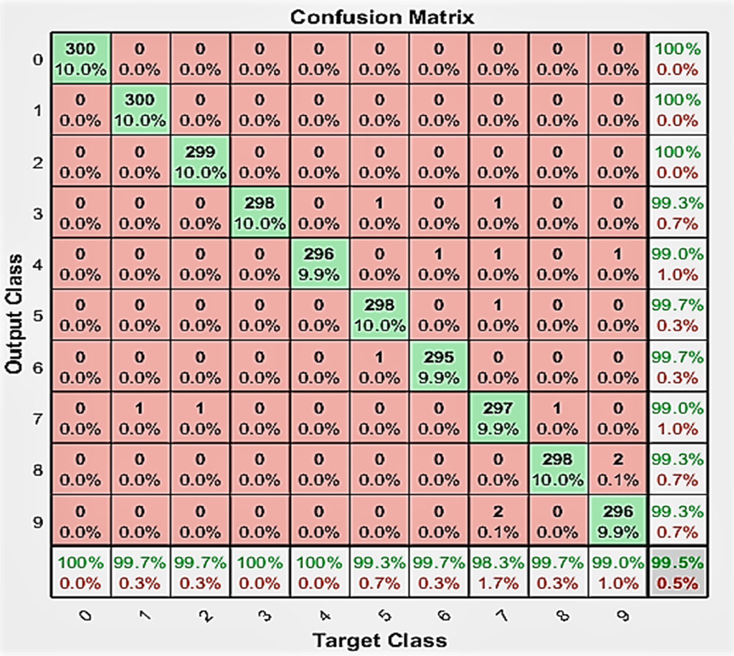
SGDM (Scottish Gradient Descent with Momentum)	Adam	RMSProp (Root Mean Square Propagation)
99.53%	98.33%	98.03%

The CNN model performance firmly depends on network optimizer. Reduced data loss, higher accuracy and less convergence time are the fundamental reasons for using optimizers. Various optimizers exist. Stochastic Gradient Descent (SGD) is the most used optimizer in deep learning as it is faster. Computational redundancy can be avoided by (SGD) optimizer [28]. By adding the ‘momentum’ parameter with SGD algorithm, optimization acceleration is achieved [28]. Adaptive moment estimation is known as ‘Adam’ optimizer [28], which is an improved version of stochastic gradient algorithm [28]. Adam optimizer is efficient and less memory-consuming [34]. It carries both AdaGrad and RMSProp feature combinations [34]. Continuously, per weight of squared gradients are calculated by moving average in root mean square propagation or RMSProp optimization [35]. It can improve the network performance by calibration. The proposed best CNN model is achieved for the classification with SGD optimizer. For experimental purposes, two other optimizers were applied to the best CNN model (architecture 5) with 0.01 learning rate to explore the performance, which is shown in Table 4. That model provides 98.33% accuracy when the network used 'Adam optimizer' and 98.03% accuracy when the network implemented 'RMSProp optimizer'. This experiment observation confirms that SGD optimizer provides the best validation accuracy for this network model.

TABLE 5
EFFECT OF DIFFERENT EPOCH VALIDATION ACCURACY OF BEST CNN MODEL (USING MNIST DATASET)

Epoch	Total Iterations	Completed Iterations	Validation Accuracy
5	370	370	99.21%
10	740	720	99.53%
15	1110	450	99.3%

Table 5 shows the effect of various epochs on the same model of architecture-5. Epoch indicates how many times the algorithm will work for the training datasets. Upgrade of weights is done per epoch. Theoretically, the increment of epoch number is able to improve the accuracy but too many epochs can cause an overfitting problem for the network. Table 5 also shows that when the epoch number was increased, the accuracy was increased. However, when the number of epochs reached a certain threshold, it was unable to fulfil the total iterations and the accuracy did not improve much than the previous. So, the optimum number of epochs must be set.



	0	1	2	3	4	5	6	7	8	9	
0	300 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
1	0 0.0%	300 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	0 0.0%	299 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	0 0.0%	298 10.0%	0 0.0%	1 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	99.3% 0.7%
4	0 0.0%	0 0.0%	0 0.0%	0 0.0%	296 9.9%	0 0.0%	1 0.0%	1 0.0%	0 0.0%	1 0.0%	99.0% 1.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	298 10.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	99.7% 0.3%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	295 9.9%	0 0.0%	0 0.0%	0 0.0%	99.7% 0.3%
7	0 0.0%	1 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	297 9.9%	1 0.0%	0 0.0%	99.0% 1.0%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	298 10.0%	2 0.1%	99.3% 0.7%
9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.1%	0 0.0%	296 9.9%	99.3% 0.7%
	100% 0.0%	99.7% 0.3%	99.7% 0.3%	100% 0.0%	100% 0.0%	99.3% 0.7%	99.7% 0.3%	98.3% 1.7%	99.7% 0.3%	99.0% 1.0%	99.5% 0.5%

Fig. 9 Confusion matrix of the validation accuracy of architecture 5 for MNIST dataset

V. DISCUSSION

The result of the proposed five CNN models shows that architecture 5 is the best model for the handwritten digit classification perfectly. This model proved the best result of 99.53% validation accuracy using the MNIST dataset. This best model or architecture 5 consists of four convolutional layers, two max-pooling layers and two fully connected layers, where the last fully connected layer is the output layer for ten-class classification. The proposed best result was obtained from the five CNN models and the best CNN model yielded the highest accuracy. A filter at the size of 3×3 remained the same for all convolutional layers in the architecture 5 CNN model. The 28×28 size of MNIST dataset was input to this model. 32 of 3×3 kernels were applied, and the feature maps were obtained, and then were convoluted again with the same size of kernel and 64 filters were added.

The second convolution layer's feature maps were then sub-sampled. After that, the values were passed through 84 of 3×3 size filters. These feature maps are convoluted again with 124 of 3×3 size filters. Finally, values were used for sub-sampling again. Then, the feature maps were passed into a fully connected layer with 50 neurons. The final fully connected layer used 10 neurons for the ten-class classification. This model was trained and validated with 12,000 image data. The obtained accuracy was 99.53% within 10 epochs. FC layers were used for the labelling to classify further. The feature value was taken and passed to the next layer to make the final prediction. Max-pooling layer was used to reduce the feature information, which was obtained from the previous two convolution layers. It helped to reduce the network size and provided fast training.

Architecture 1 took the most time than others as it has a max-pooling layer after 3 consecutive convolutional layers. But architecture 5 has only one max-pooling layer after 2 consecutive convolutional layers. Therefore, the information is reduced, and less time is taken. In this way, the FC layers and max-pooling layers have an impact on the best model. The obtained best model results in high performance without performing any pre-processing cost. This result is much better than the existing classifiers.

TABLE 6
COMPARISON OF PROPOSED BEST CNN MODEL WITH OTHER EXISTING WORK ACCURACY

References	Methodology of classification	Pre-processing steps existed or not	Accuracy
[36]	Neural Network+ Fuzzy Logic	Noise removal, centralization, normalization	56.6%
[37]	MLP	Thresholding, median filter, normalization	80.00%
[22]	Histogram feature extraction+ SVM	None	95.60%
[19]	CNN	None	95.7%
[38]	SVM	None	96.93%
[39]	HOG+ SVM	Image resizing	97.25%
[3]	DNN	Thresholding, morphological operation, slant correction, segmentation	97.7%
[21]	CNN	None	98%
[40]	DNN	Noise reduction, segmentation	98.08%
[41]	DC-ELM	None	98.43%
[42]	Active learning approach	Image resizing	98.54%
[15]	CNN	None	98.85%
[43]	DCT+ SVM	None	98.88%
[10]	LeNet-5	None	99.10%
[16]	CNN	None	99.21%
[4]	CNN	None	99.21%
[23]	CNN+ SVM	None	99.28%
[18]	CNN	None	99.30%
[44]	Gabor Filter+ CNN	None	99.32%
[20]	MLP	None	99.32%
[45]	HOG+ SVM	None	99.36%
[46]	CNN	Normalization	99.4%
Proposed best model	**CNN**	**None**	**99.53%**

Table 6 demonstrates that there are existing methods for classifying handwritten digits and the proposed best CNN model is capable of achieving 99.53% accuracy. This accuracy is enhanced compared to the previous works. There are several reasons. First, the proposed method does not use any pre-processing steps that mean it does not have any effect on the result. Next, the network modelling is the MNIST dataset. Also, CNN is one of the best feature extractor and classifiers among other machine learning algorithms [46]. There is a hypothesis that an enhanced layer can provide a better accuracy, but this must be correlative with the image size. MNIST is a well-recognized dataset and consists of 28 × 28 pixels image data. So, higher accuracy cannot be obtained by using an increased number of CNN layers [4]. For that reason, the proposed five shallow CNN models were proposed and the best model (architecture 5), which consists of 4 convolutional layers, was achieved.

Architecture 5 uses the highest convolutional layers than other architecture models in a proper way to extract the features perfectly from an image. Much smaller networks are unable to extract features perfectly and so they result in a lower accuracy. Another reason for the improved accuracy is setting the proper optimizer for the network (as shown in Table 4). Again, a proper learning rate also affects the accuracy. All things considered, the proposed simplified CNN model is effective enough because it can classify the handwritten digits easily and accurately.

Nevertheless, it should be noted that the proposed best CNN model deals only with MNIST dataset for this experiment purpose. Therefore, it limits the results of the experiment. Other types of dataset were not involved in the current study to observe the result of CNN model. Future research will benefit from working with data augmentation and other datasets to explore the CNN model performance in greater depth.

VI. CONCLUSIONS

The main concern of this work is to improve the English handwritten digit classification accuracy by using deep learning. Handwriting classification has various practical applications. The current research used 12,000 standard

MNIST images in Dataset-1 and another 29,400 handwritten data in Dataset-2 and proposed 5 different CNN models for classification. SGDM was used for network optimization. One of these proposed CNN models (architecture 5) has improved the validation accuracy to 99.53% by using MNIST standard dataset. This model consists of four convolutional layers, two max-pooling layers and two fully connected layers. The filter numbers of each convolutional layer were increased with the depth of the network. The best CNN model is simple and more accurate than other proposed models. The model is further trained with large unbiased datasets to enhance the validation accuracy even more. The best CNN model was applied with two other optimizers, but the results show that the SGDM provided the highest accuracy. The accuracy of the best CNN model was compared with the existing methods. The result shows that the proposed model is effective enough to classify handwritten digits accurately. This model is also less time-consuming as it avoids pre-processing costs and a deep number of layers and has the capability of providing better results for different datasets.

Author Contributions: *Angona Biswas*: Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing. *Md. Saiful Islam*: Methodology, Software, Validation, Data Curation, Investigation, Writing - Review & Editing, Supervision.

Funding: This research received no specific grant from any funding agency.

Acknowledgments: Authors are grateful to the Chittagong University of Engineering and Technology and department of Electronics and Telecommunication Engineering. Sincere gratitude to Arif Istiaq (Dept. ETE, CUET, batch'15) for his suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

REFERENCES

[1] Mahmoud M. Abu Ghosh, Ashraf Y. Maghari, "A Comparative Study on Handwriting Digit Recognition Using Neural Networks," 2017 IEEE International Conference on Promising Electronic Technologies, pp. 77-81, 2017.

[2] Kaensar C., "A Comparative Study on Handwriting Digit Recognition Classifier Using Neural Network, Support Vector Machine and K-Nearest Neighbor", 9th International Conference on Computing and InformationTechnology (IC2IT2013), Advances in Intelligent Systems and Computing, Springer, vol 209, pp. 155-163.

[3] Teddy Surya Gunawan, Ahmad Fakhrur Razi Mohd Noor, Mira Kartiwi, Development of English Handwritten Recognition Using Deep Neural Network, Indonesian Journal of Electrical Engineering and Computer Science, vol. 10, no. 2, , pp. 562-568, May 2018.

[4] Saqib Ali, Zeeshan Shaukat, Muhammad Azeem, Zareen Sakhawat, Tariq Mahmood, Khalil ur Rehman, "An efcient and improved scheme for handwritten digit recognition based on convolutional neural network", SN Applied Sciences, vol. 1, no. 9, August, 2019. DOI: https://doi.org/10.1007/s42452-019-1161-5

[5] N. Shobha Rani, N. Chandan, A. Sajan Jain, H. R. Kiran, "Deformed character recognition using convolutional neural networks", International Journal of Engineering & Technology, vol. 7, no. 3, pp. 1599-1604, 2018. DOI: 10.14419/ijet.v7i3.14053.

[6] Adarsh Trivedi, Siddhant Srivastava, Apoorva Mishra, Anupam Shukla, Ritu Tiwari, "Hybrid evolutionary approach for Devanagari handwritten numeral recognition using Convolutional Neural Network", The 6th International Conference on Smart Computing and Communications, Elsevier, vol. 125, pp. 525–532, 2018. DOI: 10.1016/j.procs.2017.12.068.

[7] Nibaran Das, Ram Sarkar, Subhadip Basu, Mahantapas Kundu, Mita Nasipuri, Dipak Kumar Basu, "A genetic algorithm based region sampling for selection of local features in handwritten digit recognition application", Applied Soft Computing, Elsevier, vol. 12, pp. 1592-1606., 2012. DOI: 10.1016/j.asoc.2011.11.030.

[8] Ali A. Alani, "Arabic Handwritten Digit Recognition Based on Restricted Boltzmann Machine and Convolutional Neural Networks", Multidisciplinary Digital Publishing Institute (MDPI), vol. 8, pp. 1-13, 2017. DOI: 10.3390/info8040142.

[9] Ashadullah Shawon, Md. Jamil-Ur Rahman, Firoz Mahmud, M.M Arefin Zaman, "Bangla Handwritten Digit Recognition Using Deep CNN for Large and Unbiased Dataset", International Conference on Bangla Speech and Language Processing (ICBSLP), IEEE, September 2018. DOI: 10.1109/ICBSLP.2018.8554900.

[10] Durjoy Sen Maitra, Ujjwal Bhattacharya and Swapan K. Parui, "CNN Based Common Approach to Handwritten Character Recognition of Multiple Scripts", 2015 13th International Conference on Document Analysis and Recognition (ICDAR), IEEE, pp. 1021-1025, August 2015. DOI: 10.1109/ICDAR.2015.7333916.

[11] In-Jung Kim, Xiaohui Xie, "International Journal on Document Analysis and Recognition (IJDAR)", International Journal on Document Analysis and Recognition (IJDAR), vol. 18, pp. 1-13, 2015. DOI: 10.1007/s10032-014-0229-4.

[12] Hubert Cecotti, "Active graph based semi-supervised learning using image matching: application to handwritten digit recognition", Pattern Recognition Letters, ELSEVIER, vol. 73, pp. 76-82, 1 April 2016. DOI: 10.1016/j.patrec.2016.01.016.

[13] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar and Pierre-Alain Muller, “Transfer learning for time series classification”, IEEE International Conference on Big Data 2018, pp. 1-11, November 2018. DOI: 10.1109/BigData.2018.8621990.

[14] Xiao-Xiao Niu, Ching Y. Suen, “A novel hybrid CNN–SVM classifier for recognizing handwritten digits”, Pattern Recognition, ELSEVIER, vol. 45, pp. 3018-3025, 19 October 2011. DOI: 10.1016/j.patcog.2011.09.021.

[15] Ranjan Jana and Siddhartha Bhattacharyya, “Character Recognition from Handwritten Image Using Convolutional Neural Networks”, Recent Trends in Signal and Image Processing, Advances in Intelligent Systems and Computing, Springer, Singapore, vol. 922, pp. 23-30, March 2019. DOI: 10.1007/978-981-13-6783-0_3.

[16] Fathma Siddique, Shadman Sakib, Md. Abu Bakr Siddique, “Recognition of Handwritten Digit using Convolutional Neural Network in Python with Tensorflow and Comparison of Performance for Various Hidden Layers”, 5th International Conference on Advances in Electrical engineering (ICAEE), IEEE, pp. 541-546, September 2019. DOI: 10.20944/preprints201903.0039.v2.

[17] Matthew Y.W. Teow, “Understanding Convolutional Neural Networks Using A Minimal Model for Handwritten Digit Recognition”, IEEE 2nd International Conference on Automatic Control and Intelligent Systems (I2CACIS 2017), pp. 167-172, October 2017. DOI: 10.1109/I2CACIS.2017.8239052.

[18] Vladimir Golovko, Mikhno Egor, Aliaksandr Brich, and Anatoliy Sachenko, “A Shallow Convolutional Neural Network for Accurate Handwritten Digits Classification”, Communications in Computer and Information Science, Springer, vol. 673, pp. 77-85, February 2017. DOI: 10.1007/978-3-319-54220-1_8.

[19] Dong-yuan Ge, Xi-fan Yao, Wen-jiang Xiang, Xue-jun Wen, En-chen Liu, “Design of High Accuracy Detector for MNIST Handwritten Digit Recognition Based on Convolutional Neural Network”, International Conference on Intelligent Computation Technology and Automation (ICICTA), IEEE, pp. 658-662, March 2020. DOI: 10.1109/ICICTA49267.2019.00145.

[20] Saeed AL-Mansoori, “Intelligent Handwritten Digit Recognition using Artificial Neural Network”, Int. Journal of Engineering Research and Applications, vol. 5, no. 5, pp. 46-51, May 2015. DOI: 10.13140/RG.2.1.2466.0649.

[21] Muthukrishnan Ramprasath, M.Vijay Anand, Shanmugasundaram Hariharan, “Image Classification using Convolutional Neural Networks”, International Journal of Pure and Applied Mathematics, vol. 119, no. 17, pp. 1307-1318, 2018.

[22] Eva Tuba, Milan Tuba, Dana Simian, “Handwritten Digit Recognition by Support Vector Machine Optimized by Bat Algorithm”, WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016, pp. 369-376, 2016.

[23] Savita Ahlawata , Amit Choudhary, “Hybrid CNN-SVM Classifier for Handwritten Digit Recognition”, International Conference on Computational Intelligence and Data Science (ICCIDS 2019), Procedia Computer Science, Elsevier, Vol. 167, pp. 2554-2560, 2020. DOI: 10.1016/j.procs.2020.03.309.

[24] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning”, Nature, vol. 521, pp. 436-444, May 2015.

[25] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do & Kaori Togashi, “Convolutional neural networks: an overview and application in radiology”, Insights Imaging, Springer, vol. 9, pp. 611–629, June 2018. DOI: 10.1007/s13244-018-0639-9

[26] Matthew Stewart, “Simple Introduction to Convolutional Neural Networks”, Towards Data Science, February 2019. Available at: https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac

[27] Danqing Liu, “A Practical Guide to ReLU”, Medilum, November 2017. Available at: https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7

[28] Savita Ahlawat, Amit Choudhary, Anand Nayyar, Saurabh Singh and Byungun Yoon, “Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN)”, Sensors, MDPI, vol. 20, pp. 1-18, June 2020. DOI: 10.3390/s20123344.

[29] Saad ALBAWI , Tareq Abed MOHAMMED, Saad AL-ZAWI, “Understanding of a Convolutional Neural Network”, 2017 International Conference on Engineering and Technology (ICET), IEEE, March 2018. DOI: 10.1109/ICEngTechnol.2017.8308186.

[30] Thomas Wood, “What is the Softmax Function”, Softmax Function, Deep AI, June 2019. Available at: https://deepai.org/machine-learning-glossary-and-terms/softmax-layer.

[31] N. Kwak, “Introduction to Convolutional Neural Networks (CNNs),”2016. Available at: http://mipal.snu.ac.kr/images/5/5e/Intro2CNNs.pdf

[32] “MNIST as .jpg”, Kaggle, 2016. Available at: https://www.kaggle.com/scolianni/mnistasjpg

[33] “Handwritten Digit Classification”, Kaggle, 2020. Available at: https://www.kaggle.com/pradheeprio/handwritten-digit-classification

[34] Diederik P. Kingma, Jimmy Lei Ba, “ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION”, 3rd International Conference for Learning Representations, San Diego, pp. 1-15, 2015.

[35] Thomas Kurbiel and Shahrzad Khaleghian, “Training of Deep Neural Networks based on Distance Measures using RMSProp”, Mathematics, Computer Science, ArXiv, 2017.

[36] Ahmad Al-Hmouz, Ghazanfar Latif, Jaafar Alghazo, and Rami Al-Hmouz, “Enhanced Numeral Recognition for Handwritten Multi-language Numerals Using Fuzzy Set-Based Decision Mechanism”, International Journal of Machine Learning and Computing, vol. 10, no. 1, pp. 99-107, January 2020. DOI: 10.18178/ijmlc.2020.10.1.905

[37] B. El Kessab, C. Daoui, B. Bouikhalene, M. Fakir, K. Moro, “Extraction Method of Handwritten Digit Recognition Tested on the MNIST Database”, International Journal of Advanced Science and Technology, vol. 50, pp. 99-110, January 2013.

[38] Chayaporn Kaensar, “A Comparative Study on Handwriting Digit Recognition Classifier Using Neural Network, Support Vector Machine and K-Nearest Neighbor”, 9th International Conference on Computing and InformationTechnology, AISC 209, pp. 155-163, January 2013. DOI: 10.1007/978-3-642-37371-8_19

[39] Ebrahimzadeh, Mahdi Jampour, "Efficient Handwritten Digit Recognition based on Histogram of Oriented Gradients and SVM", International Journal of Computer Applications, vol. 104, no.9, pp. 10-13, October 2014.

[40] Mahmoud M. Abu Ghosh, Ashraf Y. Maghari, “A Comparative Study on Handwriting Digit Recognition Using Neural Networks”, 2017 International Conference on Promising Electronic Technologies, IEEE, pp.77-81, 2017. DOI: 10.1109/ICPET.2017.20

[41] Shan Pang and Xinyi Yang, “Deep Convolutional Extreme Learning Machine and Its Application in Handwritten Digit Classification”, Computational Intelligence and Neuroscience, Hindawi, vol. 3, pp. 1-10, August 2016. DOI: 10.1155/2016/3049632

[42] Hubert Cecotti, “Active graph based semi-supervised learning using image matching: application to handwritten digit recognition”, Pattern Recognition Letters, vol. 73, pp. 76-82, February 2016.DOI: 10.1016/j.patrec.2016.01.016

[43] Bouchra EL QACIMY, Mounir AIT KERROUM and Ahmed HAMMOUCH, “Feature Extraction based on DCT for Handwritten Digit Recognition”, IJCSI International Journal of Computer Science Issues, vol. 11, Issue 6, no. 2, pp. 27-33, November 2014.

[44] Andr´es Calder´on, Sergio Roa and Jorge Victorino, “Handwritten Digit Recognition using Convolutional Neural Networks and Gabor filters”, e International Congress on Computational Intelligence CIIC, 2003.

[45] Hamayun A. Khan, “MCS HOG Features and SVM Based Handwritten Digit Recognition System”, Journal of Intelligent Learning Systems and Applications, vol. 09, no. 02, pp. 21-33, January 2017. DOI: 10.4236/jilsa.2017.92003

[46] Mohd Razif Shamsuddin, Shuzlina Abdul-Rahman and Azlinah Mohamed, “Exploratory Analysis of MNIST Handwritten Digit for Machine Learning Modelling”, 4th International Conference 2018, Soft Computing in Data Science (SCDS),pp. 134-145, 2018.