




Article

An Effective and Improved CNN-ELM Classifier for Handwritten Digits Recognition and Classification

Saqib Ali ¹, Jianqiang Li ¹, Yan Pei ^{2,*} , Muhammad Saqlain Aslam ³, Zeeshan Shaukat ¹ 
and Muhammad Azeem ⁴ 

¹ Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China; saqibsaleem788@hotmail.com (S.A.); lijianqiang@bjut.edu.cn (J.L.); zee@emails.bjut.edu.cn (Z.S.)

² Computer Science Division, University of Aizu, Aizu-wakamatsu, Fukushima 965-8580, Japan

³ Department of Computer Science and Information Engineering, National Central University, Taoyuan 32001, Taiwan; saqlain@g.ncu.edu.tw

⁴ Department of Information Technology, University of Sialkot, Punjab 51040, Pakistan; muhammad.azeem@uskt.edu.pk

* Correspondence: peiyan@u-aizu.ac.jp

Received: 21 September 2020; Accepted: 17 October 2020; Published: 21 October 2020



Abstract: Optical character recognition is gaining immense importance in the domain of deep learning. With each passing day, handwritten digits (0–9) data are increasing rapidly, and plenty of research has been conducted thus far. However, there is still a need to develop a robust model that can fetch useful information and investigate self-build handwritten digit data efficiently and effectively. The convolutional neural network (CNN) models incorporating a sigmoid activation function with a large number of derivatives have low efficiency in terms of feature extraction. Here, we designed a novel CNN model integrated with the extreme learning machine (ELM) algorithm. In this model, the sigmoid activation function is upgraded as the rectified linear unit (ReLU) activation function, and the CNN unit along with the ReLU activation function are used as a feature extractor. The ELM unit works as the image classifier, which makes the perfect symmetry for handwritten digit recognition. A deeplearning4j (DL4J) framework-based CNN-ELM model was developed and trained using the Modified National Institute of Standards and Technology (MNIST) database. Validation of the model was performed through self-build handwritten digits and USPS test datasets. Furthermore, we observed the variation of accuracies by adding various hidden layers in the architecture. Results reveal that the CNN-ELM-DL4J approach outperforms the conventional CNN models in terms of accuracy and computational time.

Keywords: optical character recognition; handwritten self-build digits images; deep learning; MNIST digits; feature extraction

1. Introduction

Recognizing handwritten digits from their images has been gaining great importance in the 21st century. Handwritten digits are used in various online handwritten applications like extracting postal zip codes [1], handling bank cheque amounts [2], and identifying vehicle license-plates [3], etc. All these domains are dealing with datasets and therefore demand high recognition accuracy with smaller computational complexity. It has been reported that deep learning models have more merits as compared to shallow neural designs [4–9]. Differences between DNN and SNN are described in Table 1. The objective of a handwriting digits recognition scheme is to transform handwritten characters images into machine-understandable formats. Generally, handwritten digits [10] are diverse in terms of orientation, size, and distance from the margins, thickness, security systems, and strokes,

which increase the complexity in recognizing handwritten numbers. Due to this diversity, handwritten numerals recognition is a challenging task for researchers. In the last few years, many machine learning and deep learning algorithms were developed for handwritten digits recognition (HDR). Boukharouba et al. [11] proposed a novel handwritten feature extraction technique using a support vector machine. In this model, the vertical and horizontal direction of handwritten digits were merged with the freeman chain code method, and the model doesn't require any digits normalization process. Mohebi et al. [12] proposed an HDR system by using self-organizing maps and obtained improved results compared with former self-organizing map (SOM)-based algorithms. Alwzway et al. [13] classified Arabic handwritten digits by implementing a robust deep belief neural network (DBNN) based open-source deep learning framework, Caffe, and achieved 95.7% accuracy, which is not up to mark.

Table 1. Shallow neural network vs. deep neural network.

Factors	Shallow Neural Network (SNN)	Deep Neural Network (DNN)
Feature Engineering	1. Individual feature extraction process is required. Various features cited in the literature are histogram oriented gradients, speeded up robust features, and local binary patterns.	1. Replace the hand-crafted features and directly work on the entire input. Thus, more practical for complex datasets.
Data Size Dependency	2. Needs a lesser quantity of data.	2. Needs vast volumes of data.
Size of hidden layers	3. Single hidden layer is required to fully connect the network.	3. Multiple hidden layers which may be fully connected.
Requirements	4. Give more importance to the quality of features and their extraction process. 5. More dependent on human expertise.	4. Automatically detects the significant features of an object, e.g., an image, handwritten character or a face. 5. Less human involvement.

Adhesh Garg et al. [14] present an efficient CNN model with several convolutions ReLU and pooling layers for a random dataset of English handwritten characters which is trained on the MNIST dataset, and this work obtained 68.57% testing accuracy. Ayushi Jain et al. [14] proposed a new method that introduced rotational invariance using multiple instances of convolutional neural networks (RIMCNN). They applied the RIMCNN model for classifying handwritten digits and rotated captcha recognition. This model achieved 99.53% of training accuracy. Akhtar et al. [15] presented a novel feature extraction technique whereby both support vector machines (SVM) and K-nearest neighbors (K-NNs) classifiers were used for offline handwritten digits recognition. Experimental results show 96.18% accuracy for SVM and 97% for K-NNs. Alex Krizhevsky et al. [16] presented a two-layer deep belief network (DBN) architecture that trained 1.6 million tiny images and achieved high classification performance by using CIFAR-10. Arora et al. [17] compared and CNN for the classification of handwritten digits. Results demonstrated that the CNN classifier performed better than the FNN. Malik and Roy [18] proposed a new approach that was artificial neural network (ANN)- and ELM-based for MNIST handwritten digits. The test accuracy of both the model was 96.6% and 98.4% respectively. Ali et al. [19] designed a model for recognizing the Sindhi handwritten digits, using multiple machine learning approaches. Their experimental results illustrate that the random forest classifier (RFC) and decision tree (DT) perform effectively as compared to other ML approaches. Bishnoi et al. [20] suggested a novel method for offline HDR using various databases like NIST and MNIST, whereby every digit was classified through four regions, including right, left, upper, and lower. These four regions curves were used for identifying images. Cruz et al. [21] presented a new method of feature extraction for handwritten digits identification and used ensemble classifiers. Overall, six feature sets were extracted, and this novel model was tested using the MNIST database. Nevertheless, most conventional methods discussed above were used freely available datasets like MNIST and CIFAR or other self-build datasets, including Arabic, Bangla, Sindhi, etc. However, very few works have been done on self-build datasets,

especially handwritten digits (0–9). Moreover, the reported work also has gaps in terms of accuracy and computational time, which need to be further improved.

Deep Learning4j (DL4J) is an open source, Java-based, distributed deep-learning library. It is also written in Scala that can be amalgamated with Hadoop and Spark [22]. DL4J is created in a way that can use distributed GPUs and CPUs platforms. It gives the capability to work with arbitrary n-dimensional arrays, and use the CPU and GPU resources. Distinct from various other frameworks, DL4J divides the updater algorithm from the optimization algorithm. This permits us to be flexible while trying to find a combination that works best for data and problem.

In the present work, we have proposed a self-build handwritten digit recognition system based on the symmetrical CNN-ELM algorithm. In comparison with other traditional classification algorithms, such as SVM, backpropagation (BP), ELM has fast in training speed as well as high training precision in short running time [23]. The proposed CNN-ELM architecture is split into two parts, namely features extraction and classification, as shown in Figure 1. Initially, an input image is given to the convolutional neural network for features extraction, and then the image is classified into one of the output classes. The whole process of the proposed method is as follows, firstly CNN unit with the ReLU activation function was used for feature extraction from handwritten digit images. Secondly, ELM is replaced with the last layer of CNN fully connected layer to classify the digits (0–9) based on feature vector obtained. In addition, this study made a contrast between different numbers of hidden layers for handwritten digits' recognition to validate CNN-ELM-DL4J architecture efficiency. Moreover, a self-build handwritten digits dataset of 4478 samples and a USPS test dataset are utilized to test the proposed model. The results indicate that the proposed framework recognized a self-build dataset and achieved state-of-the-art test accuracy in a short computational time.

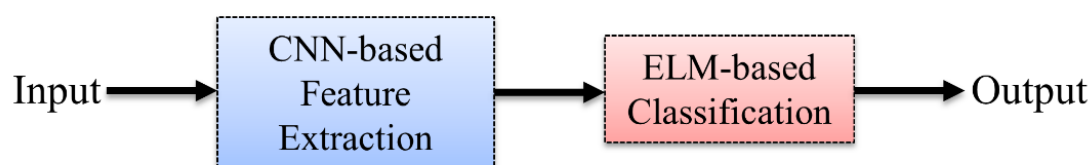


Figure 1. The pipeline of the Proposed CNN-ELM.

2. Related Work

In the previous two decades, research has been ongoing in deep learning. Using several machine learning algorithms, there have been countless trends in classifiers building on various datasets for image recognition and detection. In particular, deep learning on different datasets has shown improvement in accuracy. Deep learning algorithms like CNNs are broadly used for recognition. The MNIST dataset is a benchmark for handwritten digit that is used by numerous researchers to assess multiple leading-edge machine learning concepts. Several research papers are published in which the MNIST dataset is used for directing experimentations based on CNN and ELM pattern classification models, and they are described below.

Tan et al. [24] have proposed an SDAGD (“stochastic diagonal approximate greatest descent”) to train the weight parameters in CNN. Hamid et al. [25] used three different classifiers, namely KNN, SVM, and CNN, to assess the performance on MNIST datasets. The performance of Multilayer perceptron on that platform was not up to mark as it wasn’t able to accurately recognize digit 6, 9, and stuck in the local optimal rather global minimum didn’t obtain. With the implementation of Keras modality, it was reported that accuracy was improved on CNN as other classifiers, performed accurately. Xu et al. [26] researched on improving the overfitting in CNN. Gosh et al. [27] conducted a comparative study on MNIST dataset and implemented DBF (“deep belief networks”), DNN (“deep neural networks”), and CNN. It was concluded that with an accuracy rate of 98.08% performance of DNN was best among others as they had some error rates as well as differences in their time of

execution. Polania et al. [28] studied DBF and restricted Boltzmann machines (RBM's) for compressed sensing, and proposed a new scheme in their research.

Deng [29] presented a detailed survey on deep learning algorithms, architectures, and applications. All types, including generative, hybrid and discriminative architectures along with their algorithms, were discussed in detail. CNN, recurrent neural networks (RNN), autoencoders, DBNs, and RBMs were discussed with their various applications. Teow [30] has presented a minimal easily understandable CNN model for the recognition of handwritten digit. Yann le Cunn et al. [31] presented a thorough overview of deep learning and its algorithms. The algorithms like RNN, CNN, and backpropagation, along with multi-layer perceptron, have been discussed in all aspects with illustrations. These reported studies have demonstrated the trend of unsupervised learning in the field of artificial intelligence (AI).

Ercoli et al. [32] via obtaining multi k-means technique have premeditated hash codes and used them for repossession of visual descriptors. Krichevsky [16] on the CIFAR-10 dataset uses a 2-layer Convolutional Deep Belief Network (CDBN). The prototype obtained 78.90% accuracy in classification of a said dataset on a GPU unit. Abouelnaga et al. [33] constructed a collaborative classifier on K-nearest neighbor. They used a combination of KNN and CNN to reduce the overfitting by Principal Component Analysis (PCA). Improved accuracy of 0.7% was obtained by combining these two classifiers. Wang et al. [34] have discussed a new approach of optimization for edifice correlations between filters in CNN's. Chherawala et al. [35] proposed a vote weighted RNN's model to regulate the implication of feature sets. That model is an application of RNN and the significance of that model was determined by combinations of weighted votes. Its features were extracted from the images of Alex's word and then those features were used for handwriting recognition. Katayama and Yamane [36] suggested that the CNN architecture trained by rotated and un-rotated images undergo classification by the assessment of feature map obtained from its convolutional part. Pang and Yang [37] proposed a rapid learning model known as deep convolutional extreme learning machine (DC-ELM), by taking two datasets MNIST and USPS. Results show that the DC-ELM method improves testing accuracy and significantly decreases the training time. He et al. [38] present an effective model based on a combined CNN and regularized extreme learning machine, known as CNN-RELM, by utilizing ORL and NUST face databases. The proposed CNN-RELM model outperforms CNN and RELM. Xu et al. [39] constructed a sample selection-based hierarchical extreme learning machine (H-ELM) model for the classification task. They use a combination of FCM with CNN and H-ELM for data classification. Results reveal that the sample selection method achieves higher prediction results with a small training dataset, in a significantly short training time for the MINIST, CIFAR-10, and NORB databases. Das et al. [40] evaluate the performance of the ELM model on handwritten character recognition databases such as ISI-Kolkata Bangla characters, MNIST, ISI Kolkata Odia digits, and a new established NIT-RKL Bangla dataset. Shifei Ding et al. [41] investigate a novel convolutional extreme learning machine with a kernel (CKELM) model based on deep learning for solving various issues. KELM is not efficient for feature extraction, and DL takes excessive time for the training process. Results conclude that the performance of CKELM is higher than ELM, RELM, and KELM, especially in terms of accuracy.

3. Frameworks

3.1. Convolution Neural Network Framework

Nowadays, machine learning algorithms are trendy in the field of image segmentation and image classification because it cannot alter the topological structure of the images. CNN is a deep learning neural network, which is applying in various areas like, pattern recognition, speech analysis [42].

The conventional structure of CNN architecture comprises five layers which are shown in Figure 2. The first layer, which is the input layer has a normalized pattern of $S \times S$ size matrix. The feature map links the inputs of its prior layer. The convolution features derive from convolutional layer are the input for the max-pooling layer. Every neuron in the feature map shares the same kernel and the same

weights [43]. For example, by using 4 as kernel size, max-pooling ratio 2, stride 2, and padding of zero, all feature map layers shrink its features size S to $(S-4)/2$ from the previous feature size.

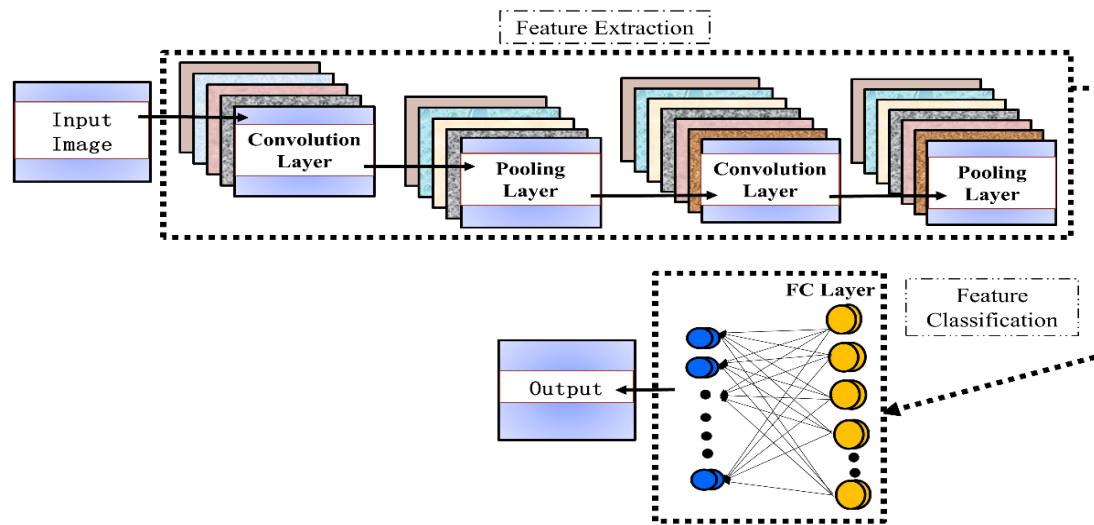


Figure 2. Structure of CNN.

There are some special structural features in the CNN architecture, including downsampling, weight sharing, and local sensing domain. Each layer has a single local perception domain neuron.

Generally, these neurons of each layer are only related to a certain domain, which is of 5×5 rectangular area the network input layer. Because of these special structural attributes, each neuron extracts the structural features of the input image. The training parameters of the CNN network can significantly shrink through weight sharing feature. Down-sampling is also an effective and unique feature of the CNN model, which is suitable for extracting images, reducing noise, and also have the capability to reduce the feature dimension. The CNN model constructs in a sequence of the input layer, hidden layers, and output layer. Two hidden layers: the convolution layer extracts features from the image, and the downsampling layer selects the optimized features from extracted features.

3.2. Extreme Learning Machine Framework

Extreme Learning Machine (ELM) is a feedforward neural networks. It is a fast learning algorithm which is established for a single-hidden layer that can recognize images. During the training process, no need to adjust or update the parameters, simply adjust the hidden layer nodes to find the best solution [44]. In contrast with the conventional classification methods like CNN and SVM [45], ELM has the power of very fast running and efficient learning speed, robust generalization capability, and few parameter adjustments. For a single hidden layer neural network, formally, suppose that we have a set of N arbitrary distinct samples (x_i, T_i) , where $x_i = [x_{i1}, x_{i2}, x_{i3}, \dots, x_{in}]^t \in \mathbb{R}^n$, $T_i = \{T_{i1}, T_{i2}, \dots, T_{im}\} \in \mathbb{R}^m$, For L number of hidden layers nodes, a single hidden layer can be described as follows,

$$O_i = \sum_{j=1}^L \gamma_j G(W_j \cdot x_i + b_j), \quad i = 1, 2, 3, \dots, N \quad (1)$$

where the activation function is denoted by $G(x)$, $W_j = [W_{j1}, W_{j2}, \dots, W_{jn}]^t$ belongs to input weights, γ_j is the output weight, bias is donated by b_j . $(W_j \cdot x_i)$ is indicated the inner product of inputs weights and input samples. A single hidden layer is used to reduce the error of output. It can be mathematically expressed as,

$$\min \sum_{i=1}^N \|O_i - T_i\| \quad (2)$$

If γ_j , W_j and b_j are exist, then it can be regarded and expressed as a matrix by the following Equation's,

$$\sum_{j=1}^L \gamma_j G(W_j \cdot x_i + b_j) = T_i, \quad i = 1, 2, 3, \dots, N \quad (3)$$

$$H\gamma = T. \quad (4)$$

By training the single hidden layer neural network, we can get \hat{W}_j , \hat{b}_j and $\hat{\gamma}_j$ which makes the Equation as follows (5)

$$\|H(\hat{W}_j, \hat{b}_j)\hat{\gamma}_j - T_i\| = \min\|H(W_j, b_j)\gamma_j - T_i\| \quad (5)$$

Here, $j = 1, 2, \dots, L$; it is equivalent to the minimum loss function

$$\min E = \min \sum_{i=1}^N \left(\sum_{j=1}^L \gamma_j G(W_j \cdot x_i + b_j) - T_i \right)^2 \quad (6)$$

The ELM algorithm does not require any adjustment for parameters. After randomly determining the input weights W_j and bias b_j , the γ of the hidden layer and output matrixes H are uniquely decided.

3.3. Combined and Improved CNN-ELM-DL4J Framework

In the convolution layer of Figure 2, the kernel is convoluted on the entire image and provides an output by using the activation function. Usually, convolution and subsampling layers are come out alternately in CNN. All output feature maps of the convolution layer are connected to the input feature maps. The output of convolution layer feature maps is obtained as follows.

$$X_i^n = f \left(\sum_{j \in M_i} X_i^{n-1} \cdot W_{ji} + \varnothing_i \right) \quad (7)$$

where n belongs to the number of convolution layer, convolution kernel is denoted by W_{ji} , \varnothing_i is used as a bias, and W_j belongs to the input map. The activation function is represented by $f()$. The sigmoid function is a typical CNN activation function. This increases the training time of the network. Therefore, an improved, easy to derive, unsaturated, and nonlinear ReLU function [46] is used in each convolution layer. ReLU function also reduces the overfitting issue and faster the convergence speed of the whole CNN architecture. The ReLU function is derived using a mathematical equation:

$$f(Z) = \max(z, 0) = \max \left(\sum_{j \in M_i} X_i^{n-1} \cdot W_{ji} + \varnothing_i, 0 \right) \quad (8)$$

In the present study, classification accuracy was obtained through the implementation of an enhanced CNN framework integrated with the ELM algorithm. The CNN unit is subjected to the extract features from the handwritten images and gives output while the ELM utilizes the output generated from the CNN unit as input and thus generates results by classifying the images.

4. Material and Methods

4.1. Used Datasets

For the experimental study, the well-renowned freely available MNIST dataset is utilized. This database consists of 70,000 images. In our experimental study, we have employed 60,000 images for training and 10,000 images for testing. The dataset was already normalized, i.e., there is no need for further pre-processing. Figure 3 represents the handwritten images from the MNIST dataset. Moreover,

in this project, the USPS test dataset of 2007 samples are also used for testing purposes. USPS [47,48] comprises 7291 training samples and 2007 testing samples in grayscale for the digits 0 to 9.

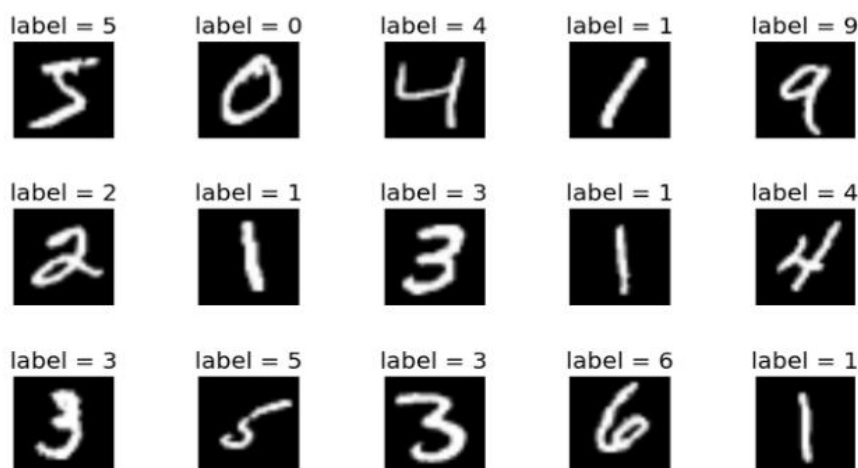


Figure 3. Typical MNIST Images with labels.

4.2. Own Test Dataset and Preprocessing

A self-build test dataset was created that contain 4478 handwritten digits images. Among all images, around five hundred photographs per digit are created for each numeral (0–9). The dataset was constructed by 5 university students. The ages of participants vary from 15 to 30 years. We cannot use our own dataset directly for experimentation because data items are untidy and different sizes. First, we have set grayscale image to 28×28 size, then transpose the colors in a way that the front became white and the background became black. While creating the dataset, it was considered to make each image as natural as ordinary people write the standard handwritten digits in their daily routine. We have used a self-build dataset only for testing purpose. The whole dataset depicts ten output classes for (0–9) digits.

4.3. CNN-ELM-DL4J Model Details

The proposed CNN-ELM structure mainly comprises the following layers: an input layer, a convolutional layer, a pooling layer, a fully connected layer, a Softmax layer, and ELM classification layer.

An input layer is the input of the neural network. Before entering this layer, it must be decided how much the image or input should be preprocessed. Networks like LeNet-5, for example, work well on images with little preprocessing.

The convolutional layer is the essential layer of CNN. In this layer, the images are transformed into a set of representative features. The main objective is to reduce the images into something easier to process, without losing their important characteristics, which means creating a feature map. The element involved in carrying out the convolution operation in the convolutional layer is named neuron, filter, or kernel. This element, which is a square matrix smaller than the image itself, is in charge of taking square patches of pixels and passing them through the filter with a certain stride till the image is completely parsed and, significant patterns in the pixels are found. The convolution consists of taking the dot product of the filter with the patch of the image, where the value of the filters can assign importance to some aspects of the image, so they can be able to differentiate one image from other. These values will tend to be learnable values, called weights, and will be reinforced by some other learnable values, called biases, which are constant values.

There are some commonly used activation functions like those in the linear unit family, for example, the ReLU function. The ReLU function has the characteristic of giving an output of zero for any negative input (or input of zero) while providing the same output value to any positive input. A down-sampling

(data reduction) operation is performed in the pooling layer, where the size of the feature map is reduced. There are different ways of down-sampling of the data, however max-pooling is the usually used option.

The convolutional and sub-sampling layers are succeeded by one or more fully connected layers, where each neuron attaches to all the neurons in the previous layer. The features of the image, extracted by the preceding layers, are combined to recognize large patterns, and the last ELM layer, combines the features to classify the images.

The number of outputs of the layer is equal to the number of classes in the target data (digit recognition is a 10-class recognition problem). The feature vector acquired from previous Fc_1 layer as the input for the ELM algorithm. ELM use some function to train the training set. After obtaining trained parameters, the predict function of ELM used for classification of test sets. In the end, the training and the validation set recognition accuracy was achieved.

The architecture used in this paper is a variation of the LeNet-5, and it was decided to implement this type of CNN-ELM architecture due to the nature of the character recognition problem (the LeNet-5 architecture was created to work specifically with a handwritten digit recognition problem). The structure of the improved CNN-ELM for handwritten digits image recognition is depicted in Figure 4, and the detailed setting of each layer is represented in Table 2.

Table 2. Description of the implemented architectures.

Layers	Parameters (CNN)	Parameters (CNN)
Input	$28 \times 28 \times 1$	$28 \times 28 \times 1$
CONV_1	Filters: $3 \times 3 \times 1 \times 8$ Activation: ReLU Stride: 2	Filters: $3 \times 3 \times 1 \times 8$ Activation: ReLU Stride: 2
POOL_1	Process: Downsampling Size: 2×2 Stride: 2	Process: Downsampling Size: 2×2 Stride: 2
CONV_2	Filters: $3 \times 3 \times 8 \times 16$ Activation: ReLU Stride: 2	Filters: $3 \times 3 \times 8 \times 16$ Activation: ReLU Stride: 2
POOL_2	Process: Downsampling Size: 2×2 Stride: 2	Process: Downsampling Size: 2×2 Stride: 2
CONV_3	Filters: $3 \times 3 \times 16 \times 32$ Activation: ReLU Stride: 1	Filters: $3 \times 3 \times 16 \times 32$ Activation: ReLU Stride: 1
FC-1	Filters: $1 \times 16 \times 6$ Stride: 1	Filters: $1 \times 16 \times 6$ Stride: 1
Softmax → ELM →	Classification Process -	- Classification Process
Output	Predicted Class	Predicted Class

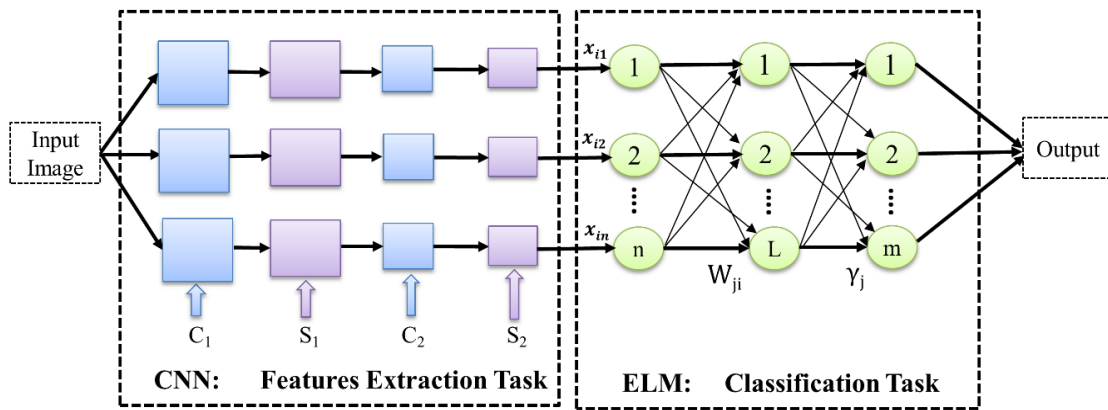


Figure 4. Structure of the improved CNN-ELM for handwritten digits image recognition.

4.4. The Training of CNN

In this phase, we have trained the CNN model, which is depicted in Figure 5. The associated features and the parameters are adjusted using the gradient descent method following errors among the actual output and the envisage output. The training process of the CNN model stops if the least error or an extreme number of iterations is reached, at which point the model is saved for the next step. Feature map of CNN can be adjusted and obtain as follows.

- i. For n -th layer of the convolutional layer, m -th feature map derived by the following equation,

$$x_i^L = f\left(\sum_{x_i^{n-1} \in N_m} x_j^{n-i} * k_{jm}^n + b_m^n\right) \tag{9}$$

where N_m belongs to the input set, f belongs to the nonlinear activation function, k_{jm}^n is the b_m^n convolutional filter, and b_m^n is the bias.

- ii. Similarly, for n -th number of subsampling layers, its m -th feature map is obtained by

$$x_m^n = f(w_m^n \text{down}(x_m^{n-1}) + b_m^n) \tag{10}$$

where w_m^n belongs to weights, $\text{down}(\cdot)$ is a pooling function, and b_m^n is the bias.

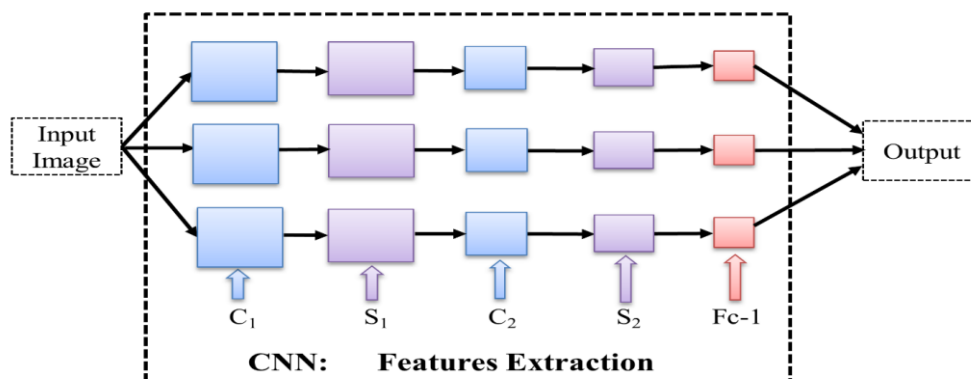


Figure 5. Training Topology of CNN.

5. Results and Discussion

In the experimental part, the model was trained on freely accessible MNIST dataset while the testing/validation of the framework was carried out by self-build handwritten and USPS digit datasets. Moreover, the results were analyzed through a confusion matrix. In addition to this, the architecture

was validated by changing the number of hidden layers. Finally, accuracy comparison of the proposed framework with the reported literature depicted that the state of the art accuracy has been achieved by combining CNN architecture with the ELM algorithm.

5.1. Digits vs. Error Rate

The error rate plot allows one to compute more statistical inference. The line graph represents the error rate versus numerals (digits) for both CNN and CNN-ELM networks in Figure 6a. One can infer from this diagram that the error rate for digit zero, two, and six is lowest 0. This might be attributed to the less cursive handwriting styles for digit 0, 2, and 6. Meanwhile, the highest error rate is found for digit 8 (0.94%) for our proposed (CNN-ELM) network owing to its resemblance with digit 3 and 5. However, the bare CNN network shows higher error rates in contrast to an extreme learning-based convolutional neural network. To justify this statement, the plot of handwritten digits vs. correctly and incorrectly classified images as depicted in Figure 6b. According to this plot, the digit zero, two, and six have the fewest incorrectly classified images, which directly means the lowest error rate. The reverse is the case for digit 8.

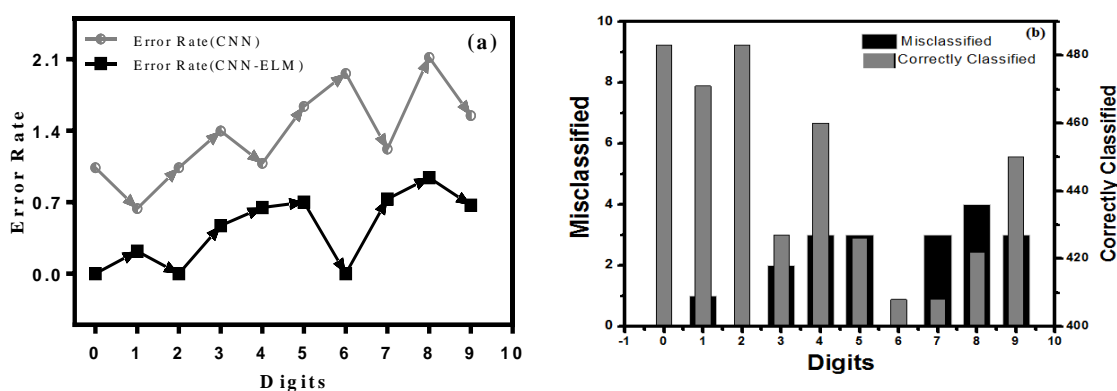


Figure 6. (a) Plot of error rate vs. Digits, (b) Plot of handwritten digits vs. incorrectly and correctly classified images.

5.2. Training and Validation Accuracy

We have observed the training and validation accuracy of CNN and CNN-ELM models simultaneously. From Figure 7a,b, the accuracy is very random during the training and validation period for each digit. The training accuracy of the CNN-ELM model is dominated by the training accuracy of CNN model. For instance, maximum training accuracy was recorded 98.90% for the digit five for CNN and 99.10% for digit zero and 5 for CNN-ELM seen in Figure 7a. Similarly, in Figure 7b, one can observe that the validation accuracy in CNN-ELM for all the digits is surpassing the validation accuracy for bare CNN. Altogether, the validation accuracy of the proposed model is dominated by the validation accuracy of CNN model. This happens because of efficient training which is done by adjusting the hyperparameters and selecting the appropriate one. These results are concordant with the error rate plot Figure 6a. Thus, all results evidence the proposed CNN-ELM model as a more efficient and superior model than the others, especially the bare convolutional neural networks.

The training set size highly affects network accuracy. The accuracy increases as more dataset are available to train the model. The training and testing dataset is not as much affected by the addition of more data in the training set.

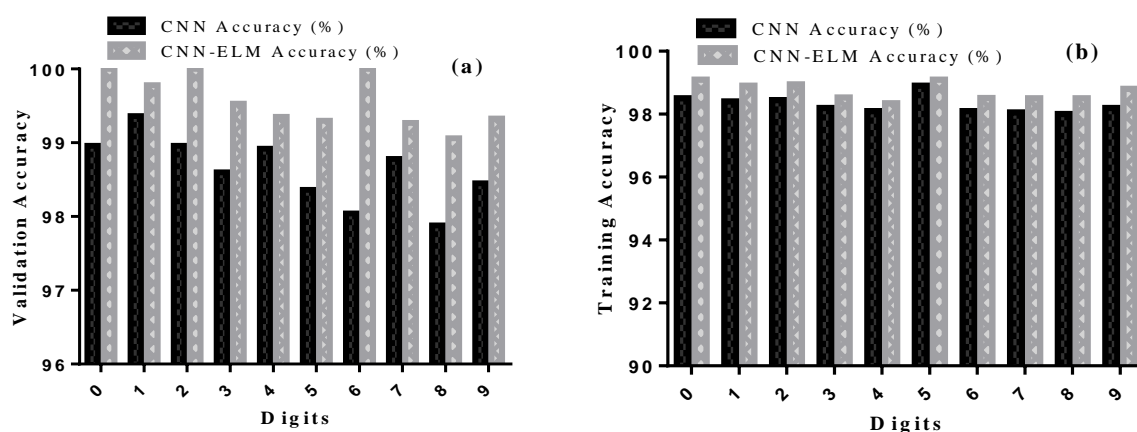


Figure 7. (a) Digits vs. training accuracy, (b) Digits vs. Validation accuracy.

5.3. Analysis through Confusion Matrix

The self-build handwritten numerals dataset is presented in the confusion matrix, as shown in Table 3. A self-build dataset comprises around 4500 (total number of images for each digit is highlighted in red colour) handwritten images were used for testing purpose; only 19 images were misclassified. For the digits 0, 2, and 6 recognition rate is 100% because not a single image is misclassified, Among 473 images of digit 1, only one image is wrongly predicted as 7. Similarly, for digit 3, two images were misclassified as digits 2 and 9. The complete details of incorrectly predicted images are shown in the table below. A total of 19 images is wrongly classified among the whole test dataset. The maximum number of wrong prediction is highlighted in pink colour. In the confusion matrix, we have shown that which misclassified image is categorized for which class. After vigilant surveillance of the patterns of these images, the causes behind these misclassifications are quite understandable.

Table 3. Confusion Matrix representing the classification performance of CNN-ELM model.

Numerals	Predicted Class										Accuracy (%)	
	0	1	2	3	4	5	6	7	8	9		
0	483										100	
1		472						1			99.78	
2			483								100	
3				1	429					1	99.53	
4					463		3				99.35	
5						1	429		2		99.30	
6								410			100	
7									411		99.27	
8										426	99.06	
9											453	99.33

5.4. Comparison of Different Number of Hidden Layers

Moreover, some additional experiments were also performed by increasing the number of layers to check the influence of the number of hidden layers on accuracy. According to reported literature by the increasing of hidden layers could give more accuracy [24]. In contrary, our proposed framework showed a decrement in accuracy with the increment of hidden layers. Figure 8a,b indicate that accuracy for the architecture constituting seven hidden layers is smaller than five and six hidden layers architecture for both CNN and CNN-ELM-DL4J models. However, the accuracy of the framework constituting five hidden layers is the highest. We also observe by increasing hidden layers, which leads to the complexity of the network, which also increases the computational time. This unusual approach is due to the amount of dataset. When experiments are performed on an enormous dataset, the framework with a high number of hidden layers may give more accuracy. In our case, the testing dataset contains only 4478 images; it can be seen that the accuracy through five hidden layers is highest.

Therefore, the proposed model needs a few parameters to accomplish higher recognition accuracy for self-build handwritten images, so it is computationally efficient. The accuracy comparison for various classification approaches is illustrated in Table 4.

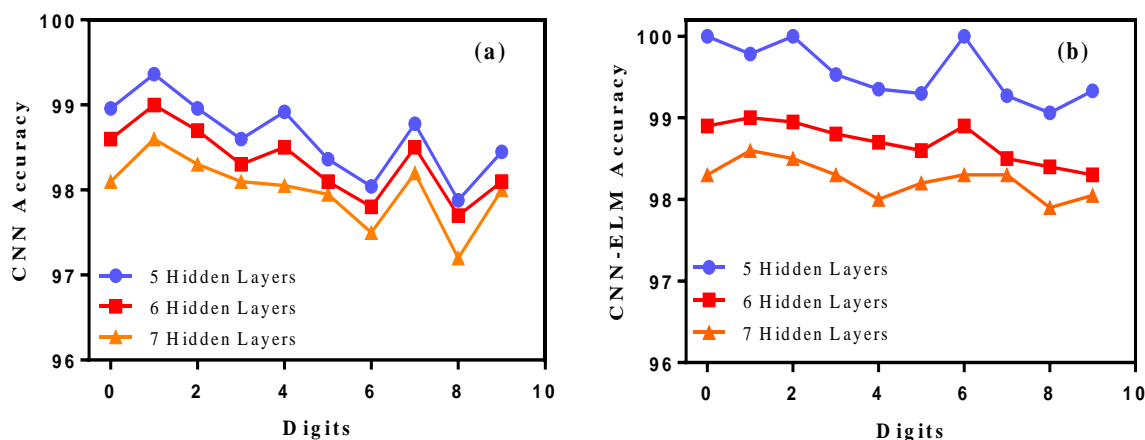


Figure 8. (a) CNN model comparison of hidden layers, (b) CNN-ELM model comparison of hidden layers.

Table 4. Classification accuracy Comparison.

Ref.	Approach	Database	Size of Sample	Testing Time (s)	Classification Accuracy (%)
[18]	ELM	MNIST	Small	-	98.4
[39]	H-ELM	MNIST	-	20.43	99.1
[39]	FCM-CNN-H-ELM	MNIST	-	16.79	98.7
[40]	ELM	MNIST	Large	-	97.7
[41]	CKELM	MNIST	Small	-	96.8
[44]	CNN-ELM	MSTAR	Small	-	100
[49]	CNN	MNIST	Large	58	99.2
[50]	Homo-ELM	MNIST	Small	-	97.0
[50]	Homo-ELM	NIST19	Small	-	98.3
[51]	Multiple fusion CNN	MNIST	-	-	98.0
This Work	CNN-ELM-DL4J	USPS	Large	26.52	99.7
This Work	CNN-ELM-DL4J	MNIST	Large	24.23	99.8
This Work	CNN-ELM-DL4J	Self-build	Small	11.27	99.6

6. Conclusions

The various deep learning-based model has been employed to recognize the handwritten digit from its image thus far. However, there is still a need for an efficient and effective model in terms of recognition accuracy, computational time, and high efficiency for feature extraction. Herein, a state-of-art convolutional neural network (CNN) with an extreme learning machine architecture was implemented to train the MNIST images and self-build handwritten numerals images used for model validation. Experimental results demonstrate that the CNN-ELM-DL4J algorithm is better than conventional CNN models in terms of recognition accuracy and computational time. By using the ELM algorithm, our model is computationally efficient as compared to simple CNN and other machine learning networks. Furthermore, we have explored the effect of a various number of hidden layers on the model's efficiency. From the results, it is concluded that adding more hidden layers led to an elevation in network complexity and computational time. Thus, the framework with an optimum number of hidden layers will give higher accuracy. For future work, experimental results can be improved and become more efficient by increasing/changing the dataset images and/or further tuning the network with appropriate parameters.

Author Contributions: Conceptualization, S.A.; methodology, S.A.; software, S.A.; formal analysis, Y.P.; resources, J.L.; writing—original draft preparation, S.A.; writing—review and editing, M.S.A.; Y.P.; Z.S. and M.A.; supervision, J.L. and Y.P.; project administration, J.L.; funding acquisition, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This study is supported by the National Key R&D Program of China with project no. 2017YFB1400803.

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

References

1. Billah, M.; Ruman, M.K.; Sadat, N.; Islam, M.M. Bangladeshi Post Office Automation System Using Neural Network. In Proceedings of the 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox's Bazar, Bangladesh, 7–9 February 2019; pp. 1–4.
2. Dansena, P.; Bag, S.; Pal, R. Differentiating pen inks in handwritten bank cheques using multi-layer perceptron. In Proceedings of the 2017 International Conference on Pattern Recognition and Machine Intelligence, Kolkata, India, 5–8 December 2017; pp. 655–663.
3. Selmi, Z.; Halima, M.B.; Alimi, A.M. Deep learning system for automatic license plate detection and recognition. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; pp. 1132–1138.
4. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–26 June 2005; pp. 886–893.
5. Xiao, J.; Zhu, X.; Huang, C.; Yang, X.; Wen, F.; Zhong, M. A new approach for stock price analysis and prediction based on SSA and SVM. *Int. J. Inf. Technol. Decis. Mak.* **2019**, *18*, 287–310. [[CrossRef](#)]
6. Wang, D.; Huang, L.; Tang, L. Dissipativity and synchronization of generalized BAM neural networks with multivariate discontinuous activations. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 3815–3827. [[PubMed](#)]
7. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [[CrossRef](#)]
8. Kuang, F.; Zhang, S.; Jin, Z.; Xu, W. A novel SVM by combining kernel principal component analysis and improved chaotic particle swarm optimization for intrusion detection. *Soft Comput.* **2015**, *19*, 1187–1199. [[CrossRef](#)]
9. Li, Y.-H.; Aslam, M.S.; Yang, K.-L.; Kao, C.-A.; Teng, S.-Y. Classification of Body Constitution Based on TCM Philosophy and Deep Learning. *Symmetry* **2020**, *12*, 803. [[CrossRef](#)]
10. Bengio, Y.; Courville, A.; Vincent, P. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828. [[CrossRef](#)]
11. Boukharouba, A.; Bennia, A. Novel feature extraction technique for the recognition of handwritten digits. *Appl. Comput. Inform.* **2017**, *13*, 19–26. [[CrossRef](#)]
12. Mohebi, E.; Bagirov, A. A convolutional recursive modified Self Organizing Map for handwritten digits recognition. *Neural Netw.* **2014**, *60*, 104–118. [[CrossRef](#)]
13. Alwzawzy, H.A.; Albehadili, H.M.; Alwan, Y.S.; Islam, N.E. Handwritten digit recognition using convolutional neural networks. *Int. J. Innov. Res. Comput. Commun. Eng.* **2016**, *4*, 1101–1106.
14. Jain, A.; Subrahmanyam, G.R.S.; Mishra, D. Rotation invariant digit recognition using convolutional neural network. In Proceedings of the 2018 2nd International Conference on Computer Vision & Image Processing, Chengdu, China, 16–18 June 2018; pp. 91–102.
15. Akhtar, M.S.; Qureshi, H.A.; Alquhayz, H. High-quality wavelets features extraction for handwritten arabic numerals recognition. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2019**, *9*, 700–710. [[CrossRef](#)]
16. Krizhevsky, A.; Hinton, G. Convolutional deep belief networks on cifar-10. Unpublished Work. **2010**, *40*, 1–9.
17. Arora, S.; Bhatia, M.S. Handwriting recognition using Deep Learning in Keras. In Proceedings of the 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida (UP), India, 12–13 October 2018; pp. 142–145.
18. Malik, H.; Roy, N. Extreme Learning Machine-Based Image Classification Model Using Handwritten Digit Database. In *Applications of Artificial Intelligence Techniques in Engineering*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 607–618.

19. Ali, I.; Ali, I.; Subhash, A.K.; Raza, S.A.; Hassan, B.; Bhatti, P. Sindhi Handwritten-Digits Recognition Using Machine Learning Techniques. *Int. J. Comput. Sci. Netw. Secur.* **2019**, *19*, 195–202.
20. Bishnoi, D.K.; Lakhwani, K. Advanced approaches of handwritten digit recognition using hybrid algorithm. *Int. J. Commun. Comput. Technol.* **2012**, *1*, 45–50.
21. Cruz, R.M.; Cavalcanti, G.D.; Ren, T.I. Handwritten digit recognition using multiple feature extraction techniques and classifier ensemble. In Proceedings of the 2010 17th International Conference on Systems, Signals and Image Processing, Rio de Janeiro, Brazil, 17–19 June 2010; pp. 215–218.
22. Kochura, Y.; Stirenko, S.; Alienin, O.; Novotarskiy, M.; Gordienko, Y. Comparative analysis of open source frameworks for machine learning with use case in single-threaded and multi-threaded modes. In Proceedings of the 2017 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), Lviv, Ukraine, 5–8 September 2017; pp. 373–376.
23. Huang, G.-B.; Zhu, Q.-Y.; Siew, C.-K. Extreme learning machine: A new learning scheme of feedforward neural networks. In Proceedings of the 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541), Budapest, Hungary, 25–29 July 2004; pp. 985–990.
24. Tan, H.H.; Lim, K.H.; Harno, H.G. Stochastic diagonal approximate greatest descent in neural networks. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 1895–1898.
25. Hamid, N.A.; Sjarif, N.N.A. Handwritten recognition using SVM, KNN and neural network. *arXiv* **2017**, arXiv:1702.00723.
26. Xu, Q.; Pan, G. SparseConnect: Regularising CNNs on fully connected layers. *Electron. Lett.* **2017**, *53*, 1246–1248. [[CrossRef](#)]
27. Ghosh, M.M.A.; Maghari, A.Y. A comparative study on handwriting digit recognition using neural networks. In Proceedings of the 2017 International Conference on Promising Electronic Technologies (ICPET), Deir El-Balah, Palestine, 16–17 October 2017; pp. 77–81.
28. Polania, L.F.; Barner, K.E. Exploiting restricted Boltzmann machines and deep belief networks in compressed sensing. *IEEE Trans. Signal Process.* **2017**, *65*, 4538–4550. [[CrossRef](#)]
29. Deng, L. A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Trans. Signal Inf. Process.* **2014**, *3*, 1–30. [[CrossRef](#)]
30. Teow, M.Y. Understanding convolutional neural networks using a minimal model for handwritten digit recognition. In Proceedings of the 2017 IEEE 2nd International Conference on Automatic Control and Intelligent Systems (I2CACIS), Kota Kinabalu, Malaysia, 21 October 2017; pp. 167–172.
31. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)]
32. Ercoli, S.; Bertini, M.; Del Bimbo, A. Compact hash codes for efficient visual descriptors retrieval in large scale databases. *IEEE Trans. Multimed.* **2017**, *19*, 2521–2532. [[CrossRef](#)]
33. Abouelnaga, Y.; Ali, O.S.; Rady, H.; Moustafa, M. CIFAR-10: KNN-based Ensemble of Classifiers. In Proceedings of the 2016 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 15–17 December 2016; pp. 1192–1195.
34. Wang, H.; Chen, P.; Kwong, S. Building correlations between filters in convolutional neural networks. *IEEE Trans. Cybern.* **2016**, *47*, 3218–3229. [[CrossRef](#)]
35. Chherawala, Y.; Roy, P.P.; Cheriet, M. Feature set evaluation for offline handwriting recognition systems: Application to the recurrent neural network model. *IEEE Trans. Cybern.* **2015**, *46*, 2825–2836. [[CrossRef](#)] [[PubMed](#)]
36. Katayama, N.; Yamane, S. Recognition of rotated images by angle estimation using feature map with CNN. In Proceedings of the 2017 IEEE 6th Global Conference on Consumer Electronics (GCCE), Nagoya, Japan, 9–12 October 2018; pp. 1–2.
37. Pang, S.; Yang, X. Deep convolutional extreme learning machine and its application in handwritten digit classification. *Comput. Intell. Neurosci.* **2016**, *2016*, 1–11. [[CrossRef](#)] [[PubMed](#)]
38. He, C.; Kang, H.; Yao, T.; Li, X. An effective classifier based on convolutional neural network and regularized extreme learning machine. *Math. Biosci. Eng. MBE* **2019**, *16*, 8309–8321. [[CrossRef](#)] [[PubMed](#)]
39. Xu, X.; Li, S.; Liang, T.; Sun, T. Sample selection-based hierarchical extreme learning machine. *Neurocomputing* **2020**, *377*, 95–102. [[CrossRef](#)]
40. Das, D.; Nayak, D.R.; Dash, R.; Majhi, B. An empirical evaluation of extreme learning machine: Application to handwritten character recognition. *Multimed. Tools Appl.* **2019**, *78*, 19495–19523. [[CrossRef](#)]

41. Ding, S.; Guo, L.; Hou, Y. Extreme learning machine with kernel model based on deep learning. *Neural Comput. Appl.* **2017**, *28*, 1975–1984. [[CrossRef](#)]
42. Sukittanon, S.; Surendran, A.C.; Platt, J.C.; Burges, C.J. Convolutional networks for speech detection. In Proceedings of the 2004 Eighth International Conference on Spoken Language Processing, Jeju Island, Korea, 4–8 October 2004; pp. 1077–1080.
43. Lauer, F.; Suen, C.Y.; Bloch, G. A trainable feature extractor for handwritten digit recognition. *Pattern Recognit.* **2007**, *40*, 1816–1824. [[CrossRef](#)]
44. Wang, P.; Zhang, X.; Hao, Y. A Method Combining CNN and ELM for Feature Extraction and Classification of SAR Image. *J. Sens.* **2019**, *2019*, 1–9. [[CrossRef](#)]
45. Niu, X.-X.; Suen, C.Y. A novel hybrid CNN-SVM classifier for recognizing handwritten digits. *Pattern Recognit.* **2012**, *45*, 1318–1325. [[CrossRef](#)]
46. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the 2012 Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
47. Maji, S.; Malik, J. *Fast and Accurate Digit Classification*; Technical Report No. UCB/EECS-2009-159; Electrical Engineering and Computer Sciences Department, University of California at Berkeley: Berkeley, CA, USA, 2009; pp. 1–11. Available online: <http://www2.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-159.pdf> (accessed on 20 October 2020).
48. Kusetogullari, H.; Yavariabdi, A.; Cheddad, A.; Grahn, H.; Hall, J. ARDIS: A Swedish historical handwritten digit dataset. *Neural Comput. Appl.* **2019**, 1–14. [[CrossRef](#)]
49. Ali, S.; Shaukat, Z.; Azeem, M.; Sakhawat, Z.; Mahmood, T.; ur Rehman, K. An efficient and improved scheme for handwritten digit recognition based on convolutional neural network. *SN Appl. Sci.* **2019**, *1*, 1125. [[CrossRef](#)]
50. Wang, W.; Gan, Y.; Vong, C.-M.; Chen, C. Homo-ELM: Fully homomorphic extreme learning machine. *Int. J. Mach. Learn. Cybern.* **2020**, *11*, 1–10. [[CrossRef](#)]
51. Zhao, H.-h.; Liu, H. Multiple classifiers fusion and CNN feature extraction for handwritten digits recognition. *Granul. Comput.* **2020**, *5*, 411–418. [[CrossRef](#)]

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).