A MPSO-Aided Technique for Robust Recognition Using Deep Convolution Neural Networks and Its Application in Solar Energy Sector

M.Blessy Queen Mary[1], S.G.Sam Stanley[2]
[1]Assistant Professor/Information Technology, Government College of Technology, Coimbatore, Tamil Nadu, India
Email: blessy.stanley123@gmail.com
[2]Associate Professor/Department of Mechanical Engineering, PARK College Of Engineering and Technology, Coimbatore, Tamil Nadu, India
Email: pcetonlineclass@gmail.com
Mary, M. Q. & Stanley, S. (2026). A MPSO-Aided Technique for Robust Recognition Using Deep Convolution

https://doi.org/10.55041/ijsmt.v2i7.110

Cite this Article: Mary, M. Q. & Stanley, S. (2026). A MPSO-Aided Technique for Robust Recognition Using Deep Convolution Neural Networks and Its Application in Solar Energy Sector. International Journal of Science, Strategic Management and Technology, *02*(8), 1-9. https://doi.org/10.55041/ijsmt.v2i7.11

Neural Networks and Its Application in Solar Energy Sector. International Journal of Science, Strategic Management and Technology, *02*(8), 1-9. https://doi.org/10.55041/ijsmt.v2i7.11

Abstract
This article introduces a robust face recognition technique that leverages the micro-batch particle swarm optimization (MPSO) for optimal parameter selection and deep convolution neural networks which lead to considerable improvement in recognition accuracy and minimization of computation cost. The conventional approaches to recognize the pattern, faced the limitations in estimating the recognition accuracy as there is no knob to fine tune the parameters, hence an additional layer of information is provided to augment the deep convolution neural networks (CNN) with the optimized weights gained from micro-batch particle swarm optimization (MPSO) algorithm. The utilization of micro-batch particle swarm optimization aimed to optimize the end results on convolution neural networks to improve the recognition accuracy. The proposed method has been applied to YALE database and the experimental research can be carried out which gained maximum accuracy with minimal execution time and the final result was one step ahead than conventional methods. These methods can also be applied in Solar Energy sector.
Keywords: Face Recognition, Particle Swarm Optimization, Deep Convolution Neural Networks, Principal Component Analysis

1. INTRODUCTION

In recent decades, face recognition has emerged as a prominent field due to its wide range of applications, including biometrics, information security systems, criminal identification, smart cards, image processing, and authentication systems. Face recognition stands out for its non-intrusive nature, allowing images to be captured, processed, and identified even without the subject's knowledge or active involvement.

Research into face recognition for real conditions on related technologies and general evaluation standards began in the 1991s [1]. Despite ongoing advancements, deep neural networks have been effectively used for face recognition, focusing on face detection identification and tracking by PRDIT algorithm to improve system performance [2]. For the ethical application of biometrics, facial recognition technology is used [3]. To optimize space utilization, face recognition techniques have been employed for compressed images [4]. Additionally, genetic algorithm is used for face recognition to assess strengths and identify areas for improvement [5].

The structure of this paper is organized as follows: Section 2 presents description of BFO, SVM and RBF components and Section 3 describes feature selection process of the proposed tuned bacterial foraging algorithm with SVM trained by RBF networks in detail and Section 4 presents the performance evaluation of the proposed algorithm. Then finally, conclusions are drawn in Section 5.

2. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) is a population-based stochastic optimization algorithm motivated by intelligent collective behaviour of some animals such as flocks of birds [6]. Particle Swarm Optimization algorithm learned from animal's activity or behaviour which is used to solve optimization problems. In PSO, each member of the population is called a particle and the population is called a swarm. Starting with a randomly initialized population and moving in randomly chosen directions, each particle goes through the searching space and remembers the best previous positions of itself and its neighbours. Particles of a swarm communicate good positions to each other as well as dynamically adjust their own position and velocity derived from the best position of all particles. The next step begins when all particles have been moved. Finally, all particles tend to fly towards better and better positions over the searching process until the swarm move close to an optimum of the fitness function The PSO method is becoming very popular because of its simplicity of implementation as well as ability to converge to a good solution. It does not require any gradient information of the function to be optimized and uses only primitive mathematical operators. As compared with other optimization methods, it is faster, cheaper and more efficient. PSO is well suited to solve the non-linear, non-convex, continuous, discrete, integer variable type problems.

It has been explored by researchers for its theory, issues and applications. It is improved to solve the best feature compensation coefficient of the face recognition algorithm [7]. In PSO, particles are attracted to the best solution by the combination of attraction as an individual and as any neighbourhood particle as they traverse through the search space. For each entity of particle, there is a neighbourhood particle which is viewed as a subset of particles so that to communicate with the original particle. Euclidean neighbourhood model stood for first PSO model in which the particle distance can be computed to evaluate which one is nearer to ease the process of communication. Since the Euclidean neighbourhood model is computationally less efficient, the topological neighbourhood is introduced which in turn called as 'gbest' model or global neighbourhood. In this model, every particle in the search space is related and acquired instruction from all other particles in an effective way.

A particle in the search space is contained with three vectors: location of the particle in the N-dimensional search space $\vec{x}_i = (x_{i1}, x_{i2}, \dots, x_{iN})$, position where the particle found at its best $\vec{b}_i = (b_{i1}, b_{i2}, \dots, b_{iN})$ and the speed of the particle $\vec{s}_i = (s_{i1}, s_{i2}, \dots, s_{iN})$. After initialization, the particles traverse around the search space and the entire swarm is refined at every stage of time by improving its speed and location of each particle in the N-dimensional space.

$$s_i = s_i + c\epsilon_1(b_i - x_i) + c\epsilon_2(b_g - x_i) \quad (1)$$

$$x_i = x_i + s_i \quad (2)$$

wherec is a constant; ϵ_1 and $\epsilon_2$are independent random numbers provoked at each iteration and $\vec{b}_g$ is the best location observed by the neighbour particle. An efficient artificial neural network is developed to select a block of feature instead of single feature to ensure the distinctiveness and variations of features with application to face recognition systems [8].

2.1 Micro Particle Swarm Optimization (MPSO)

In particle swarm optimization, the number of particles are traversed along the search space to attain the best position which depends on three factors (a) own velocity (b) own past best position (c) global or neighbours past best position. If the total particles used in the particle swarm optimization technique are limited to three or five, then it is called micro particle swarm optimization (MPSO) which has the following properties:

- When a network is characterized by numerous parameters, particle swarm optimization seems difficult to implement and micro-particle swarm optimization will be the only possible alternative.
- Velocity needed in particle swarm optimization depends on the back-propagation output.

2.2 Structure of Particle Swarm Optimization

The flow diagram of particle swarm optimization is depicted in Figure 1. As shown in the Figure, the first step in the optimization process is initialization. After initializing particle position and random velocity, set the constants. The second step is the computation of fitness values. Check whether the values are better than current best position values. If is yes, allot the value as new best position; Otherwise keep the previous value. Set particle's best position to global best position and compute the velocity of each particle. Revision of particle's data values by its velocity value can be done and the process is iterated as the time allowed closing.

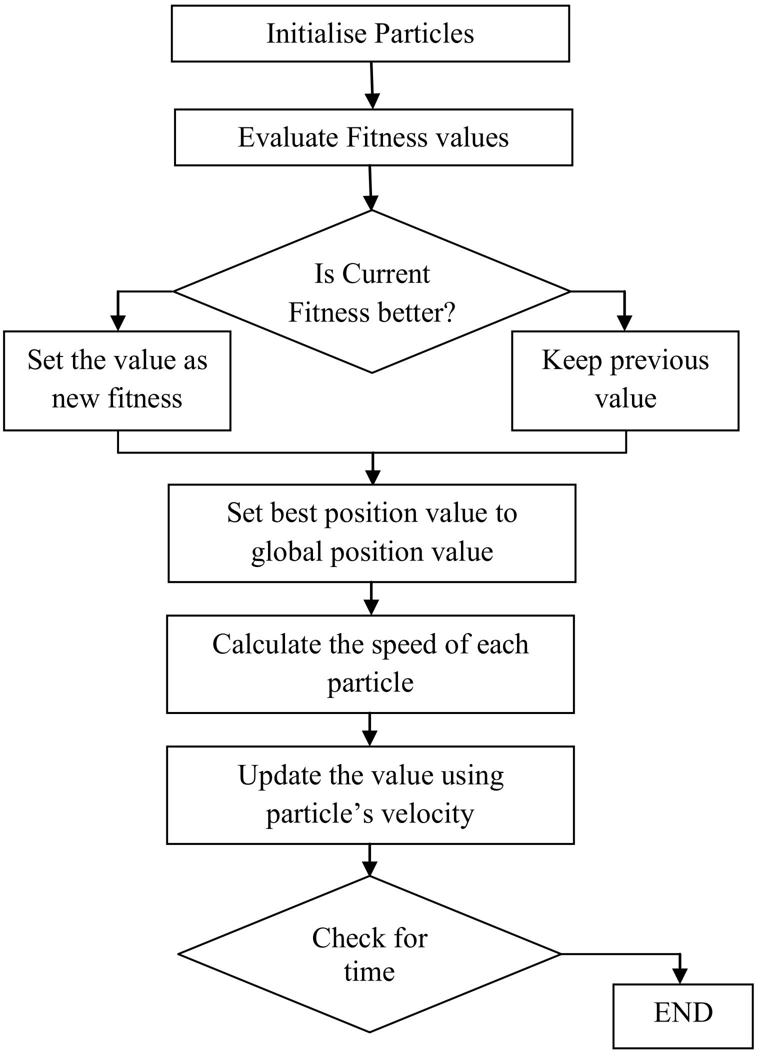

Figure. 1 Flow diagram of Particle Swarm Optimization

3. DEEP CONVOLUTION NEURAL NETWORK

Convolution Neural Network is an extension of multilayer perceptron (MLP) specially designed to meet the challenges in the field of pattern recognition. The performance of convolution neural network in dimensionality reduction and feature extraction are extraordinary well as compared to conventional approaches. The architecture of deep convolution neural network is shown in the Figure 2.

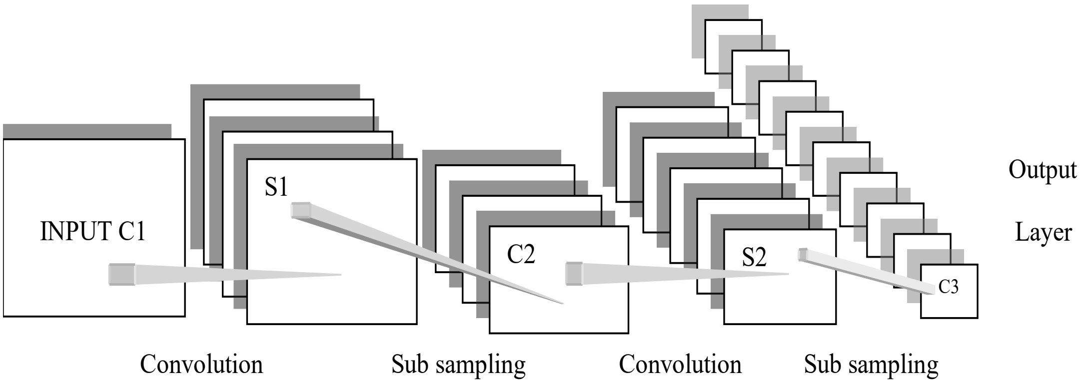

Figure 2 Structure of deep convolution neural network

As shown in Figure 2, the structure of the network is arranged as convolution layers and sub sampling layers. The processing layers hold three convolution layers (C1, C2 and C3) sandwiched between two sub sampling layers (S1 and S2) and an output layer. In case of convolution layer, each neuron has specific connection to the input section in the preceding layer which is served as receptive field. Uniform sharing of weights tends all neurons with related features acquire data from varying input sections. In case of sub sampling layer, the features are down sampled spatially and the map size is reduced by a constant value.

A quantum deep convolution neural network is designed which provided the exponential acceleration in dimensionality reduction [9]. Convolution neural network contributed partial invariance and extracted large features from hierarchical set of layers. This procedure was able to provide fast classification and normalization so that the system exhibited better classification performance than other methods. The block diagram of the system used in the above method is presented in the Figure 3.

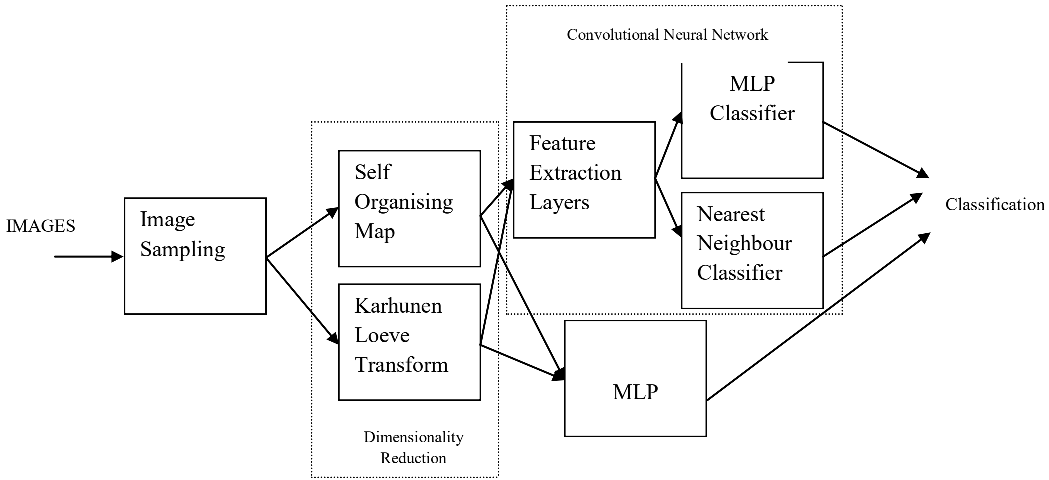

Figure 3 Block diagram of face recognition system using convolution neural network

The optimal architecture of convolutional neural networks is searched using Particle Swarm Optimization (PSO) for enhanced image classification [10]. Optimized hyperparameters of convolutional neural networks were determined by multi-level particle swarm optimization [11]. It was concluded that the results obtained were optimized so that the accuracy can be increased. This can be achieved by selecting the appropriate parameters on deep convolution networks by the use of particle swarm optimization.

3.1 DEEP CONVOLUTION NEURAL NETWORK TRAINING

The training of deep convolution neural network with population based search technique is the main challenge because of the computational raising by a factor with the population size to obtain appropriate values of weights and biases. In general, the training of deep learning networks can be done by back propagation algorithms since it gives estimated partial derivatives of the error with regard to corresponding weights and biases. The convergence can be quickly attained by varying momentum terms with partial derivatives. Based on the complexity, the networks are trained with population based search techniques such as particle swarm optimization or genetic algorithms with mini-batch or micro-batch.

Since the architecture of deep learning networks is complex, the difficulty increases in gradient based approach and convergence of the network. So, there is a need to select suitable training algorithm for such type of architectures.

4. PROPOSED IMAGE CLASSIFICATION TECHNIQUE

A simple deep learning network is designed to classify the images with more accuracy. The essential functions employed for designing the network are principal component analysis (PCA), indexing and pooling. The overall architecture is named as PCA Network (PCANet). Two variants of PCANet are also taken into account to ease the comparison with the same topology but the selection of filters is through random or linear discriminant analysis. Chan et al. (2015) proposed an unsupervised convolutional deep neural network in which its filter has no requirements in regularized parameters and optimization solvers. Moreover this simplicity enables mathematical and computational analysis to verify its effectiveness.

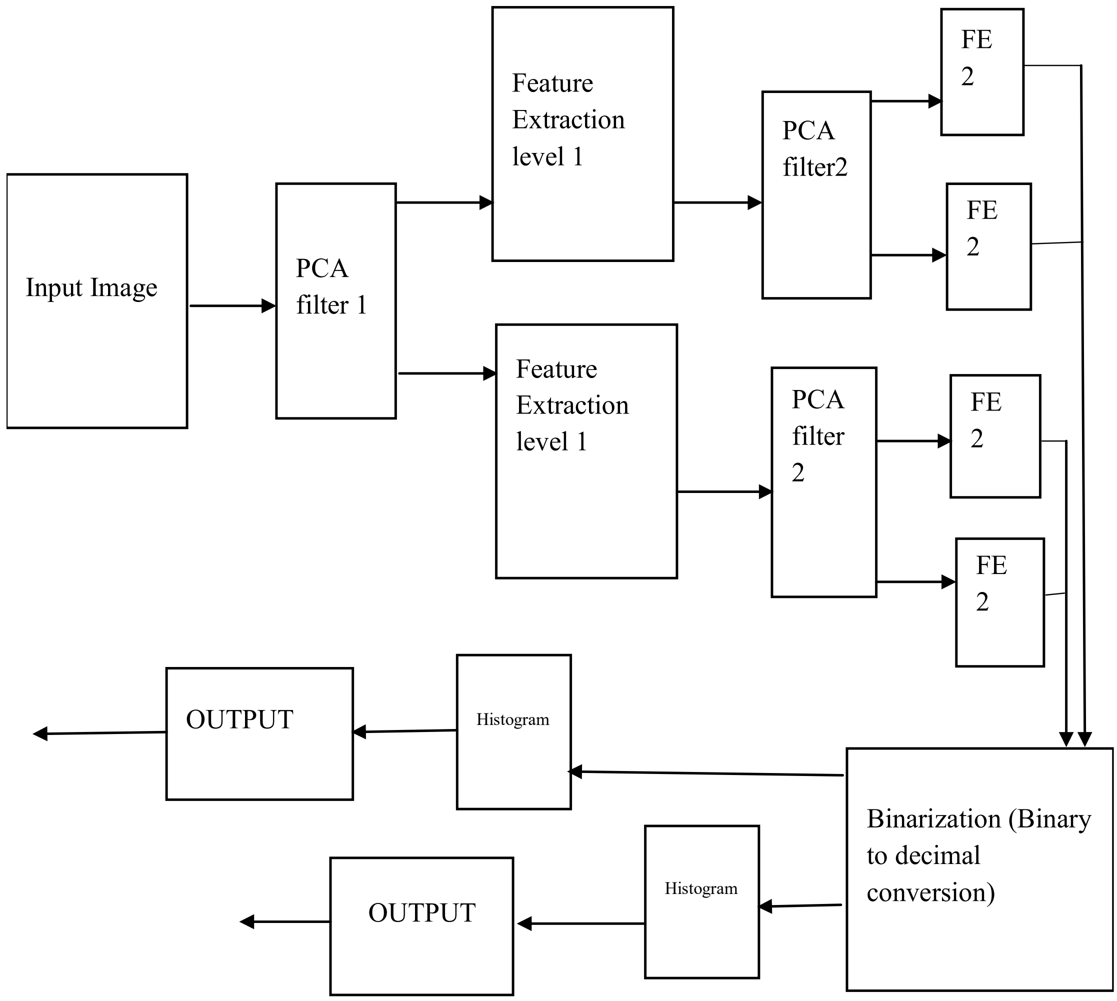

Figure 4 **Illustration of proposed architecture extracts features from an image**

The process of feature extraction from an image in the proposed architecture is illustrated in the Figure 5. The convolution filter in each stage is chosen to be the PCA filters; the nonlinear layer is assumed for quantization and feature pooling while the block-wise histograms of the binary codes are considered as the final output features of the network. The resultant effect of this method allowed the system with robust classification, which in turn leads the system to recognize the images in an effective way.

The proposed deep convolution incorporates the goodness of Oriented Principal Component Analysis (OPCA), thus provide robustness to intra-class variability.

- One of the variants of the network called LDANet is designed in such a way that the PCA filters are replaced with the filters that are learned from linear discriminant analysis (LDA).
- The other variant RandNet is developed with the same architecture as deep network but the filters are substituted by random learning.

Finally feature extraction process can be done with the help of Eigen values using Principal Component Analysis (PCA) and then classification or categorization of face images can be completed through Micro batch particle swarm Optimization trained Back Propagation Neural Network (BPNN) which in turn represents the proposed deep learning neural network.

International Journal of Science, Strategic Management and Technology
Volume 02 Issue 08 August-2026 | ISSN: 3108-1762 (Online) | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

4.1 Step by step procedure of the proposed algorithm

Since Back Propagation Neural Network is a multi-layered feed forward neural network, BPNN learning algorithm adjusts the weights and bias of each of the neurons during the training. This characteristic feature decreases the Mean Square Error (MSE) and is directed to calculate the desired output.

- Assign the value of all weights randomly ranging from -1.0 to1.0.
- Then assign binary input values to each of the neurons in the inputlayer.
- At this step, the activation of each and every neuron can be done by the followingsteps:

○ Multiply weight values of the connections with the output values of the preceding neurons, and summarize thesevalues.

○ After that, pass the result to an activation function, which analyses the output value of thisneuron.

- Repeat all thesteps until the final output layer isachieved.
- Compare the computed output to the desiredoutput and calculate the value of mean squareerror.
- Update the values of all weights in each layer. Go to step1.
- The algorithm ended with the complete matching of output patterns with the input patterns.

The flow diagram of the proposed method is presented in the Figure 6.

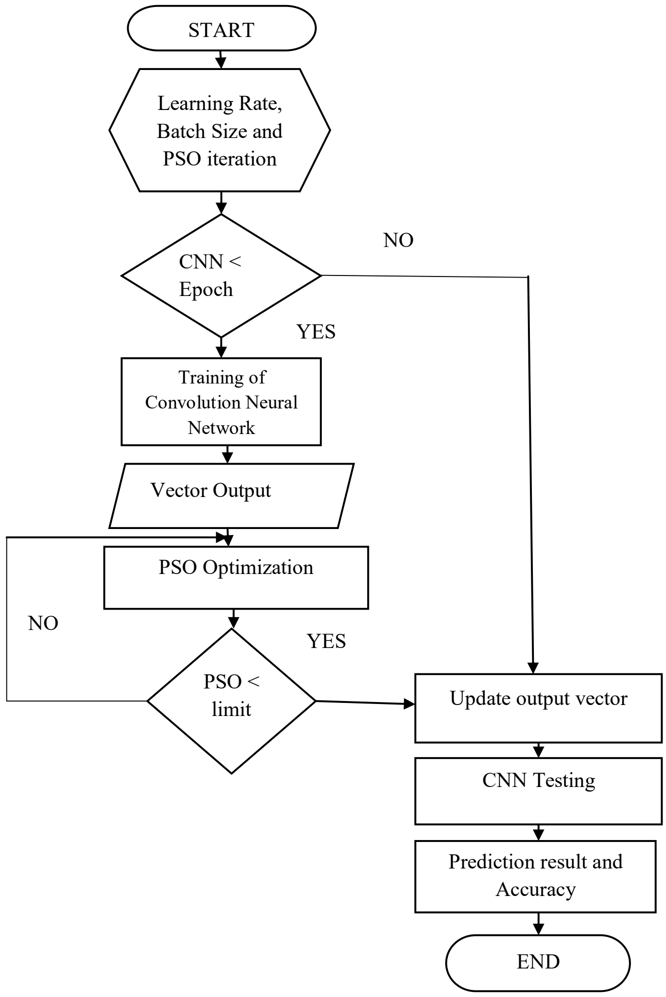

Figure 5 Flow diagram of the proposed method

Advantages

- Advanced designs of multistage features ornetworks thereby increase the accuracy.
- Eliminates the necessity of training data and replace the PCA filters at each layer with random filters of the samesize.
- Enhances the supervision of the learned filters by incorporating the information of class labels in the training data and learn the filters based on the idea of multi-class linear discriminant analysis (LDA).
- BPNN classifier works well on larger scale datasets and when lighting variation is small.

4.2 EXPERIMENTAL ANALYSIS OF PROPOSED SYSTEM

The sample images of face used for this experimental investigation had been taken from ORL, Japanese Female Facial Expression (JAFFE) and Cohn – Kanade (CK) databases. Initially the features of face from the face database are selected. Simple description of the data sets and sample screen shots are given below.

```
 ====== PCANet Parameters =======

PCANet =

          NumStages: 2
          PatchSize: [5 5]
         NumFilters: [40 8]
      HistBlockSize: [8 8]
    BlkOverLapRatio: 0.5000
            Pyramid: [4 2 1]


 ====== PCANet Training =======
Computing PCA filter bank and its outputs at stage 1...
Computing PCA filter bank and its outputs at stage 2...
Extracting PCANet feasture of the 100th training sample...
Extracting PCANet feasture of the 200th training sample...

Accuracy up to 986 tests is 57.71%; taking 0.23 secs per testing sample on average.
Accuracy up to 987 tests is 57.65%; taking 0.23 secs per testing sample on average.
Accuracy up to 988 tests is 57.59%; taking 0.23 secs per testing sample on average.
Accuracy up to 989 tests is 57.53%; taking 0.23 secs per testing sample on average.
Accuracy up to 990 tests is 57.58%; taking 0.23 secs per testing sample on average.
Accuracy up to 991 tests is 57.62%; taking 0.23 secs per testing sample on average.
Accuracy up to 992 tests is 57.56%; taking 0.23 secs per testing sample on average.
Accuracy up to 993 tests is 57.60%; taking 0.23 secs per testing sample on average.
Accuracy up to 994 tests is 57.55%; taking 0.23 secs per testing sample on average.
Accuracy up to 995 tests is 57.49%; taking 0.23 secs per testing sample on average.
Accuracy up to 996 tests is 57.43%; taking 0.23 secs per testing sample on average.
Accuracy up to 997 tests is 57.37%; taking 0.23 secs per testing sample on average.
Accuracy up to 998 tests is 57.41%; taking 0.23 secs per testing sample on average.
Accuracy up to 999 tests is 57.46%; taking 0.23 secs per testing sample on average.
Accuracy up to 1000 tests is 57.50%; taking 0.23 secs per testing sample on average.

 ===== Results of PCANet, followed by a linear SVM classifier =====
     PCANet training time: 136.97 secs.
     Linear SVM training time: 21.41 secs.
     Testing error rate: 42.50%
     Average testing time 0.23 secs per test sample.
```

Figure 6 PCANet Training

From the figure 7, it is clear that the training time of the network is 137 s and average testing time for each test sample is 0.23 s.

ORL: It was developed by AT&T Laboratories, Cambridge, U.K and is contained with 400 images of 40 distinct persons. The images were taken at different time instances with high degree variability in lights, face expressions and different postures. Sample of face images are shown in the Figure 8.

Japanese Female Facial Expression (JAFFE): This database contains 213 images of 7 facial expressions posed by 10 Japanese female models structured by Michael Lyons, Miyuki Kamachi and JiroGyoba in Kyushu University. An example is given in Figure 9.

Cohn – Kanade (CK): It was created by Kanade, Cohn &Tian (2000) and there are 486 Facial Action Coding systems (FACS) of image sequences from 97 subjects. Each image sequence is composed of images of varying expressions starting from neutral and proceeds to target expression. Example of experimental images is presented in the Figure 10.

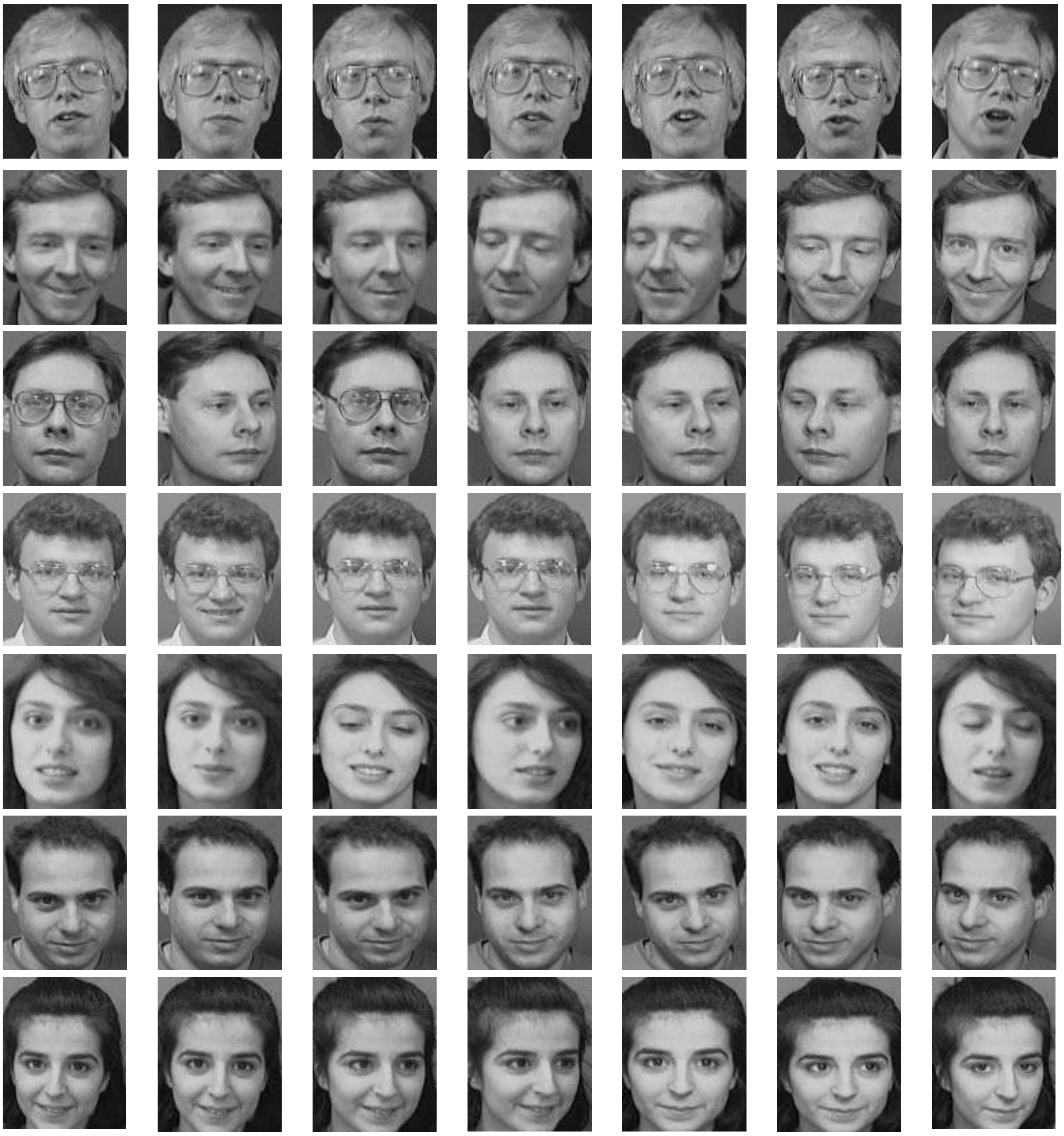

Figure 8 Sample face images with different expressions, posture and illumination (ORL)

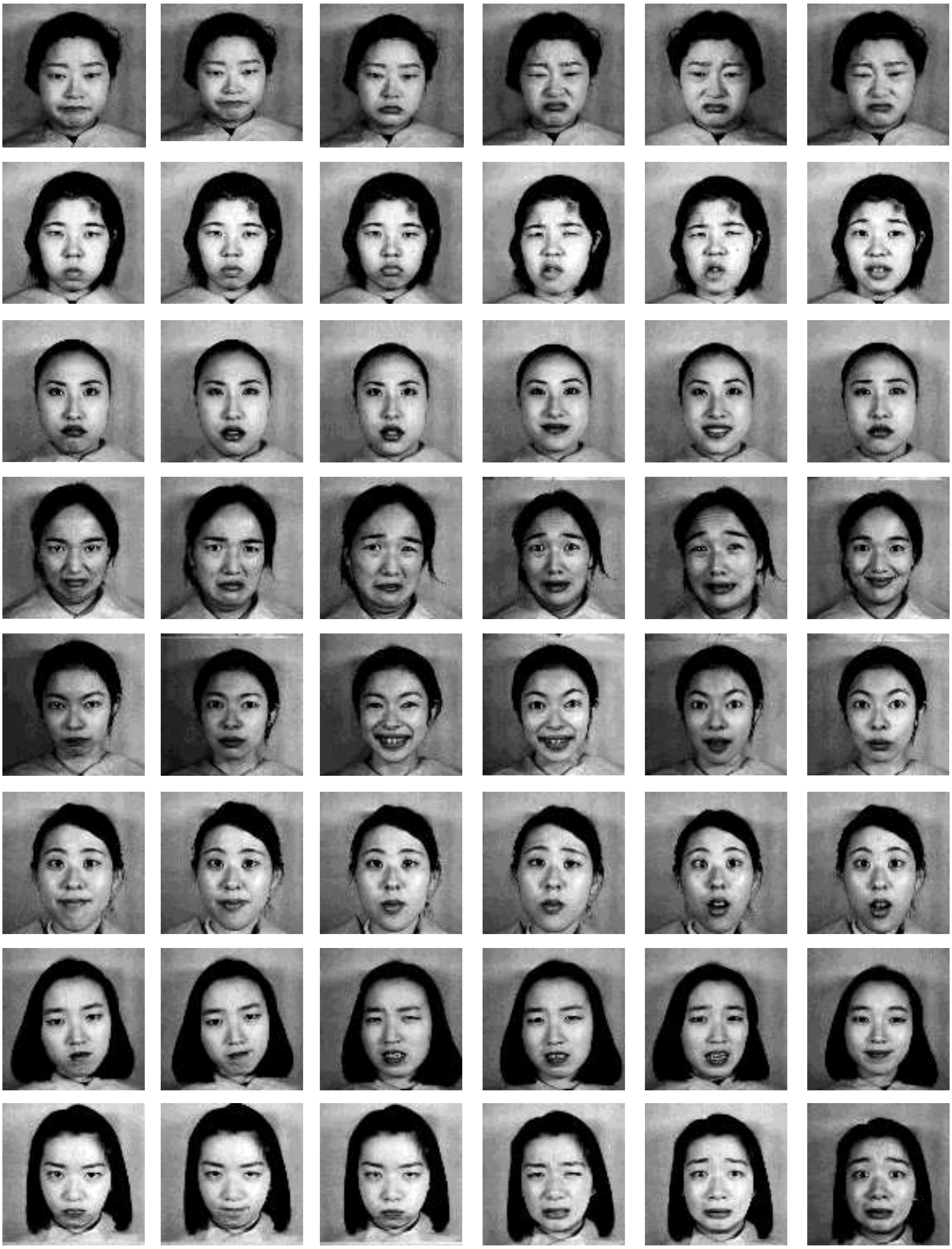

Figure 9 **Samples of face images with different expressions, posture under same illumination (JAFFE)**

International Journal of Science, Strategic Management and Technology
Volume 02 Issue 08 August-2026 | ISSN: 3108-1762 (Online) | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

Figure 70 Samples of face images with different expressions, posture and illumination (COHN-KANADE)

Recognition Accuracy (first stage): When experimenting with the above face datasets for recognition accuracy, the number of filters can be made to vary in two stages. The recognition rate with varying number of filters in the first stage is depicted in the Figure 11. In the first stage, the number of filters can be differed by a factor of 2. From the Figure, it is clear that as the number of filters increased, the recognition rate of the proposed method (PCANet) is also increased and reached 98% as maximum recognition rate. As compared to other methods such as LDANet and RandNet, the proposed method proved as optimum.

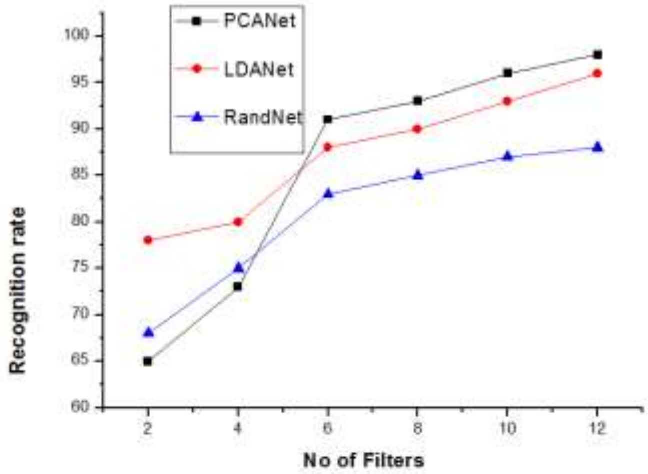

Figure 81 Recognition rate of proposed method with varying number of filters (First stage)

Recognition Accuracy (second stage): The recognition rate with varying number of filters in the second stage is depicted in the Figure 12. In the second stage, the number of filters can be differed by a factor of 5. From the figure, it is clear that as the number of filters increased, the recognition rate of the proposed method (PCANet) is also increased and reached a maximum of 99.3% as recognition rate. The proposed method can be compared to other methods such as LDANet and RandNet. It is obvious that the proposed method performed extremely well when it is made to compete with other methods.

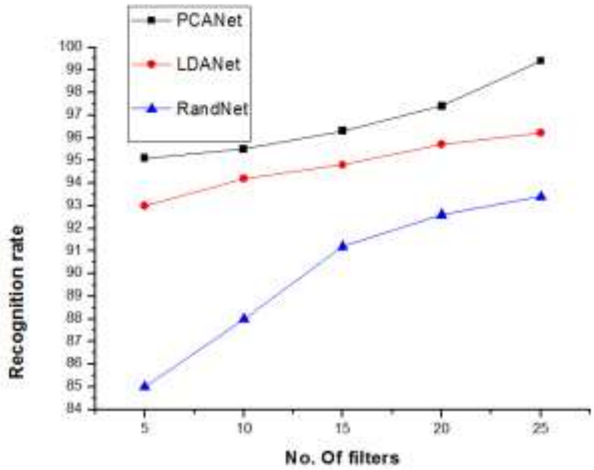

Figure 92 Recognition rate of proposed method with varying number of filters (Second stage)

Error rate (first stage): When experimenting with the above face datasets to determine error rate, the number of filters can be made to vary in two stages. The error rate with varying number of filters in the first stage is depicted in the Figure 13. In the first stage, the number of filters can be differed by a factor of 2. From the Figure, it is clear that as the number of filters increased, the error rate of the proposed method (PCANet) is decreased and reached a minimum of 0.7%. As compared to other methods such as LDANet and RandNet, the proposed method exhibit minimum error rate and hence proved it as optimum.

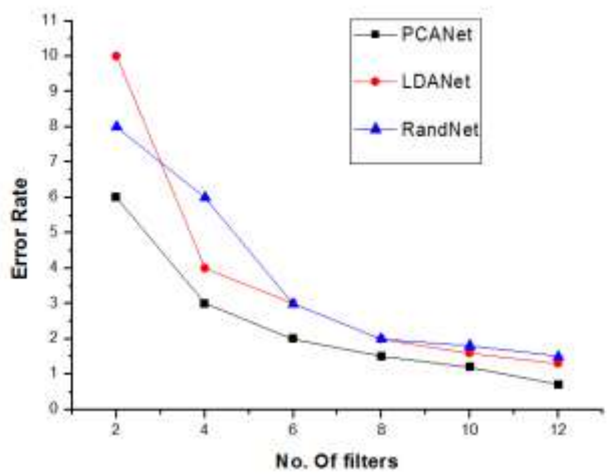

Figure 103 Error rate of proposed method with varying number of filters (First stage)

Error rate (second stage): The error rate with varying number of filters in the second stage is depicted in the Figure 14. In the second stage, the number of filters can be differed by a factor of 5. From the Figure, it is clear that as the number of filters increased, the error rate of the proposed method (PCANet) is decreased and reached a minimum value. The error rate of proposed method as compared to other methods such as LDANet and RandNet is minimum and thus confirmed itself an optimum algorithm for recognizing images.

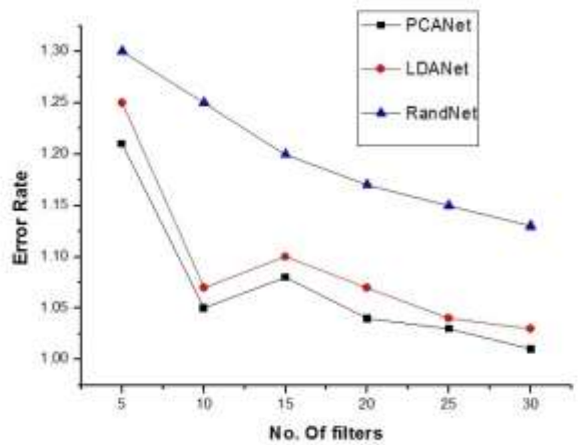

Figure 114 Error rate of proposed method with varying number of filters (Second stage)

5 CONCLUSION

Hence an effective face recognition algorithm using deep convolution neural network and micro particle swarm optimization is experimentally investigated and the reports are comforting as it demonstrated the value of the proposed algorithm. It is apparent that the proposed algorithm improved the recognition accuracy while at the same time decreased the error rate. In this proposed research, the recognition accuracy can be achieved by deep convolution neural network all the way through micro particle swarm optimization algorithm.

REFERENCES

[1]Li, Lixiang, et al. "A review of face recognition technology." IEEE access 8 (2020): 139110-139120.

[2] Manjula, V. S., and L. D. S. S. Baboo. "Face detection identification and tracking by PRDIT algorithm using image database for crime investigation." Int. J. Comput. Appl 38.10 (2012): 40-46.

[3]Smith, Marcus, and Seumas Miller. "The ethical application of biometric facial recognition technology." Ai & Society 37.1 (2022): 167-175.

[4]Quinn, George W., Patrick J. Grother, and George W. Quinn. Performance of face recognition algorithms on compressed images. US Department of Commerce, National Institute of Standards and Technology, 2011.

[5]Zhi, Hui, and Sanyang Liu. "Face recognition based on genetic algorithm." Journal of Visual Communication and Image Representation 58 (2019): 495-502.

[6]Wang, Dongshu, Dapei Tan, and Lei Liu. "Particle swarm optimization algorithm: an overview." Soft computing 22.2 (2018): 387-408.

[7]Zhang, Yanhu, and Lijuan Yan. "Face recognition algorithm based on particle swarm optimization and image feature compensation." SoftwareX 22 (2023): 101305.

[8]Chalabi, Nour Elhouda, et al. "Particle swarm optimization based block feature selection in face recognition system." Multimedia Tools and Applications 80.24 (2021): 33257-33273.

[9]Li, YaoChong, et al. "A quantum deep convolutional neural network for image recognition." Quantum Science and Technology 5.4 (2020): 044003.

[10]Wang, Bin, et al. "Evolving deep convolutional neural networks by variable-length particle swarm optimization for image classification." 2018 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2018.

[11]Singh, Pratibha, Santanu Chaudhury, and Bijaya Ketan Panigrahi. "Hybrid MPSO-CNN: Multi-level particle swarm optimized hyperparameters of convolutional neural network." Swarm and Evolutionary Computation 63 (2021): 100863.