Open Access
Full Length Article

ORIGINAL ARTICLE

A Novel Method for Face Recognition Based on Features and Gestures

Sumaira Tabassum[1], Qamar Nawaz[2], Isma Hamid[3], Syed Mushhad Gilani[4]

[1]MS Student, Department of Computer science, University of Agriculture Faisalabad, Pakistan.
[2]Lecturer, Department of Computer science, University of Agriculture Faisalabad, Pakistan.
[3]Assistant Professor, Department of Computer Science, National Textile University Faisalabad, Pakistan.
[4]Assistant Professor, University Institute of Information Technology, PMAS, Arid Agriculture University Rawalpindi, Pakistan

ABSTRACT

This work proposes a novel method to recognize the face of an individual by extracting features from a poorly illuminated frontal facial image. The accuracy of the face recognition system is highly dependent on robust feature extraction. For this purpose, a pre-trained deep neural network model 'FaceNet NN4' with a minimum set of weights and fewer number of layers has been used to extract features from a given input image. The model is optimized by using triplet loss function and 128-dimensional encodings for each image have been computed to represent facial features. The distance matrix is generated by calculating the differences between encodings using standard L2 distance formulae. Based on the distance matrix, the relative similarity of image pairs taken into consideration for the recognition task. Reducing weights and number of layers have greatly contributed to the model's effectiveness in terms of time while achieving a good accuracy rate. All the experiments are carried out on the YaleFace dataset. Experimental evaluation and analysis exhibit that the proposed method acquires a better recognition rate with improved efficiency.

Keywords: Face Recognition, Facial Expressions, Feature Extraction, Deep Learning, Deep Convolution Neural Network Model

Author`s Contribution
[1,2,3,4]Manuscript writing, Data Collection, Data analysis & interpretation, Conception, synthesis, planning of research and discussion

Address of Correspondence
Qamar Nawaz
Email: qamar@uaf.edu.pk

Article info.
Received: Jan 21, 2019
Accepted: Dec 07, 2019
Published: December 30,2019

Cite this article: *Tabassum S, Nawaz Q, Hamid I, Gilani SM. A Novel Method for Face Recognition Based on Features and Gestures. J. Inf. commun. technol. robot. appl.2019; 10(2):25-35.*

Funding Source: *Nil*
Conflict of Interest: *Nil*

INTRODUCTION

Face Recognition is a process of identifying the identity of a person by their face and is an emergent area of research in many of today's modern world security systems. Humans can do this task very easily even under the varying lighting conditions or in an uncontrolled situation, but this task is difficult to execute using computers due to the involvement of complex mathematical procedures. Face Recognition also plays a very important role in biometric identification [1] which refers to a method of automatically identifying or verifying an individual by their traits or considering their physical characteristics.

As every individual has a unique face with distinctive facial features [2]. Therefore, distinguish a face on the

bases of facial features and gestures [3], particularly in an uncontrolled condition is emergent research problem. Predominantly, Face Recognition is conceived as a process that detects faces [4] from an image or series of images, bringing off some alignment, extract features and then perform recognition tasks. The aim of face recognition procedures is to design and develop an efficient algorithm that uproots the eminent features, characteristics, and textures of face to make a comparison between new face images with the stored ones, and afterward perform recognition tasks in a revamping way.

Any Face recognition process mainly described by the following steps (i) Face acquisition (ii) extraction and representation of facial features data [5] and (iii) perform a recognition task, where pre-processing may aid in achieving the good quality of recognition. Face acquisition tasks can be further divided into two main steps i.e. face detection [6][7] and Head-pose estimation [8]. Before performing the recognition task, it is indispensable to detect whether a face is present in the image or not [9]. Face detection deals with identifying the sizes and locations of faces in digital images, which serve as a key steps for processing information of a face. It has been colossally applied in Human-Computer Interaction (HCI), identity authentication, pattern recognition, and video surveillance, etc [10].

After detecting a face in an input image, the facial features are extracted to recognize a face. The accuracy of the FR system extremely relies on robust feature extraction method. Facial Expression Recognition (FER) procedure depends either on geometry-based features, appearance-based features or aggregation of both. Geometric-based features are described in terms of the shape of the face and extracted from geometrical elements such as the nose, eyebrow, and mouth, etc. These features include edge features, corner features, ridges, blobs, image texture, salient points and so on. Appearance-based features are described in terms of the texture of the face and are caused by different expressions such as wrinkles and furrows etc. Figure 1 shows pictorial representation of FR process.

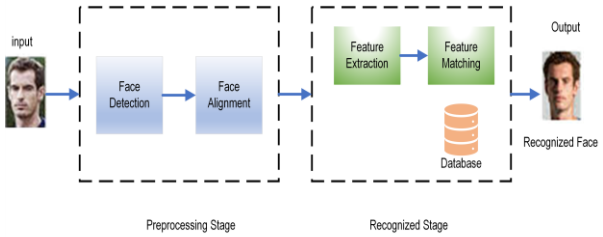

Figure 1.Face Recognition Process

Although face recognition has ultimate advantages over other biometric techniques, still improvements are required to achieve desired outcomes in terms of accuracy and efficiency. Various expostulations which are needing to be solved while developing face recognition system include: Illumination variations, Frontal vs pose variations, Expressions Variations, Aging, and Occlusions. In the domain of face recognition, the final goal of researchers is to design and develop an innovative face recognition system with satisfactory performance. Over the past decades, many studies have been conducted especially with the advent of Deep Learning (DL) in many domains [11][12]. In this context, many methodologies have come in view, but in recent years DL and Deep Convolutional Neural Network (DCNN) gained noticeable attention because of their efficiency and high accuracy rates [13]. Consequently, vast progress and encouraging outcomes have been attained in the domain of face recognition, and current systems have obtained a noticeable degree of maturity while operating under many uncontrolled conditions. But researches are still unable to operate adequately in all non-favorable situations that are usually come across by real-world applications.

Despite the success of several face recognition systems, some of the issues still remain to be resolved that critically exert influence on the accuracy of the system. Two of the most influential issues in any face recognition systems are (1) Illumination variations and (2) Expression variations. Due to the variations in lighting conditions, a similar face may appear differently. More explicitly, we can say that the variations induced by the illumination can be greater than the differences among individuals, which may lead to misclassifying the identity of an input image.Images with diverse Facial Expressions (FE) such as smile, sad, disgust, crying, with opened and

closed eyes, and even slight variations in FE can drastically influence system performance [14].

In this paper, we proposed a novel method for face recognition that is based on extracted features through a pre-trained deep 'FaceNet NN4' model. The study is focused on the frontal normal images and face images with multiple expressions and illumination variations. The Partial occlusion effect from input images has been reduced by performing image enhancement. Haar cascade classifier is used to detect a face from the input image. Facial features are extracted using a pre-trained model with a reduced set of weights and fewer number of layers to reduce model size while freezing the weights of all other lower-level layers of the original model [15]. The model is optimized with the help of a triplet loss function, to validate its accuracy and efficiency. Extracted features are represented as 128-D encoding. Matching scores between facial features of different face images are computed through L2 distance formulae and a Distance matrix [16] is computed. All the experiments have been performed on the YaleFace dataset. From experimental evaluation, it is found that using a reduced size model, a better recognition rate with improved efficiency has been achieved. The proposed study is focused on the inference phase rather than the training of the model to test the accuracy of sample data [17].

The rest of the paper is organized as follows: Section II gives a brief literature review of the face recognition task. Section III discusses the proposed methodology. Section IV describes experimental evaluation and results of the reduced size DCNN model. Section V gives a brief conclusion of the proposed work.

LITERATURE REVIEW

Primarily face recognition task is based on face detection and robust feature extraction and their representation. For face detection, Deep Belief Networks (DBN) is applied to probe the task correlation between different parts of the face and satisfactory results were attained in case of face pose variations and occlusion [18]. The idea of DBN is also used for classification purposes for X-Ray images [19]. An extensive review of the dominantly used facial landmarks detection algorithm is conducted and then these algorithms are classified into three leading categories: Constrained Local Model, holistic method, and Regression-based models [20].

The precise facial features extraction and their efficacious representation are crucial steps to attain better performance for Face Recognition [21]. For example, the effect of the Histogram of Oriented Gradients (HOG) descriptor has been analyzed in the domain of the FER problem to recognize facial expressions [22]. A combined approach that uses the advantages of the Facial Action Coding System (FACS) and Linear Binary Pattern (LBP) has been presented for the representation of FE features from a rough scale to the finer one [23]. Many Deep NN [24] models have been presented to attain eminent features for the face recognition task. For vigorous features extraction and representation, an innovative Deep Neural Network (DNN) model [25] was developed by reducing the reconstruction error. An innovative algorithm that uses deep convolutional features was designed for face verification under unconstrained conditions [26]. For food recognition, a multi-view multi-scale features aggregation method was presented that was based on deep visual, mid-level and high-level semantic features [27].

Three diverse DL models for high-level feature extraction are investigated in face verification tasks [28]. The effect of curvelet and wave transformation has been studied in terms of raising the performance of Elman Neural Network (ENN) [29] for face recognition. Robust hierarchical convolutional features were exploited in order to perform visual tracking [30]. A deep approach was employed in conjunction with a k-nearest classifier to mitigate class imbalance in various tasks as FR and attribute prediction [31].

RESEARCH METHODOLOGY

The main objective of the proposed method is to recognize a face based on extracted facial features from a given frontal input image with partial occlusion effect. Face Recognition is a task of positively identifying a face from the input image by making a comparison with a dataset of already captured images. For effective analysis and to reduce the partial occlusion effect, input images are enhanced on the base of the pixel's average brightness values. A face is to be detected from a preprocessed input image by employing the Haar-

Cascade Classifier. Robust facial features are extracted by using a pre-trained model with a minimum set of weights and a reduced number of layers to accommodate the small-scale dataset. These facial features are epitomized as 128-D encodings of the face image. In the end, L2 comparison is to be performed between embeddings of the specified input image with the stored dataset. After that, the matching scores, in the form of the distance matrix is computed. By considering the calculated distance, the relative similarity of image pairs is considered to perform the recognition task as shown in Figure 2. Triplet loss function is incorporated to compile and optimize the model and to evaluate its validity and accuracy by reducing the error in prediction. The proposed method comprises the following steps:

Step 1. Enhance the input image (Pre-Processing)

Step 2. Detect the face from the preprocessing image.

Step 3. Extract features through a pretrained model with a reduced set of weights and the minimum number of layers and representing these features as 128-D encodings.

Step 4. Compute L2 distance among extracted encodings to generate distance matrix.

Step 5. Perform face recognition tasks

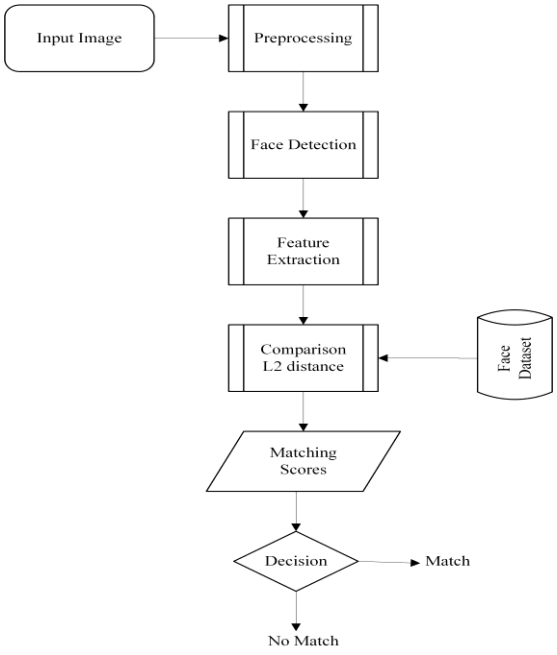

Figure 2. Designed Methodology

3.1. Preprocessing

The objective of pre-processing is to improve imagery data by enhancing the effect of the images or discard unrelated or redundant features without changing the nature of the image. The use of an effective pre-processing algorithm may have remarkable positive effects to get better quality of facial feature extraction.

In our study, as a pre-processing, we perform contrast enhancement on input images that are captured with dark background in order to reduce partial occlusion effect. At first overall brightness value of an image has been computed at the pixel level. If the calculated brightness value is less than the threshold value, then the enhancement effect has been employed. This may resolve the illumination effect to some extent. The contrast enhancement of images has been performed with the help of a mathematical expression based on just multiplication and addition operations using equation 1.

$$O = \alpha * I + \beta \quad (1)$$

Where O is the output image and I is input image. And α represents image contrast. The value of α is equal to 1 means no contrast, if its value is greater than 0 and less than 1 mean low contrast, and if its value is greater than 1 means high contrast. Where β represents image brightness, and its value ranges from -127 to +127.

3.2. Face Detection

Face detection refers to whether a face is present in the input image or not. Before applying the recognition task, it is vital to detect a face from an input image because any other element except face in the image may deteriorate the accuracy of the Face Recognition algorithm. Many algorithms exist for face detection. In our study, we have used Haar-cascade classifier for face detection and extraction of Haar-like features, because of its improved accuracy.

Haar Cascade method is an effective feature-based machine learning approach for detecting a face in the input image [32]. In Haar features, first, the pixel values inside the black area are added together; then the values in the white area are summed up, as shown in Figure 3. Then the total value of the white area is subtracted from the total value of the black area. This result is used to categorize image sub-regions. Each feature is represented by a single value and can be calculated using equation 2.

Haar feature value = sum of pixel (Rect_White) - sum of pixel (Rect_Black) (2)

Different weak classifiers are then further combined to achieve a strong classifier to detect a face using equation 3.

$$F(x) = w_1 f_1(x) + w_2 f_2(x) + w_3 f_3(x) \quad (3)$$

Where *f(x)* represent weak classifier of vector x, and w is their corresponding weights.

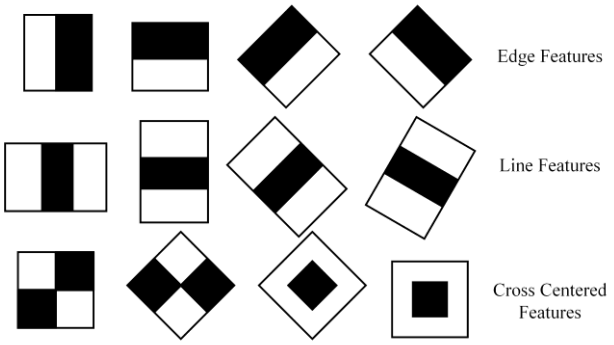

Figure 3. Haar-like features

3.3. Extracting Features

In our study, facial features are extracted from the input image through a pre-trained FaceNet Deep Neural Network Model based on work proposed in [33]. Our contribution is based on NN4 architecture that accepts an input image of size 96×96 with a reduced number of layers and fewer weights than the original one. While weights of all other layers are kept frozen to extract more general features by considering the idea of Transfer Learning [15].

The proposed methodology based on NN4 architecture of Deep FaceNet model. The core component of this model is the use of triplet loss as an embedding function. Based on triplet loss function, FaceNet model directly trains its output to be a 128-D embedding space [34][35]. The NN4 architecture of the FaceNet model is based on Inception modules which are considered its pivotal parts. The original model has seven Inception modules to extract features. We have reduced the model size by eliminating one inception module and its corresponding weights. Therefore, the proposed model uses a lesser set of weights and is more efficient in comparison to the original model and retrains almost same accuracy. The following subsection gives detail of each layer involved in the model.

- **Input Layer**

The input layer defines the format of the input image with no parameter to learn. The model takes input image of size 96 x 96 having the shape (m, Nc, N_H, N_W) = (m, 3, 96, 96). Where m represents no. of a batch of input. Nc represents the number of channels of the input image. Its value is 3 as RGB with 3 colour channels. N_H represents the height of the input image and N_w represents the width of the input image.

- **Convolutional Layer**

The convolutional layer is the very first layer in any model and considered as core block of any CNN [36]. The importance of a convolutional layer is to extract activation features from the given input image by applying some filters. A filter is just a vector of weights through which input is convolved. The convolutional layer takes a punch of filters and applies to a given input image to create different activation features. In the convolutional layer, the convolution operation is a building block which is basically a mathematical linear operation that is applied on two inputs and generates an output. The output size of the convolutional layer is calculated using equation 4.

$$O = \frac{(I - K + 2*P)}{S} + 1 \quad (4)$$

Where O and I represent the size of output and input feature map respectively. K represents the size of the kernel. And P and S represent padding and striding, which are two significant parameters of the convolution operation. The striding parameter controls how many numbers of units a filter shifts each time while convolving around an input matrix. By default, the value of this parameter is 1 which mean it process each and every pixel, so dimensions of output are the same as the input. Whereas padding refers to wide convolution and adds symmetry of zeros around the border of the input matrix to preserve the information. The number of learning parameters in each convolutional layer is calculated using equation 5.

$$param = channel\_in * filter\_size * F + F \quad (5)$$

According to the equation 5, number of parameters are 9474 in conv1 layer when input image has three channels, kernel size is 7x7, and number of filters are 64.

- **Non-Linearity (ReLu)**

Principally, convolution is a linear operation. Hence, it is vital to add some non-linearity in the model. For employing non-linearity, the model uses the Rectified

Linear Unit (ReLu) [37] as an activation function. The ReLu function simulates a node only when its input value is greater than a certain quantity otherwise it will generate 0 as represented in equation 6.

$$f(y) = \begin{cases} 0 & if\ y \le 0 \\ y & otherwise \end{cases} \quad (6)$$

- **Pooling Layer**

The pooling layer is another nonlinear layer to fetch pooled features from convolved ones and predominantly used for down-sampling [38]. When this layer is added to a model, it reduces the dimensionality of images by reducing the number of pixels in the output image while preserving the significant information from the previous convolutional layer. Pooling may be performed as Max Pooling or Average Pooling. The pre-trained Deep model uses Max pooling operation after the intermediate layers and Average Pooling operation in Top Layers. The pooling operation can be performed with any number of filters of any size with similar stride. The output size of the Max pool layer is calculated using equation 7.

$$O = \frac{I - P}{S} + 1 \quad (7)$$

Where O and I represent the size of output and input matrix respectively. P is the pooling size and S is stride.

- **Inception Layer**

This is the core part of the model and is based on GoogLeNet architecture. GoogLeNet is a collection of inception modules and its architecture is wider than deeper. Originally this architecture contains 22 layers with 9 inception modules, based on the idea that a DCNN has been created by stacking up multiple inception layers. This architecture attains state of the art in the domain of detection and classification of objects. Each inception module takes different filters of multiple sizes and performs convolution operation parallel on the same level and concatenate them as a single output. To reduce the computation power, the concept of dimensionality reduction was introduced by adding an additional 1×1 convolution before the large filter size convolution. It has a smaller number of parameters and size as compared to other state of the art methods which makes it practically easier.

- **Encoding Face Images into a 128-D Vector**

In our study, the 128-D encodings for each image are created by feeding these images into a pretrained FaceNet model with a reduced set of weights. The FaceNet model follows the architecture of the inception module and is already trained on an extensive amount of face data. The inception module generates the encodings for each fed image as a 128-D vector in the form of an output matrix of shape (m, 128) as shown in Figure 4. Images that are stored in databases are also passed through the inception block to generate 128-D encodings vector for each dataset image. These feature vectors are then used to make a comparison between two images. There exist adequate similarities among the encodings of images of the same individual and dissimilarities between the encodings of two different images [39].

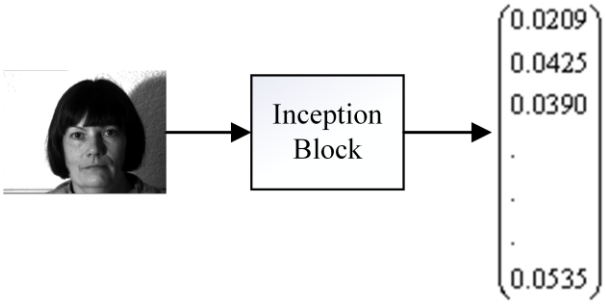

Figure 4. Feature Representation of an input image as 128-D Encoding

- **The Triplet Loss**

The FaceNet model is directly optimized through a novel triplet loss function [33]. A loss function determines how accurate an algorithm models the provided dataset. It is a measure to evaluate the performance of the model by calculating the absolute difference among the actual value and the predicted value by the model. In Artificial Neural Network (ANN) a loss function generates a non-negative value which is used to measure the degree of variation among the actual label and predicted value. The decrease in non-negative value of the function, increases the robustness of the model.

Neural Network (NN) that is designed to generate encodings for features representation uses triplet loss function for the direct training and optimization of the model. The objective of triplet loss is to minimize the distance between the encodings of same person's images (one is called anchor Ai whereas other is called positive Pi). And to maximize the distance between the encodings

of different person's images (called negative Ni) by α margin which is enforced among positive and negative sets. The triplet loss function helps to determine the relative similarity of image pairs. The triplet loss function is described using equation 8.

$$\sum_{i}^{N}\left[\| f(Ai)-f(Pi)\|^2 - \| f(Ai)-f(Ni)\|^2 + \alpha\right]_+ \quad (8)$$

3.4. Computing L2 Distance

After extracting and representing robust facial features as feature vectors, L2 distance is computed to generate a distance matrix. Extracted features are characterized as 128-D encodings. If computed distance among encodings of images is less than threshold, then the identity of a person is recognized otherwise the person is not recognized. This concept is visualized in Figure 5. The similarities among the encodings are measured through the standard Euclidean distance formula using equation 9.

$$\sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \quad (9)$$

Where qi and pi represent encoding vectors of distinct images.

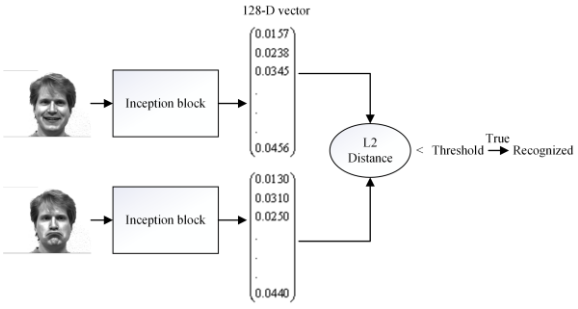

Figure 5. To Find Similarity between Two Encodings by Computing L2 Distance

RESULTS AND DISCUSSION

The performance of the proposed system has been measured on the widely used YaleFaces Dataset. All the images in the data set are resized as 96 x 96 and stored in PNG format because the pre-trained DL model requires all the input of shape 96 x 96 with 3 channels.

4.1. YaleFaces Dataset

The YaleFaces Dataset consists of 165 distinctive grayscale face images of 15 different individuals in GIF format. It comprises 11 diverse images for every 15 distinct individuals with label subject01, subject02, etc. to subject15 as shown in Figure 6. This database is publicly obtainable by the Kaggle website and can be download free from the official website of kaggle in zip format. The pixel values intensity of gray-scale images is usually denoted by numbers in the range from 0 to 255. Where 0 represents black color and 255 means white color. These images for every individual are taken under different lighting conditions with variations in expressions. Our database directory contains one image per subject to reflect the concept of one-shot learning, and the test directory comprises eleven images for every individual to test the validity of the model with a reduced number of parameters in term of its accuracy and efficiency.

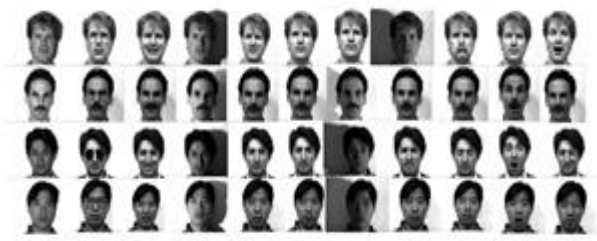

Figure 6. Example of YaleFace Dataset Analysis

The first part of the pre-trained model consists of a stack of layers. An input layer takes an image of a size 96x96 with bit depth 32. In the lower convolutional layer, there are 64 filters of size 7x7 with stride size 2. The concept of Batch Normalization (BN) is employed to mitigate the effect of 'internal Covariate Shift' [40]. Furthermore, this layer is followed by the addition of a non-linear activation (e.g. the ReLu function), Zero Padding with a kernel size of 1x1, and finally, max-pooling to perform down sampling with a kernel size of 3 x 3 and stride size 2. In 2nd convolutional layer, there are 64 filters of size 1x1 with stride size 1x1, along with BN, non-linearity and zero Padding of kernel size 1 x 1. The 3rd convolutional layer has 192 filters with kernel size 3x3 and stride size 1x1, followed by BN, activation function, zero-padding with size 1 x 1, and then perform some down-sampling. Each of these layers takes 2-D input, also known as feature map and translate it into another form with the help of convolution operations, nonlinearity, and down-sampling. Table 1 depicts the detailed summary of lower layers of pretrained DCNN.

Table 1: Summary of Lower Layers DCNN			
Layer Type	DCNN Layers Detail		
	Properties	Learning	No. of Parameters
Input layer	96 x 96	No	---
Convolution 1	64 filters 7x7 dimensions Stride 2 x 2	Yes	9472
Batch Normalization 1 (non-linearity+ zero padding)	Axis =1	Yes	256
MaxPoolng 1	3 x 3 kernel dimensions	No	---
Convolution 2	64 filters 1 x 1 dimensions Stride 1 x 1	Yes	4160
Batch Normalization 2 (Non-linearity + zero padding)	Axis =1	Yes	256
Convolution 3	192 filters 3 x 3 dimensions Stride 1 x 1	Yes	110784
Batch Normalization 3 (Non-linearity +Zero Padding)	Axis =1	Yes	768
MaxPoolng2	pool size=3 stride 2 x 2	No	---

The second part of the model consists of stacks of six inception modules [41]. Table 2 depicts the total number of parameters that are learned during each inception module. It is clear that the number of parameters has increased with each inception module, and more specific features are extracted at top-level layers.

Table 2: Total Number of Parameters Generated by each Inception Module	
Layer Type	Inception Module Details
	Number of Parameters
Inception 1a	165,168
Inception 1b	229,568
Inception 1c	399,712
Inception 2a	548,608
Inception 2b	719,840
Inception 3a	794,688

Table 3 illustrates the total number of parameters that are generated by the proposed system and the total number of trainable and non-trainable parameters and then compare it with the original model. It is clear that the proposed system generates a reduced number of parameters without dropping its accuracy [42]. The result shows that the proposed system is more efficient than the original one as shown in Figure 7 while accelerating on YaleFaces Dataset.

Table 3. Comparison Between Proposed Model and FaceNet NN4 Model		
parameter	Number of Parameters	
	Proposed System	FaceNet NN4
Total params Trainable params Non-trainable params	3,077,616 3,069,968 7,648	3,743,280 3,733,968 9,312

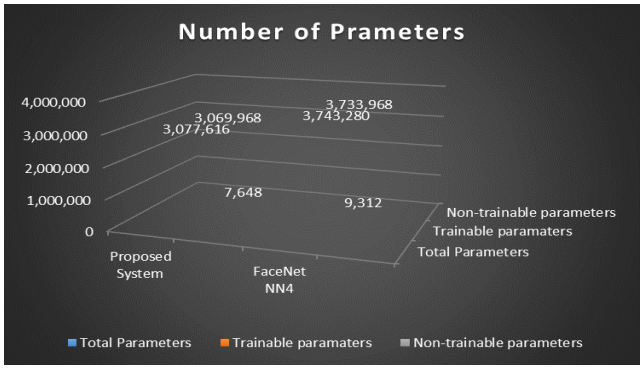

Figure 7. Comparison between Numbers of Parameters

4.2. Experiments and Results

At first, the proposed algorithm computes 128-D encodings for each image stored in database directory where each directory contains one image per individual. The reason to store one image per directory is to employ the concept of one-shot learning [43. Then encodings for the query image have also been calculated and L2 distance is calculated using standard Euclidean distance formulae by making a comparison between the encodings of input image and the encodings of each image stored in the database directory. The minimum distance among encodings depicts that images belong to the same identity and are recognized.

Table 4 depicts the difference among encodings of face images with different variations in facial expressions along with poor background illumination.Subject01 to subject05 are the labels that are assigned to the images

that are stored in the database directory as a unique identity.

When a query image passed through a system, it's encoding is compared with all images stored in the database. Each comparison generates a value that represents how close or far these images are to each other and then figure out the image identity. To accurately handle illumination variations, the images with the poor background are enhanced in brightness.

The system performs an enhancement process only for those images that come with poor background lighting conditions. Results have shown that all the face images of under observation individuals are verified and recognized with better recognition rates. It can also be observed in Table IV that an image belongs to the same individual has the closest distance than those that belong to different images.

On experiments, subject1 gives 100% accuracy, as all undergoes images have been recognized accurately. Table 5 depicts the overall recognition rate for each experimented subject. Eleven distinct images for every individual have been tested. If an image has not been recognized, then its difference with the false recognized image has been computed. In the end, the mean difference for all non-recognized images for a particular subject has been computed to find an error rate. For simplicity, overall 55 images have been tested from which 50 images are recognized accurately, hence gives a 90.9% accuracy rate. Further, it has been observed that the false recognition rate is very small for all non-recognized images, which can be reduced using more robust weights in the future. The proposed system gives better accuracy rates when tested with image set of different individuals, which shows the effectiveness of the proposed system.

Table 4: Distance Matrix with Expressions and Illumination Variations

Query images	Database images				
	Subject 01	**Subject 02**	**Subject 03**	**Subject 04**	**Subject 05**
Image 01	2.0659472	2.3786935	2.5139886	2.3941534	2.5309929
Image 02	1.6492216	1.1440132	2.7322497	1.3582390	1.7593024
Image 03	1.5704883	1.6332557	0.9625925	1.1706589	1.6790965
Image 04	1.3710334	1.4815318	2.0409090	0.9550065	1.1679941
Image 05	1.7746427	1.9647748	2.1431587	1.5037690	0.9523586

Table 5: Overall Recognition Rate with Limited Dataset

	No. of input images with facial expression and illumination variations	Recognized images	Mean Difference between False Recognition	Accuracy %
Subject 01	11	11	0	100.00
Subject 02	11	9	0.237634	81.81
Subject 03	11	10	0.041402	90.91
Subject 04	11	10	0.121559	90.91
Subject 05	11	10	0.163612	90.91

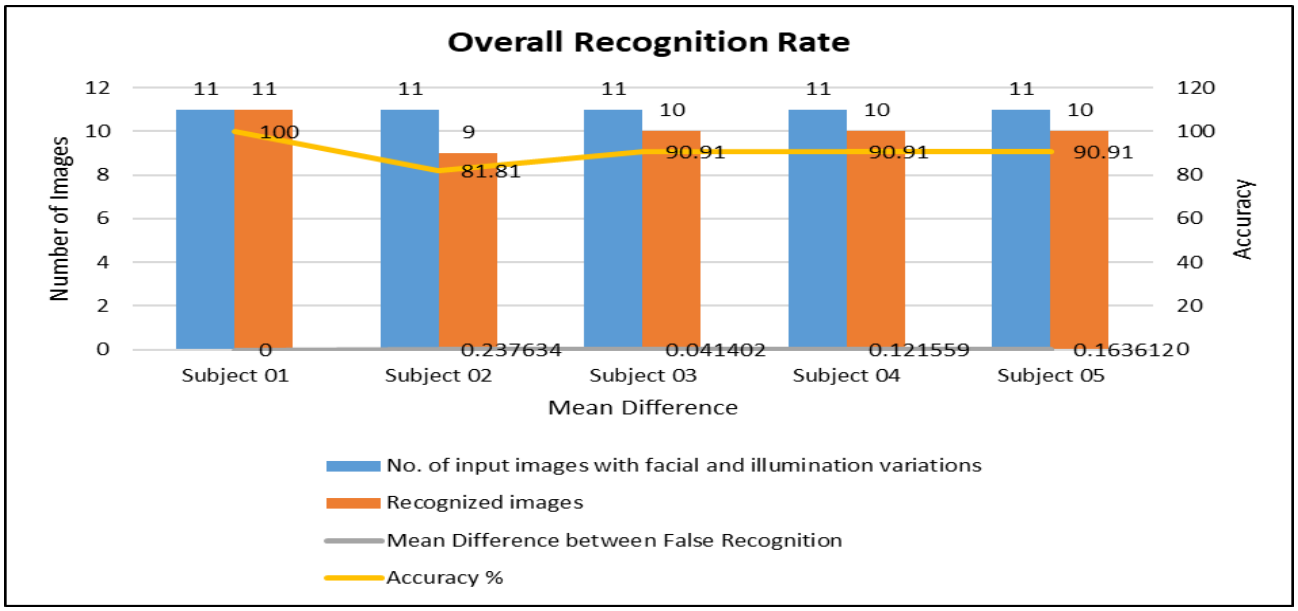

Figure 8. Overall Recognition Rate of Proposed System with Limited Dataset

CONCLUSION

This study proposed a novel pre-trained DCNN model for face recognition and investigated it performance. The proposed model is able to recognize faces in images varying in expressions and background illumination. The proposed model uses reduced weights set and Haar Cascade Classifier to detect faces in images. The features extracted by the model are represented as 128-Dencodings and difference between two encodings is calculated using L2 formulae. A Triplet loss function is used for model optimization. Experiments are performed on publically available face dataset. Experiment results show that the proposed model is efficient and achieved better recognition rate.

Acknowledgement

The research work is supported by Department of Computer Science, University of Agriculture, Faisalabad Pakistan.

REFERENCES

1. V. I. Syryamkim, D. N. Kuznetsov, and A. S. Kuznetsova, "Biometric identification," IOP Conf. Ser. Mater. Sci. Eng., vol. 363, no. 1, 2018
2. Y. Lin et al., "Improving person re-identification by attribute and identity learning," Pattern Recognit., vol. 95, pp. 151–161, 2019.
3. M. Simão, P. Neto, and O. Gibaru, "EMG-based online classification of gestures with recurrent neural networks," Pattern Recognit. Lett., vol. 128, pp. 45–51, 2019.
4. Y. Chen et al., "SimpleDet: A Simple and Versatile Distributed Framework for Object Detection and Instance Recognition," vol. 20, pp. 1–8, 2019.
5. I. Hupont and M. Chetouani, "Region-based facial representation for real-time Action Units intensity detection across datasets," Pattern Anal. Appl., vol. 22, no. 2, pp. 477–489, 2019.
6. C. Garcia and M. Delakis, "A neural architecture for fast and robust face detection," Object Recognit. Support. by user Interact. Serv. Robot., vol. 2, no. 11, pp. 44–47, 2004.
7. C. Chen, W. Wong, and C. Chiu, "A 0 . 64 mm 2 Real-Time Cascade Face Detection Design Based on Reduced Two-Field Extraction," vol. 19, no. 11, pp. 1937–1948, 2011.
8. W. W. Kim, S. Park, J. Hwang, and S. Lee, "Automatic head pose estimation from a single camera using projective geometry," ICICS 2011 - 8th Int. Conf. Information, Commun. Signal Process., pp. 0–4, 2011.
9. Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, "Object Detection with Deep Learning: A Review," IEEE Trans. Neural Networks Learn. Syst., vol. 30, no. 11, pp. 3212–3232, 2019.
10. Y. Ban, S. K. Kim, S. Kim, K. A. Toh, and S. Lee, "Face detection based on skin color likelihood," Pattern Recognit., vol. 47, no. 4, pp. 1573–1585, 2014.
11. Y. Chen et al., "An Instruction Set Architecture for Machine Learning," ACM Trans. Comput. Syst., vol. 36, no. 3, pp. 1–35, 2019.
12. Y. Xu, X. Zhou, S. Chen, and F. Li, "Deep learning for multiple object tracking: A survey," IET Comput. Vis., vol. 13, no. 4, pp. 411–419, 2019.

14. Z. Chiba, N. Abghour, K. Moussaid, and A. El, “Intelligent approach to build a Deep Neural Network based IDS for cloud environment using combination of machine learning algorithms,” Comput. Secur., vol. 86, pp. 291–317, 2019.
15. M. Kawaler and A. Czyżewski, “Database of speech and facial expressions recorded with optimized face motion capture settings,” J. Intell. Inf. Syst., vol. 53, no. 2, pp. 381–404, 2019.
16. J. Tian and Y. Li, “Convolutional Neural Networks for Steganalysis via Transfer Learning,” Int. J. Pattern Recognit. Artif. Intell., vol. 33, no. 2, 2019.
17. H. J. Ye, D. C. Zhan, and Y. Jiang, “Fast generalization rates for distance metric learning: Improved theoretical analysis for smooth strongly convex distance metric learning,” Mach. Learn., vol. 108, no. 2, pp. 267–295, 2019.
18. T. Chang, H. Li, G. Wen, Y. Hu, and J. Ma, “Facial expression recognition sensing the complexity of testing samples,” Appl. Intell., vol. 49, no. 12, pp. 4319–4334, 2019.
19. M. Li, C. Yu, F. Nian, and X. Li, “A Face Detection Algorithm Based on Deep Learning,” Int. J. Hybrid Inf. Technol., vol. 8, no. 11, pp. 285–296, 2016.
20. N. Shankar, S. Sathish Babu, and C. Viswanathan, “Femur Bone Volumetric Estimation for Osteoporosis Classification Using Optimization-Based Deep Belief Network in X-Ray Images,” Comput. J., 2019.
21. Y. Wu and Q. Ji, “Facial Landmark Detection: A Literature Survey,” Int. J. Comput. Vis., vol. 127, no. 2, pp. 115–142, 2019.
22. B. Kong, J. Supančič, D. Ramanan, and C. C. Fowlkes, “Cross-Domain Image Matching with Deep Feature Maps,” Int. J. Comput. Vis., vol. 127, no. 11–12, pp. 1738–1750, 2019.
23. P. Carcagnì, M. Del Coco, M. Leo, and C. Distante, “Facial expression recognition and histograms of oriented gradients: a comprehensive study,” Springerplus, vol. 4, no. 1, 2015.
24. L. Wang, R. F. Li, K. Wang, and J. Chen, “Feature representation for facial expression recognition based on FACS and LBP,” Int. J. Autom. Comput., vol. 11, no. 5, pp. 459–468, 2014.
25. M. Unser, “A representer theorem for deep neural networks,” J. Mach. Learn. Res., vol. 20, pp. 1–30, 2019.
26. Z. Yu, T. Li, N. Yu, Y. Pan, H. Chen, and B. Liu, “Reconstruction of hidden representation for robust feature extraction,” ACM Trans. Intell. Syst. Technol., vol. 10, no. 2, 2019.
27. J. C. Chen, V. M. Patel, and R. Chellappa, “Unconstrained face verification using deep CNN features,” 2016 IEEE Winter Conf. Appl. Comput. Vision, WACV 2016, 2016.
28. S. Jiang, S. Member, W. Min, L. Liu, and Z. Luo, “Multi-Scale Multi-View Deep Feature Aggregation for Food Recognition,” IEEE Trans. Image Process., vol. PP, no. XX, p. 1, 2019.
29. N. Bouchra, A. Aouatif, N. Mohammed, and H. Nabil, “Deep Belief Network and Auto-Encoder for Face Classification,” Int. J. Interact. Multimed. Artif. Intell., vol. 5, no. 5, p. 22, 2019.
30. A. S. Abdullah, M. A. Abed, and I. Al-Barazanchi, “Improving face recognition by elman neural network using curvelet transform and HSI color space,” Period. Eng. Nat. Sci., vol. 7, no. 2, pp. 430–437, 2019.
31. C. Ma, J. Bin Huang, X. Yang, and M. H. Yang, “Robust Visual Tracking via Hierarchical Convolutional Features,” IEEE Trans. Pattern Anal. Mach. Intell., vol. PP, no. c, p. 1, 2018.
32. C. Huang, Y. Li, C. L. Chen, and X. Tang, “Deep Imbalanced Learning for Face Recognition and Attribute Prediction,” IEEE Trans. Pattern Anal. Mach. Intell., vol. PP, no. c, pp. 1–1, 2019.
33. P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 1, no. C, 2001.
34. F. Schroff and J. Philbin, “Paper Reviews Call 002 -- FaceNet: A Unified Embedding for Face Recognition and Clustering.”
35. M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 8689 LNCS, no. PART 1, pp. 818–833, 2014.
36. G. Zeng, Y. He, Z. Yu, X. Yang, R. Yang, and L. Zhang, “InceptionNet/GoogLeNet - Going Deeper with Convolutions,” Cvpr, vol. 91, no. 8, pp. 2322–2330, 2016.
37. J. Gu et al., “Recent advances in convolutional neural networks,” Pattern Recognit., vol. 77, pp. 354–377, 2018.
38. G. E. Hinton, “(n.d.) Vinod Nair, Georffrey E. Hinton, Rectified Linear Units Improve Restricted Boltzmann Machines,” no. 3.
39. O. Yildirim and U. B. Baloglrf, “Regp: A new pooling algorithm for deep convolutional neural networks,” Neural Netw. World, vol. 29, no. 1, pp. 45–60, 2019.
40. M. Bicego and E. Grosso, “On the importance of local and global analysis in the judgment of similarity and dissimilarity of faces,” Image Vis. Comput., vol. 92, p. 103813, 2019.
41. S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” 2015.
42. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, pp. 2818–2826, 2016.
43. S. Xu, S. Zagoruyko, and N. Komodakis, “Exploring weight symmetry in deep neural networks,” Comput. Vis. Image Underst., vol. 187, no. January, p. 102786, 2019.
44. E. van der Spoel et al., “Siamese Neural Networks for One-Shot Image Recognition,” Aging (Albany. NY)., vol. 7, no. 11, pp. 956–963, 2015.