



# Face Recognition based on Convolutional Neural Network

Surya Prasada Rao Borra, N. V. S. S. Pradeep, N. T. S. Raju, S. Vineel, V. Karteek

**Abstract:** Recognition holds great significance to give biometric authentications that are utilized in various applications particularly in attendance and security. A gathered database of the subjects is converted applying image processing methods to make this task. This paper suggests a cascade object detector based face detection and convolutional neural network alexnet based face recognition that can recognize the faces. The techniques used for face recognition are machine learning-based methods because of their great precision as associated with different methods. Face detection is the initial level before face recognition that is done utilizing a cascade object detector classifier. Face recognition is performed utilizing Deep Learning's sub-field that is Convolutional Neural Network (CNN). It is a multi-layer network which is used to train the network, to perform a particular task using classification. Check learning of a trained CNN model that is AlexNet is used for face recognition.

**IndexTerms:** Convolutional Neural Network, Face Recognition, Face Detection, MATLAB

## I. INTRODUCTION

Face recognition is a mechanism through which the characteristics of specific individuals are identified by a vision system. Face recognition holds great importance in daily life such as video surveillance, security, attendance and so on. As it is the simplest way of authentication without any knowledge of person about computer, it is used in wide range. As the Artificial Intelligence gets attention in the today's era and due to the non-intrusive characteristics, the face recognition holds great significance in human identification with respect to biometric techniques. Face recognition can be quickly halted in an unobstructed environment, even if the person does not have knowledge on the computer and subjects.

The research is still going on to develop better face recognition techniques, it is discerned to have been discussed in several research papers, for example[1],[2].

Revised Manuscript Received on March 17, 2020.

\* Correspondence Author

**Surya Prasada Rao Borra\***, Dept. Of ECE, PVP Siddhartha Institute of Technology, Kanuru, Vijayawada, India. Email: suryaborra1679@gmail.com

**Nelluri Venkata Siva Sai Pradeep**, Dept. Of ECE, PVP Siddhartha Institute of Technology, Kanuru, Vijayawada, India.

**Nidra Taraka Satya Raju**, Dept. Of ECE, PVP Siddhartha Institute of Technology, Kanuru, Vijayawada, India.

**Saraswathi Vineel**, Dept. Of ECE, PVP Siddhartha Institute of Technology, Kanuru, Vijayawada, India.

**Varkala Karteek**, Dept. Of ECE, PVP Siddhartha Institute of Technology, Kanuru, Vijayawada, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

Popular methods focused on simple learning have been challenged with difficulties such as show variability, facial disguises, view lighting, background image ambiguity, and shifts in face appearance as in the references[3]. Deep learning is getting major strides in determining issues that hold for several years limited the AI community's greatest attempts. Face recognition has proven its advancement for designing complicated designs in multidimensional data and hence it is more validated in domains like technology, management and public fields. It is a unique algorithm which has beaten all other algorithms in the field of face recognition due to its architecture and records it has created in the field of machine learning semantic segmentation and so on.

There are many deep learning methods like Deep Belief Network (DBN)[6], Accumulated Autoencoder[7] and Convolutional Neural Network (CNN).

CNN is the frequently used algorithm for object and face recognition. CNN is a type of artificial neural network which works on principle of convolution of signals. It helps in extracting the features of input data so that number of features needed are increased. LeCun is the father of CNN. Handwriting recognition is the first application developed using CNN[ 8]. His algorithm becomes guidance to many other scientists to innovate new technologies in many other fields and base for many applications. Krizhevsky, Sutskever, and Hinton delivered the best outcomes when they published their work in ImageNetCompetition[ 9]. By using Graphical Processing Units (GPUs) which has great computational power, CNN has done extraordinary cutting edge finishes over many fields, including semantic segmentation, and edge detection, image recognition, scene recognition.

This paper mainly discuss about the best suitable algorithm for face recognition which yields high accuracy compared to other recognition algorithms. Within this article, the overall structure of the face recognition method is composed of mainly three steps.

Pre-processing stage:

- i) Colour space conversion
- ii) Resizing the image
- iii) Extraction of facial features

In our system, Softmax Classifier is used for classification depends on CNN feature extraction.

## II. METHODOLOGY

Convolutional Neural Network is one among the most common algorithms of deep learning which is a type of machine learning in which a model is taught to perform classification of a text, video, sound or image tasks directly.

CNNs are especially used for finding patterns which helps in recognizing scenes, objects and faces in pictures. We learn from image data directly, by using patterns images are identified and the need for extracting manual features is removed.

CNNs mainly depend on applications in which object recognition and computer vision are present, such as face-recognition applications and self-driving vehicles, depending on your requirement a pretrained model with your dataset can be used to create a CNN or you can either use scratch to create convolutional neural network.

**A) Architecture of CNN:**

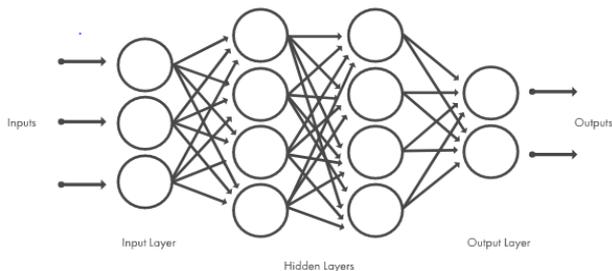
Architecture of CNN contains array of layers with all layers connected sequentially to define the layout of a deep network. For example, a deep network that classifies 28-by-28 grayscale images into 10 groups.

```
Layers = [
    ImageinputLayer([28 28 1])
    convolution2dLayer(3,16,'Padding',1)
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)
    convolutin2dLayer(3,32,'Padding',1)
    batchNormalizationLayer
    reluLayer
    fullyConnectedLayer(10)
    softmaxLayer
    classificationLayer];
```

**B) Layers of CNN:**

Collection of artifacts in layer is called as layers. The layers can be used as an input to the trainNetwork training feature.

To define the layout of a neural network in which all the layers are connected sequentially we need to build an array of layers directly. A LayerGraph object can be used to describe a network architecture in which all the layers have multiple inputs or outputs.



**Fig. 1. Layers of CNN**

**i)Image Input Layer:**

The standardization of data is applied by feeding images into a network by the layer of image.

The image size is specified by input size arument. The image's number of color channels, height and width represents the size of an image. For example, the number of channels for a grayscale image are 1 and for a color image, the number of channels are 3.

**ii) Convolutional Layer:**

convolutional sliding filters to the data are applied by the convolutionary 2-D layer . Various components are present in the convolutional layer.[10]

Neurons present in the convolutionary layer links the previous layers input images or the outputs to subregions. While scanning through an image, the features localized by the regions are learnt by the layer.

**iii) Batch Normalization Layer:**

Each input channel across a mini-batch is returned to its normal state by batch normalization layer. By using the batch normalization layer(ReLU layers) between convolutional layers and non-linearities, results in speeding the training of CNNs and reduces sensitivity.

Next, by subtracting the mean of the mini-batch and dividing by standard deviation of the mini batch, each channels activations are normalised. The layer then changes the input by an apprenticed offset  $\beta$  and scales it by an apprenticeship scale factor  $\gamma$ .  $\gamma$  and  $\beta$  are learning parameters which are changed during network training.

**iv) ReLU Layer:**

A ReLU layer performs a threshold operation for each input variable, where any value that is less than zero is set to zero.

A nonlinear activation function, such as a rectified linear unit (ReLU), defined by a ReLU layer, is usually followed by traditional and batch normalization layers. A ReLU layer operates a threshold for each variable, where any input value less than zero is set to zero,

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \dots\dots\dots(1)$$

The layer ReLU does not alter its input size.

**v) Cross Channel Normalization Layer (Local Response Normalization) :**

A channel-wise local response (cross-channel) layer of normalization performs normalization on the channel-wise.

Normalization of the channel-wise local response is conducted by this layer. Usually, it follows the layer of ReLU activation. Each element will be replaced with a normalized value obtained using from the elements of several adjacent channels (elements in the standardization window). That is, trainNetwork calculates a normalized value  $x'$  for every element  $x$  in the input

$$x' = \frac{x}{\left(K + \frac{\alpha * ss}{windowChannelSize}\right)^{\beta}} \dots\dots\dots(2)$$

Where  $\alpha$ ,  $K$  and  $\beta$  are the normalization hyper parameters, and  $ss$  is the number of the element squares in the standardization window[11]. You may define the size of standardization window using the crossChannelNormalizationLayer function's windowChannelSize statement. You can also specify the hyperparameters using the name-value pair arguments for Alpha, Beta, and K.

**vi) Dropout Layer:**

With a obtained probability, input elements are randomly set to zero by dropout layer.

The dropout layer sets input elements to zero randomly during the training time followed by the dropout mask  $rand(size(X)) < Probability$ ,  $X$  is denoted by the input layer and scaling is done for the remaining elements by  $1/(1-Probability)$ . Between iterations, underlying network architecture changes more effectively by using this operation and overfitting of network is prevented [11], [12]. As the number gets increases,



the elements also gets increased for being dropped when training.

The layer's output at prediction time is equal to its input.

Dropout layer doesn't perform any learning similar to max or average pooling in its layer.

vii) Fully Connected Layer:

In this layer, input is multiplied by a matrix for weight and for bias, add a vector. More number of fully connected layers are followed by the convolutionary (and down-sampling) layers.

Binding of neurons present in fully connected layer with neurons present in previous layer happens in this layer. Each layer incorporates all the features (local information) acquired throughout the picture from previous layers to recognize the bigger patterns. Incorporation of functions to the network happens in last fully connected layer which are used for classification of the images for the classification problems. That's why the outputSize argument of the network which contains last fully connected contains the number of subjects or objects in the database. In order to eliminate regression problems, we need to make the output size same as the number of input variables. Thus we eliminate regression problems.

viii) Softmax and Classification Layers:

A softmax function is applied to the data for softmax layer. softmaxLayer is the argument used to build a softmax layer.

Classification layer is used to calculate the cross entropy loss of mutually exclusive classes for multi-class classification problems. Use ClassificationLayer to build a classification layer.

The softmax and classification layers comprehensively obey all classification problems.

The activation function of the output unit is softmax function:

$$y_r(x) = \frac{\exp(a_r(x))}{\sum_{j=1}^k \exp(a_j(x))}$$

.....(3)

Where  $0 \leq y_r \leq 1$  and  $\sum_{j=1}^k y_j = 1$

For multi-class classification problems softmax function is used which is activation of the output system that is after the last fully connected layer:

$$P(c_r|x, \theta) = \frac{P(x, \theta|c_r)P(c_r)}{\sum_{j=1}^k P(x, \theta|c_j)P(c_j)} = \frac{\exp(a_r(x, \theta))}{\sum_{j=1}^k \exp(a_j(x, \theta))}$$

.....(4)

Where  $0 \leq P(c_r|x, \theta) \leq 1$  and  $\sum_{j=1}^k P(c_j|x, \theta) = 1$

Moreover,

$$a_r = \ln(P(x, \theta|c_r)P(c_r)), P(x, \theta|c_r)$$

Where  $a_r$  is the conditional probability of class r given in the sample.

ix) Regression layer:

For regression problems a regression layer is used to find the half mean squared error loss. For more complicated problems about regression A regression layer will obey a fully connected final layer. The mean squared error for a single observation is given by:

$$MSE = \sum_{i=1}^R (t_i - y_i)^2 / R$$

.....(5)

R represents the number of answers,  $t_i$  represents output of the aim and  $y_i$  represents response I expect from the network. The regression layer loss function for picture and sequence-to - one regression network is the half-mean-squared-error of estimated response, not standardized by R:

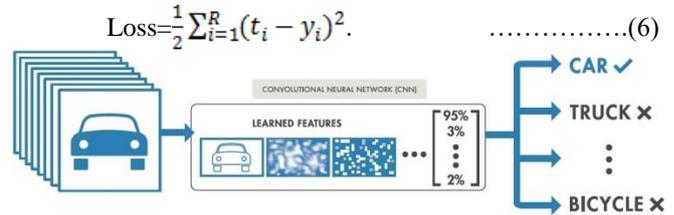


Fig. 2. Deep learning workflow

**C) Number of Neurons:**

In a convolutionary layer, the total number of Neurons (output size) is given by product of Map Size and Number of Filters.

Where Map Size is defined as multiplication of output height and width which gives the total amount of neurons present in the network.

Let us take an example, assume a color image of 32-by-32-by-3 is given as input. For a convolution layer with eight filters each having a size of 5-by-5, for a given filter the number of weights is given by  $5 * 5 * 3 = 75$ , and for a given layer the total number of parameters is given by  $(75 + 1) * 8 = 608$ .

In each direction, if the stride is 2 and size 2 padding is defined then 16-by-16 feature map is obtained for each.

This is because  $(32 - 5 + 2 * 2) / 2 + 1 = 16.5$ , and the padding at some bottom and right of the picture is discarded.

At last,  $16 * 16 * 8 = 2048$  number of layered neurons in the network.

Normally, neuron outcomes flows through some sort of nonlinearity layers, like ReLU layer.

**D) Learning Parameters:**

Learning rates are able to change by using name-value pair. If require a constant learning rates irrespective of variable then it will automatically takes global training options.

These all things can be done while defining the convolutional layer.

**E) Number of Layers:**

Convolutionary Neural Network contains any number of layers which depends upon our requirement.

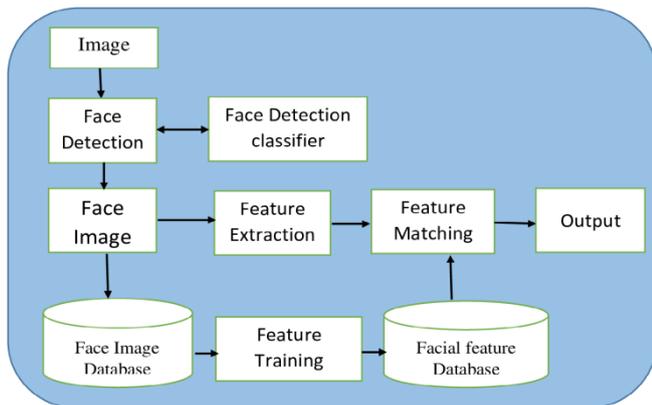
If the application is more complex then we need more number of layers.

If the application is less complex then we need less number of layers.

**III. THE PROPOSED ALGORITHM**

The block schema of the proposed CNN recognition algorithm is given below





**Fig. 3. Block diagram of Face recognition using CNN**

The algorithm mainly consists of three steps as below:

- 1) Resize the input images from any pixel size to 277\*277.
- 2) Build a CNN structure with twenty five layers made up of Input image, Convolutional, ReLU, Cross Channel Normalization, Fully Connected, Dropout, Softmax, Classification Output in required number of times.
- 3) After extracting all features, use Softmax classifier for classification.

The structure of proposed CNN algorithm feature extraction block is illustrated in fig

- 1 'data' Image Input
- 2 'conv1' Convolution
- 3 'relu' ReLu
- 4 'norm1' Cross Channel Normalization
- 5 'pool1' Max Pooling
- 6 'conv2' Convolution
- 7 'relu2' ReLu
- 8 'norm2' Cross Channel Normalization
- 9 'pool2' Max Pooling
- 10 'conv3' Convolution
- 11 'relu3' ReLu
- 12 'conv4' Convolution
- 13 'relu4' ReLu
- 14 'conv5' Convolution
- 15 'relu5' ReLu
- 16 'pool5' Max Pooling
- 17 'fc6' Fully Connected
- 18 'relu6' ReLu
- 19 'drop6' Dropout
- 20 'fc7' Fully Connected
- 21 'relu7' ReLu
- 22 'drop7' Dropout
- 23 'fc8' Fully Connected
- 24 'prob' Softmax
- 25 'output' Classification Output

#### IV. DATABASE

The image database contains 4 subjects in the database. Each subject contains nearly 294 frames which are resized to 277\*277 pixels size from input. The trained data is pretrained using alexnet for deep cnn feature extraction by

softmax classifiers. Database contains all the angles of each face for extracting features of every frame. For example in this paper, the image database contains 4 subjects as follows



**Fig. 4. Images captured using webcam**

The above images are captured using cascade object detector.

Cascade Object Detector:

The classification of cascades consists of stages, where each stage is an ensemble of poor learners. The poor learners are basic classifiers called stumps of decision. Each stage is trained using a Boosting technique. Boosting offers the ability to train a highly accurate classifier by taking a weighted average of the poor learner's decisions.

Increasing stage of the classifier marks a region identified as either positive or negative by the current location of the sliding window. Positive means an object has been found, while negative indicates that no objects have been identified. If the label is negative, this area will be marked total, and the detector will move the window to the next location. If the label is positive, then the classifier will move the region to the next step. When the final stage classifies the region as accurate, the detector identifies an object located at the current window spot.

The stages are planned to remove negative samples as quickly as possible. Here the main assumption is that there is no point of interest the vast majority of screens. Alternatively, the true positive is rare and worth taking the time to verify.

- i) If a positive result is correctly identified, a true positive exists.
- ii) When a negative sample is wrongly marked as positive, a false positive occurs.
- iii) When a positive result is wrongly marked as negative, a false negative occurs.

#### V. EXPERIMENTAL RESULTS

The CNN is designed with 25 layers such as input layer, Convolution, ReLU, Cross Channel Normalization, Max Pooling, Convolution, ReLU, Cross Channel Normalization, Max Pooling, Convolution, ReLU, Convolution, ReLU, Convolution, ReLU, Max Pooling, Fully Connected, ReLU, Dropout, Fully Connected, Softmax, Classification Product. The face recognition can be performed by training images from data i.e.,

resize the images to 277\*277 and transfer them to CNN layers to preprocess the trained data and then to softmax classifier to obtain the characteristics of the subjects. The results are as follows:

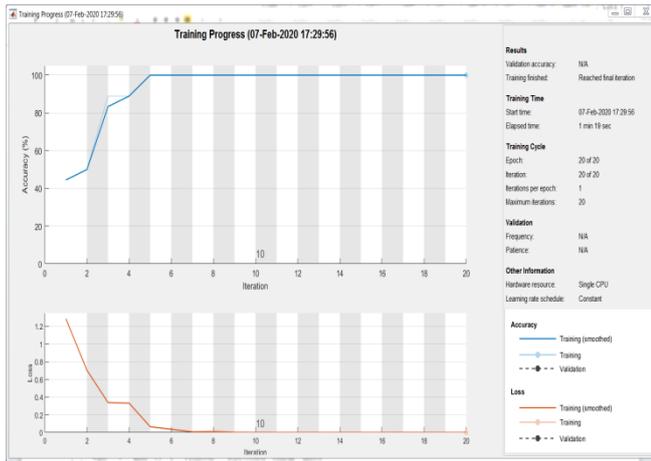
**Table. 1. The characteristics of proposed algorithm**

Training on single CPU.  
Initializing image normalization.

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate
1	1	00:00:06	44.44%	1.2862	0.0010
20	20	00:01:19	100.00%	9.8602e-05	0.0010

accuracy =  
1

The training process of the CNN can be shown in below Fig. 5



**Fig. 5. Training progress for the proposed algorithm**

The review of proposed CNN architecture in terms of loss is near equals to 0 and accuracy reaches 100%. Thus by note, the efficiency of the introduced algorithm is as desired.

**VI. CONCLUSION**

The system is developed with twenty five layered CNN. As complexity increases, number of layers. For more complex operations multilayered CNN is required to perform with greater efficiency. There is a quicker retraining process to this face recognition device. The epochs used for training in this system are 20 epochs. The accuracy of the system is 100% for every 4 subjects. Yet, the MATLAB platform is not very fitting for this method as there was a sequence of degeneration where the normal time to train was not compatible. This face recognition technology could be developed in a different framework for future work; including python, for example, and could be expanded as a real-time system. The program should be proficient in obtaining new images and saving them into the image database.

**REFERENCES**

1. S. G. Bhele and V. H. Mankar, "A Review Paper on Face Recognition Techniques," Int. J. Adv. Res. Comput. Eng. Technol., vol. 1, no. 8, pp.2278–1323, 2012.
2. V. Bruce and A. Young, "Understanding face recognition," Br. J. Psychol., vol. 77, no. 3, pp. 305–327, 1986.
3. C. Geng and X. Jiang, "Face recognition using sift features," in Proceedings - International Conference on Image Processing, ICIP, pp. 3313–3316, 2009.
4. Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Learning Deep Representation for Face Alignment with Auxiliary Attributes,"

5. IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 5, pp. 918–930, 2016.
6. M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3367–3375, 2015.
7. R. Xia, J. Deng, B. Schuller, and Y. Liu, "Modeling gender information for emotion recognition using Denoising autoencoder," in ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing – Proceedings, pp. 990–994, 2014.
8. Y. Bengio, "Learning Deep Architectures for AI," vol. 2, no. 1, 2009.
9. Y. LeCun, "Backpropagation Applied to Handwritten Zip Code Recognition," Neural Comput., vol. 1, no. 4, pp. 541–551, Dec. 1989
10. A. Krizhevsky, I. Sutskever, and H. E. Geoffrey, "ImageNet Classification with Deep Convolutional Neural Networks," Adv. Neural Inf. Process. Syst. 25, pp. 1–9, 2012.
11. Murphy, K. P. Machine Learning: A Probabilistic Perspective. Cambridge, Massachusetts: The MIT Press, 2012.
12. Krizhevsky, A., I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." Advances in Neural Information Processing Systems. Vol 25, 2012.
13. Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." Journal of Machine Learning Research. Vol. 15, pp. 1929–1958, 2014.

**AUTHORS PROFILE**



**Surya Prasada Rao Borra** working as Associate Professor in the department of ECE, Prasad V. Potluri Siddhartha Institute of Technology, Kanuru, Vijayawada. He has 18 years of teaching experience. He is working in the area of Image Processing. He has published 14 research papers in various national and international journals and presented papers in national and international conferences in the area of Image Processing and VLSI technology.



**Nelluri Venkata Siva Sai Pradeep** is pursuing Bachelor of Technology in the department of ECE, Prasad V. Potluri Siddhartha Institute of Technology, Kanuru, Vijayawada. He is member of IETE Student forum and a Certified Labview Associate Developer in National Instruments.



**Nidra Taraka Satya Raju** is pursuing Bachelor of Technology in the department of ECE, Prasad V. Potluri Siddhartha Institute of Technology, Kanuru, Vijayawada. He is member of IETE Student forum.



**Saraswathi Vineel** is pursuing Bachelor of Technology in the department of ECE, Prasad V. Potluri Siddhartha Institute of Technology, Kanuru, Vijayawada. He is member of IETE Student forum.



**Varkala Kartteek** is pursuing Bachelor of Technology in the department of ECE, Prasad V. Potluri Siddhartha Institute of Technology, Kanuru, Vijayawada. He is member of IETE Student forum.