Face Recognition Using Deep Convolutional Network and One-shot Learning

Joyassree Sen[1], Bappa Sarkar[2], Mst. Ashrafunnahar Hena[3], Md. Hafizur Rahman[4]

[1] Assistant Professor, CSE Department, Islamic University, Bangladesh
[2] Assistant Professor, CSE Department, Islamic University,Bangladesh
[3] Assistant Professor, EEE Department, Islamic University, Bangladesh
[4] Database Programmer, ICT Cell, Islamic University, Bangladesh

Abstract

Among the most successful application of images analysis and understanding, face recognition has recently received significant attention, especially during the past few years. Facial recognition technology (FRT) has emerged as an attractive solution to address many contemporary needs for identity and verification of identity claims. Face recognition is the identification of humans by the unique characteristics of their faces. FRT technology is the least intrusive and fastest bio-metric technology. It works with the most obvious individual identifier for the human face. With increasing security needs and with the advancement in technology extracting information has become much simpler. The system proposed in this paper uses the power of Convolution Neural Network (CNN) to encode the face and produce a vector matrix. Then we use tripled loss function to calculate the distance between input and trained image to predict the face.

Keyword: *CNN, FRT, ANN, Machine learning, Conv*

I. *A. Introduction*

Recognition is one of the most useful functions of our visual system. We recognize materials (marble, orange peel), surface properties (rough, cold), objects (my car, a willow tree), and scenes (a thicket of trees, my kitchen) at a glance and without touching them. We recognize both individuals keep learning more throughout our life. As we learn, we organize both objects and categories into useful and informative taxonomies and relate them to language. Replicating these abilities in the machines that surround us would profoundly affect the practical aspects of our lives, mostly for the better. Certainly, this is the most exciting and difficult puzzle that faces computational vision scientists and engineers in this decade. In this paper, we build face recognition system using deep convolutional neural network and one shot learning technique.

B. Problem statement

Human can easily detect and recognize various objects but this is a difficult task for the computer. In computer vision recognition various object such as human face is challenging one, because the human face is a dynamic objects that comes in many forms and colors. In past few decades various methods were implemented to solve this task. The conventional face recognition pipeline consists of four stages: face detection, face alignment, feature extraction (or face representation) and classification. Perhaps the single most important stage is feature extraction. In this paper, we use deep convolutional neural network for feature extraction and one-shot technique to verify face which use only one image to recognize faces [1].

C. a) Geometric based feature extraction

In geometric-based approach, the local features (local statistics and locations include mouth, eyes, eyebrows, and nose are at first extracted from face images. The most important geometric based methods are Active appearance graph models.

Active shape model (ASM)

A model for face recognition based on the local matching gabor consisting of three main modules and one of which is active shape model where image alignment is done and is used to align the face has been proposed.

Active appearance model (AAM)

This model contains a statistical model of the shape and the grey level appearance of the object of interest. These models were generated by combining the model of shape variation with a model of appearance variations in a shape normalized form.

Scale- invariant feature transform (SIFT)

A model for fuzzy match index for scale invariant feature transform features which involve all the SIFT key points in decision making process were presented. This SIFT is primarily designed for object recognition applications such as face recognition, iris recognition, fingerprint identification and so on.

b) Appearance base feature extraction

This method is usually used for frontal face detection using color images-based feature extraction and appearance-based classification [8]. Face detection with certain degree of accuracy and robustness uses low level image features such as color and shape.

Local Binary Pattern (LBP)
The method for face recognition in uncontrolled environment which works with local binary pattern of facial images has been presented.

Gabor features
The Gabor based method is used which modifies the grid from which the Gabor features are extracted using mesh to model face deformations produced by varying pose and also statistical model of the scores are computed by using Gabor features to improve recognition performance.

Principal component analysis (PCA)
The utilization of correlation between pixels, columns, rows takes place but the local spatial information is not used in these techniques and it is observed that patches are more meaningful basic units than pixels for face recognition.

II. Deep learning

Deep learning [3] is a sub-field of machine learning dealing with algorithms inspired by the structure and function of the brain called artificial neural networks. In other words, it mirrors the functioning of our brains. Deep learning algorithms are similar to how nervous system structured where each neuron connected each other and passing information.

A. Neural network

Neural network is the fundamental building block of deep learning. A special case of a neural network called convolution neural net-work (CNN) [9] is the primary focus of this project. Before discussing CNNs, we will discuss how neural network work. An Artificial Neuron Network (ANN), popularly known as Neural Network is a computational model based on the structure and functions of biological neural networks. It is like an artificial human nervous system for receiving, processing, and transmitting information in terms of Computer Science [4].

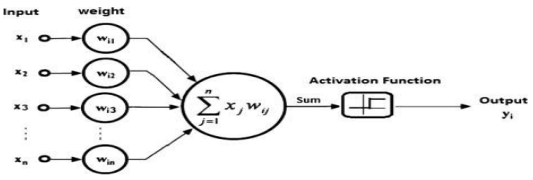

Figure 1: An Artificial Neuron

Basically, there are 3 different layers in a neural network [5].

- Input Layer (All the inputs are fed in the model through this layer)
- Hidden Layers (There can be more than one hidden layers which are used for processing the inputs received from the input layers)
- Output Layer (The data after processing is made available at the output layer)

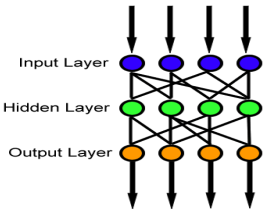

Figure 2: Neural Network Layers

a) Weight
Weights are used to connect the each neuron in one layer to the every neuron in the next layer. Weight determines the strength of the connection of the neurons. If we increase the input then how much influence does it have on the output? Weights [6] near zero mean changing this input will not change the output. Many algorithms will automatically set those weights to zero in order to simplify the network.

b) Activation function
In artificial neural network we apply mostly RELU activation function which replace the all negative values to 0 and remains same with the positive values. In the past, nonlinear function like tanh and sigmoid were used but researchers found out that relu layers work far better because the network is able to train a lot faster (because of the computational efficiency) without making a significant difference to the accuracy. [7]

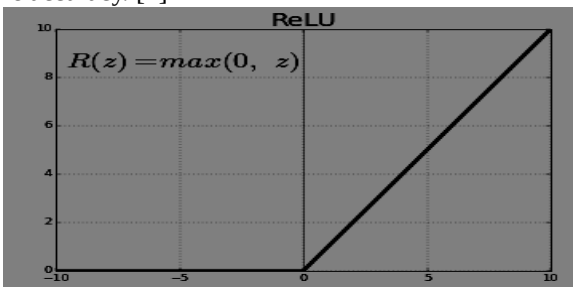

Figure 3: RELU Activation Function

c) Adam Optimizer
The Adam optimization algorithm [8] is an extension to stochastic gradient descent that has recently seen broader adoption for deep learning applications in computer vision and natural language processing. Adam is different to classical stochastic gradient descent. Stochastic gradient descent maintains a single learning rate (termed alpha) for all weight updates and the learning rate does not change during training.

B. Convolutional Neural networks

Convolutional neural network (CNN) [9] are very similar to ordinary Neural Networks but it is specially for image processing . Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity.

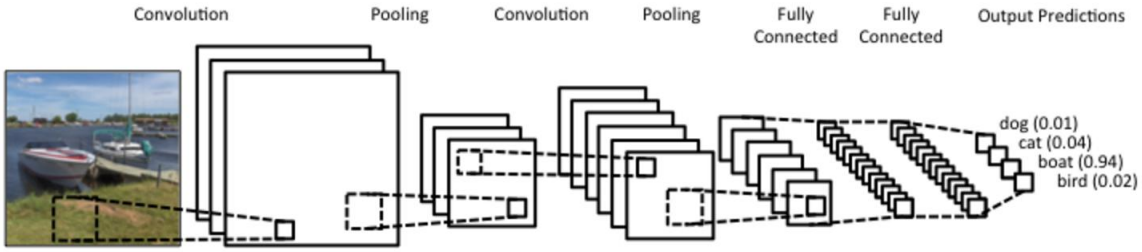

Figure 4: A Basic CNN architecture

C. Basic CNN architecture

Convolution neural network is a sequence of layers, and every layer of a convnet transforms one volume of activations to another through a differentiable function [10]. We use three main types of layers to build convnet architectures:

- Convolution Layer
- Pooling Layer
- Fully Connected Layer

a) Convolution Layer

The Convolution layer is the core building block of a Convolutional Network that does most of the computational heavy lifting. It performs the convolution operations over the input volumes.
Mathematical equation for CNN operation is:

$$(I * K) = \sum_{i=1}^{h} \sum_{j=1}^{w} K_{ij} * I_{x+i+1,\ y+j+1}$$

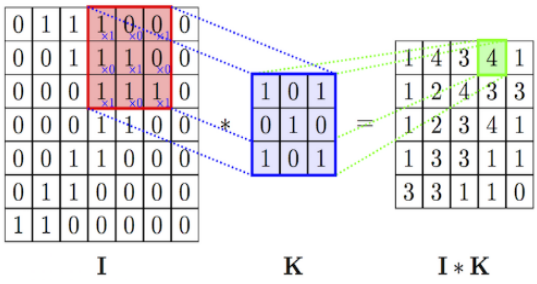

Figure 5: Basic CNN Operation

Mathematical calculation between the first 3x3 with the above image matrix's highlighted part (shown in different color) produce a result 4.The calculation is follows:
1*1+0*0+0*1+1*0+1*1+0*0+1*1+1*0+1*1 = 4

b) Pooling layer

The pooling layer is used to reduce the spatial dimensions, but not depth, on a convolution neural network model, basically this is what convolutional layer gain. Some projects don't use pooling, especially when they want to "learn" some object specific position.

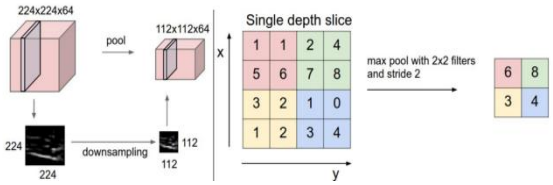

Figure 6: Pooling Example

c) Fully connected layer

The objective of a fully connected layer is to take the results of the convolution/pooling process and use them to classify the image into a label (in a simple classification example).The output of convolution/pooling is flattened into a single vector of values, each representing a probability that a certain feature belongs to a label. The fully connected part of the CNN network goes through its own back propagation process to determine the most accurate weights. Each neuron receives weights that prioritize the most appropriate label. Finally, the neurons "vote" on each of the labels and the winner of that vote is the classification decision.

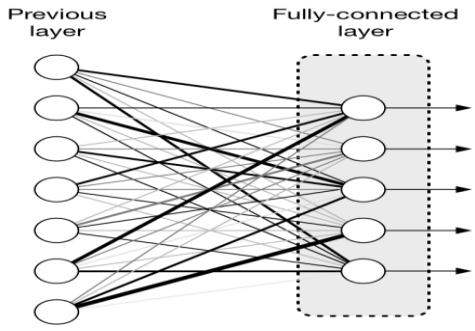

Figure 7: Fully Connected Layer

D. Dropout

Dropout [8] is a technique where randomly selected neurons are ignored during training. They are "dropped-out" randomly. This means that their contribution to the activation of downstream neurons is temporally removed on the forward pass and any weight updates are not applied to the neuron on the backward pass.

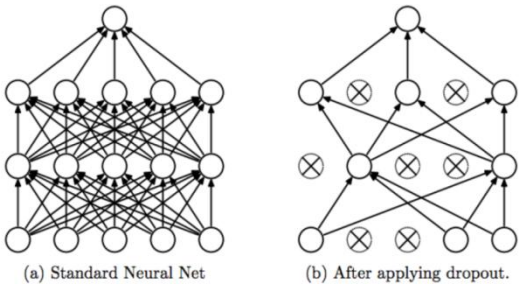

Figure 8: Dropout Example

E. Transfer Learning

Transfer learning [10] is a machine learning technique where a model trained on one task is re-purposed on a second related task. Transfer learning makes use of the knowledge gained while solving one problem and applying it to a different but related problem. For example, knowledge gained while learning to recognize cars can be used to trucks.

There are two types transfer learning method

- Pre-Training
- Fine Tuning

F. One Shot Learning

One-shot learning is a classification task where one, or a few, examples are used to classify many new examples in the future. This characterizes tasks seen in the field of face recognition, such as face identification and face verification, where people must be classified correctly with different facial expressions, lighting conditions, accessories, and hairstyles given one or a few template photos. Modern face recognition systems approach the problem of one-shot learning via face recognition by learning a rich low-dimensional feature representation, called a face embedding, which can be calculated for faces easily and compared for verification and identification tasks.

III. Methodology

We use VGG face model to extract feature from the face and then fed into Siamese network. We use CNN for face perception work. There are several pre-trained models which can readily be utilized for feature extraction, e.g. VGGF, VGG16, VGG19. In our case, for feature extraction, we have utilized the VGGF pre-trained model [3] which we discuss below. Thus, the methodology we adopt here uses the pre-trained VGGF model for feature extraction which is followed by CS or linear SVM for classification.

A. The VGG-Face model

As mentioned above, there are several pre-trained models for CNN and one of the most popular and widely used in face recognition is the VGGF model [9] is developed by Oxford Visual Geometry Group. The model was trained on a huge dataset containing 2.6M face images of more than 2.6 K individuals. The architecture of VGGF comprises of layers, starting from the input layer up to the output layer. The input should be a color image and as the pre-processing step, an average is normally computed from the input image. In general, the VGGF contains thirteen convolutional layers, each layer having a special set of hybrid parameters. Each group of convolutional layers contains max-pooling layers and there are also 15 rectified linear units (ReLUs). After these layers, there are three fully connected layers namely the FC6, FC7 and FC8. The first two have 4096 channels, while FC8 which has 2622 channels are used to classify the identities. The last layer is the classifier which is a softmax layer to classify an image to which the individual face class belongs to. We illustrate the architecture of this further in the following.

layer	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
type	input	conv	relu	conv	relu	mpool	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	relu	mpool	conv
name	–	conv1_1	relu1_1	conv1_2	relu1_2	pool1	conv2_1	relu2_1	conv2_2	relu2_2	pool2	conv3_1	relu3_1	conv3_2	relu3_2	conv3_3	relu3_3	pool3	conv4_1
support	–	3	1	3	1	2	3	1	3	1	2	3	1	3	1	3	1	2	3
filt dim	–	3	–	64	–	–	64	–	128	–	–	128	–	256	–	256	–	–	256
num filts	–	64	–	64	–	–	128	–	128	–	–	256	–	256	–	256	–	–	512
stride	–	1	1	1	1	2	1	1	1	1	2	1	1	1	1	1	1	2	1
pad	–	1	0	1	0	0	1	0	1	0	0	1	0	1	0	1	0	0	1

layer	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
type	relu	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	softmx
name	relu4_1	conv4_2	relu4_2	conv4_3	relu4_3	pool4	conv5_1	relu5_1	conv5_2	relu5_2	conv5_3	relu5_3	pool5	fc6	relu6	fc7	relu7	fc8	prob
support	1	3	1	3	1	2	3	1	3	1	3	1	2	7	1	1	1	1	1
filt dim	–	512	–	512	–	–	512	–	512	–	512	–	–	512	–	4096	–	4096	–
num filts	–	512	–	512	–	–	512	–	512	–	512	–	–	4096	–	4096	–	2622	–
stride	1	1	1	1	1	2	1	1	1	1	1	1	2	1	1	1	1	1	1
pad	0	1	0	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0

Figure 9: VGG Face Model Layer Architecture

B. Feature extraction using the VGGF model

As we know, CNNs learn features through the training stage and then use such features to classify images. Each convolutional (conv) layer [3] learns different features. For example, one layer may learn about entities such as edges and colours of an image while further complex features may be learned in the deeper layers. For example, a result of conv layer involves numerous 2D arrays which are called channels. In VGGF, there are layers, of them are convolutions and the remaining layers are mixed between Relu, pooling, fully connected and the last layer is the softmax. However, after applying the conv layer to an input image, which has filters with size x, the features can be extracted for classification purposes.

C. Siamese Network for One-Shot Learning

The word "Siamese" means joined or connected. A network that has been popularized given its use for one-shot learning is the Siamese network. A Siamese network [5] is an architecture with two parallel neural networks, each taking a different input, and whose outputs are combined to provide some prediction.

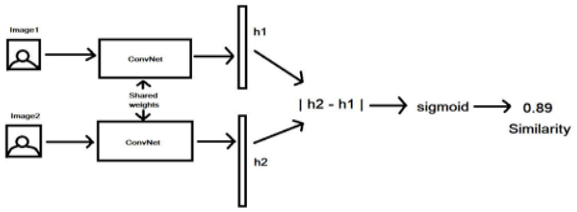

Figure 10: Siamese Network

There are two common ways to find the distance of two vectors: cosine distance and Euclidean distance.

a) Cosine distance

Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. Cosine distance between two vectors A and B is

$$\cos\theta = \frac{A.B}{|A|.|B|}$$

Similarity = 1 - $\cos\theta$

b) Euclidean distance

Euclidean distance or Euclidean metric [7] is the "ordinary" straight-line distance between two points in Euclidean space. With this distance, Euclidean space becomes a metric space.

IV. Implementation & Result

We use python programming language along with necessary development tools and different useful machine learning/deep learning libraries. Python is a great general-purpose programming language on its own, but with the help of a few popular libraries it becomes a powerful environment for scientific computing. We choose python to build our model because python has many highly developed deep learning libraries which helps us building this object detection model easily and more. Python libraries for deep learning:

- Numpy
- Keras
- TensorFlow
- OpenCV
- Matplolib

As development tools, we use Jupyter Lab which is an open-source web application that allows creating and sharing documents that contain live code equations, visualization and narrative text. Uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning and much more it. It is a great tool for exploratory data analysis and widely used for data scientist.

A. Model

We used Keras deep learning libraries to construct this model. Even though research paper is named Deep Face, researchers give VGG-Face name to the model. This might be because Facebook researchers also called their face recognition system Deep Face without blank. VGG-Face is deeper than Facebook's Deep Face; it has 22 layers and 37 deep units. The structure of the VGG-Face model [3] is demonstrated below. Only output layer is different than the image net version we might compare. Let's visualize the model.

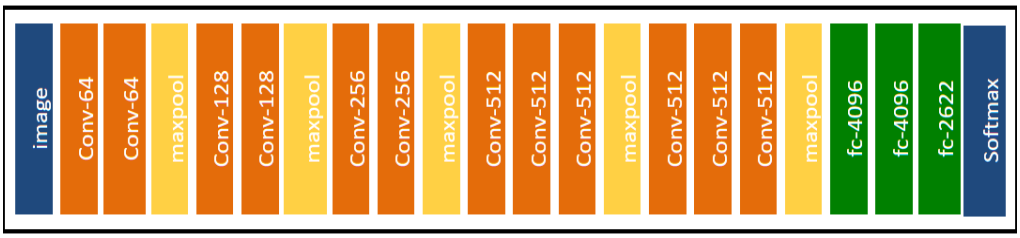

Figure 11: VGG Face Model

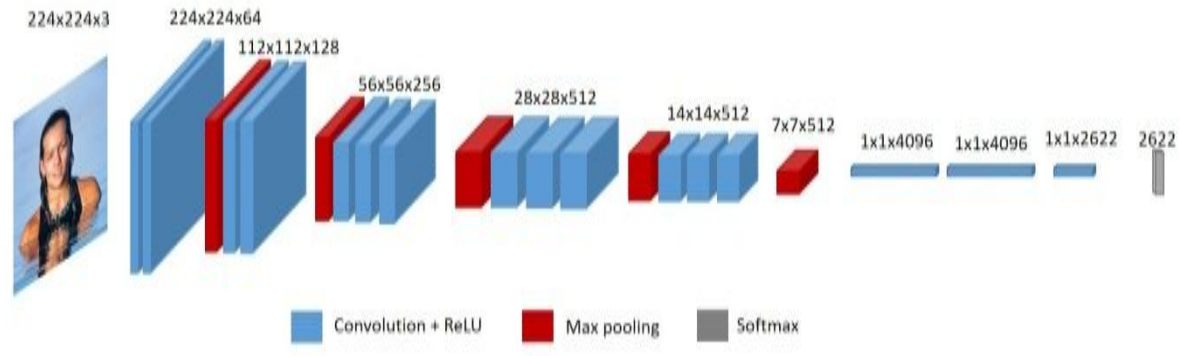

Figure 12: VGG Face Model Visualization

CNN layers extract feature from the image and fed them to the next layers. Output layer give a probabilistic output base on the previous fully connected layer.

B. Preprocessing

Before image fed into the model we have to preprocess the image for the network. Notice that VGG model expects 224x224x3 sized input images. Here, 3rd dimension refers to number of channels or RGB colors. Besides, preprocess input function normalizes input in scale of [-1, +1]. As model expects 224x224x3 sized image, we have to convert our image this format. Keras and Numpy have wonderful function to do this task.

C. Vector similarity

We've represented input images as vectors. We will decide both pictures are same person or not based on comparing these vector representations. Now, we need to find the distance of these vectors. There are two common ways to find the distance of two vectors: cosine distance and Euclidean distance. Cosine distance is equal to 1 minus cosine similarity. No matter which measurement we adapt, they all serve for finding similarities between vectors.

Figure 13: Vector Similarity between faces

D. Training

For training the network we have to set our image database. For this paper we train our model with below images.

Figure 14: Train Image (Anamul'face)

Figure15: Train Image (Phoebe'face)

E. Result

The output of our model mainly varies with the threshold level of the distance matrices. In this model we use 0.40 as threshold value. The number of layers and filters we use the main parameter to improve the accuracy and effectiveness to this model [9]. The more layer we use to get more accurate output but the time consume trade of is the main curse. Here the result:

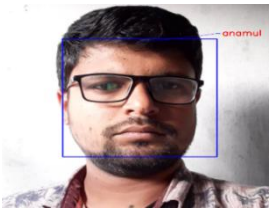

Figure 15 : Test Result-1

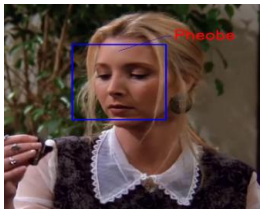

Figure 16: Test Result-2

V. Conclusion and Discussion

Face recognition has always been challenging topic for both science and fiction. A woman has her hair dyed or worn a hat to disguise. Deep learning tasks [8] usually expect to be fed multiple instances of a custom class to learn (e.g. lots of pictures of someone). This makes face recognition task satisfactory because training should be handled with limited number of instances – mostly one shot of a person exists. Moreover, adding new classes should not require reproducing the model. Critical amount of lower light may hamper to get output. The rotation of person from the back is able to recognize the face accurately. The more variation of data also causes error. Such as if we train our model by a person who wears glass but if we test our model without wearing glass this can be caused wrong output. If someone trains the model with beard face and test the model without beard face it also may cause error and vice versa. While our study has revealed many challenges for current face recognition research, the current study has several limitations. One, we did not examine the effect of face image size on algorithm performance in the various conditions. Minimum size thresholds may well differ for various permutations, which would be important to determine. Two, the influence of racial or ethnic differences on algorithm performance could not be examined due to the homogeneity of racial and ethnic backgrounds in the databases. While large databases with ethnic variation are available, they lack the parametric variation in lighting, shape, pose and other factors that were the focus of this investigation. Three, faces change dramatically with development, but the influence of change with development on algorithm performance could not be examined. Fourth, while we were able to examine the combined effects of some factors, databases are needed that support examination of all ecologically valid combinations, which may be non-additive. In this paper we use deep attribute based representation for one-shot face recognition. The deep attribute representations are obtained by fine-tuning a deep CNN [1] for face recognition on data for specific attributes such as gender and shape of face. While, specific face information is challenging, it is far easier to obtain attribute related information.

We observed that the face features when further adapted by various attributes yield consistent improvement in accuracy for one-shot recognition. This was observed for one-shot recognition using one-shot similarity kernel based techniques. In future, we would be interested in exploring the kind of attributes that are useful for improving face recognition [5]. Such as image augmentation technique can apply to improve the performance. Exemplar-SVM based method can be applied.

VI. References

[1] Fei-Fei L, Fergus R, Perona P. "*One-shot learning of object categories*". IEEE transactions on pattern analysis and machine intelligence. 2006 Feb 21;28(4):594-611.

[2] Gosavi VR, Sable GS, Deshmane AK. "*Evaluation of feature extraction techniques using neural network as a classifier: a comparative review for face recognition*". Int. J. Sci. Res. Sci. Technol. 2018;4(2):1082-91.

[3] Lauzon FQ. "*An introduction to deep learning*". In2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA) 2012 Jul 2 (pp. 1438-1439). IEEE.

[4] Balaji SA, Baskaran K. "*Design and development of artificial neural networking (ANN) system using sigmoid activation function to predict annual rice production in Tamilnadu*". Ar Xiv preprint arXiv:1303.1913. 2013 Mar 8.

[5] Yegnanarayana B. "*Artificial neural networks*". PHI Learning Pvt. Ltd.; 2009 Jan 14.

[6] Kutsuna T. "*Linearly Constrained Weights: Reducing Activation Shift for Faster Training of Neural Networks*".

[7] Sharma A. "*Understanding activation functions in neural networks*". Medium. com blog. 2017 Mar;30.

[8] Brownlee J. "*Gentle introduction to the adam optimization algorithm for deep learning*'. Machine Learning Mastery. 2017 Jul 3.

[9] Karpathy A. Stanford university cs231n: Convolutional neural networks for visual recognition. url: http://cs231n. stanford. edu/syllabus. html. 2018.

[10] Stewart M. "*Simple introduction to convolutional neural networks*". Towards Data Science. 2019 Feb, 27.