Research Article

International Journal of Advanced Technology and Engineering Exploration, Vol 8(78)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2021.874073

An efficient CNN model with squirrel optimizer for handwritten digit recognition

T. Senthil[1]*, C. Rajan[2] and J. Deepika[3]
Assistant Professor, Department of ECE, Bannari Amman Institute of Technology, Sathyamangalam, Erode, Tamilnadu, India[1]
Associate Professor, Department of IT, K.S Rangasamy College of Technology, Tiruchengode, Namakkal, Tamilnadu, India[2]
Associate Professor, Department of IT, Bannari Amman Institute of Technology, Sathyamangalam, Erode, Tamilnadu, India[3]

Received: 10-April-2021; Revised: 26-May-2021; Accepted: 27-May-2021

Abstract

Automatic handwritten digit recognition provides significant contributions towards many real-time applications starting from the vehicle's number plate to doctor's prescription. However, the real challenge in these applications highly depends on the factors such as accuracy rate and time. Considering this significance, a novel handwritten digit recognition method is proposed without the adoption of any pre-processing steps like noise prediction, segmentation, and feature selection/extraction. The purpose of eliminating these preliminary steps is to reduce the computational complexity as the utilization of the Deep Learning (DL) approach helps to reduce the computational complexity of directly performing classification. Here, a novel Layered Convolutional Neural Networks (LCNN) model with an efficient Squirrel Optimizer (LCNN-SO) is modeled to attain better classification and global solution during the handwritten digit recognition. This cascaded model is a simple emerging DL-based one with multiple layers. The proposed LCNN-SO model adds multiple layers over the CNN model to focus on accurate classification and optimizing the layers to achieve better results using squirrel optimizer. The layered stages of CNN with the optimizer are trained and constructed to recognize the various kinds of digit isolation over the input data. Here, Special Database 1 and Special Database 2 are used to analyze and classify the input data for providing non-segregated digits for further processing in real-time applications. The simulation is carried out in MATLAB 2018 environment and metrics like accuracy, elapsed time, precision, recall, and F-measure are evaluated. The outcomes of these metrics are 98.5%, 99%, 99.5%, and 99.50% respectively. The anticipated LCNN-SO model gives better prediction accuracy when compared to existing approaches like Convolutional Neural Networks + Long-Short Term Memory (CNN+LSTM), pre-trained Convolutional Neural Networks + Multi-Layered Perceptron (CNN+MLP), pre-trained CNN+LSTM, pre-trained Convolutional Neural Networks +Support Vector Machine (CNN+SVM), Dense trajectories with Histogram of Gradients (HoG), and Convolutional Neural Networks + Orthogonal Learning Particle Swarm Optimization (OLPSO) respectively.

Keywords

Classification, Global convergence, Layered-CNN, Optimization and Recognition accuracy.

1.Introduction

The modern way of handwritten digit recognition has various potential real-time applications like management of students' score sheets, reading the tax form, postal code reading, and cheque processing [1]. Generally, diverse numeral strings are made of overlapping, segregated, connected, or wrecked digits. Segmentation, feature selection, and feature extraction of digit handwritten images with diverse numeral string lengths are analyzed by various researchers [2]. Those existing approaches are designed to predict the unconstrained handwritten digit which includes all the four pre-processing stages before classification.

The most challenging tasks with this prediction model are segmentation and feature extraction as these steps pretend to give better classification accuracy of isolated digits [3].

Generally, the numbers of digits over, the strings are unknown and therefore the optimal boundary between them is extremely complex to identify. Some existing

*Author for correspondence

feature extraction approaches rely on naturally inspired heuristic models to produce potential features for classification purposes [4]. Thus, the optimal selections of features are complex to predict due to their varying nature. Some generally used approaches over feature extraction are to enhance the classification results and optimal results are generated. Vellasques et al. [5] adopt a Support Vector Machine (SVM) classifier for eliminating redundant features and therefore reduce the consequence of redundancy of features during the classification process.

Ribas et al. [6] discuss the functionality of many existing segmentation processes based on appropriate segmentation and computational measure for various isolated digits. It is observed that the hybridization of various feature extraction models helps to enhance the classification outcomes. Thus, the foremost challenge is to build a classifier model that performs feature extraction internally to reduce the computational complexities and time needed for execution.

The complexities encountered during the selection and design of various appropriate segmentation and feature extraction models trigger the investigators to model a complexity-free classifier model. Thus, DL comes into the screen to perform classification.

Ciresan [7] used two layers of Convolutional Neural Networks (CNN). The first layer-based CNN model is used to predict the connected or overlapped digits where the second layer-based CNN model is used for classifying the digits of touching the strings. By examining the classifier score, the input data are classified by comparing the touching digits and isolated digits. The training of these two-layered CNN models is carried out without any negative samples. Ciresan handles this issue with the measurement of differences among the finest scores of the CNN network model. The input numeral strings from NIST’19 dataset with the adoption of two-digit classifier increases the prediction rate when compared to other classifier model used for digit isolation.

Hochuli et al. [8] provide the improved version of the two-layered CNN model designed by Ciresan with the training of a task-specific classifier model. This classifier model is termed as a length classifier which is adopted to compute the total amount of overlapped digits in the given string while the other classifiers are trained for predicting the segregated digits (two or three overlapped digit classes). The complexity arose with the classifier length, which is eliminated with the integration of classifier output to compute the results. The experimental outcomes for these overlapped digits from the NIST'19 dataset provide the advantages of this improved version. Thus, the results show that the modeling of this lengthy classifier leads to a huge misclassification rate. There are many investigations performed over the field of handwritten digit recognition; however, there is not much research for analyzing the entire document as the digital handwritten prediction is resourceful for various applications like official documents, bank cheques, medical prescriptions, etc. Also, it is helpful in police or detective (forensics) departments like the prediction of actual documents from the forged documents, handwriting-based person identification, and so on. The recent advancement in the technological field motivates to perform the handwritten digit prediction process using Deep Learning (DL) as it gives higher prediction accuracy than existing Machine Learning approaches.

The deep learning approach attains the superior performance than conventional approaches for various prediction problems. But, the existence of huge variations encountered in handwritten digit prediction problems uplifts the need and adoption of the deep learning-based classification model. This motivates the design of a simple Layered CNN (LCNN) model with an optimization approach to deal with the complexities encountered in the previous CNN models.

The concept behind the simple CNN modelling is presented in the prevailing pioneer work given by Viola and Jones [9] to sort out the problems over face detection. Here, the cascaded-trained classifier model with Ada-boost and HAAR feature extraction model is used. The upgraded or the modified version of this simple CNN model follows the cascaded nature with the inclusion of robust features. Also, indeed of traditional handcrafted features, CNN uses the cascaded model to learn the features and captures the visual more appropriately.

Zhang et al. [10] modelled a deep cascaded framework that uses the interconnection among the various image characteristics for enhancing the network performance. The structure adopted for this model is made of CNN which is inspired by the resourceful outcomes attained from the cascaded CNN model to handle the complex prediction problem. This work anticipates a novel layered-CNN with the integration of the optimizer model to predict the components of input handwritten digits.

The objectives of this work are discussed below:

1) To model a simple layered CNN model to predict the overlapping among the handwritten digits without the use of any expensive segmentation problem, but with the inclusion of an optimizer model.
2) To propose a novel LCNN model with an efficient Squirrel Optimizer (LCNN-SO) for evaluating the number of overlapping digits in every component.

The objective is fulfilled by avoiding the complex segmentation and feature extraction model as the layered components are extracted from the simple CNN and passed to the proposed LCNN and the squirrel optimizer model. The features are extracted and analyzed by the LCNN itself. The layers are trained and used to classify the overlapped/isolated digits by forwarding the tough regions to the successive layers.

The successive layers receive the complex regions of digit images and discriminate them with the digit classes by optimizing them. The proposed LCNN model is trained to label the handwritten digits by considering the isolated digits as negative samples. More layers are added to classify the digits more appropriately.

The experimental outcomes attained with the LCNN-SO model shows the efficiency of the designed model when compared to other methods like CNN+ Long-Short Term Memory (LSTM), pre-trained CNN+ Multi-Layered Perceptron (MLP), pre-trained CNN+LSTM, pre-trained CNN+SVM, Dense trajectories with histograms of oriented gradients (HoG), and CNN+ Orthogonal Learning Particle Swarm Optimization (OLPSO) approaches. Moreover, the proposed LCNN-SO framework enhances the execution time with the elimination of three-stage pre-processing. However, an optimization model is integrated with the CNN model to attain global solutions with the available number of the training dataset. Finally, the proposed LCNN-SO model attains the better performance by reducing computational complexity.

The remaining section is categorized as, section 2 shows the literature reviews on existing DL classifier model and three-stage pre-processing techniques for handwritten digit recognition. Section 3 contains proposed layered CNN with an optimizer model. Section 4 discussed the numerical results and the discussion part for the same. Section 5 shows the conclusion with future research directions.

2.Literature review

From the earlier '90s, handwritten digit prediction has been considered a major research field. Various efforts are given for languages like Arabic, English, Bangla, Hindi, Odia, and Lanna dhamma, and so on. There are various handwritten prediction models which need huge attention from researchers. Recently, deep learning has gained the huge attention of researchers due to its growing prediction accuracy. Here, unsupervised learning has been determined as the predominant solution for complexities encountered during the labelling of training data.

Coates and Ng [11] employed a conventional clustering model to provide feature specification. The inputs are normalized with patches for training images to learn various kinds of lower features with centroids. Then, the images are specified by predicting the similarity among the k-centroids and every local patch to generate an extremely large amount of image responses. The feature specification is done by polling the local features spatially with k-response images. Indeed, of this message's simplicity, it is extremely complex to construct an effective deep network model owing to the lack of various topological orders with k-means centroids and increased with several image responses.

Lawrence et al. [12] discuss the topological neuron orders over SOM grids to provide a suitable and appropriate SOM model for higher data dimensionality with feature representation and data visualization. The integration of CNN with SOM is made to deal with the face prediction problem. Here, SOM is adopted for the quantization of facial image patches over the topological space to diminish the dimensionality by increasing the robustness for the least variation. Here, CNN is trained with input images that are mapped and learned to analyze larger features.

Tan et al. [13] conduct experimentation with self-organizing map (SOM) for the prediction of invariant face occlusion. Here, the topological properties are mapped to the local patches and neuron coordinates. The modified K-Nearest Neighbor (KNN)model intends to give an ensemble classifier model to develop an appropriate match among the facial images with the topological representation. The hierarchical model was designed for feature extraction visually using map dimension. It is merged with Gabor filters for the construction of hierarchical representation for digit recognition.

Zhao et al. [14] modelled a stacked multi-layer organizing map for dealing with the background images. The author used a deep mapping model for automatic classification and feature extraction from data streaming. The author anticipates the hierarchical model for organization mapping for human recognition. Here, CNN and SOM model are developed for sign language prediction. Recently, various attempts are made to predict handwritten word recognition in various scripts with a holistic approach.

Dasgupta et al. [15] applied Arnold transformation to scuttle the input image and Hough transforms to attain various features from those input images. Then, the author used multi-class SVM for classifying the cursive letters. This model is applied to an English database that maintains 7744 images with diverse classes.

Tamen et al. [3] anticipate the Chebyshev model with contour and statistical features for Arabic text recognition. Here, classification is performed with various classifiers like an extreme learning machine, multi-layer perceptron, and attains superior results using the IFNENIT database which contains Arabic handwritten with 21 classes.

Jino et al. [16] anticipates a novel CNN for predicting words (Malayalam) and validates it on 314 classes from the available dataset. The features attained with SVM provide a robust accuracy. But the model does not consider the batch normalization concept; thereby it works poorly over a dataset of huge size.

Barua et al. [17] anticipate a holistic model for predicting handwritten words. The extracted features are ratio, area, pixel ratio, density, centroid, and long-run from the provided images. However, MLP is used to predict words from the input script specifying the state capitals and Indian' union territories.

Sueiras et al. [18] adopt segmentation with features of long-term memory to categorize English words. The author uses two different databases known as RIMES and IAM databases to verify the model efficiency.

Coates and Ng [11] employed a conventional clustering model to provide feature specification. The inputs are normalized with patches for training images to learn various kinds of lower features with centroids. Then, the images are specified by predicting the similarity among the k-centroids and every local patch to generate an extremely large amount of image responses. The feature specification is done by polling the local features spatially with k-response images. Indeed, of this message's simplicity, it is extremely complex to construct an effective deep network model owing to the lack of various topological orders with k-means centroids and increased with several image responses.

For the handwritten prediction of bangle words, both the holistic and analytical models are anticipated for the past few decades over major existing works. Moreover, the contemporary analysis of this model is to predict the bangle of handwritten words.

Bhowmik et al. [19] utilized a Genetic Algorithm (GA) and Markov model (MM) for isolation of bangle word prediction. This model attains 79% accuracy which includes 120 west Bengal hometown names where the city names are given as class names with 300 samples for all classes. Similarly, the elliptical features are given for five diverse classifiers to holistically predict the bangle handwritten text. The validation of these methods is done with the newly constructed dataset with 50 samples from word classes and attains an accuracy of 78% with MLP. Then, the author recommends the utilization of the HoG feature descriptor with a sequential optimization process for the bangle handwritten prediction model. The prediction accuracy obtained is 90% from the online accessible dataset with 10,000 samples (20 city names).

Recently, the author has initiated the design of a holistic model with shape-dependent feature descriptors that merges the tetragonal, elliptical, vertical pixel and density-based histogram models. Here, SVM and MLP classifiers are used for attaining prediction accuracy. For evaluation purposes, the author considers 18,000 samples of bangle handwritten word images that include roughly 120 city names where the prediction accuracy is 84% approximately. Similarly,

Sahoo et al. [4] discuss negative refraction properties with shape-feature descriptors to recognize the accuracy holistically. It includes 80 handwritten city names over bangle scripts. This model attains an accuracy of 88% over the 12,000 handwritten word sample by merging the prevailing logistic classifiers and self-organizing mapping model. From the above analysis, it is known that the feature extraction, segmentation process reduces the prediction accuracy. Therefore, it paves the way for modeling a deep learning-based classification model along with the optimizer to attain global solutions. Thus, a deep learning-based layered CNN model with squirrel search optimization is proposed to improve the prediction accuracy with reduced execution time.

Saha et al. [20] in 2019 presented a novel method using seven layered Deep CNN with substantial recognition precisions.

Recently, deep learning has gained the huge attention of researchers due to its growing prediction accuracy. Here, unsupervised learning has been determined as the predominant solution for complexities encountered during the labelling of training data. There are several other methods discussed in the same direction [21−25].

Table 1 shows the comparison of various existing methods used for digit recognition. From the earlier '90s, handwritten digit prediction has been considered a major research field. Various efforts are given for languages like Arabic, English, Bangla, Hindi, Odia, and Lanna dhamma, and so on. There are various handwritten prediction models which need huge attention from researchers.

Table 1 Comparison of various existing approaches

S. no.	Authors	Language	Methods	Year	Pros	Cons
1	Bag and Harit [26]	Arabic	Histogram of Oriented Gradient and SVM	2018	It shows both the photometric and geometric variations	Shows variations in large spatial regions
2	Elleuch et al. [27]	Arabic	Convolutional Neural Network	2016	Automatically detects the features	Larger training data required
3	Jauhiainen et al. [28]	Arabic	Automatic Hashing Text alignment	2019	It ensures both the original and duplicate copy	Lower security level
4	Kumar et al. [29]	Arabic	Deep Convolutional Neural Network	2019	It learns distinctive features	Requires expensive GPU
5	Al-Thubaity et al. [30]	Arabic	Bi-directional Long Short-Term Memory	2020	Predicts the time series	Takes longer time to train the data
6	Mahmood et al. [31]	Urdu	Convolutional Neural Networks + Multi-Layered Perceptrons	2020	It converts input to most essential outputs	Requires too many parameters
7	Husnain et al. [32]	Urdu	Convolutional Neural Networks + Support Vector Machine	2019	Shows clear margin among the classes	Not suitable for larger dataset
8	Parija et al. [33]	Indian	Support Vector Machine + Decision tree	2018	Output is easy and interprets without statistical knowledge	It is unstable and small variations lead to huge changes
9	Bahrani et al. [34]	Persian	Convolutional Neural Network	2019	Computationally efficient	Does not encode the orientation and position of input objects
10	Bossaghzadeh [35]	Persian	Convolutional Neural Networks	2020	Does not require any human intervention	Requires huge training data

3.Methods

The proposed methodology comprises three essential modules. They are dataset acquisition (special database 1, and special database 2), classification module (layered CNN), and optimization module (squirrel searching optimization). Similarly, the performance metrics like accuracy, recall, precision, F-measure, and execution time is evaluated and compared with various existing approaches. The block diagram of the proposed LCNN-SO model is given in *Figure 1*.

a. Dataset consideration

The handwritten digits database (MNIST) is considered for this research work (*Figure 2*). It has 60,000 training and 10,000 testing samples. This MNIST is the sub-set of the larger dataset from NIST, which is constructed [21]. This database comprises two diverse sources known as NNIST's special database 1, and special database 3. The images over this dataset are normalized for size (fixed size). The black and white images are normalized to 20×20 to fit the pixel box and maintain the aspect ratio. The outcome images contain grey levels which are attained from anti-aliasing approaches [22]. The center of pixel mass and image translation is positioned with 28×28 fields. Some existing classifier models reduce the error rate when the input digits are centered by bounding box and other center mass. The samples are collected from highschool students, while the latter samples are gathered from census bureau employees [36]. The samples are chosen for testing and training as the same writers are not involved in both sets. The training set holds more than 250 samples from the writers, while the remaining are provided for testing. The original images need to undergo pre-processing. Image normalization is performed to fit the 20×20-pixel boxes for maintaining the aspect ratio. Next, anti-aliasing is applied to acquire black and white images. It is next transformed into grayscale images. Finally, blank padding is performed to make the image fit with a 28×28-pixel box. Thus, the centre of mass of the digits is fitted with the digit-matched centre.

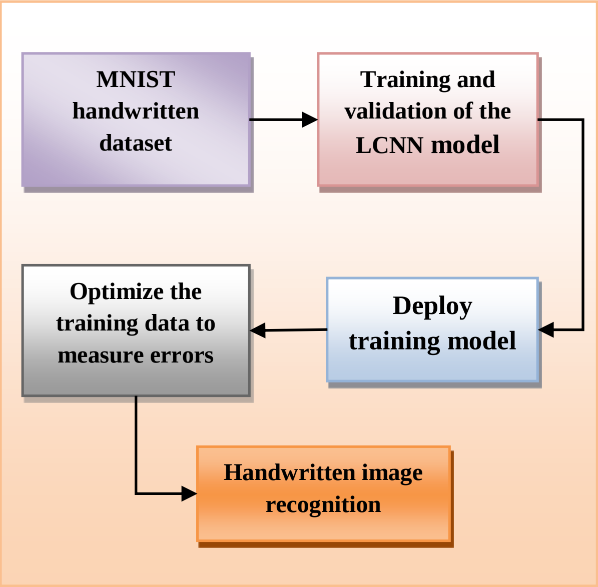
MNIST handwritten dataset
Training and validation of the LCNN model
Deploy training model
Optimize the training data to measure errors
Handwritten image recognition

Figure 1 Block diagram of LCNN-SO model

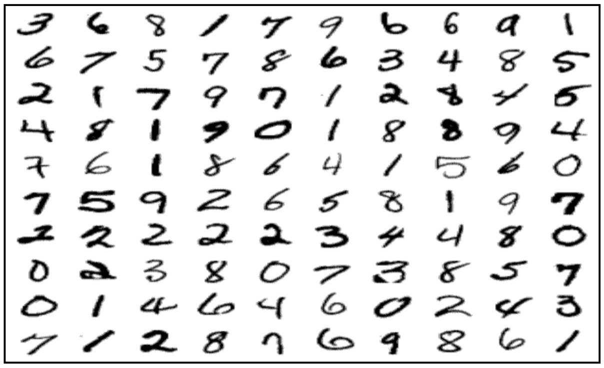

Figure 2 MNIST database sample – size normalized [21]

b. Layered CNN model

The total amount of input images over the MNIST handwritten digits database needs to be increased as DL needs a reasonable dataset. Here, normalization is a standard process that is applied when the numbers of images are artificially increased due to translation [23]. This process is essential to avoid over-fitting issues. The generated dataset is used for the validation and training process as shown in *Figure 4*. Here, pre-trained DL models, namely AlexNet, and GoogLeNet are used for validation and training dataset. AlexNet and GoogLeNet is a series of CNN with a repetitive convolution layer, followed by a non-linear activation function. It is termed a Rectified Linear Unit (ReLU). The ReLU and convolution operation is mathematically expressed as in Equation (1) & (2):

$$X_j^l = B_j^l + \sum_j^l K_{ij}^l * X_i^{l-1} \qquad (1)$$

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

Here, X_j^l is an activation map as an outcome of convolution operation (layer $'l'$) with kernel K_{ij}^l with activation map of prior layer X_i^{l-1} and B_j^l is biased. Generally, ReLU is followed by maximum pooling. It diminishes the activation map dimensionality. The mapping is performed from pooling regions. Some additional layers are added over the AlexNet which is known as batch normalization. This model is composed of 2 fully connected layers (4096 neurons) followed by successive connected layers. Here, the number of neurons is directly proportional to the number of classes, the classification layer, and the softmax layer. The outcome from the softmax layer is mathematically expressed with the probability, class vectors, and classification layers to determine the class labels of the input image. It is shown in Equation (3):

$$P_j = \max_{\mathrm{i\epsilon R_j}} \alpha_i \qquad (3)$$

$$y_m = \frac{e^{T_m}}{\sum_{n=1}^{M} e^{T_n}} \qquad (4)$$

Here, y_m is a possible class with the $'T'$ vector of the $'M'$ dimension. Similarly, the GoogleNet architecture is composed of an inception module that contains parallel convolution layers. Here, the features are extracted parallel and concatenated with the inception module. Therefore, it diminishes the computational complexity of the LCNN model. The modules are given in stack format with 27 layers. The depth of this layered architecture is increased with the adoption of the stacking convolution layer. Thus, the loss and accuracy are quickly saturated as shown in *Tables 5* and *6* (section 4). This process is performed quickly because of its vanishing gradients during the backpropagation process. The residual connections are established among the convolution layer to resolve the gradient problem.

•Convolutional layer

It processes the input image (handwritten digit) through a convolutional filter which activates the features extracted from the images. This reduces the complexity encountered during separate segmentation and feature extraction. The convolutional layer output is expressed as in Equation (5):

$$M_j^p = f\left(\sum_{i\epsilon M_j} M_i^{l-1} * k_{ij}^l + B_j^l\right) \qquad (5)$$

Here, $'l'$ layers, k_{ij} is a convolutional kernel, B_j^l is biased, M_j^p is input. Here, some parameters like weight and bias are typically trained with the proposed LCNN model (unsupervised learning model). The raw inputs obtained from the dataset are given to the convolutional layer via filters and activates the extraction of features. Here, the utilities are used to convolve the raw input images with intermediate mapping.

•Pooling layers

It simplifies the output with the adoption of a non-linear down-sampling process. It diminishes the total number of network parameters that need to be learned. The probability of evaluating every image region is expressed as in Equation (6):

$$P_i = \frac{\propto_i}{\sum k\varepsilon S_j \alpha_k} \qquad (6)$$

Here, $'F'$ is feature map, S_j is pooling region, $'i'$ is the index used for pooling purposes. It is expressed in Equation (7):

$$a_{xy}^{l,k} = (m,n,x,y)\epsilon P(\alpha_{m,n}^{l-1,F} w(x,y)) \qquad (7)$$

Here, $\propto, l, x, y$ is neuron activation mapping with $'f'$ feature maps over l^{th} layer where $w(x,y)$ is a weighted function.

•Fully connected layers

It flattens the network with the extracted features into a 1D vector that specifies image-level features for prediction purposes.

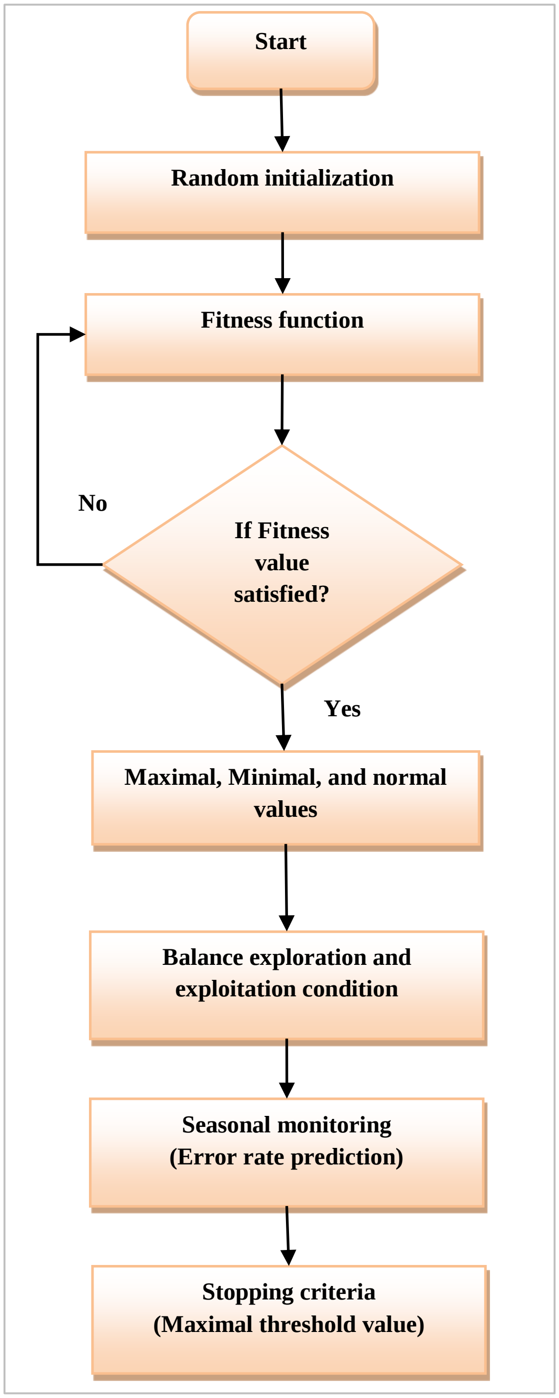
Start
Random initialization
Fitness function
No
If Fitness value satisfied?
Yes
Maximal, Minimal, and normal values
Balance exploration and exploitation condition
Seasonal monitoring (Error rate prediction)
Stopping criteria (Maximal threshold value)

Figure 3 Flow diagram of Squirrel optimizer

C. Squirrel search optimizer

Generally, optimization algorithms are essential for DL because the complexity of handling the data during the training process will consume more hours, days, and weeks.

The optimization algorithm performance can directly influence the training efficiency of the proposed LCNN model. This work considers squirrel optimizer as this optimizer shows better performance in handling the seasonal condition while preserving huge population diversity [24]. It balances both exploration and exploitation by fine-tuning the network model. This process initiates with the random location initialization based on the population [25]. The flying squirrel location is monitored with the vector specification and dimensional search space. The flow diagram of SO is given in *Figure 3*.

•Random initialization

When the number of data is increased while adopting the DL model; it increases the memory access and leads to a huge amount of execution time. The squirrel location is expressed as in Equation (7):

$$FS = \begin{bmatrix} FS_{1,1} & FS_{1,2} & \dots \dots & \dots & FS_{1,d} \\ \dots & \dots & \dots \dots & \dots & \dots \\ FS_{n,1} & FS_{n,2} & \dots \dots & \dots & FS_{n,d} \end{bmatrix} \qquad (7)$$

Here, $FS_{i,j}$ is the dimension of an i^{th} flying squirrel. It is used to allocate the location of all squirrels over the forest.

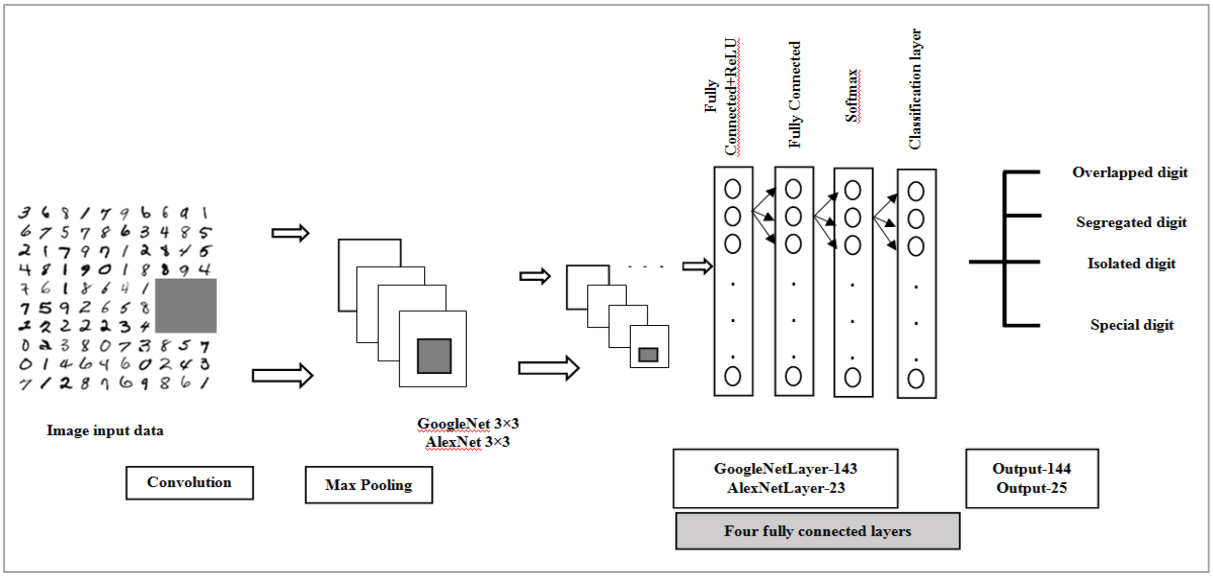

Figure 4 LCNN architecture model

•Fitness function

The fitness value is used to describe the quality of food source based on the probability of survival. The fitness functions are expressed as in Equation (8):

$$f = \begin{bmatrix} f_1\left(\left[FS_{1,1}, FS_{1,2}, \dots, FS_{1,d}\right]\right) \\ f_2\left(\left[FS_{2,1}, FS_{2,2}, \dots, FS_{2,d}\right]\right) \\ \dots . \\ \dots \\ f_n\left(\left[FS_{n,1}, FS_{n,2}, \dots, FS_{n,d}\right]\right) \end{bmatrix} \qquad (8)$$

•Declaration

After storing the fitness value, the array is sorted in ascending order. The minimal fitness value is provided for the hickory nut tree, the best fitness value is considered as an acorn tree, and the remaining fitness values are given to normal trees. The foraging nature of the squirrel is often influenced by the predator's occurrence (error rate). This issue is handled by the employment of a location updating mechanism with the probability measure of the predator's presence.

•Seasonal monitoring

The seasonal variation can influence the foraging activities of the squirrel as it suffers from heat loss issues. Generally, they maintain a high body temperature. Thus, the seasonal monitoring condition is used to eliminate the condition of being trapped over local optimal solutions. The minimal value influences the exploration and exploitation capabilities of the proposed SO. Here, the higher value enhances exploration and the lower value improves the exploitation capability. Thus, the stability balance is established with the adoption of SO.

•Stopping criteria

The tolerance towards the training process is essential to measure the convergence which is determined to be more permissive; however, the smaller threshold is depicted among the consecutive outcomes. When the maximal execution time is reached the stopping criteria is triggered. The squirrel optimizer functionality is shown in Algorithm 1.

Algorithm 1:

```
//begin the process
Step 1. Initialize the parameters with weighted vectors in a uniformly distributed manner   //input data from the dataset
Step 2. Generate random location for the squirrels by determining the neighborhood weighted vectors with $'FS'$.
Step 3. Compute fitness value using initial population as in Equation (8)
Step 4. Sort the location with target vectors    //based on fitness value
Step 5. Perform sorting with referral points    //minimal, maximal, and normal
Step 6. Randomly select the values of squirrel based on the neighborhood update //minimal – hickory nut tree, maximal – acorn tree, others- normal trees
        While (“stopping criteria not fulfilled”)
Step 7. Perform the random position by adjusting the weighted vectors;
        end while
Step 8. Compute seasonal monitoring ‘c’ and generate the weighted vector;
Step 9. Update the minimal seasonal value
Step 10. If the maximum threshold value is reached, then update neighborhood weighted
         vector;  //trigger stopping criteria
Step 11.     Attain final optimal solution;
Step 12. else goto Step 2;
//end the process
```

4.Results

The numerical results obtained from the experimentation of the proposed LCNN-SO approach is discussed in this section. Here, classification is carried out directly with the elimination of pre-processing steps. The simulation is carried out in MATLAB 2018b environment which runs on Windows 10 PC, Intel Core i7 processor, CPU (2.50 GHz). The input data are acquired from the MNIST special database 1, and special database 2 to carry out training and testing models. This dataset is composed of 60,000 training and 10,000 testing images. *Table 2* reveals the layer details of the two most popular CNN models Alexnet and GoogleNet. However, the proposed LCNN+SO follows a similar pattern as these models. The samples are normalized with a fixed image size of 28×28. *Table 3* and *4* depicts the prediction accuracy of the proposed LCNN+SO model. This is compared with existing approaches like CNN+LSTM, Pre-trained CNN+MLP, Pre-trained CNN+LSTM, pre-trained CNN+SVM, Dense trajectories with HoG, and CNN+OLPSO respectively. The major disadvantage with CNN+LSTM is that it does not encode the orientation and position of the object and it cannot be applied to the spatially invariant the input data. It also requires a huge amount of training data. The drawback associated with CNN+MLP is that it deals with too many parameters as the layers are fully connected. The nodes have connected, which results in inefficiency and redundancy. The features extracted from dense trajectories are extremely inefficient due to higher redundancy of trajectory and background interference susceptibility. While in the case of CNN+OLPSO, the computational complexity is higher. The accuracy attained from the above-mentioned models are 63%, 85%, 91%, 96%, 91.25%, and 98%. The accuracy of these models is 35.5%, 13.5%, 7.5%, 2.5%, 7.25%, and 0.5% are lower than LCNN-SO respectively. Some performance metrics like precision, recall, F-measure is also computed for learning approaches like Naive Bayes (NB), multilayer perceptron’s, k-star, CNN with OLSVM, and LCNN-SO model respectively as in *Table 5*.

Table 2 Layers of AlexNet and GoogleNet

	AlexNet			**GoogleNet**		
Layer	**Function**	**Filter size**	**Stride**	**Function**	**Filter size**	**Stride**
Conv 1	Convolution	11×11×3	4	Convolution	11×11×3	4
Pool 1	Max pooling	3×3	2	Max pooling	3×3	2
Conv 2	Convolution	5×5×48	1	Convolution	5×5×48	1
Pool 2	Max pooling	3×3	2	Max pooling	3×3	2

	AlexNet			GoogleNet		
Conv 3	Convolution	3×3×256	1	Convolution	3×3×256	1
Conv 4	Convolution	3×3×192	1	Convolution	3×3×192	1
Conv 5	Convolution	3×3×192	1	Convolution	3×3×192	1
Pool 5	Max pooling	3×3	2	Max pooling	3×3	2

5.Discussions

The accuracy of NB, MLP, k-star, CNN + Online Algorithm for SVM (OLSVM), is 30.5%, 20.5%, 16.5%, and 0.5% which is less than the LCNN-SO model. Similarly, the precision of these approaches is 31.8%, 20.8%, 16.6%, and 0.24%, which is less than the proposed LCNN-SO model. The recall values are 32.5%, 21.16%, 17.1%, and 0.4%, which is less than the LCNN-SO model. Then, the F-measure of LCNN-SO is 33.5%, 21.2%, 17.3%, and 0.23% higher than other models. Most of the existing models adopt the MNIST database, chars 74K, and Urdu handwritten dataset which is available online for free access. In most of these cases, pixel features and geometric features are considered. Here, the proposed DL model considers the image pixels for computation purposes. The accuracy and error rate of the existing models are given where the error rate of CNN + Stochastic Gradient Descent, CNN+Adam, CNN+Adagrade, CNN+Adadelta is 0.22%, 0.19%, 0.54%, 0.22%, and 0.18% respectively. However, the accuracy of CNN (char 74K+GoogleNet), CNN (Urdu handwritten dataset + AlexNet), CNN+OLSVM is 88%, 77%, 98%, which is 10.5%, 21.5%, and 0.5% lesser than the proposed LCNN+SO model respectively. *Figure 5* to *Figure 11* shows the graphical representation of various performance measures.

The recognition rate of 0-9 digits were 98.9%, 99.2%, 98.3%, 99.5%, 98.6%, 99.6%, 96.5%, 99.67%, 99.5%, and 98.5 respectively; while the error rates were 0.23, 0.54, 0.57, 0.59, 0.42, 0.89, 0.36, 0.78, 0.12, and 0.99 for digits 0-9 respectively. The execution time was reduced for the proposed LCNN-SO as shown in *Table 6*. The complexity of this model was reduced due to the elimination of the pre-processing stages. *Table 7* and *Table 8* explain the training and validation accuracy along with the loss measurements. Here, 10 epochs were considered for training and validation accuracy.

Table 3 Accuracy computation

S. No	Existing Vs Proposed	Accuracy (%)
1	CNN + LSTM	63%
2	Pre-trained CNN + MLP	85%
3	Pre-trained CNN + LSTM	91%
4	Pre-trained CNN + SVM	96%
5	Dense trajectories with HoG	91.25%
6	CNN + OLPSO	98%
7	LCNN-SO	98.5%

Table 4 Performance metrics results

Methods	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)
NB	68%	67.2%	67%	66%
MLP	78%	78.2%	78.34%	78.3%
k-star	82%	82.4%	82.4%	82.2%
CNN + OLSVM	98%	98.76%	99.1%	99.27%
LCNN-SO	98.5%	99%	99.5%	99.50%

Table 5 Comparison of CNN architecture for handwritten digit recognition

Approach	Database	Features	Error rate and accuracy
CNN+Stochastic gradient descent	MNIST	Pixel	0.22%
CNN+Adam	MNIST	Pixel	0.19%
CNN+Adagrad	MNIST	Pixel	0.54%
CNN+Adadelta	MNIST	Pixel	0.22%
Deep learning	MNIST	Pixel	0.18%
Simple CNN	Chars 74k	Pixel	88% (GoogleNet)
CNN	Urdu handwritten dataset	Pixel and geometric	77% (AlexNet)
CNN + OLSVM	MNIST	Pixel	98%

Approach	Database	Features	Error rate and accuracy
LCNN+SO	MNIST	Pixel	98.5%

Table 6 Error rate and execution time comparison

Digits	Recognition rate	Error rate	Execution time (S)
0	98.9	0.23	4
1	99.2	0.54	3
2	98.3	0.57	5
3	99.5	0.59	3
4	98.6	0.42	2
5	99.6	0.89	6
6	96.5	0.36	4
7	99.67	0.78	3
8	99.5	0.12	2
9	98.5	0.99	5

Table 7 Training accuracy and loss measurements

Epochs	1	2	3	4	5	6	7	8	9	10
Training accuracy	84.6	91.03	91.65	95.48	93.74	99.1	92.56	94.13	96.59	97.13
Training loss	53.6	28.45	28.97	14.6	15.69	14.78	13.64	14.85	9.86	9.58

Table 8 Validation accuracy and loss measurements

Epochs	1	2	3	4	5	6	7	8	9	10
Training accuracy	91.63	94.3	97.8	95.8	97.6	97.8	99.6	99	99.3	98.5
Training loss	25.6	18.9	9.47	13.8	7.38	5.86	3.56	1.55	1.65	1.56

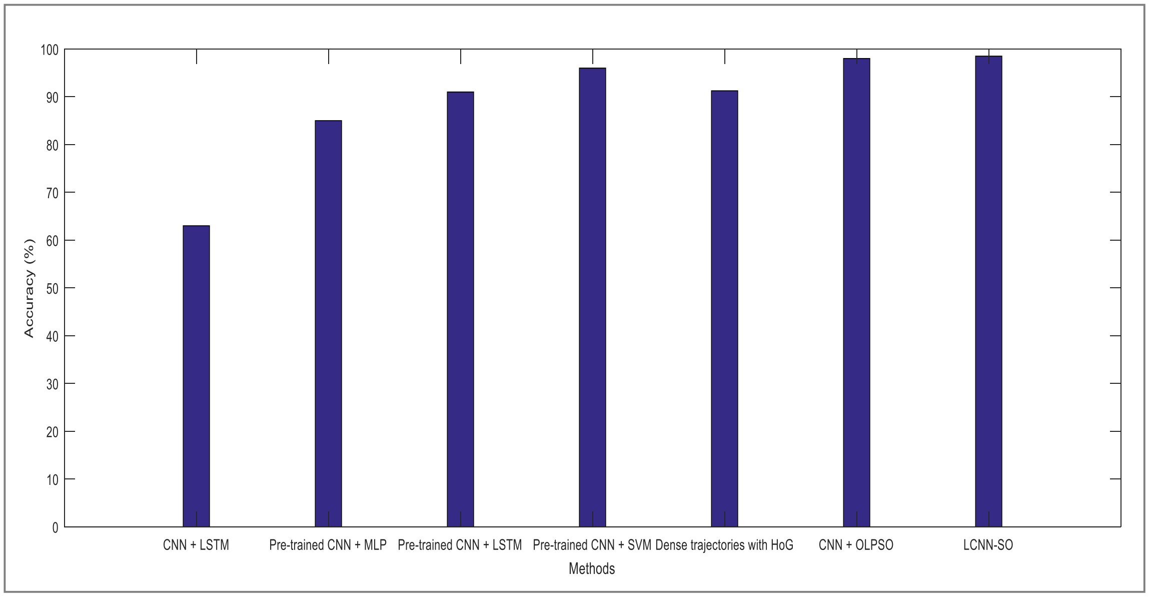

Figure 5 Accuracy comparison

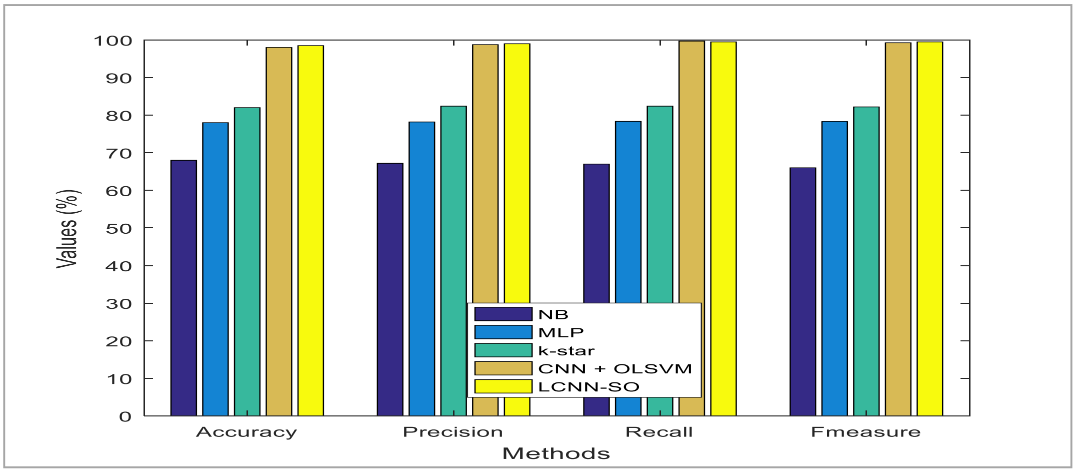

Figure 6 Performance metrics comparison

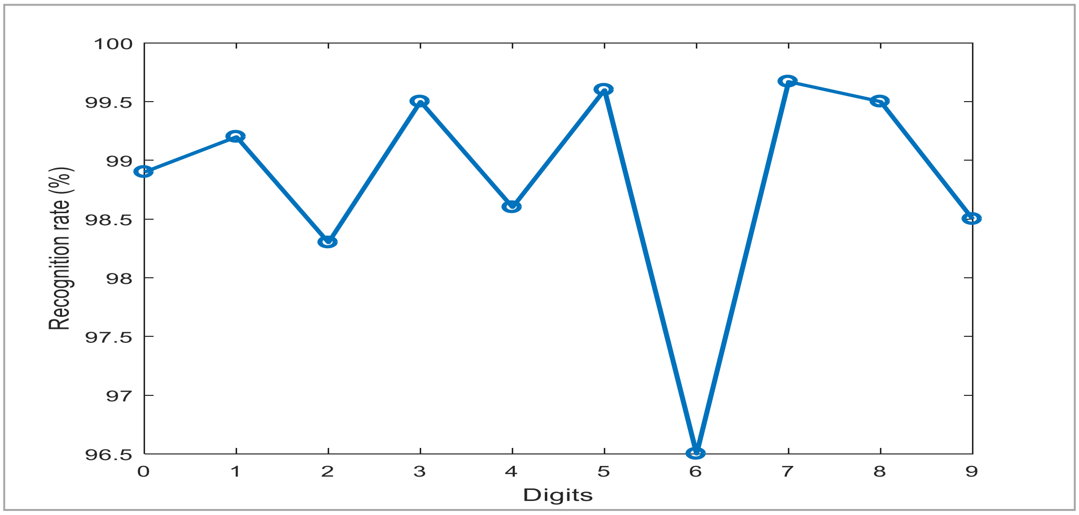

Figure 7 Recognition rate

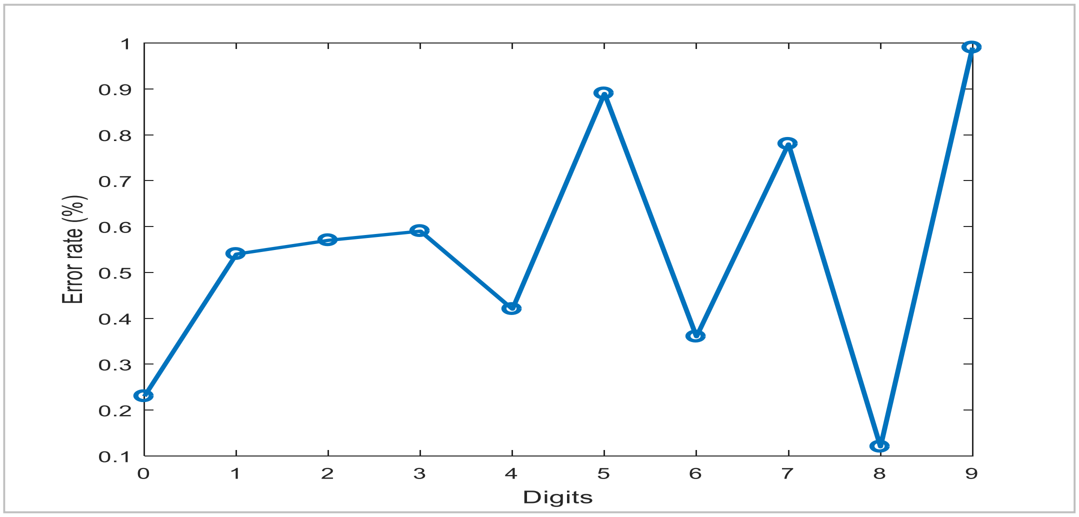

Figure 8 Error rate computation

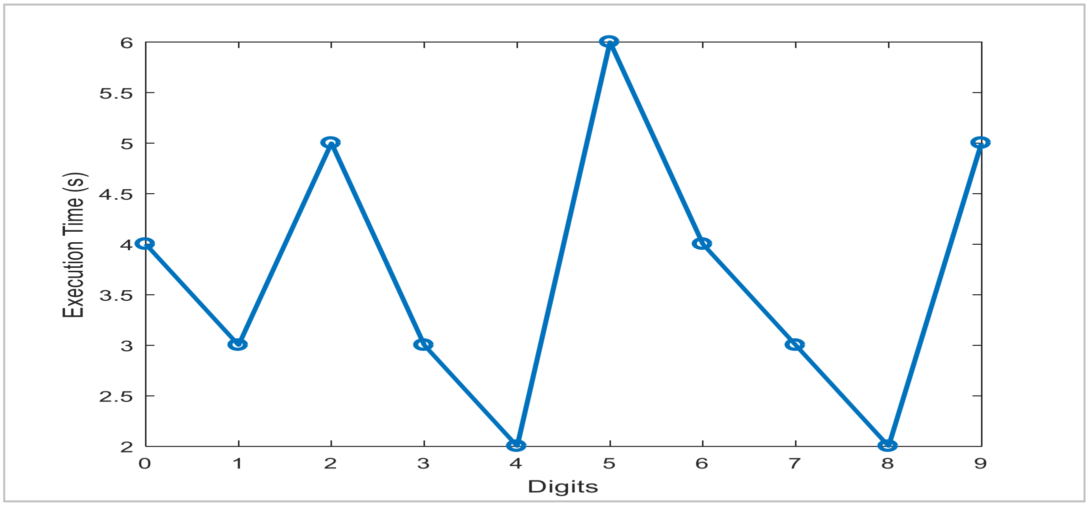

Figure 9 Execution time (Sec) computation

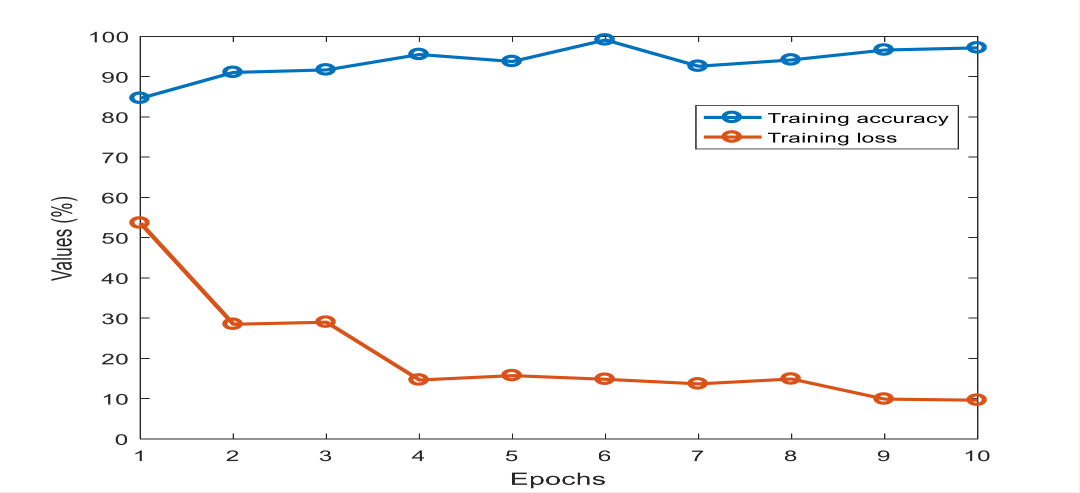

Figure 10 Training accuracy and loss computation

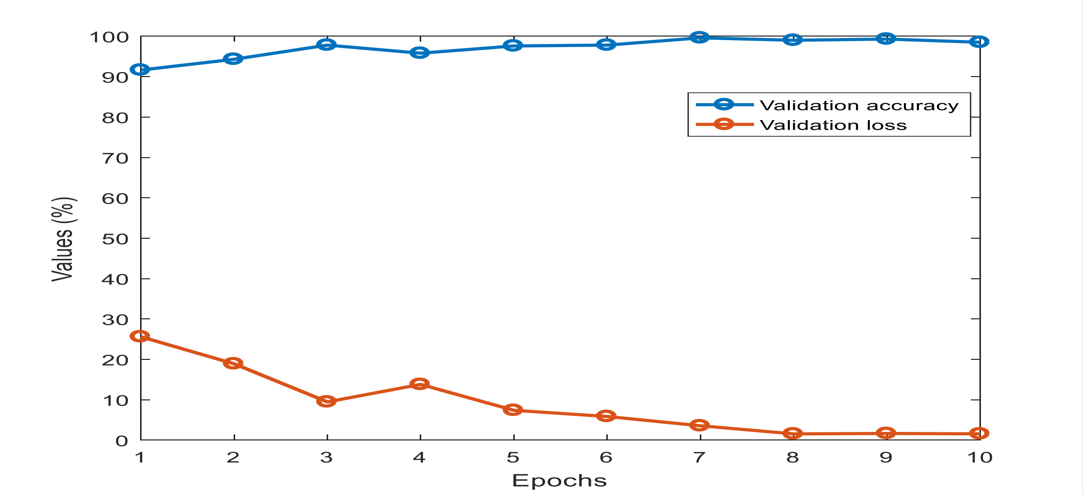

Figure 11 Validation accuracy and loss computation

6.Conclusion and future work

This research work discussed a novel handwritten digit recognition system known as LCNN for predicting isolated and overlapped digits without higher computational complexity. Here, the preprocessing stages are ignored to reduce the complexity of prediction. The given CNN performs optimization with Squirrel Optimizer (SO) thereby producing better accuracy in digit recognition. Exploration and exploitation are managed by the SO algorithm. Thus, the error rate was reduced by execution time. The input samples were taken from Special Database 1 and Special Database 2 (MNIST) and were given to the classifier model. This model avoids training of separate classifiers to categorize isolated and overlapped digits. The experimentation was done with MATLAB simulation environment which compares the output with various existing approaches like CNN+LSTM, pre-trained CNN+MLP, pre-trained CNN+LSTM, pre-trained CNN+SVM, Dense trajectories with HoG, and CNN+OLPSO respectively. The prediction accuracy of the proposed LCNN-SO model is 98.5%, which is higher than other approaches. The proposed LCNN-SO model gives a better trade-off in contrast to these above-mentioned models. In future, hybrid optimizers may be used for still improving the prediction accuracy and lower the error rate in lesser execution time. The major limitation of this proposed model is the adoption of the same dataset for evaluation as there is no proper dataset for digital handwritten data.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

References

[1] Sampath AK, Gomathi N. Fuzzy-based multi-kernel spherical support vector machine for effective handwritten character recognition. Sādhanā. 2017; 42:1513-25.

[2] Bhowmik S, Malakar S, Sarkar R, Basu S, Kundu M, Nasipuri M. Off-line Bangla handwritten word recognition: a holistic approach. Neural Computing and Applications. 2019; 31:5783-98.

[3] Tamen Z, Drias H, Boughaci D. An efficient multiple classifier system for Arabic handwritten words recognition. Pattern Recognition Letters. 2017; 93:123-32.

[4] Sahoo S, Nandi SK, Barua S, Bhowmik S, Malakar S, Sarkar R. Handwritten bangla word recognition using negative refraction based shape transformation. Journal of Intelligent & Fuzzy Systems. 2018; 35(2):1765-77.

[5] Vellasques E, Oliveira LS, Britto AS, Koerich AL, Sabourin R. Filtering segmentation cuts for digit string recognition. Pattern Recognition. 2008; 41(10):3044-53.

[6] Ribas FC, Oliveira LS, Britto AS, Sabourin R. Handwritten digit segmentation: a comparative study. International Journal on Document Analysis and Recognition. 2013; 16(2):127-37.

[7] Ciresan D. Avoiding segmentation in multi-digit numeral string recognition by combining single and two-digit classifiers trained without negative examples. In international symposium on symbolic and numeric algorithms for scientific computing 2008 (pp. 225-30). IEEE.

[8] Hochuli AG, Oliveira LS, Britto AS, Sabourin R. Handwritten digit segmentation: is it still necessary? Pattern Recognition. 2018; 78:1-11.

[9] Viola P, Jones MJ. Robust real-time face detection. International Journal of Computer Vision. 2004; 57(2):137-54.

[10] Zhang K, Zhang Z, Li Z, Qiao Y. Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters. 2016; 23(10):1499-503.

[11] Coates A, Ng AY. Learning feature representations with k-means. In neural networks: tricks of the trade 2012 (pp. 561-80). Springer, Berlin, Heidelberg.

[12] Lawrence S, Giles CL, Tsoi AC, Back AD. Face recognition: a convolutional neural-network approach. IEEE Transactions on Neural Networks. 1997; 8(1):98-113.

[13] Tan X, Chen S, Zhou ZH, Zhang F. Recognizing partially occluded, expression variant faces from single training image per person with SOM and soft k-NN ensemble. IEEE Transactions on Neural Networks. 2005; 16(4):875-86.

[14] Zhao Z, Zhang X, Fang Y. Stacked multilayer self-organizing map for background modeling. IEEE Transactions on Image Processing. 2015; 24(9):2841-50.

[15] Dasgupta J, Bhattacharya K, Chanda B. A holistic approach for off-line handwritten cursive word recognition using directional feature based on arnold transform. Pattern Recognition Letters. 2016; 79:73-9.

[16] Jino PJ, Balakrishnan K, Bhattacharya U. Offline handwritten Malayalam word recognition using a deep architecture. In soft computing for problem solving 2019 (pp. 913-25). Springer, Singapore.

[17] Barua S, Malakar S, Bhowmik S, Sarkar R, Nasipuri M. Bangla handwritten city name recognition using gradient-based feature. In proceedings of the 5th international conference on frontiers in intelligent computing: theory and applications 2017 (pp. 343-52). Springer, Singapore.

[18] Sueiras J, Ruiz V, Sanchez A, Velez JF. Offline continuous handwriting recognition using sequence to sequence neural networks. Neurocomputing. 2018; 289:119-28.

[19] Bhowmik TK, Parui SK, Roy U. Discriminative HMM training with GA for handwritten word recognition. In

international conference on pattern recognition 2008 (pp. 1-4). IEEE.

[20] Saha C, Faisal RH, Rahman MM. Bangla handwritten digit recognition using an improved deep convolutional neural network architecture. In international conference on electrical, computer and communication engineering 2019 (pp. 1-6). IEEE.

[21] http://yann.lecun.com/exdb/publis/index.html#lecun-98. Accessed 10 March 2021.

[22] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998; 86(11):2278-324.

[23] Lu Y, Yi S, Zeng N, Liu Y, Zhang Y. Identification of rice diseases using deep convolutional neural networks. Neurocomputing. 2017; 267:378-84.

[24] Guo Y, Liu Y, Oerlemans A, Lao S, Wu S, Lew MS. Deep learning for visual understanding: a review. Neurocomputing. 2016; 187:27-48.

[25] Vernes K. Gliding performance of the northern flying squirrel (Glaucomys sabrinus) in mature mixed forest of eastern Canada. Journal of Mammalogy. 2001; 82(4):1026-33.

[26] Bag S, Harit G. A survey on optical character recognition for Bangla and Devanagari scripts. Sadhana. 2013; 38(1):133-68.

[27] Elleuch M, Maalej R, Kherallah M. A new design based-SVM of the CNN classifier architecture with dropout for offline Arabic handwritten recognition. Procedia Computer Science. 2016; 80:1712-23.

[28] Jauhiainen T, Lui M, Zampieri M, Baldwin T, Lindén K. Automatic language identification in texts: a survey. Journal of Artificial Intelligence Research. 2019; 65:675-782.

[29] Kumar M, Jindal MK, Sharma RK, Jindal SR. Character and numeral recognition for non-Indic and Indic scripts: a survey. Artificial Intelligence Review. 2019; 52(4):2235-61.

[30] Al-Thubaity A, Alkhalifa A, Almuhareb A, Alsanie W. Arabic diacritization using bidirectional long short-term memory neural networks with conditional random fields. IEEE Access. 2020; 8:154984-96.

[31] Mahmood Z, Safder I, Nawab RM, Bukhari F, Nawaz R, Alfakeeh AS, et al. Deep sentiments in Roman Urdu text using recurrent convolutional neural network model. Information Processing & Management. 2020; 57(4).

[32] Husnain M, Saad MMM, Mumtaz S, Jhanidr MZ, Coustaty M, Muzzamil LM, et al. Recognition of Urdu handwritten characters using convolutional neural network. Applied Sciences. 2019; 9(13):1-21.

[33] Parija B, Pattanayak R, Tripathy SK. Faults classification in a microgrid using decision tree technique and support vector machine. In second international conference on green computing and internet of things 2018 (pp. 145-8). IEEE.

[34] Bahrani A, Majidi B, Eshghi M. Coral reef management in persian gulf using deep convolutional neural networks. In international conference on pattern recognition and image analysis 2019 (pp. 200-4). IEEE.

[35] Bossaghzadeh A. Improving persian digit recognition by combining deep neural networks and SVM and Using PCA. In international conference on machine vision and image processing 2020 (pp. 1-5). IEEE.

[36] Wu S, Wei W, Zhang L. Comparison of machine learning algorithms for handwritten digit recognition. In international symposium on intelligence computation and applications 2017 (pp. 532-42). Springer, Singapore.

T. Senthil was born at Namakkal in 1988. He received his Bachelor's in Electronics and Communication Engineering under Anna University, Chennai. He also received his Master's degree in Applied Electronics under Anna University, Chennai. At present he is working as Assistant Professor in Bannari Amman Institute of Technology. Sathyamangalam, Erode, Tamilnadu. His areas of interest are Image Processing, Machine Learning and Communication. He has published many articles in international journals and authored book chapters relating to his research field.
Email: t.senthilece@gmail.com

Dr. C. Rajan received his Bachelor's degree in B.Tech. Information Technology under Anna University, Chennai. He also received his Master's and Ph.D. degree under Anna University, Chennai. At present he is working as Associate Professor in K.S Rangasamy College of Technology, Tiruchengode, Namakkal, Tamilnadu. His areas of interest are Soft Computing, Network security and Wireless Networks. He has authored many Books, Book chapters and research articles. He is an editor in many reputed journals and is working as editor for many Journals.
Email: rajancsg@gmail.com

J. Deepika was born at Namakkal district during the year 1988. She received her Bachelor's in computer science and Engineering under Anna University, Chennai. She also received her Master's degree in Software Engineering under Anna University, Chennai. At present she is working as Assistant Professor in Bannari Amman Institute of Technology. Sathyamangalam, Erode, Tamilnadu. Her areas of interest are Big Data, Machine Learning and Data Mining and Analytics. She has authored Book chapters and contributed her research field through many research articles. She has acted as reviewer in many conferences.
Email: deepi.realmail@gmail.com