Face Recognition using Eigenfaces and Artificial Neural Networks

Raghunatha Rao, Ghodkari Chowdary, Dept of Computer Science, University of Florida

***Abstract*— Face Recognition finds its place in large number of applications. Due to this humungous use it has been an active area for researchers. Scientists and researchers have been working from past three decades developing novel techniques which increase the efficiency in face recognition systems. These systems are widely used in access and security control to authenticate, social media and advertising to classify and identify people, law enforcement for surveillance and go on. Many techniques based on the density based, geometric based, temporal based, etc. are developed over time and evaluated their performance. In this paper we implement Face Recognition systems one using Eigenfaces and other using Artificial Neural Networks and their results are summarized for recognizing the faces.**

***Index Terms*—Face Recognition, Pattern Recognition, Eigenfaces, Artificial Neural Networks ANN, Principal Component Analysis, FaceNet, Support Vector Machines**

I. Introduction

Face Recognition has been an active problem involving in various industries from past decades. It finds it place in surveillance and automatic recognition by the law enforcement, used in social media for human interaction, security systems for giving access to right people, design of systems with human interaction and so on. The face recognition has brought many researchers from disciplines such as computer vision, pattern recognition, machine learning and image processing. Human images of a same person vary in various factors such as illumination, expression, scale and posture. Researchers are working to overcome these differences in human images from the past two decades in order obtain better accuracy.

Face Recognition involves given picture of a person, it helps us to identify whether there is a person face captured in it, locate his face in the picture and able to identify who he/she is. Face Recognition is achieved by following steps: 1. Face Detection 2. Feature Extraction 3. Face Recognition. Face detection involves to checks if there is human face in the picture, if it exists then locate it. Faces in pictures are taken at different scales and angles, so proper scaling and orientation of the face paths are needed. This is also achieved by the Face detection. The patches of faces from face detection are at illuminations, with different expressions and also may contain clutter and occlusion. To overcome all these problems, feature extraction is done to get better information, reduce the dimension of the images to one single scale, extract the salience features in the images and denoise it. Now comes the Face recognition step, where the face patch of the picture is now compared against face database. A face database is needed for face recognition, this database involves the images on people taken at different poses and angle in different illumination conditions. The features of these images are extracted and stored in the face database which is used in the face recognition. The face patch extraction from features extraction is used to compare over the face classes in the database. Many algorithms have been developed and tested to deal this classification and identification problem in comparing the input picture over face database for Face identification and Face verification. In this paper, we plan to see a literature survey of various pattern recognition techniques developed over time, develop Face recognition systems using eigenfaces and artificial neural networks.

II. Description & Literature Survey

This section gives an overview of various technique on human face that perform very good over frontal faces, but each technique has its disadvantages. One of first approach developed for face recognition is the Eigenfaces [1] at The Media Laboratory, MIT in 1991. The Eigenface algorithm based on the Principal Component Analysis was used to represent the images. Many experiments have found that eigenfaces approach is only suitable for images having the frontal faces but various studies show proven techniques with different poses as well [7]. Another approach is using the neural networks is well prevalent in the industry after explosion of Artificial Intelligence into many fields. The ANN consists of neurons which are arranged in form of layers. The accuracy of these networks is boosted with deep learning and supervisory methods. Use of neural networks can be well suited as it lessens complexity of implementation. It learns from the training samples from images taken during different conditions and increases its accuracy. ANN take more time for training; it is the major hurdle is using the neural networks [8].

Use of Hidden Markov Models has also been successful for face recognition. Markov chain is used to compute probability for sequence of observable events. Faces are divided into features such as eyes, nose, mouth, etc., these are labeled as the states of the hidden markov model. As the HMMs require one dimensional sequence, the images two dimensional need to be converted into 1D sequences [9].

Template Matching has also been evaluated for face recognition. Basic approach can be test image which is represented a two-dimensional array is compared using metrics such as Euclidean distance, with single template representing whole image. We can use more than one template from different viewpoints to represent a human face [10].

Support Vector Machines (SVM) are also applied to face recognition systems. SVM is supervised learning technique that performs classification and regression over the input. SVMs can give better results when used over a large data set is used for training [11].

In this paper, we try to implement face recognitions using the eigenfaces and artificial neural networks and summarize their results.

III. Data Set

We have chosen 'Labeled Faces in the Wild', from Computer Vision Lab, University of Massachusetts. This data set consists around 1300 images of faces that are collected over the web. Each picture has been labeled with the name of person it has. Around 1680 of people in the data set have more than one picture. We have chosen this dataset for its large volume of images of different people. We shall divide the data set into testing and training data sets with better training ratio, because more training gives better results in recognition.

Figure 1 Source: Sample Images from dataset. Ref [18]

IV. Eigenfaces for Face Recognition

Face recognition using eigenfaces was invented around 30 years ago at Media Laboratory, MIT. The approach is to transform the face images into sets of feature images called eigenfaces. The eigenfaces are the principal components of the training set. The recognition of the new image is done by projecting the new image on the space formed by eigenfaces and then we classify the image by comparison the features in the eigenfaces space and with position of the features in the new image.

A. Introduction

A general face recognition system works by notion that it has the get the essential features in an image and transform it in an efficient way. This transformed image is compared against a database of similar transformed images. So, mathematically we calculate the eigenvectors of the covariance matrix of the set of images. These eigenvectors are considered as set of features which collectively denote the variation in the face images. Each image in the collection contributed to the eigenvector according to its features, this eigenvector collectively called eigenface.

Every picture in the training data can be shown as the linear combination of the eigenfaces. Therefore, the number of eigenfaces is same as the number of images in the training set. We can also approximate the eigenvectors which have more contribution on to the eigenface accordingly. So, this based on the idea that collection of the faces can be formed from collection of weights from each face and set of standard images.

The entire process is summarized as:

- We calculate the eigenfaces from the training set of face images, this forms a face space.
- Whenever a new image is encountered, according to input image we find a set of weights and M eigenfaces by superimposing our input image onto each of the eigenfaces.
- If the image is close to the face space, we conclude there is a face in the image.
- If the input image has a face, then we classify the weight pattern as the known face or an unknown one.

B. Algorithm

Let A = {a_1, …, a_n} be random vectors, a_i belongs to R_d

- Compute the mean μ.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} a_i$$

- Compute the covariance matrix S.

$$S = \frac{1}{n}\sum_{i=1}^{n}(a_i - \mu)(a_i - \mu)^T$$

- Compute the eigenvalues λ_i and eigenvectors ν_i of S.

$$S\nu_i = \lambda_i \nu_i \, , i = 1,2, \dots , n$$

- The eigenvectors are the ordered based on their descending values. The principal components are the eigen vectors corresponding to k largest eigenvalues. The principal components of observed vector a are given by

$$y = W^T(a - \mu) \; where \; W = (\nu_1, \dots, \nu_k)$$

- The reconstruction from the PCA basis is given by:

$$a = Wy + \mu, where \; W = (\nu_1, \dots, \nu_k)$$

- The face recognition is gone by [12]:
  - Training samples are projected into the PCA.
  - The input image is also projected into the PCA.
  - Now we find the nearest neighbor between the projected training images and the query image.

C. Implementation

- Read all the faces {A_1, ..., A_n} from the training set. These faces are taken under different conditions and must be normalized and resampled to a common pixel resolution (r x c). This is preprocessing for our implementation.

- Now represent every image A_i as a vector G, which is of size 1 x rc. This vector corresponds to r x c image A_i.

- Compute the mean face μ. The mean vector consists of average of each image from the training data set.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} a_i$$

- Now we perform the Principal Component Analysis (PCA) in following way.
  - Subtract the mean from each image. Every image from training set A_i must be subtracted from mean face μ.
  - Calculate the covariance matrix S.

$$S = \frac{1}{n}\sum_{i=1}^{n}(a_i - \mu)(a_i - \mu)^T$$

  - Compute the eigenvalues λ_i and eigenvectors v_i of the covariance matrix S.

$$Sv_i = \lambda_i v_i \,, i = 1,2, \dots, n$$

- Choose the principal components. The number of principal components are chosen randomly by setting a threshold on variance.
- The top k principal components are chosen after sorting in decreasing order of the eigenvalues

$$y = W^T(a - \mu) \; where \; W = (\, v_1, \dots, v_k \,)$$

- Representing faces on the basis. Each picture can be represented as a linear combination of eigenfaces. The reconstruction from the PCA basis is given by:

$$a = Wy + \mu, where \; W = (\, v_1, \dots, v_k \,)$$

- Face recognition using the eigenfaces. Given an input image vector, we first project it into the eigen space. Now we find the nearest neighbor between the projected training images and projected query image. We have used Euclidean distance as the similarity metric to measure closest image to input image.

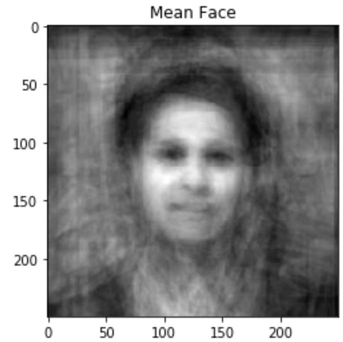

Figure 2 Mean Face generated

D. Results

The above discussed model was executed successfully and following results were obtained on the data. Figure 3 shows eigenfaces which have the highest eigenvalues computed over the dataset.

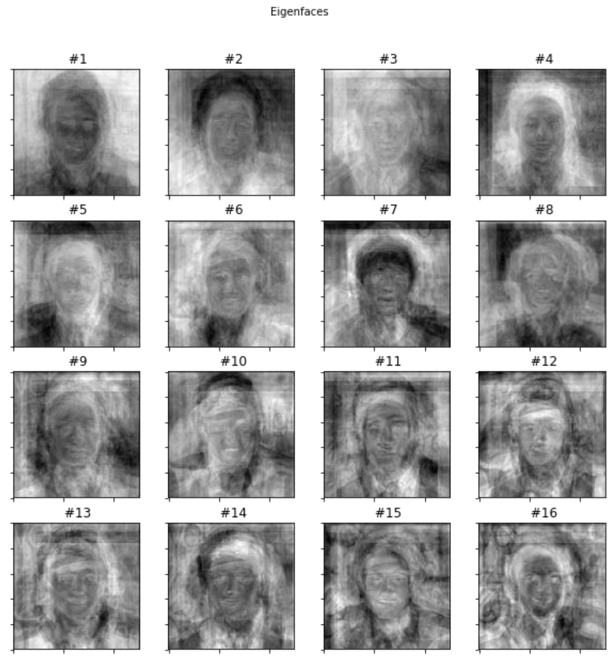

Figure 3 EigenFaces generated

Figure 4 represents graph between number of components and the cumulative variance. We observe that only 25 components are needed to cover 90% of the variance. So, using all 35 components we can recover the essential features of the data.

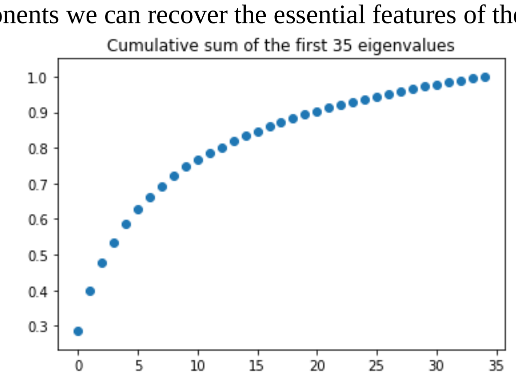

Figure 4 Cumulative Sum of first highest Eigenvalues

Figure 5 shows results of a test case executed to check the performance of the model. Eigenfaces model was able to recognize the faces correctly finding the correct person.

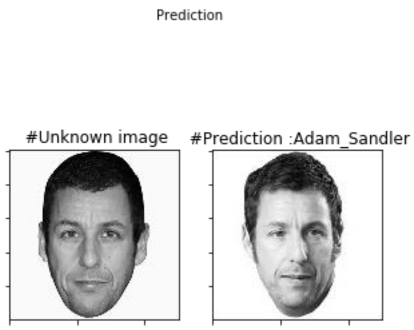

Figure 5 Results of the EigenFaces Algorithm

V. Artificial Neural Networks for Face Recognition

Artificial Neural Networks are tools from machine learning which are inspired from the brain and tend to replicate human learning. Due to their generalization and good learning ability ANNs are widely used in the industry. ANN consists of many neurons connected together as layers. Each neuron has inputs, an activated function and some initial weights. ANN consists of input and output layers, may also contain hidden layers which transform the input into some useful output. ANN works based on the backpropagation algorithm, which allows the neurons in the hidden layers to adjust if the output does not match.

A. Introduction

We are using FaceNet, a Convolution Neural Network developed at Google. It uses two types of CNN architectures, one Zeiler & Fergus Architecture [14] and the other Inception Model [15]. FaceNet facilitates a unified embedding for face recognition, verification and clustering [13]. FaceNet does the mapping of each image into Euclidean space so that the distances in the space represents a similarity metric. This means that images of a person X is closer to other images of person X when compared with images of other people. FaceNet trains the CNNs using Stochastic Gradient Descent with standard backprop uses ReLU as activation function and triplet loss function. ReLU is popular activation function for neural networks apart from Sigmoid and Tanh. Unique idea regarding FaceNet is the learning process, which is done by mapping the images to Euclidean space and create embeddings over usage of any hidden layers. Once the task of embeddings is done the remaining tasks like verification and recognition can be achieved using any other standard techniques. So, the FaceNet just creates embeddings that are directly used for face recognition and verification.

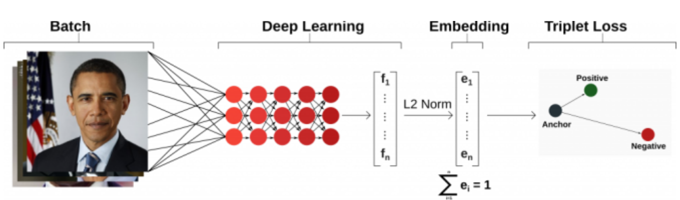

Figure 6 FaceNet Ref [19]

B. Preprocessing

The entire dataset needs to be done some preprocessing work before its use for the neural network. As the images in the dataset are of different sizes and pixels, we need to make sure that the images contain a human face. For detecting face in image, we are using Multi-Task Cascaded Convolution Neural Network (MTCNN) [16] a face detector algorithm, implemented for Keras library in Python. The MTCNN does of job of detecting face in an image, later we crop resize the face portion to common scale for better testing.

Figure 7 Original Image

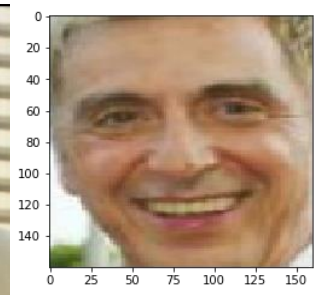

Figure 8 Processed Image

C. Implementation

1. A pretrained FaceNet [17] neural network is being used for our implementation to reduce the overhead time required for training. The FaceNet neural network which is used has been trained over a set of one million images of celebrity faces and has been widely test for good accuracy.

2. We are using this neural network to create embeddings over the preprocessed images which act as feature vectors for our classification.

3. Now, we are using the Support Vector Machine [21] from the Scikit-learn library over the embeddings which are our feature vectors over which our classification is done.

4. The result model is now tested using a random image from the dataset which gives the ideal matching faces for the input.

D. Results

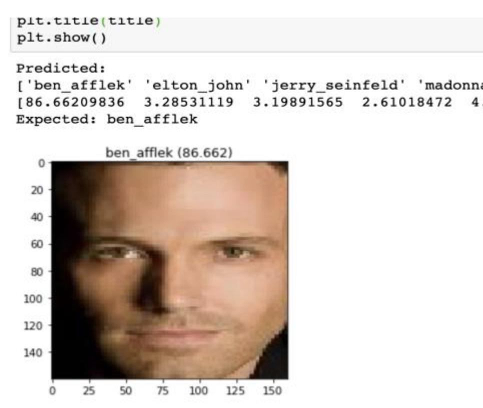

Figure 9 Results of FaceNet

The proposed model has been executed successfully over the dataset. Figure 9 shows results obtained on execution of FaceNet model. The model is working correctly in determining the face in the image.

VI. Summary & Conclusion

In this paper, we have successfully executed face recognition systems one using eigenfaces and other using artificial neural networks. Both models were able to predict input image successfully but sometimes they were inaccurate. We observe that accuracy of model using artificial neural networks around 88% is much better over the eigenfaces one which performs poorly around 63%. The eigenfaces was based on statistical model named Principal Component Analysis (PCA) and the FaceNet was based on the artificial neural networks which is computational model built on artificial neurons.

The success rate of the eigenface model can be increased by training it over relatively larger data of face images or it can also be combined with other existing techniques to create hybrid models. We have used pretrained FaceNet neural network for our implementation, a rigorous training on the input dataset over FaceNet neural network can also be done to achieve better results for face recognition.

References

[1] Turk, M. and Pentland, A., 1991, January. Face recognition using eigenfaces. In Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition (pp. 586-587).

[2] Tolba AS, El-Baz AH, El-Harby AA. Face recognition: A literature review. International Journal of Signal Processing. 2006 Feb;2(2):88-103.

[3] Moghaddam B, Jebara T, Pentland A. Bayesian face recognition. Pattern recognition. 2000 Nov 1;33(11):1771-82.

[4] Lawrence S, Giles CL, Tsoi AC, Back AD. Face recognition: A convolutional neural-network approach. IEEE transactions on neural networks. 1997 Jan;8(1):98-113.

[5] Guo, Guodong, Stan Z. Li, and Kapluk Chan. "Face recognition by support vector machines." In Proceedings fourth IEEE international conference on automatic face and gesture recognition (cat. no. PR00580), pp. 196-201. IEEE, 2000.

[6] Costa, L., Menezes, C., Santos, A., Araújo, M. and Campos, G., 2020, January. A comparative study involving classifiers and dimensionality reduction techniques applied to facial recognition. In Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional (pp. 832-843). SBC.

[7] Sharif, Muhammad, S. Mohsin and Muhammad Younas Javed. "A Survey: Face Recognition Techniques." (2012).

[8] Madan Lal, Kamlesh Kumar, Rafaqat Hussain Arain, Abdullah Maitlo, Sadaquat Ali Ruk and Hidayatullah Shaikh, "Study of Face Recognition Techniques: A Survey" International Journal of Advanced Computer Science and Applications (IJACSA), 9(6), 2018.

[9] F. Samaria and F. Fallside, "Face identification and feature extractionusing hidden markov models," Image Processing: Theory and Application, G. Vernazza, ed., Elsevier, 1993.

[10] R. Bruneli and T. Poggio, "Face recognition: features versus templates," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 15, pp. 1042-1052, 1993.]

[11] Guodong Guo, S. Z. Li and Kapluk Chan, "Face recognition by support vector machines," Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580), Grenoble, France, 2000, pp. 196-201

[12] Face Recognition with Opencv https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html

[13] Schroff, F., Kalenichenko, D. and Philbin, J., 2015. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 815-823)

[14] Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." In European conference on computer vision, pp. 818-833. Springer, Cham, 2014.

[15] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

[16] K. Zhang, Z. Zhang, Z. Li and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," in *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499-1503, Oct. 2016.

[17] Nyoki-mtl/keras-facenet
Nyoki-Mtl - https://github.com/nyoki-mtl/keras-facenet

[18] Guo, Huimin & Schwartz, William & Davis, Larry. (2011). Face verification using large feature sets and one shot similarity. 2011 International Joint Conference on Biometrics, IJCB 2011. 1-8. 10.1109/IJCB.2011.6117498.

[19] A Facenet-style Approach To Facial Recognition
Madio - https://towardsdatascience.com/a-facenet-style-approach-to-facial-recognition-dc0944efe8d1

[20] Face Recognition Using One-shot Learning
Chm - https://mc.ai/face-recognition-using-one-shot-learning/

[21] Joachims, Thorsten. "Svmlight: Support vector machine." SVM-Light Support Vector Machine http://svmlight. joachims. org/, University of Dortmund 19, no. 4 (1999).