International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6531
***Article Received**: 24 January 2023 **Revised**: 26 February 2023 **Accepted**: 05 March 2023*

A Novel Approach for Optimization of Convolution Neural Network with Particle Swarm Optimization and Genetic Algorithm for Face Recognition

Kalaiarasi P[1], P. Esther Rani[2]
[1]Research Scholar, Department of Electronics and Communication Engineering
Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology
Avadi,Chennai 600062, India
[1]Email: vtd603@veltech.edu.in
[2]Professor, Department of Electronics and Communication Engineering
Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology
Avadi, Chennai 600062 India
[2] Email:drpestherrani@veltech.edu.in

Abstract — Convolutional neural networks are contemporary deep learning models that are employed for many various applications. In general, the filter size, number of filters, number of convolutional layers, number of fully connected layers, activation function and learning rate are some of the hyperparameters that significantly determine how well a CNN performs.. Generally, these hyperparameters are selected manually and varied for each CNN model depending on the application and dataset. During optimization, CNN could get stuck in local minima. To overcome this, metaheuristic algorithms are used for optimization. In this work, the CNN structure is first constructed with randomly chosen hyperparameters and these parameters are optimized using Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) algorithm. A CNN with optimized hyperparameters is used for face recognition. CNNs optimized with these algorithms use RMSprop optimizer instead of stochastic gradient descent. This RMSprop optimizer helps the CNN reach global minimum quickly. It has been observed that optimizing with GA and PSO improves the performance of CNNs. It also reduces the time it takes for the CNN to reach the global minimum

Keywords-CNN, Hyperparameter Optimization, Genetic algorithm,Particle Swarm Optimization, Face recognition.

I. INTRODUCTION

In biometric applications, face recognition is one of the most popular and secure methods. Geometric characteristics were used in earlier face recognition systems. As feature vectors of a face, facial features are represented along with geometrical traits like the distances between the mouth, nose, and eyes. These separations signify the recognition of a facial picture [1]. The feature vectors are represented using template-based techniques in addition to geometric characteristics [2]. The study continued with techniques including PCA, LDA, and ICA that had a high rate of recognition [3]. Support Vector Machine (SVM) approach was used as a classifier in neural networks for face recognition when machine learning was first developed. The differences between the same individual and the differences between various people are expressed by this SVM. The performance was improved when compared to the earlier techniques [4]. Today, CNNs perform effectively in a variety of applications, including speech recognition, picture classification, phrase classification, and medical applications. Many researchers have created CNN designs as AlexNet [5], GoogLeNet [6], VGGNet [7], ResNet [8], and Squeezenet [9]. The same network, however, cannot be utilized for various applications. Depending on the dataset and application, the network needs to be modified.

1.1 CNN

This section provides a quick overview of the CNN's overall architecture and operating parameters [10]. Fig. 1 depicts the general architecture of CNN. A common CNN design consists of three layers: the input layer, hidden layer, and output layer. Each pixel in the picture is sent to the input layer's neurons in the first layer, which goes by the name of input layer. Convolutional and pooling layers are contained in the second layer, often known as the hidden layer. The convolutional layer has m × m-sized kernels, which are filters. In this layer, the filters are moved vertically from left to right and horizontally from top to bottom over the input image.

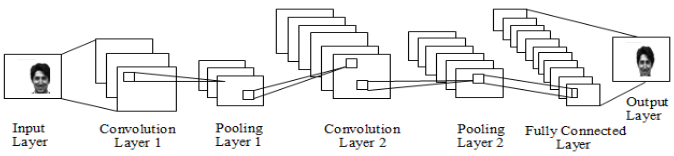

Figure 1: CNN Architecture

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6531
***Article Received**: 24 January 2023 **Revised**: 26 February 2023 **Accepted**: 05 March 2023*

The filter's values are convoluted with the input image's values at every location where it is present. Any one of the following methods can be used to minimize the picture size in the pooling layer: Average pooling and Max pooling. In contrast to Max pooling, which chooses the pixel with the highest value, Average pooling averages the pixel values in the particular convolution area. The picture size is decreased from the input size by this pooling. The new matrix created after pooling is called feature maps. The amount of feature maps is influenced by the employment of filters, which in turn is influenced by the CNN architecture. A stride value that indicates the spacing of filters must be assigned before sliding the filters onto the photos. The size of feature maps is influenced by this stride value. A function called padding is used to prevent the size of feature maps from being reduced in case of inputs with less pixel range. The fully connected layer follows the convolutional layer. Any number of fully connected layers and convolutional layers are possible in a CNN design. The classification layer, which carries out the categorization, is the fully connected layer. Instead of being represented as a matrix in this layer, all of the feature maps are placed horizontally. By giving probability distributions matching to the class that the input image belongs to, this layer flattens the feature maps produced by the previous convolution layer. The classification layer then assigns each input picture to the class that best fits it. In order to increase the effectiveness of the CNN, factors like filter size, filter number, stride size, and activation function must be set.

1.2 Genetic Algorithm

Darwin's idea of evolution influenced the development of the genetic algorithm, a meta-heuristic search method [11]. At first, populations of individuals are created at random, and each individual's fitness function is computed. Based on the application, CNN's optimization problem evaluates this fitness function. Then, the populations of individuals who are eligible are chosen to be the parents so that an offspring can be produced from these parents. Crossover and mutation processes produce these progeny. The produced offspring are currently being assessed for fitness to determine which are qualified to serve as the parent population for the following generation. The GA can be ended after the best solution for the particular problem has been found by repeating these processes [12].

1.3 Particle Swarm Optimization

The stochastic optimization approach known as "particle swarm optimization" (PSO) is based on the social behavior of swarms of insects, fish, birds, and fish, as well as the behavior of fish and other animals. They follow a common strategy for locating food in these swarms. Each swarm adapts its search pattern in response to the learning experiences of both its own and the other swarms in the group. PSO has a local best value and a global best value for each individual swarm, or particle. Particles use these values to calculate the velocity and new solution in the search space. A new dimension, velocity, and location are calculated for each generation [13]. Until the best outcome is obtained, this procedure is repeated.

II. Related Works

In several research projects, the metaheuristic algorithms are employed to optimize the CNN's structure and parameters. A new approach suggested in [14] for autonomously creating CNN architecture for image classification. In order to assure a deeper CNN during evolution and fitness evaluation, an encoding approach was used employing a skip connection in the CNN. This approach, which was tested on benchmark datasets CIFAR 10 and CIFAR 100, decreases the CNN parameter by 10 times and the computing resource to a one percent. In order to configure a CNN, the hyper parameters are manually assigned in [15]. Using the grid search approach and the random search method, the model was evaluated for optimization. The findings demonstrate that random search outperforms grid search in terms of GA optimization. The constructed CNN was tested using datasets and demonstrated accuracy of about 80%. Using a hybrid evolutionary computation technique, [16] developed a shortcut connection in CNN. The term "shortcut connection" refers to the connection of a layer's output to another layer that is not the layer after it. This method enhances deep CNN training.

In [17] face recognition was treated as a two dimensional recognition issue. The variations in the face images are represented by a feature space termed face space, which holds the representation of the facial characteristics. Eigen faces are used to represent these face spaces which include the faces' Eigen vectors. The capacity to learn and identify faces is possessed by these vectors. The average identification rate for a collection of 2500 photos with changes in lighting, orientation, and size was 81.3%. Fisher's linear discriminant approach was suggested in [18] as a way to improve the identification rate under various lighting conditions. With computing requirements akin to those of the Eigen face approach, this technique differentiates the classes of the face picture in low-dimensional space even when illuminated differently. This Fisher face gets a lower error rate of 5.3% and decreases the space in the photos.

The face feature histogram sequences were represented using LBP in [19]. In the experiment, the histogram sequence was categorized, and the sequences were turned into vectors using the Carle square dissimilarity measure and PCA algorithms. The study showed that the PCA method achieved a higher accuracy of 89.5%. In order to achieve greater accuracy compared to LDA, PCA, ICA, and other hybrid methods like

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6531
***Article Received**: 24 January 2023 **Revised**: 26 February 2023 **Accepted**: 05 March 2023*

Gabor-ICA, ICA-SVM, and a select few others, a hybrid method from Gabor and Non-negative matrix factorization was developed [20]. This novel hybrid algorithm had a 95% accuracy rate for the ORL dataset. To identify faces with varied effects, including occlusions, expressions, illuminations, and positions, convolutional neural networks were utilized in [21]. On the FERET dataset, a 4-layered CNN was evaluated, and it outperformed all prior research with an accuracy of 85.13%.

By having a quick learning curve and a robust network for classification, the Deep Convolutional - Optimized Kernel Extreme Learning Machine technique presented in [22] Machine (DC-KELM) was proven to deliver better results faster. Using a polynomial kernel, the hidden layer's output was calculated. In this study, Particle Swarm Optimization was used to improve the classifier KELM's parameters. For the CMU PIE, AT&T, UMIST and Yale datasets, the error rate was determined to be 0.2, 0.5, 21 and 8.89 respectively using the suggested technique. Additionally, less time was needed for training. The velocity operator and a unique encoding approach were optimized in [23] using the PSOCNN. Compared to previous evolutionary systems, the suggested approach was able to converge more quickly and automatically create DCNNs for image classification. When the suggested technique was tested using the MNIST datasets, the error rate was 5.90, which was lower than the rates of the most recent models. Using PSO and the steepest gradient descent algorithm, [24] suggested an automated technique for choosing network structures. MNIST and Kaggle datasets were trained and tested using the automatically built network. Excellent outcomes were attained in both training and testing, it has been noticed. The PSO method was used to optimize an auto encoder in [25]. With less computer resources and no manual input, the PSO algorithm seeks out the ideal architecture. For the MNIST, CIFAR-10, and STL-10 datasets, the improved CNN structure was assessed, and it appears to perform more accurately than the cutting-edge methods.

For sign language recognition the CNN architecture is optimized with PSO to find the optimum parameters in [26]. The convolutional neural network's optimal parameters, utilized during the convolutional process are determined using the PSO method. The research work makes this contribution primarily through these considerations. In the first step, the parameters generated by PSO are maintained under the same conditions in each convolutional layer, and the classification rate serves as the objective function that PSO evaluates. In the second stage, the PSO generates various parameters for each layer, and the objective function is made up of the recognition rate in addition to the Akaike information criteria; the latter helps to determine the optimum network performance but takes more time. The optimized structure was implemented with the American Sign Language MNIST, Mexican Sign Language Alphabet and the American Sign Language Alphabet. This proposed methodology provides 99% recognition rate which is higher than the state-of-the-art technologies.

Indian Classical dances were classified using a hybrid Particle Swarm Grey Wolf optimized CNN in [27]. This cutting-edge technique was developed to identify the best CNN settings, including size of filters, number of hidden layers, epochs and batch size. The suggested optimized architecture is used to classify eight Indian classical dances using the benchmark datasets MNIST, CIFAR, and Indian Classical Dance (ICD). The proposed strategy increases the model's accuracy from 97.3% on the ICD Dataset to 99.4% and 91.1% on the MNIST and CIFAR benchmark datasets, respectively. In comparison to past techniques, our auto-tuned network increased performance for classifying Indian classical dance forms by 5.6% while decreasing computational expense.

III. METHODOLOGY

A. Optimizing CNN with GA

For the evolutionary process and production of progeny, this GA needs the population's size, its maximum number of generations, its CNN structure, and its faces dataset. The genetic algorithm's framework is shown in Algorithm 1. Populations are first started randomly with the specified number of population sizes. All initialized populations are assessed for the CNN architecture's suitability with the provided dataset. Following a fitness assessment, the eligible population serves as the parents, and genetic procedures like crossover and mutation are used to produce the offspring. The freshly created children that survive for fitness are now updated as the parents of the following generation. Up until the maximum number of generations is achieved, this process is repeated.

1) Population Initialization: In this stage, the specific hyper parameters of the CNN architecture are used to produce the individual populations at random. The chromosomes of an individual are represented by factors like learning rate, momentum, convolutional filter, convolutional layer, and weight decay. Fig. 2 depicts chromosome of an individual as an example. The finer points of population initialization are shown in Algorithm 2. This approach initializes the ideal values for the hyperparameters based on the training of the image dataset on the CNN using the values specified for the parameters as listed in Table 1.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6531
***Article Received**: 24 January 2023 **Revised**: 26 February 2023 **Accepted**: 05 March 2023*

Algorithm 1: Framework of GA

Input: A standard CNN structure, Size of the population, Maximum number of generations, Face image dataset

Output: A CNN structure with suitable initial weights

P_i = *Individual Populations of given size*
$G \leftarrow 0$
while G < *Maximum number ofgenerations*
do
investigate the fitness of individuals
randomly select individuals as parents
generate offspring from parents
evaluate offspring
$G \leftarrow G + 1$
End
Return the best individual

C.L	C.F	M	W.D	L.R
5	3	0.8	0.1	0.001

Figure 2: Chromosomes of an Individual

TABLE 1: HYPERPARAMETERS OF GA-CNN

Hyperparameters	Values
Learning rate (L.R)	0.001, 0.01 &0.1
Momentum (M)	0.8, 0.9
Conv. Filters (C.F)	3-5
Conv. Layers (C.L)	3-5
Weight Decay (W.D)	0.1, 0.01, 0.001
Optimizer	SGD, RMS prop

2) Evaluation: In order for the network to give high classification accuracy for face recognition, a fitness function is utilized. The CNN structure is constructed after each population has its fitness assessed. The SGD and RMS prop [28] technique is used to train the resulting network, and the loss function is computed. As a result, both the learning rate and the quantity of epochs are variable. The method for evaluating fitness is described in Algorithm 3.

3) Selection: For genetic operations, the individuals that are assessed for fitness are chosen. From the individual population, two parents are chosen at random. Every generation, the population that has just been created is assessed for fitness and chosen.

4) Offspring Generation: A new offspring is produced via crossover and mutation procedures employing the chosen people as parents. To create the offspring, the parent individuals are divided into two sections and some of the parts are switched. A mutation occurs when one or more bits in an individual are altered, resulting in the creation of an offspring. According to their fitness score, the recently produced children will now serve as the parent population for the following generation. The production of children is summarized in Algorithm 4.

The chromosomes of an individual are the hyperparameters in this case, as already indicated. Chromosomes are divided into two pieces in both of the chosen parents for the crossover surgery, and one component from each parent is switched to create the kid. For switching, a crossing probability is set. When a mutation occurs, a randomly chosen chromosome from the set of hyperparameters in Table 1 has its value modified, creating a kid. The freshly created offspring are now expected to give birth to the generation after them. Until the optimum network design and hyperparameters are determined, this procedure is repeated a maximum number of times.

Algorithm 4: Offspring Generation

Input: The individual populations with fitness score (P_o),
population retained as elite (P_e),
population retained as non-elite (P_r),
Size of the population (P_i),
mutation probability (P_m),

Output: New Population P_n

Arrange all the individual population (P_o) with fitness score in descending order
From the sorted (P_o) add top (P_e) individuals to the new population (P_n)
Select (1- P_r)% from (P_o) and to (P_n)
$P_p \leftarrow P_n$
while $Size(P_n) < P_i$ **do**
Parent1 $\leftarrow$ selects an individual randomly from parents (P_p)
Parent2 $\leftarrow$ selects a individual randomly from parents (P_p)
if $Parent\ 1 \neq Parent\ 2$ **then**
Create Children $Child\ 1$ and $Child\ 2$ from the selected parents using crossover operation
for *each offspring in Child* **do**
$n \leftarrow$ from (0,1) generate a number randomly
if $P_m > n$ **then**
randomly select some value and replace a random gene in offspring
End
End
$P_n \leftarrow P_n U\ Child$
End
End
Return P_n

B. *Optimizing CNN with PSO*

This section presents the PSO algorithm's framework. An evolutionary algorithm called PSO was put out by [29, 30] in

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6531
***Article Received**: 24 January 2023 **Revised**: 26 February 2023 **Accepted**: 05 March 2023*

1995. It includes a number of phases, including particle initialization, fitness assessment, global best, and updating the most recent velocity and weight values in the nearby area. Particle and velocity initialization at random is the first phase of PSO. The second stage involves adding the randomly produced particles to the cost function used to get the global best and local best. Global best is the smallest of local best, while local best has the lowest cost of all particles. Finally, the optimal updates to the velocity and location are made according to formulae (1) and (2).

$$(i+1) = W \times (i) + C_a \times R_a \times (lbest_{nd} - X(i)) + C \times R_b \times (gbest_{nd} - X_{nd}(i)) \quad (1)$$

where i is the iteration, W is the weight, V_{nd} is the n-th particle velocity in dimension d, C_a and C_b are constants, R_a and R_b are random numbers between 0 and 1, l_{best} and g_{best} are local best and global best

$$(i+1) = (i) + (i+1) \quad (2)$$

To solve these drawbacks in CNN, the architecture is improved with the PSO algorithm in such a way that it also optimizes the CNN's hyperparameters. With this the process, the optimal architecture is created, one with a low cost function and high accuracy with quick computing.

In addition to PSO, the CNN architecture for learning employs a separate learning method called RMS Prop. The suggested method outperforms CNN with backpropagation and CNN improved by genetic algorithms in terms of performance. The following algorithm provides the PSO framework.

PSO is the mostly used algorithm to search for a neural network and its hyperparameters by changing the position of the swarm. For the training process using mini batch learning, each particle or individual is encoded as network architecture. RMS prop was utilized for learning rather than SGD. The network's accuracy and cost function are determined after training. As a result, the individual's fitness value is updated in order to find the optimal cost function and accuracy. The following are the optimization procedures used in PSO.

1) Initialization: Initializing the PSO algorithm's parameters, including the no. of particles, the maximum no. of iterations, the co-efficients, and the inertia weight, is the first step. The network's parameters, such as its layers, learning rate, momentum, filters, and weight decay are similarly initialized in a manner similar to that. Together with these parameters, the number of epochs for training and validation are also set.

Algorithm 5: Particle Swarm Optimization

Input: Number of individuals, velocity, position, Number of iterations

Output: Calculate Old Fitness for the Global and Local Best

```
for i= 1 to maximum iteration do,
for i=1 to maximum individual do,
  %Update velocity
  (i + 1) = W × V_nd(i) + C_a × R_a × (lbest_nd
  − X_nd(i)) + C_b × R_b × (gbest_nd −
  X_nd(i)) %Update position
  (i + 1) = (i) + V_nd(i + 1)
  %Evaluate cost function
  New fitness = f(X_nd(i + 1))
      if(New fitness < Old fitness)
            Old fitness = New fitness
              X_nd = (i + 1)
      else
            New fitness = old fitness
            (i + 1) = X_nd
      End if
End for
I = min(New fitness)
lbest = X_nd(I)
End for
```

2) Population Initialization: In this method, the parameters of the network are indicated by the population of the PSO. These parameters are made up of a random number. The network architecture and associated hyperparameters are acquired once the population has been started. Now, training and validation processes are used to assess the produced architecture.

3) Evaluation: The RMS prop technique was used to train the network once it had been started, with a mini batch of data from the entire dataset. The validation dataset, which is not part of the training dataset, is used to evaluate the network after it has been trained. The examination involves calculating each individual's scores (fitness). After the scores have been determined, the prior score from a previous generation is compared with the present score. The local best is updated with the new score if the current score is higher than the old one. Similar to this, the best individual score from previous generations is compared to the current individual score to determine the global best. If the current best score is higher than the prior best score, the current score becomes the global best. The velocity and position of the people are updated with these local and global best.

4) Termination: The aforementioned three processes are repeated til the optimum network structure is produced or up

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6531
***Article Received**: 24 January 2023 **Revised**: 26 February 2023 **Accepted**: 05 March 2023*

to the number of iterations that can be accommodated. The algorithm comes to an end after the requirement has been met.

IV. RESULTS AND DISCUSSIONS

Three databases are employed for experimentation: Faces94 [31], ORL [32], and Yale [33]. Many researchers are assessing CNN on face recognition and classification using these datasets. There are 3000 photos in the Faces94 dataset, 150 classes with 20 images each. Each picture is 180 by 200 pixels and is an RGB image. ORL database has 10 images of 112 X 92 in size, totally 400 images of 40 people in the ORL database. Yale's database has 165 pictures of 15 people, each having 11 pictures at a resolution of 320 by 243. The split between the training and testing portions of the dataset is 80:20. MATLAB 2020A is the tool used for this work.

A. GA Optimized CNN

Three convolutional layers and two fully linked layers make up CNN architectural model. ReLu function was employed as the activation function. In order to reduce the size of the input pictures, pooling layers were utilized in between convolutional layers. In this study, the SGDM and RMSprop algorithms are used separately to train the CNN. RMSprop method yields superior results to SGDM. This is due to the fact that, in the case of RMSprop, the square root of the gradients is taken into account when determining the global minimum. This produces faster convergence, which offers high accuracy and takes minimal time.

The optimal CNN architecture's hyperparameters, determined by GA for the provided dataset, are shown in Table 2. They include learning rate of 0.01, momentum of 0.8, weight decay factor of 0.001, three convolutional layers, and two fully connected layers. When the learning rate is set to 0.01 and the Mini batch size is 64, the network offers high accuracy. The network needs 5 epochs to attain its greatest level of accuracy, and even when the number of epochs was increased, there was no additional rise in accuracy. The following criteria are established for GA: Population size of 20, maximum generations is 10, crossover and mutation probability as 0.8 each.

TABLE 2: HYPERPARAMETERS OF PSO-CNN

Hyperparameters	Range of values
Learning rate	0.01
Momentum	0.8
Conv. Layers	3
FC Layer	2
Weight Decay	0.001
Optimizer	SGD, RMS prop

The performance of the CNN optimized using GA is shown to produce better results when trained using the RMSprop approach. The results in Table 3 show that CNN with GA outperforms a conventional CNN with backpropagation. Even in CNN-GA, the network trained with RMSprop performs well across the board in terms of accuracy, error, and time consumption.

The performance of the proposed methodology compared to the various existing models is shown in Table 4. Eigen faces and Fisher faces are simple models with low complexity for testing and training. These techniques offer additional features with really less performance.

TABLE 3: PERFORMANCE OF CNN AND CNN-GA TRAINED BY SGDM AND RMSPROP

Network	Optimizer	Accuracy (%)	Error	Time (sec)
CNN	SGD	90.6	0.734	1532
	RMS Prop	92.3	0.481	738
CNN-GA	SGD	94.6	0.362	438
	RMS Prop	96.6	0.134	326

TABLE 4: PERFORMANCE OF CNN-GA WITH VARIOUS EXISTING ALGORITHMS

Algorithms		Accuracy (%)	
	ORL Dataset	Yale Dataset	Faces94 Dataset
PCA [15]	79.6	-	-
LBP+PCA [14]	89.5	-	-
PCA+BP [15]	94.5	-	-
Gabor+NMF [15]	95	-	-
Eigen Faces [12]	89.5	77.9	-
Fisher Faces [13]	87.7	85.2	-
CNN [17]	92.6	93.3	-
CNN-GA(Proposed)	99.7	99.3	96.6

B. PSO Optimized CNN

The PSO optimized CNN is assessed using the ORL database and Faces94 dataset, and it is found to be providing improved accuracy and computational time with a lower error rate. Comparing the new approach to CNN with backpropagation and CNN with GA, both of which were trained using SGD and RMS prop, allows for an evaluation of the proposed approach's effectiveness.. Tables 5, 6, 7 and 8 present the performance comparison. As demonstrated in Table 5, the CNN-PSO performs better than the other two models. Although CNN-GA and CNN-PSO perform similarly, the accuracy of CNN-PSO for RMS prop exceeds with 97.3% for Faces94 dataset. In Fig. 3, 4 and 5 models' accuracy, error, and time comparisons are displayed. The effectiveness of the suggested model compared to current techniques is displayed in Table 9 for ORL dataset and Fig. 6 shows the graphical representation of the same. For assessing the models in earlier publications, datasets with fewer photos were employed. For the purpose of evaluating the proposed model, 3000

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6531
***Article Received**: 24 January 2023 **Revised**: 26 February 2023 **Accepted**: 05 March 2023*

image Faces94 dataset is employed in addition to the ORL dataset.

TABLE 5: PERFORMANCE OF CNN, CNN-GA AND CNN-PSO

Network	Optimizer	Accuracy (%)	Error	Time (sec)
CNN	SGD	90.6	0.734	1532
	RMS Prop	92.3	0.481	738
CNN-GA	SGD	94.6	0.362	438
	RMS Prop	96.6	0.134	326
CNN-PSO	SGD	94.6	0.327	410
	RMS Prop	97.2	0.122	306

When it comes to datasets with fewer photos, like ORL and Yale, CNN with backpropagation also extracts more features and offers excellent accuracy, but when it comes to cases where there are more datasets, the recognition rate drops. For ORL and Yale datasets, the proposed model CNN optimized with GA offers an accuracy of 99.7% and 99.3% respectively. The proposed model provides a high recognition rate with little computing effort, even for a dataset like faces94.

TABLE 6: ACCURACY COMPARISON FOR DIFFERENT EPOCHS

Epochs	Algorithm		
	CNN-BP	CNN-GA	CNN-PSO
1	84.7	87.3	89.9
2	86.8	91.2	92.1
3	90.2	92.9	93.8
4	92.9	94.5	95.2
5	94.3	96.6	97.3

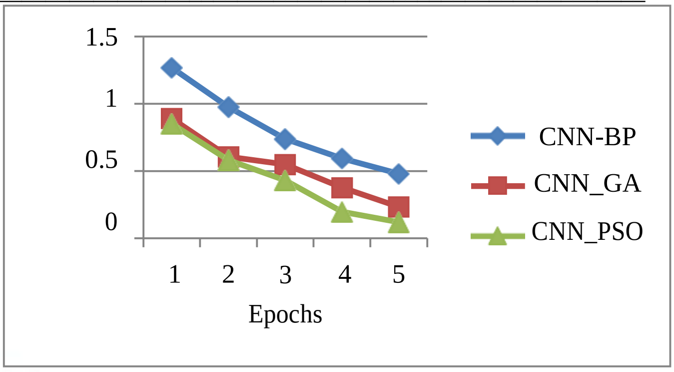

Figure 3: Accuracy Comparison

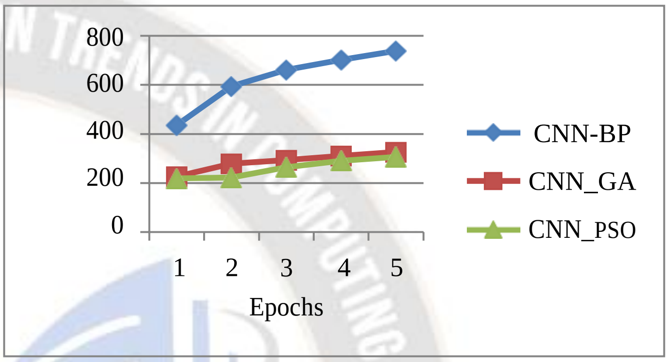

Figure 4: Error Comparison

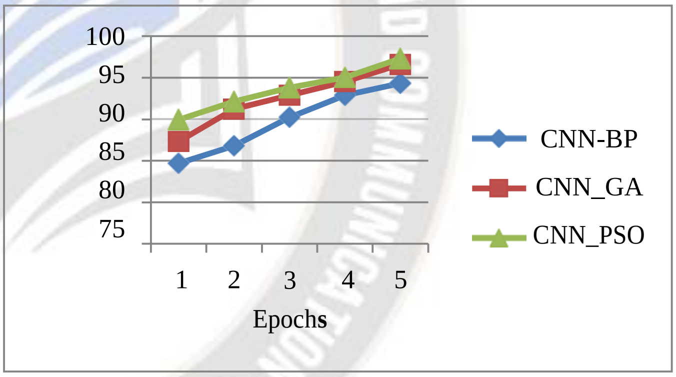

Figure 5: Time Comparison

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6531
***Article Received**: 24 January 2023 **Revised**: 26 February 2023 **Accepted**: 05 March 2023*

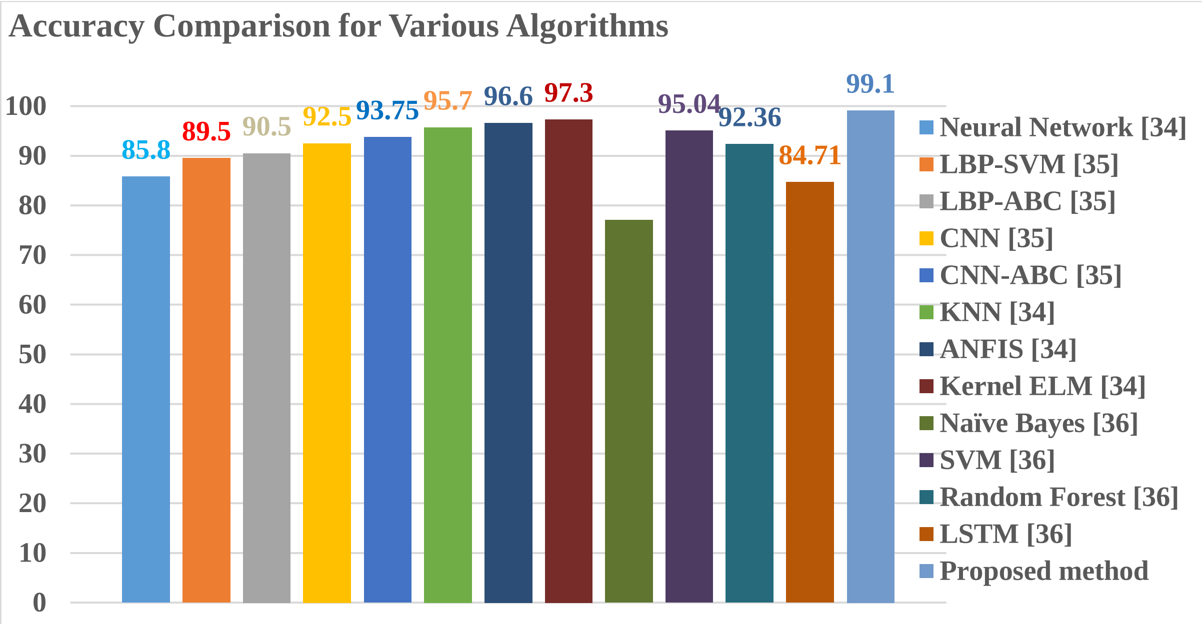

Figure 6: Accuracy Comparison of proposed with existing Algorithms

TABLE 9: PERFORMANCE OF CNN-PSO WITH EXISTING METHODS

Methods	Accuracy (%)
	ORL Dataset
Naive Bayes [36]	77.02
LSTM [36]	84.71
Neural Network [34]	85.8
LBP-SVM [35]	89.5
LBP-ABC [35]	90.5
Random Forest [36]	92.36
CNN [35]	92.5
CNN-ABC [35]	93.75
SVM [36]	95.04
KNN [34]	95.7
ANFIS [34]	96.6
Kernel ELM [34]	97.3
CNN-PSO (Proposed)	99.1

V. CONCLUSIONS

In this research, a novel method for CNN was investigated through GA and PSO optimization. CNN-BP performance is compared to that of the suggested approach for the SGD and RMS Prop learning algorithms. From the findings of the research, the CNN-PSO approach for the faces94 dataset works well in terms of accuracy, error rate, and time consumption. The accuracy of the proposed approach outperforms that of the existing techniques when it is compared to other algorithms for the ORL dataset. In the future, various datasets with a greater number of images may be utilized for evaluation, and the CNN may be improved for performance using a hybrid PSO-GA technique.

REFERENCES

[1] V. Vijayakumari, "Face Recognition Techniques: A Survey," World Journal of Computer Application and Technology, 1(2), 41-50, 2013.

[2] A. V. Nefian and M. H. Hayes, "Hidden markov models for face detection and recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 1, 141-145, 1999.

[3] G. Carvajal, W. Valenzuela and M. Figueroa, "Subspace-based face recognition in analog VLSl," Proceedings of the 20th Neural Information Processing Systems, 225-232, Vacouver, 2007.

[4] L. Juwei, K. N. Plataniotis, L. D. Hannon, M. K. lilian, M. R. Lasch and P. F. Ramig, "Machine identification of human faces," Pattern Recognition, 13(2), 97-110, 1981.

[5] A. Krizhevsky, I. Sutskever and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems, 1097-1105, 2012.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al, "Going deeper with convolutions," Proceedings CVPR IEEE, 1–9, 2015.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Proceedings International Conference on Learning Representations, 2015.

[8] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," Proceedings CVPR IEEE, 2016.

[9] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally and K. Keutzer," SqueezeNet: AlexNet-level accuracy with 50× fewer parameters and <0.5MB model size," Proceedings CVPR IEEE, arXiv:1602.07360 [cs.CV], 2016.

[10] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," The handbook of brain theory and neural networks, 3361(10). 1995.
[11] J. H. Holland, "Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence," MIT press, 1992.
[12] P. Esfahanian and M. Akhavan M, "GACNN: Training Deep Convolutional Neural Networks with Genetic Algorithm," Neural and Evolutionary Computing. arXiv: 1909.13354, 2019.
[13] Mei-Ping Song and Guo-Chang Gu "Research on particle swarm optimization: a review". Proceedings of International Conference on Machine Learning and Cybernetics, Vol.4. (IEEE Cat.No.04EX826). pp. 2236- 2241. Shanghai. China. DOI: 10.1109/ICMLC.2004.1382171, 2004.
[14] Y. Sun, B. Xue, M. Zhang, G. G. Yen, J. Lv, "Automatically Designing CNN Architectures Using the Genetic Algorithm for Image Classification," IEEE Transactions on Cybernetics, 50(9), pp.3840-3854. DOI: 10.1109/TCYB.2020.2983860, 2020.
[15] N. M. Aszemi, and P. D. D. Dominic, "Hyperparameter Optimization in Convolutional Neural Network using Genetic Algorithms," International Journal of Advanced Computer Science and Applications. 10(6). DOI:10.14569/IJACSA.2019.0100638, 2019.
[16] B. Wang, Y. Sun, B. Xue and M. Zhang, "A Hybrid GA-PSO Method for Evolving Architecture and Short Connections of Deep Convolutional Neural Networks," Proceedings CVPR IEEE, 2019.
[17] M. A. Turk and A. P. Pentland, "Face recognition using eigen faces," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Maui, HI, USA, pp. 586-591, doi: 10.1109/CVPR.1991.139758, 1991.
[18] P. N. Belhumer, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," IEEE Transactions on Pattern Analysis and Machine Intelligence. 19(7). pp. 711-720. DOI: 10.1109/34.598228, 1997.
[19] W. Ge, W. Quan and C. Han, "Face description and identification using histogram sequence of local binary pattern," Seventh International Conference on Advanced Computational Intelligence (ICACI, Wuyi, China. pp. 415-420, DOI: 10.1109/ICACI.2015.7184743, 2015.
[20] F. Purnomo, D. Suhartono, M. Shodiq, A. Susanto, S. Raharja and W. Kurniawan, "Face recognition using Gabor Wavelet and Non-negative Matrix Factorization," SAI Intelligent Systems Conference (IntelliSys), London, UK, pp. 788-792. DOI: 10.1109/IntelliSys.2015.7361230, 2015.
[21] F. Radzi, M. Khalil-Hani, S. S. Liew and R. Bakhteri, "Convolutional Neural Network for Face Recognition with Pose and Illumination Variation," International Journal of Engineering and Technology, 6(1), 2014.
[22] T. Goel and R. Murugan, R, "Classifier for Face Recognition Based on Deep Convolutional – Optimized Kernel Extreme Learning Machine," Computers & Electrical Engineering, Vol.85, 2020, DOI:10.1016/j.compeleceng.2020.106640. 2020.
[23] F. Erivaldo, F. Junior and G. G. Yen, "Particle swarm optimization of deep neural networks architectures for image classification. Swarm and Evolutionary Computation, vol.49, 2019, DOI: 10.1016/j.swevo.2019.05.010. 2019. pp 62-74.
[24] F. Ye,"Particle swarm optimization-based automatic parameter selection for deep neural networks and its applications in large-scale and high-dimensional data," PLoS ONE, 12(12), 2012.
[25] Y. Sun, B. Xue, M. Zhang and G. G. Yen, "A Particle Swarm Optimization-Based Flexible Convolutional Autoencoder for Image Classification," IEEE Transactions on Neural Networks and Learning Systems, 30(8), pp. 2295-2309. DOI: 10.1109/TNNLS.2018.2881143, 2019.
[26] J. Fregoso, C. I. Gonzalez and G. E. Martinez, "Optimization of Convolutional Neural Networks Architectures Using PSO for Sign Language Recognition". Axioms, 10(3), 139, 2021, DOI: 10.3390/axioms10030139.
[27] J. R. Challapalli and N. Devarakonda, "A novel approach for optimization of convolution neural network with hybrid particle swarm and grey wolf algorithm for classification of Indian classical dances," Knowledge and Information System, 64:2411–2434. DOI: 10.1007/s10115-022-01707-3, 2022.
[28] N. M. Aszemi and P. D. D. Dominic, "Hyperparameter Optimization in Convolutional Neural Network using Genetic Algorithms," International Journal of Advanced Computer Science and Applications, 10(6). DOI:10.14569/IJACSA.2019.0100638, 2019.
[29] E. Russell and J. Kennedy, "A new optimizer using particle swarm theory," Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan (4-6 Oct. 1995). pp: 39-43. DOI:10.1109/MHS.1995.494215.
[30] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization," Proceedings of IEEE International Conference on Neural Networks, Perth. Australia. Pp. 1942-1948, 1995.
[31] Faces94 dataset, https://cswww.essex.ac.uk/mv/allfaces/faces94.html.
[32] Yale dataset, http://vision.ucsd.edu/datasets/yale_face_dataset_original/yalefaces.zip.
[33] ORL dataset, https://www.face-rec.org/databases/
[34] S. A. Padmanabhan and J. Kanchikere, "An efficient face recognition system based on hybrid optimized KELM," Multimedia Tools Application, 79, pp: 10677–10697, 2020.
[35] W. Yulin and J. Mingvan, "Face recognition system based on CNN and LBP features for classifier optimization and fusion," The Journal of China Universities of Posts and Telecommunications. 25(1):37-47, 2018.
[36] P. G. Devarakonda and B. Bozic, "Particle Swarm Optimization of Convolutional Neural Network for Human Activity Prediction," Optimization Algorithms of Swarm Intelligence, 2022, DOI: 10.5772/intechopen.97259.