*I.J. Image, Graphics and Signal Processing,* **2022, 3, 32-46**
Published Online June 2022 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijigsp.2022.03.04

# Multi-Module Convolutional Neural Network Based Optimal Face Recognition with Minibatch Optimization

**Deepa Indrawal**
PhD. Research Scholar, Mewar University, Chittorgarh, India
Email: deepaindrawal@gmail.com

**Archana Sharma**
Research Guide, Professor, TIT, Bhopal
Email : er.archna.sharma@gmail.com

Received: 15 February 2022; Revised: 10 March 2022; Accepted: 14 April 2022; Published: 08 June 2022

**Abstract:** Technology is getting smarter day by day and facilitating every part of human life from automatic alarming, automatic temperature, and personalised choice prediction and behaviour recognition. Such technological advancements are using different machine learning techniques for artificial intelligence. Face recognition is also one of the techniques to develop futuristic artificial intelligence-based technology used to get devices equipped with personalised features and security. Face recognition is also used for keeping information of facial data of employees of any company citizens of any country to get tracked and control over crimes in unfair incidents. For making face recognition more reliable and faster, several techniques are evolving every day. One of the fastest and most dependable face recognitions is CNN based face recognition. This work is designed based on the multiple convolutional module-based CNN equipped with batch normalisation and linear rectified unit for normalising and optimising features with minibatch. Faces in CNN's fully connected layer are classified using the SoftMax classifier. The ORL and Yale face datasets are used for training. The average accuracy achieved is 94.74% for ORL and 96.60% for Yale Datasets. The convolutional neural network training was done for different training percentages, e.g., 66%, 67%, 68%, 69%, 70%, and 80%. The experimental outcomes exhibited that the defined approach had enhanced the face recognition performance.

**Index Terms:** Convolutional neural networks, softmax classifier, deep learning, batch normalisation, face recognition.

## 1. Introduction

Face recognition refers to how a visual framework perceives the face of a relevant human. Because of its application in security systems, access control, video surveillance, commercial areas, and even social organisations like Facebook has become a critical human-PC interaction instrument. Face recognition has gained popularity with the rapid growth of artificial intelligence due to its non-judgmental character. When compared to other biometric technologies, it is the most common strategy for human identification. Face recognition can be tested without the awareness of the participant in an uncontrolled environment.

When you look at the historical backdrop of face recognition, you'll notice that it's been addressed in a number of research studies, including [1,3]. Traditional shallow learning strategies have been challenged by barriers such as posture change, facial camouflages, scene illumination, the intricacy of the image background, and variations in facial demeanour, as illustrated in references [3,4]. Shallow learning methods make use of a few basic facial features and rely on artificial intelligence to extract sample features. More complex face traits can be extracted using deep learning-based approaches [5,8]. Deep learning is making significant progress in addressing challenges that have hampered artificial knowledge localisation efforts for a long time. Deep learning is making significant progress in addressing challenges that have hampered artificial intelligence's best efforts for a long time.

It is largely used in image recognition, natural language processing, semantic segmentation, and a number of other real-world scenarios [9,11]. It addresses the challenge of learning hierarchical representations with a single algorithm or a few systems. Three deep learning systems are Stacked Autoencoder, Deep Belief Network (DBN), and Convolutional Neural Network (CNN) [12,13].

In picture and facial recognition, CNN is the most commonly used algorithm. CNN is a machine learning technique that extracts features from input and increases the number of features using a convolutional neural network.

LeCun was the first to suggest CNN, and it was first used in handwriting recognition [14]. His organisation was a true inspiration for many scholars in the subject, as it was the birthplace of a substantial part of the new architecture. They had the top results when distributing their work in the ImageNet Competition [15]. It is primarily considered to be the most influential article in the field of computer vision, demonstrating that CNNs outperform handmade based recognition approaches. Thanks to the introduction of Graphical Processing Units, CNN has also achieved fantastic success in various applications, including semantic segmentation, picture identification, scene recognition, and edge detection.

Main contribution of the work is as follows:

- High accuracy face recognition using adaptive learning based multi module CNN
- With use of CNN the network is more suitable for new data sets to be added at later stages
- Number of convolution layers are less to make training process faster and keep system less complex
- No preprocessing stages, benefits for giving direct input to the CNN
- Direct image input to the system makes it feasible for diverse object datasets

## 2. Material and Methods

### *2.1 Proposed CNN System Model*

CNN usually has maximal usage in designing computer vision-related applications, e.g., image processing. In various recognition applications, the convolution operation is used with the filter to extract the features of the information image (here face). In other words, CNN's are a sort of Neural Network proven to be effective in image classification and identification. CNN's are feed-forward neural networks with several layers. CNN's are channels, bits, or neurons with programmable loads, parameters, and biases. Each track takes a few data sources, performs convolution, and adds non-linearity to the mix [16]. A typical CNN architecture is shown in Figure 1. Convolution, batch normalisation followed by pooling layer, ReLU- Rectified Linear Unit layer, SoftMax layer, ended along with fully connected layer all are mathematical techniques used in the building of CNN.

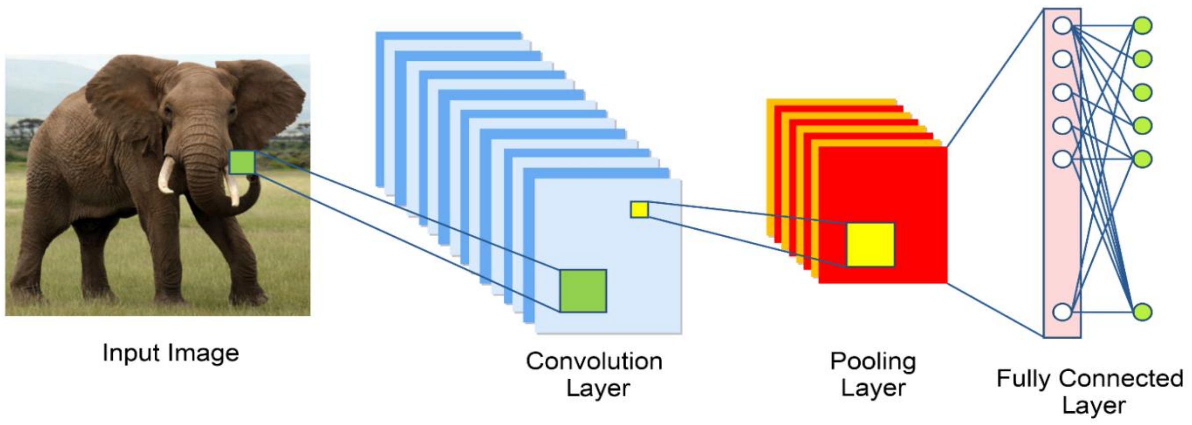

Fig.1. A traditional design for CNNs.

#### *2.1.1 Convolutional Operation:*

The convolutional layer simulates the central structural square of a convolutional network, which is responsible for most of the computing work. The main goal of the convolution layer is to feature extraction from image files. Convolution uses small squares of the face image to learn visual attributes and save the spatial relationship between pixels. A group of learnable neurons are used to tangle the face image. These neurons build a feature map or activation map in the yield image, which is then sent on to the next convolutional layer as information data.

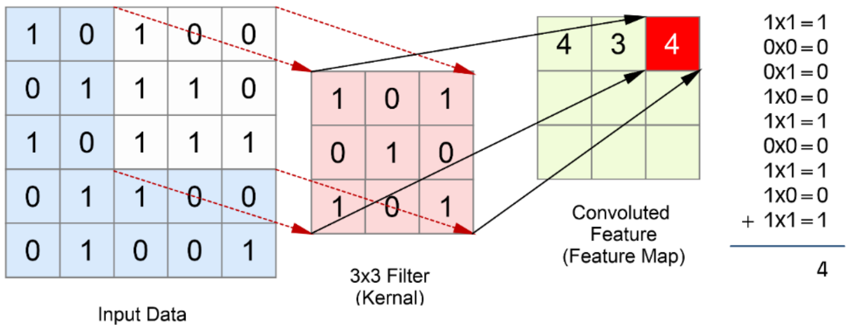

Fig.2. Operation of Convolution

The convolution operation to calculate the correlation between the kernel weights and the image pixels (or previous layer output) is shown in Figure 2. It is an element-wise multiplication operation followed by a summation between the kernel feature map and the input, as shown in the equation below.

The output of the convolution layer is formed from an input of size *N, H, W, D, Cin*, where *H* = height, *W* = width, *D* = depth, *C* = number of channels, and *N* = batch size.

$$Out\left(N_i, C_{out_j}\right) = b\left(C_{out\ j}\right) + \sum_{k=0}^{C_{in}-1} \ weight\ \left(C_{out\ j}, k\right) * \ input\ (N_i, k) \tag{1}$$

The trainable kernels are $l \times l \times l$ matrices that slide over the massive input to find meaningful patterns by building new feature maps and convolving feature maps from the preceding layer, where * is the cross-correlation operation (3D cross-correlation in our example) between two signals.

The kernel weights are trained using backpropagation to retrieve significant structures from given information, which is the idea of this procedure. The weights of a kernel are shared spatially across all points in the image. A kernel characteristic relevant for one part of an image is likewise suitable for other image sections. This is a crucial feature of CNNs, reducing the number of parameters. Practically all widely used deep learning packages perform it as a matrix operation rather than individual element-wise multiplication and summing.

### *2.1.2 Batch Normalization Operation:*

Batch normalisation is a data normalisation method that helps to improve the performance and speed of the training process in deep learning systems, e.g., CNN. During the forward pass, the multiplication between weights and the input shifts the distribution along the direction of the weights. In a deep neural network, subsequent operations result in a massive shift in the distribution, which leads to a loss in performance. This is called internal covariate shift. To tackle this problem, the batch normalisation layer was proposed. Just like the input to the network must be normalised with zero mean and unit variance, the batchNorm layer normalises the input to each layer by calculating the mean and variance in a batch-wise manner.

The batch normalisation operation normalises the elements $x_i$ of the input to calculate the mean $\mu_B$ first, and variance $\sigma_B^2$ over time, spatial and visual domain for each channel independently. Then, it calculates the normalised activations as

$$y_i = \gamma \hat{x}_i + \beta \tag{2}$$

Where the offset $\beta$ and scaling parameter $\gamma$ are learnable parameters that are updated during network training.

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{3}$$

Where $\epsilon$ is a constant, it increases numerical stability when the variance is modest. [17].

### *2.1.3 Pooling Operation:*

The pooling operation will downsample the large image by replacing subregions (e.g. 2×2 matrix) with a single value. This value can be the average of all the region or maximum value elements. The Pooling layer reduces the complexity of each activation map while maintaining the essential data. The images of people’s faces are separated into a series of non-overlapping rectangles. Each area is downsampled using a non-linear approach such as average or maximum. This layer, commonly put between convolutional layers, achieves higher generalisation, faster intermingling, and is resistant to translation and mutilation (refer to Figure 3).

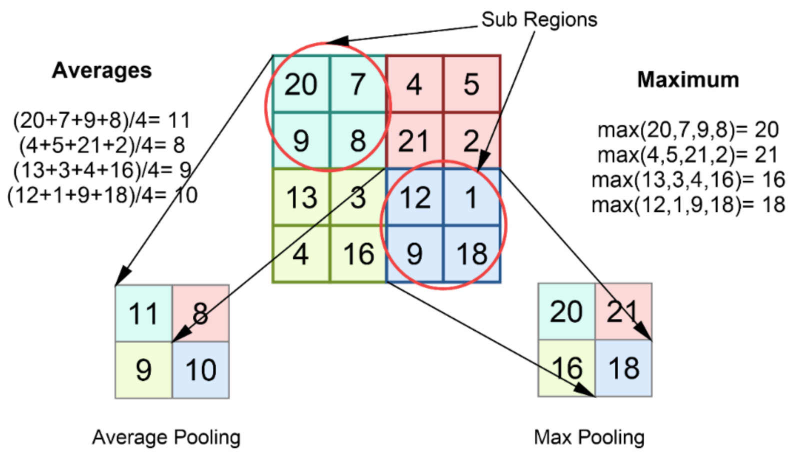

Fig.3. Average and Max Pooling Operation

### *2.1.4 ReLU Operation:*

ReLU is a non-linear process that includes rectifier-based units. It’s a clever component operation that works per pixel and replaces all negative values in the feature map with nothing.

The Rectified Linear Unit (ReLU) is an activation function broadly used in deep neural networks. It can be observed as a truncation operator applied element-wise on the input. ReLU layer has no parameters, hence no need for parameter learning. In the case of images, the ReLU function will let through features that result positive for a particular pattern and render other negative patterns to zero(refer to Figure 4). In this way, the ReLU function induces sparsity in the model, reducing complexity and memory consumption. Other popular activation functions such as sigmoid and tanh tend to saturate at their extremes.

To comprehend the operation of ReLU, let’s assume that there is an input neuron labelled as, and from that, the rectifier is characterised as:

$$f(x) = max(0, x) \tag{4}$$

In the literature for the neural system. The derivative of ReLU is:

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{5}$$

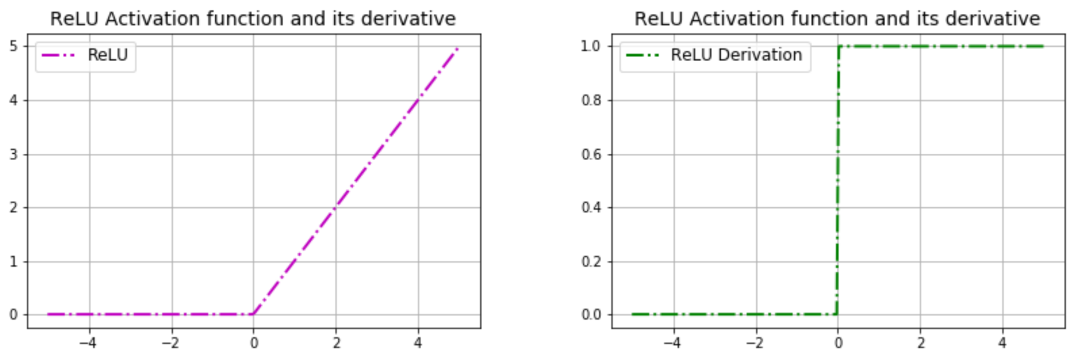

Fig.4. The plot of ReLU and its derivative

### *2.1.5 Fully Connected Layer:*

Each channel in the previous layer is connected to each channel in the following layer, as the name “fully connected layer” suggests. The yield from the convolutional, pooling, and ReLU layers exemplifies the info image’s significant level features. The FCL’s purpose is to use these features to categorise the input image into various classes based on the training dataset. The fully connected layer is the final pooling layer that looks after the characteristics of a classifier that uses the Softmax activation technique. The Fully Connected Layer has a total of one yield probability. Softmax is used to perform the activation procedure, which ensures this. Softmax compresses a vector of arbitrary real-valued scores into a vector of values that falls somewhere between zero and one.

### *2.1.6 Soft Max Operation:*

The 'softmax' function will transform a matrix of K fundamental values into a vector of K tiny values, which is summed 1 in total. These are collections of +ve, -ve, 0, or >1, but the 'softmax' turns them into probabilistic values between 0 and 1. If one of the inputs is small or -ve, the 'softmax' reduces the probability, and if the input is large, it increases the likelihood, but it will always be between 0 and 1.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{6}$$

Where
$\vec{z}$ = input vector
$z_i$= elements of the input vector
$e^{z_i}$= exponential function (standard)
$\sum_{j=1}^{K} e^{z_j}$= normalisation term
$K$= multi-class classifier's number of classes

## *2.2 Proposed Methodology*

The fundamental goal of this research is to develop an excellent recognition system that is both accurate and fast. In this paper, there are three stages to the general design of the Face recognition technique. It starts with the pre-preparing stage: loading face dataset, shuffling and splitting of the dataset for training and validation, proceeds with machine learning using extraction of facial features with *'sgdm'* optimisation algorithm, and afterwards extracted feature set is classified for recognition accuracy testing of the trained network. Softmax Classifier is responsible for implementing the final stage of our framework, which is a classification based on facial features retrieved from CNN as shown in Figure 5.

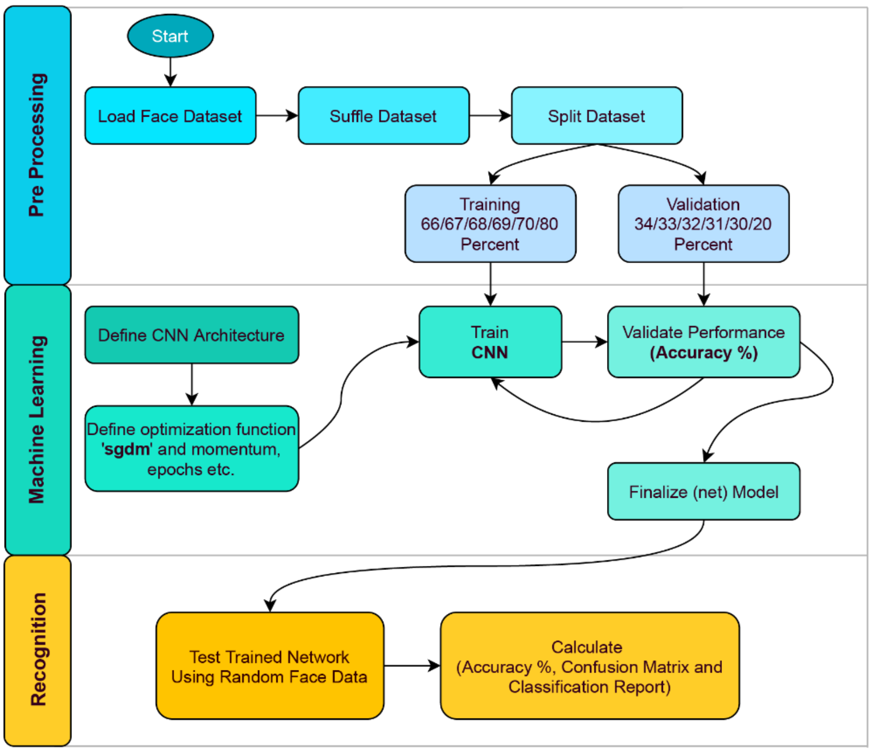
Pre Processing
Machine Learning
Recognition
Start
Load Face Dataset
Suffle Dataset
Split Dataset
Training 66/67/68/69/70/80 Percent
Validation 34/33/32/31/30/20 Percent
Define CNN Architecture
Define optimization function 'sgdm' and momentum, epochs etc.
Train CNN
Validate Performance (Accuracy %)
Finalize (net) Model
Test Trained Network Using Random Face Data
Calculate (Accuracy %, Confusion Matrix and Classification Report)

Fig.5. Stages of proposed MM-CNN face recognition system

### ***Stage 1: Pre-Processing***

This stage involves the operation covering the steps to load the input dataset of faces (Yale, ORL). Load these datasets into the simulation environment for further operation, followed by shuffling face samples after that dataset is partitioned into training and validation parts, 66% to 80% used for training and 34% to 20% for validation, respectively (refer to Figure 6).

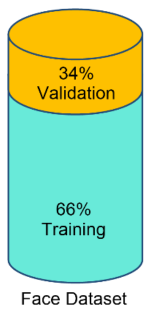

Fig.6. Splitting Face Datasets into Two Parts Training & Validation

***Stage 2: Machine Learning***

The machine learning stage comprises four parts: defining CNN architecture, defining optimisation function, network training, and network validation of network, as shown in Figure 5. This stage extracts features and trains the neural network using a defined percentage of the face dataset. The layered architecture of CNN is shown in Figures 7 and Figure 10 for Yale Dataset and ORL Datasets, respectively, with all the detailed feature map, filter size (kernel), and the number of filters of each convolution module.

Five convolution modules are being used to achieve higher accuracy for Yale face dataset, and four convolution modules for ORL face dataset. Every module has the following layers:

*A. 2D Convolution Layer*

The face images are given as input to this stage, and 2D convolution is performed with $5 \times 5$ filter (kernel) size on the image to get the 16 convoluted features or feature maps of the input face image (refer to Figure 2 for complex operation). The output of this layer is shown below in Figure. Similarly, 32, 64, 128, and 256 convoluted features (feature maps) will be extracted from the second, third, fourth and fifth convolution layers, respectively.

*B. Batch Normalization Layer*

Batch normalisation performs to normalise data received from the convolution layer, i.e., feature maps. This layer refines the features as per batch size defined. As a result, we will get a different set of normalised features.

*C. ReLU Layer*

Rectified Linear Unit or ReLU layer is responsible for getting truncated features after normalisation, and it fastens the training process of deep learning.

*D. Max Pooling Layer*

The pooling layer is implemented in a deep learning environment to reduce the sample sizes (here, face features). Here max pooling is performed, which means maximum values of subregions is considered to downsample the features. The pooling layer is only a bridge between two convolutional modules. This can't be used to connect the output or fully connected layer.

***Stage 3: Recognition***

The previous stage has done all the learning processes from the given face dataset from Yale and ORL faces. After learning (extraction of face features), the neurons are trained and classified using another three layers. The three layers that make up this system are the fully connected, softmax, and classification layers. Below is a description of how layers function.

*A. Fully Connected Layer*

After getting the output of the final convolutional module (here, the output of relu_5 layer) will be reshaped into a single column and operate with the weight matrix and bias vector to get the specified number of classes (here 28 people).

*B. Softmax Layer*

The preceding activation function in the deep learning network is used for multi-class logistic regression. In other words, used for network normalisation to get the probability distribution of network output. This layer helps predict the class of input or help to classify the input.

*C. Classification Layer*

This layer shows the weighted classification, i.e., after calculation of cross-entropy loss gives the definite class of the input.

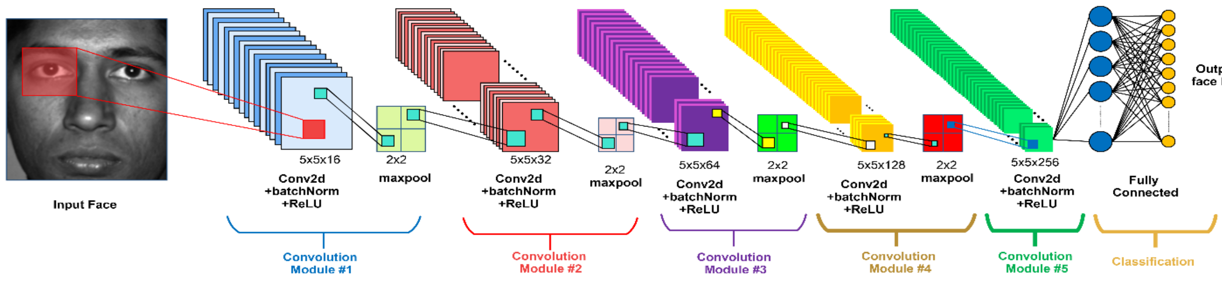

Fig.7. CNN Layered Architecture for Yale Dataset

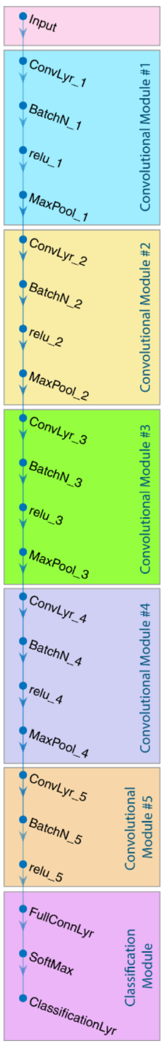

Fig.8. Proposed MM-CNN Layer Graph used for Yale Dataset

The CNN is formed using different modules: the convolution layer, batch normalisation layer, relu layer, and pooling layer. The input face information travels from the input layer to classification through these modules and gives the final recognised output; refer to Figure 8 for the Yale face dataset and Figure 11 for the ORL face dataset.

Figures 9 and 12 show the o/p feature maps of all the layers and modules. For Yale face dataset module 1, generate 16 feature maps using $5\times5$ filter size(kernel) and forward these to batch normalisation layer, relu, and pooling layers. In module 2, the filter size is kept constant at $5\times5$, and 32 feature maps are generated. In module 3, 64 feature maps are generated, and in module 4, 128 feature maps and in module 5, 256 feature maps are generated. These maps are, and kernel sizes have been decided after many trials to get optimal accuracy.

The ORL face dataset kernel size is kept 3×3 and 8, 16, 32 and 64 feature maps generated in four modules. These modules are designed with increasing order of feature maps. The increasing number of feature maps are helpful to extract the features out of features extracted from the previous stage. By this approach, essential features are filtered out. These delicate features are then normalised using a fully connected and softmax layer to get the weighted probability of features ranges between 0 and 1. This weighted vector is led to decide the face's final class(person).

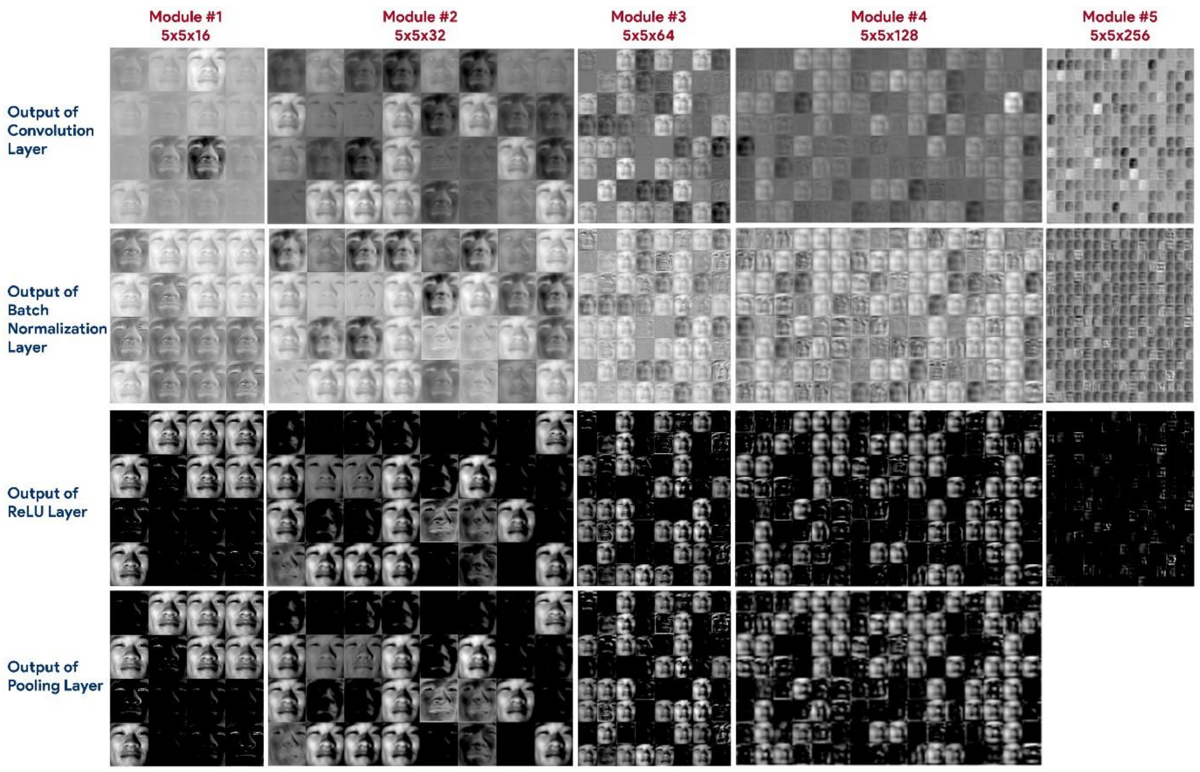

Fig.9. Layer Outcomes of Proposed MM-CNN System for Yale Dataset

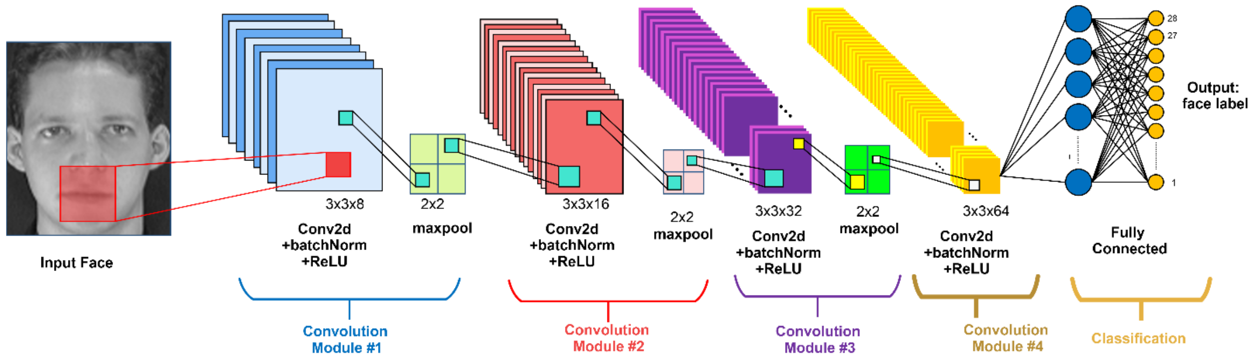

Fig.10. CNN Layered Architecture for ORL Dataset

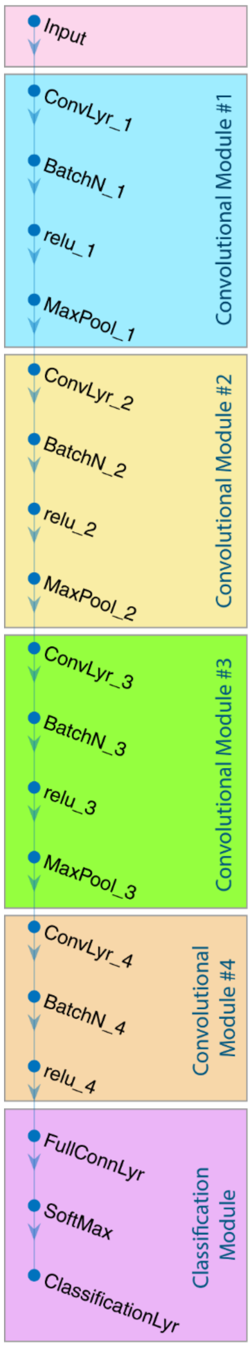

Fig.11. CNN Layer Graph used for ORL Dataset

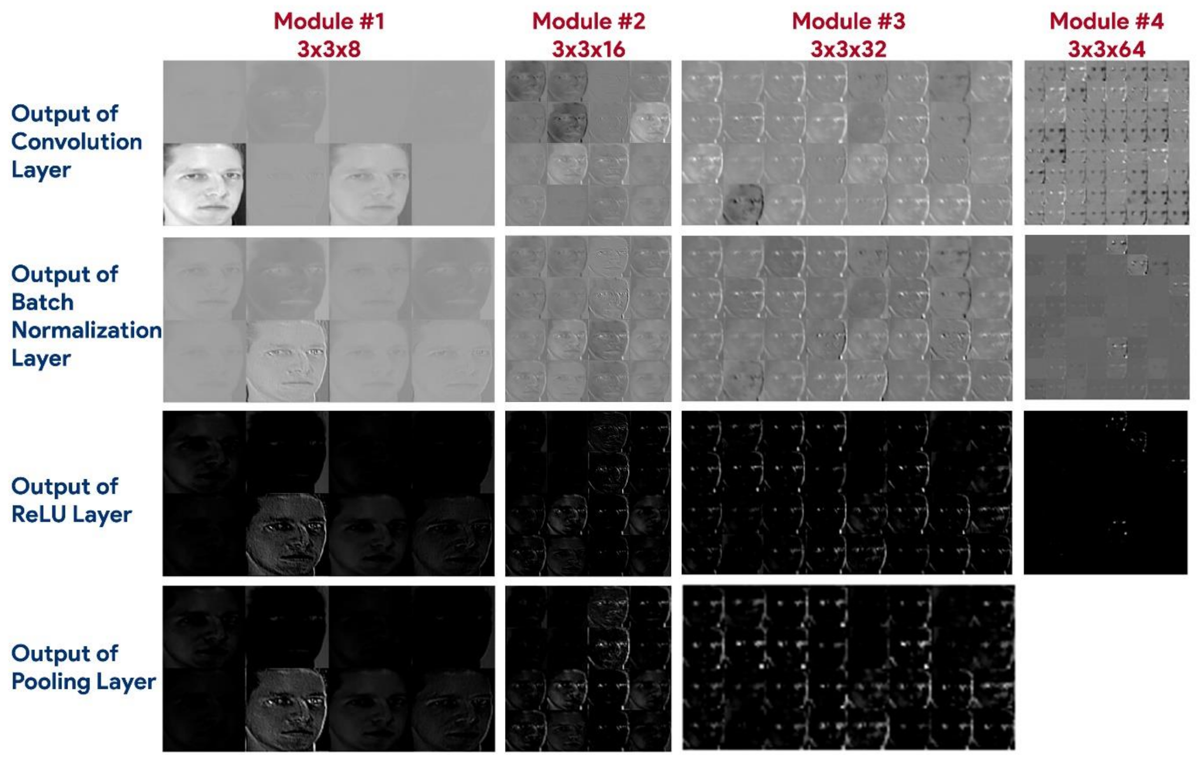

Fig.12. Layer Outcomes of Proposed MM-CNN System for ORL Dataset

## 3. Experimental Result and Analysis

The proposed algorithm for face recognition developed in this work was tested on ORL and Yale face datasets. The Yale face dataset has 16215 face photographs of 28 distinct people, each with nine postures and 64 different lighting conditions. In contrast, the ORL dataset contains 280 face images of 28 people, each with 10 photos of each class (refer to Figure 13).

For experimental evaluation, different training percentage has been taken to get the best performance of the recognition system. The benchmarks are shown in terms of accuracy (%), Mean, Standard Deviation.

$$SD = \sqrt{\frac{\sum|x-\mu|}{N}} \tag{7}$$

Where $\Sigma$ = sum of, $\mu$ = dataset mean value, $x = a$ value in the data set, and $N$ = number of data points in the population, the arithmetic mean was calculated using the following mathematical formula:

$$M = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{8}$$

$M$ = arithmetic mean

$n$ = averaged number of terms

$x_i$ = the averaged numbers' value for each item

Accuracy (A) is calculated using the following formula:

$$A = \frac{1}{k}\sum S_m \tag{9}$$

Where $k$ = number of test samples, $S_m$= correctly classified samples.

The Yale face dataset is divided into 66% and 34%. 66% of faces data samples are used for training CNN, and 34% of faces are used to validate professional networks for checking and increasing accuracy. Subsequently, this step is repeated for different training percentages and validation parts. After training and validation, achieved accuracy results for different training percentages are shown in Table 1. The average accuracy and standard deviation are shown in Table 2.

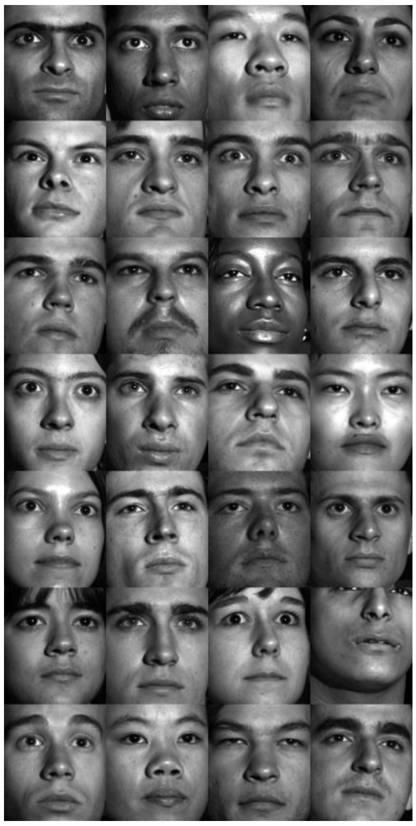

Fig.13 Face samples in Yale dataset of 28 different persons

Table 1. Accuracy Performance of Training Datasets of Yale Database Under Different Percentages

| Method | Network Accuracy for Varied Training Percentage of Dataset | | | | | |
|---|---|---|---|---|---|---|
| | 66% | 67% | 68% | 69% | 70% | 80% |
| DCN [18] | 93.00 | 93.50 | 94.50 | 94.00 | 92.10 | 94.50 |
| Two branches of CNN [19] | 91.50 | 91.80 | 90.90 | 92.90 | 93.10 | 93.10 |
| LLRLCR [20] | 90.60 | 91.90 | 92.90 | 92.30 | 91.90 | 92.90 |
| ESPCN+ CNN [1] | 95.20 | 95.30 | 94.50 | 94.20 | 94.60 | 95.30 |
| MM-CNN (Our Proposed) | 96.79 | 96.77 | 97.43 | 96.72 | 95.95 | 95.99 |

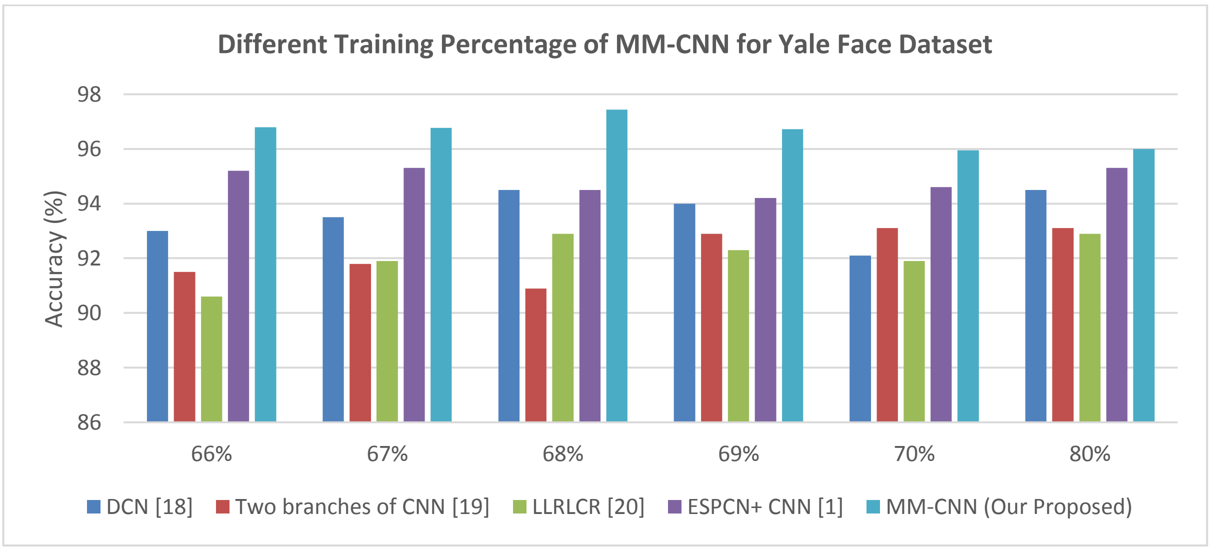

Fig.14. Accuracy for different percentages of training for Yale face dataset

Table 2. Standard Deviations (S.D.) and Accuracy of Yale Data for different methods

| Method | Average Accuracy | SD |
|---|---|---|
| ESPCN + CNN [1] | 94.90 | 0.22 |
| DCN [18] | 92.20 | 0.88 |
| LLRLCR [20] | 92.10 | 0.71 |
| Two divisions of CNN [19] | 93.60 | 0.86 |
| MM-CNN (Our Proposed) | 96.60 | 0.56 |

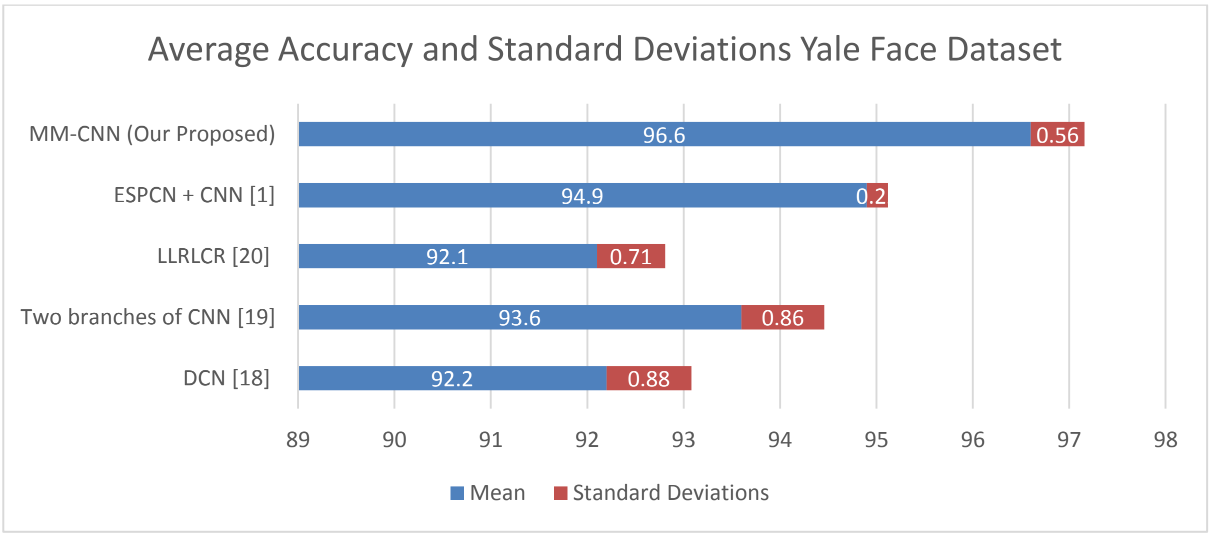

Fig.15. Average Accuracy and Standard Deviation for different methods using Yale face dataset

The ORL face dataset is divided into 66% and 34%. 66% of faces data samples are used for training CNN, and 34% of faces are used to validate professional networks for checking and increasing accuracy. Subsequently, this step is repeated for different training percentages and validation parts. After training and validation, achieved accuracy results for different training percentages are shown in Table 3. The average accuracy and standard deviation are shown in Table 4.

Fig.16. Face samples in ORL dataset of 28 different persons

Table 3. Accuracy Performance of ORL Database at Different Parts of Training

| Method | Network Accuracy for Varied Training Percentage of Dataset | | | | | |
|---|---|---|---|---|---|---|
| | 66% | 67% | 68% | 69% | 70% | 80% |
| ESPCN+CNN [1] | 92.5 | 91.6 | 93.5 | 91.5 | 92.1 | 93.5 |
| DCN [18] | 89.8 | 88.4 | 87.3 | 89.2 | 87.5 | 89.8 |
| LLRLCR [20] | 89.5 | 90.5 | 88.5 | 90.5 | 91.4 | 91.4 |
| Two branches of CNN [19] | 91.6 | 90.4 | 91.6 | 92.7 | 90.4 | 92.7 |
| MM-CNN (Our Proposed) | 94.05 | 94.05 | 96.43 | 94.05 | 95.24 | 94.64 |

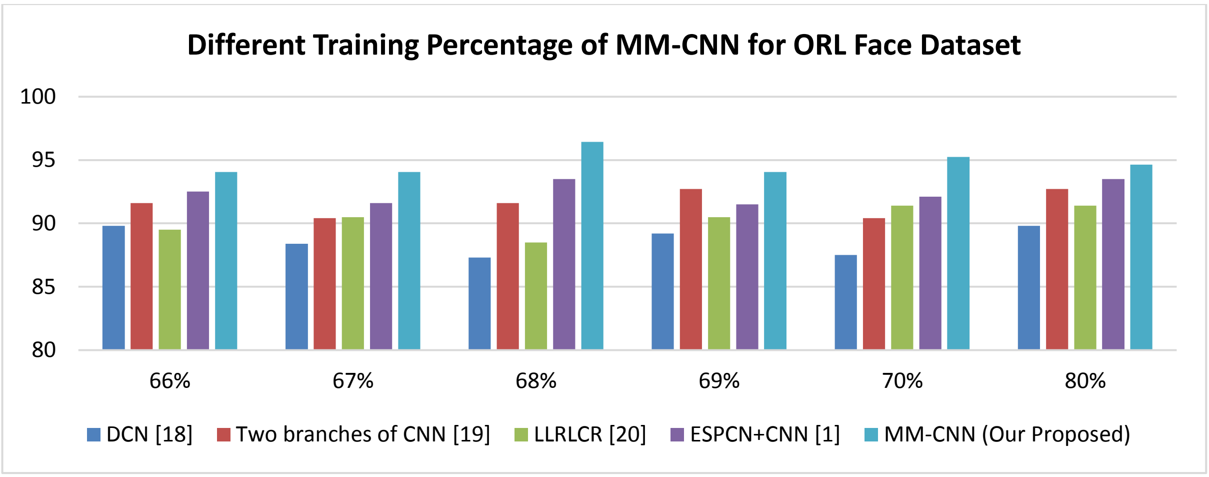

Fig.17. Accuracy for different percentages of training for ORL face dataset

Table 4. Average Accuracy and Standard Deviations(S.D.) of All Methods (ORL Dataset)

| Method | Average Accuracy | SD |
|---|---|---|
| ESPCN + CNN [1] | 92.24 | 0.81 |
| DCN [18] | 88.44 | 1.07 |
| LLRLCR [20] | 90.08 | 1.11 |
| Two branches of CNN [19] | 91.34 | 0.97 |
| MM-CNN (Our Proposed) | 94.74 | 0.56 |

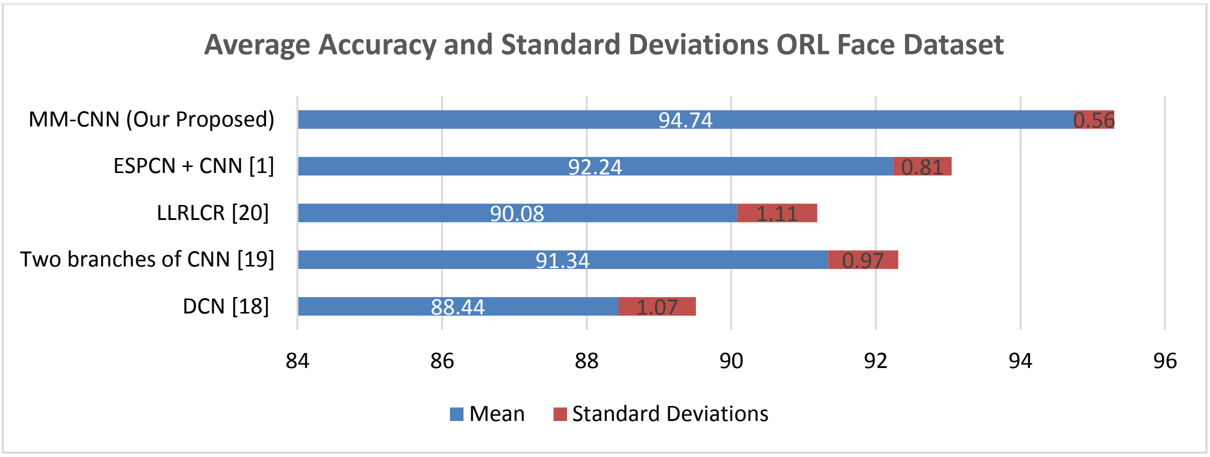

Fig.18. Average Accuracy and Standard Deviation for different methods using ORL face dataset

Table 5. Difference of proposed work with existing work

| Method | Parameters | | | |
|---|---|---|---|---|
| | Approach | Face Dataset | Preprocessing Stages | Training Network |
| ESPCN + CNN [1] | Efficient Sub-Pixel Convolutional Neural Network | ORL, Extended Yale B | LR to HR | Self |
| DCN [18] | Two-Branch Deep CNN | FERET | Down Sampling | PreTrained VGGnet |
| LLRLCR [20] | Geometric Transformations Model | - | - | - |
| Two branches of CNN [19] | Low-Rank Representation and Locality-Constrained Regression | AR, Extended Yale B | LR to HR | Self |
| MM-CNN (Our Proposed) | Multi-Module CNN with Mini-Batch Optimization | ORL, Extended Yale B | - | Self |

Face recognition system using existing approaches has compared with the proposed work on the basis of main approach, face dataset, preprocessing stages and training network has been shown in Table 5. The main pros of proposed method over existing approaches is no pre-processing stage is there which takes less complexity of system and time for training network.

## 4. Discussion and Conclusion

This research work of face recognition is developed using a convolutional neural network, having several hidden layers constituted in 5 modules for Yale face dataset because of large face images and four modules for the ORL dataset for a small number of face samples. This scheme utilises the minibatch operation, batch normalisation, and linear rectified unit and pooling operation in each module to achieve a higher accuracy level, as seen in the simulation outcomes. The accuracy level achieved for Yale faces dataset is an average of 96.60%, and the ORL faces dataset average accuracy is 94.74% considering 66% to 80% training sets. These results clearly show better accuracy than the benchmark methods shown in experimental results. The proposed CNN-based recognition system takes much time to train the network, which can be solved by using the parallel processing enabled computing setup. The proposed face recognition is tested on grayscale images that could also be tested on the colour images because the upcoming multimedia technology is transferring on RGB and CMYK colour spaces. Integration of some pre-processing stages will speed up the training process by optimising the feature data, and this will filter out the less valuable or less significant information to be stored by the neural network. Another future extension can be done by increasing the layers of the CNN, which supports more feature extraction from the objects to be recognised.

## References

[1] M. A. Talab, S. Awang, and S. A. M. Najim, “Super-Low Resolution Face Recognition using Integrated Efficient Sub-Pixel Convolutional Neural Network (ESPCN) and Convolutional Neural Network (CNN),” in *2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, Selangor, Malaysia, Jun. 2019, pp. 331–335.

[2] D. N. Parmar and B. B. Mehta, “Face Recognition Methods & Applications,” Int. J. Comput. Technol. Appl., vol. 4, no. 1, pp. 84–86, 2013.

[3] T. Meenpal, A. Balakrishnan, and A. Verma, “Facial Mask Detection using Semantic Segmentation,” in *2019 4th International Conference on Computing, Communications, and Security (ICCCS)*, Rome, Italy, Oct. 2019, pp. 1–5.

[4] M. M. Y. Zhang, K. Shang, and H. Wu, “Learning deep discriminative face features by customised weighted constraint,” *Neurocomputing*, vol. 332, pp. 71–79, Mar. 2019.

[5] D. Bhamare and P. Suryawanshi, “Review on Reliable Pattern Recognition with Machine Learning Techniques,” *Fuzzy Information and Engineering*, vol. 10, no. 3, pp. 362–377, Jul. 2018.

[6] Technical Implementation Unit for Instrumentation Development, Indonesian Institute of Sciences, Bandung, 40135, Indonesia *et al.*, “Comparing Performance of Supervised Learning Classifiers by Tuning the Hyperparameter on Face Recognition,” *IJISA*, vol. 13, no. 5, pp. 1–13, Oct. 2021.

[7] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep Face Recognition,” in *Proceedings of the British Machine Vision Conference 2015*, Swansea, 2015.

[8] G. Guo and N. Zhang, “A survey on deep learning-based face recognition,” *Computer Vision and Image Understanding*, vol. 189, p. 102805, Dec. 2019.

[9] Ming Liang and Xiaolin Hu, “Recurrent convolutional neural network for object recognition,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 3367–3375.

[10] P. O. Pinheiro and R. Collobert, “Recurrent Convolutional Neural Networks for Scene Labeling,” p. 9, 2014.

[11] Wei Shen, Xinggang Wang, Yan Wang, Xiang Bai, and Z. Zhang, “DeepContour: A deep convolutional feature learned by positive-sharing loss for contour detection,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 3982–3991.

[12] M. You, X. Han, Y. Xu, and L. Li, “Systematic evaluation of deep face recognition methods,” *Neurocomputing*, vol. 388, pp. 144–156, May 2020.

[13] Research Scholar, Department of ECE, Global Academy of Technology, Bangalore-560098, R. K, and R. J, “Performance Evaluation of Face Recognition system by Concatenation of Spatial and Transformation Domain Features,” *IJCNIS*, vol. 13, no. 1, pp. 47–60, Feb. 2021.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[15] A. Uçar, Y. Demir, and C. Güzeliş, “Object recognition and detection with deep learning for autonomous driving applications,” *SIMULATION*, vol. 93, no. 9, pp. 759–769, Sep. 2017.

[16] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” 2015.

[17] Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen, “Deep Network Cascade for Image Super-resolution,” in *Computer Vision – ECCV 2014*, vol. 8693, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 49–64. Accessed: Mar. 10, 2022.

[18] E. Zangeneh, M. Rahmati, and Y. Mohsenzadeh, “Low-Resolution Face Recognition Using a Two-Branch Deep Convolutional Neural Network Architecture,” 2017.

[19] G. Gao *et al.*, “Robust low-resolution face recognition via low-rank representation and locality-constrained regression,” *Computers & Electrical Engineering*, vol. 70, pp. 968–977, Aug. 2018.

[20] R. Tkachenko, P. Tkachenko, I. Izonin, and Y. Tsymbal, “Learning-Based Image Scaling Using Neural-Like Structure of Geometric Transformation Paradigm,” in *Advances in Soft Computing and Machine Learning in Image Processing*, vol. 730, A. E. Hassanien and D. A. Oliva, Eds. Cham: Springer International Publishing, 2018, pp. 537–565. Accessed: Mar. 10, 2022.

[21] I. Izonin, R. Tkachenko, D. Peleshko, T. Rak, and D. Batyuk, “Learning-based image super-resolution using weight coefficients of synaptic connections,” in *2015 Xth International Scientific and Technical Conference “Computer Sciences and Information Technologies” (CSIT)*, Lviv, Ukraine, Sep. 2015, pp. 25–29.

[22] J. Mohammed Sahan, E. I. Abbas, and Z. M. Abood, “A facial recognition using a combination of a novel one dimension deep CNN and LDA,” *Materials Today: Proceedings*, p. S2214785321051841, Jul. 2021.

[23] S. Banerjee and S. Das, “Mutual variation of information on transfer-CNN for face recognition with degraded probe samples,” *Neurocomputing*, vol. 310, pp. 299–315, Oct. 2018.

## Authors’ Profiles

**Dr. Archana Sharma** is PhD, from Maulana Azad National Institute of Technology, Bhopal (India ). She is currently a professor at Technocrats Institute of Technology, Bhopal (India). She is a lifetime member of Professional Institution IETE. Her research interests include wireless communication, Data communication and security, microwave Antennas and dielectric resonator antenna. She also reviewed various research papers, including the wireless personal communication journal Springer.

**Deepa Indrawal** is a PhD research scholar at Mewar University Chittorgarh Rajasthan(India). She is currently a Sr. Lecturer in Govt. Polytechnic College Raisen (M.P) India. She received her M.Tech. Degree from Maulana Aza National Institute of Technology Bhopal (India). She is a lifetime member of the Indian Society for Technical Education (ISTE). Her research interest includes image processing, Information Security in Communication. She also reviewed a research paper on Secured face recognition Systems.

**How to cite this paper:** Deepa Indrawal, Archana Sharma, "Multi-Module Convolutional Neural Network Based Optimal Face Recognition with Minibatch Optimization", International Journal of Image, Graphics and Signal Processing(IJIGSP), Vol.14, No.3, pp. 32-46, 2022.DOI: 10.5815/ijigsp.2022.03.04