Face Recognition using Eigenfaces

Konkimalla Chandra Prakash*
Department of Electrical Engineering
IIT Hyderabad
Email: ee14btech11014@iith.ac.in

Y.M.Srikar*
Department of Electrical Engineering
IIT Hyderabad
Email: ee14btech11038@iith.ac.in

Shantanu Desai #
Department of Physics
IIT Hyderabad
Email: shantanud@iith.ac.in

***Abstract*—We present an approach to the detection and identification of human faces and describe a working, near real-time face recognition system which recognizes the person by comparing characteristics of the face to those of known individuals. Our approach treats face recognition as a two-dimensional recognition problem, taking advantage of the fact that faces are are normally upright and thus may be described by a small set of 2-D characteristic views. Principal Component analysis and Fisher Linear Discriminant methods have demonstrated their success in face recognition. The representations in these subspace methods are based on second order statistics of the image set, and do not address higher order statistical dependencies such as the relationships among three or more pixels. In this paper, we investigate the use of Independent Component Analysis, Non negative matrix factorization, Local Linear embedding, Spectral embedding, MDS and TSNE and their usage as informative representations for visual recognition.We used CNN without any dimensionality reduction techniques and observed that CNN is better for classification.**

Keywords—Face recognition, eigen faces, PCA, Convolutional Neural Network.

I. INTRODUCTION

Face recognition is an easy task for humans. Experiments in [6] have shown, that even one to three day old babies are able to distinguish between known faces. So how hard could it be for a computer? It turns out we know little about human recognition to date. Are inner features (eyes, nose, mouth) or outer features (head shape, hairline) used for a successful face recognition? How do we analyze an image and how does the brain encode it. Developing a computational model of face recognition is quite difficult, because faces are complex, multidimensional, and meaningful visual stimuli. They are a natural class of objects, and stand in stark contrast to sine wave gratings, the blocks world, and other artificial stimuli used in human and computer vision research[1].

Face recognition based on the geometric features of a face is probably the most intuitive approach to face recognition. One of the first automated face recognition systems was described in [3]: marker points (position of eyes, ears, nose, ...) were used to build a feature vector (distance between the points, angle between them, ...). The recognition was performed by calculating the euclidean distance between feature vectors of a probe and reference image. Such a method is robust against changes in illumination by its nature, but has a huge drawback: the accurate registration of the marker points is complicated, even with state of the art algorithms.

* - Equal Contribution of two Authors - Professor and Guide

Some of the latest work on geometric face recognition was carried out in [2].A 22-dimensional feature vector was used and experiments on large data-sets have shown, that geometrical features alone dont carry enough information for face recognition.

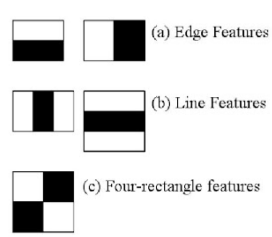
Although face recognition is a high level visual problem, there is quite a bit of structure imposed on the task. We take advantage of some of this structure by proposing a scheme for recognition which is based on an information theory approach, seeking to encode the most relevant information in a group of faces which will best distinguish them from one another. The approach transforms face images into a small set of characteristic feature images, called eigenfaces , which are the principal components of the initial training set of face images. Recognition is generally performed by projecting a new image into the subspace spanned by the eigenfaces (face space) and then classifying the face by comparing its position in face space with the positions of known individuals.Our variant uses Convolutional Neural Networks without any dimensionality reduction because of their recent success in computer vision field.

II. FACE DETECTION

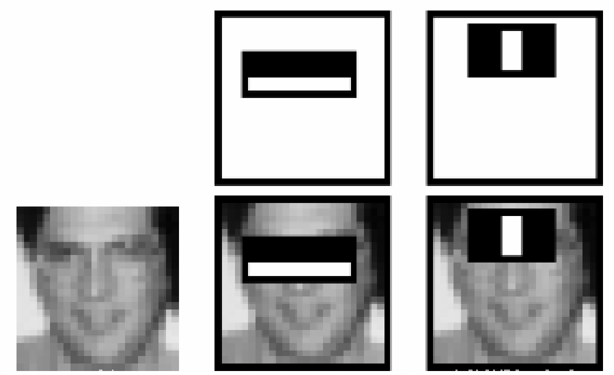
Before recognizing a person in an Image we need to know whether the person is present in the Image, if yes ,we need a bounding box (box around the face) around the person's face. This problem is commonly stated as face detection in literature.

General Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

We use this for face detection here.Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier.Then we need to extract features from it. For this, haar features shown in below image are used.



Amongst all features we calculated, there were many haar features but most of them were irrelevant. For example, consider the image below. Top row shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose.



For this they introduced the concept of Cascade of Classifiers. Instead of applying all the many features on a window, group the features into different stages of classifiers and apply one-by-one. (Normally first few stages will contain very less number of features). If a window fails the first stage, discard it. We don't consider remaining features on it. If it passes, apply the second stage of features and continue the process. The window which passes all stages is a face region.

III. Eigenfaces for recognition

First we describe the commonly used technique Eigenfaces for face recognition.The idea of using eigenfaces was motivated by a technique developed by Sirovich and Kirby [6] for efficiently representing pictures of faces using principal component analysis. They argued that a collection of face images can be approximately reconstructed by storing a small collection of weights for each face and a small set of standard pictures.

In the language of information theory, we want to extract the relevant information in a face image, encode it as efficiently as possible, and compare one face encoding with a database of models encoded similarly. A simple approach to extracting the information contained in an image of a face is to somehow capture the variation in a collection of face images, independent of any judgement of features, and use this information to encode and compare individual face images.

In mathematical terms, we wish to find the principal components of the distribution of faces, or the eigenvectors of the covariance matrix of the set of face images. These eigenvectors can be thought of as a set of features which together characterize the variation between face images. Each image location contributes more or less to each eigenvector, so that we can display the eigenvector as a sort of ghostly face which we call an eigenface. Each face image in the training set can be represented exactly in terms of a linear combination of the eigenfaces. The number of possible eigenfaces is equal to the number of face images in the training set. However the faces can also be approximated using only the best eigenfaces - those that have the largest eigenvalues, and which therefore account for the most variance within the set of face images. The primary reason for using fewer eigenfaces is computational efficiency

The Eigenfaces method described in [4] took a holistic approach to face recognition: A facial image is a point from a high-dimensional image space and a lower-dimensional representation is found, where classification becomes easy. The lower-dimensional subspace is found with Principal Component Analysis, which identifies the axes with maximum variance. While this kind of transformation is optimal from a reconstruction standpoint, it doesnt take any class labels into account. Imagine a situation where the variance is generated from external sources, let it be light. The axes with maximum variance do not necessarily contain any discriminative information at all, hence a classification becomes impossible. So a class-specific projection with a Linear Discriminant Analysis was applied to face recognition in [5]. The basic idea is to minimize the variance within a class, while maximizing the variance between the classes at the same time.

The problem with the image representation we are given is its high dimensionality. Two-dimensional p q grayscale images span a m = pq-dimensional vector space, so an image with 100 100 pixels lies in a 10, 000-dimensional image space already. Thats way too much for any computations, but are all dimensions really useful for us? We can only make a decision if theres any variance in data, so what we are looking for are the components that account for most of the information. The Principal Component Analysis (PCA) was independently proposed by Karl Pearson (1901) and Harold Hotelling (1933) to turn a set of possibly correlated variables into a smaller set of uncorrelated variables. The idea is that a high-dimensional dataset is often described by correlated variables and therefore only a few meaningful dimensions account for most of the information. The PCA method finds the directions with the greatest variance in the data, called principal components.

A. *Algorithm for Eigenfaces*

Let $\{x_1, x_2, ...x_n\}$ be a random vector with observations $x_i \in R^d$

1. Compute the mean μ

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

2. Compute the covariance matrix S

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

3. Compute the eigenvalues λ_i and eigenvectors υ_i of S

$$S\upsilon_i = \lambda\upsilon_i, i = 1, 2, ...n$$

4. Order the eigenvectors by the descending order of their eigenvalues. The k principal components are the eigenvectors corresponding to the k largest eigenvalues.

The k principal components of the observed vector x are then given by:

$$y = W^T(x - \mu)$$

where $W = (\upsilon_1, \upsilon_2, ...\upsilon_k)$. The reconstruction from PCA basis can be done by

$$x = Wy + \mu$$

The Eigenfaces method then performs face recognition by:

1. Projecting all training samples into the PCA subspace .

2. Projecting the query image into the PCA subspace .

3. Finding the nearest neighbor between the projected training images and the projected query image .

IV. Convolutional Neural Network

Inspired by the success of convolutional neural network in computer vision applications such as object recognition, scene labelling, image classification, action recognition, human pose estimation, document analysis, we have employed this to for face recognition. The convolutional neural network (CNN) is a type of feed-forward neural network or a sequence of multiple layers which is inspired by biological processes involved in visual cortex[13]. It reduces the dependency on hand-crafted features and learns useful features from the data itself. It is a combination of feature extractor and classifier and mainly consists of convolution, pooling and fully connected layers. A brief description of the generic layers used in CNN model is given below.

A. Convolution layer

It extracts the useful features by convolving the input feature map with different filters. The weights of the filters are randomly initialized from Gaussian distribution with mean zero and variance close to zero. Then these weights are gradually learnt during back-propagation.

B. ReLu

This is an non linear activation function and generally performs better than sigmoid non linearity. For its response being similar to that of the neurons, ReLU is widely used in deep learning. It is defined as
-

$$f(x) = max(0, x)$$

C. Pooling layer

It is a form of nonlinear sub-sampling which is used to reduce the size of the input feature map. This in turn results in less number of parameters and computations. It also provides slight translation invariance.

D. Fully connected layer

High level reasoning in Neural Networks is done by fully connected layer. It is similar to traditional MLP where all input neurons are connected to previous layer. All the output features of previous layer are taken to determine the correlation between features and a particular class.

E. Softmax

It is used to represent categorical distribution. It is used as a classifier where each class is outputted with some probability then particular class is predicted based on highest probability. It is defined as
-

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \text{ for } j = 1, 2, ...K$$

V. Methods used

A. Convolutional Neural Network

We used a CNN for face recognition whose input image is preprocessed by reshaping into 32x32, as input size for CNN must be constant.Stacking of the above mentioned layers or non-linearities is structured first then they are trained with back propagation.Architectures and accuracies are mentioned in results section.

B. Principal Component Analysis followed by SVM

Principal components analysis is a procedure for identifying a smaller number of uncorrelated variables, called ”principal components”, from a large set of data. The goal of principal components analysis is to explain the maximum amount of variance with the fewest number of principal components. Principal components analysis is commonly used in the social sciences, market research, and other industries that use large data sets.

C. Independent Component Analysis followed by SVM

Independent component analysis attempts to decompose a multivariate signal into independent non-Gaussian signals.This is done by assuming that the sub-components are non-Gaussian signals and that they are statistically independent from each other.

$$x = As$$

Here x contains different combinations of sources ’s’ our goal is to find the inverse of this transformation to get back ’s’ i.e;

$$s = Wx$$

where,

$$W = A^{-1}$$

D. Non negative matrix factorization followed by SVM

Problem with PCA is that Eigen Vectors are defined relative to mean and principle components can be positive or negative.

Non negative matrix factorization applies additional constraint that the W and Y are non negative matrices for data X. Where W and Y are,

$$X = WY$$

E. Locally LInear Embedding followed by SVM

Locally Linear Embedding is unsupervised learning algorithm to embed higher dimensional space to lower dimensionality space while preserving local neighborhood of each point.This relation is determined by relation of each point with k nearest neighbors.

$$E(W) = ||X - WX||^2$$

It will try to minimize is reconstruction error from neighboring points.The W matrix is wisely chosen to not to get stuck in the obvious solution of identity matrix by making only k nearest neighbour corresponding coefficients to be non zero.

F. TSNE followed by SVM

T-SNE is an ML algorithm for dimensionality reduction.It is a nonlinear dimensionality reduction well-suited for embedding high-dimensional data into a space of two or three dimensions, which can then be visualized in a scatter plot.

First, t-SNE constructs a probability distribution over pairs of high-dimensional objects in such a way that similar objects have a high probability of being picked, whilst dissimilar points have an extremely small probability of being picked. Second, t-SNE defines a similar probability distribution over the points in the low-dimensional map, and it minimizes the KullbackLeibler divergence between the two distributions with respect to the locations of the points in the map. We used SVC with grid search to find the best results.

G. Spectral Embedding followed by SVM

Spectral Embedding (also known as Laplacian Eigenmaps) is one method to calculate non-linear embedding. It finds a low dimensional representation of the data using a spectral decomposition of the graph Laplacian. The graph generated can be considered as a discrete approximation of the low dimensional manifold in the high dimensional space. Minimization of a cost function based on the graph ensures that points close to each other on the manifold are mapped close to each other in the low dimensional space, preserving local distances.

Transform the raw input data into graph representation using affinity (adjacency) matrix representation.We can use Unnormalized graph laplacian,

$$L = D - A$$

or normalized graph laplacian

$$L = D^{-1/2}(D - A)D^{-1/2}$$

Eigen value decomposition is done on graph laplacian.

H. Multidimensional scaling followed by SVM

Multidimensional scaling seeks a low-dimensional representation of the data in which the distances respect well the distances in the original high-dimensional space.

In general, is a technique used for analyzing similarity or dissimilarity data. MDS attempts to model similarity or dissimilarity data as distances in a geometric spaces. The data can be ratings of similarity between objects, interaction frequencies of molecules, or trade indices between countries.

VI. Results

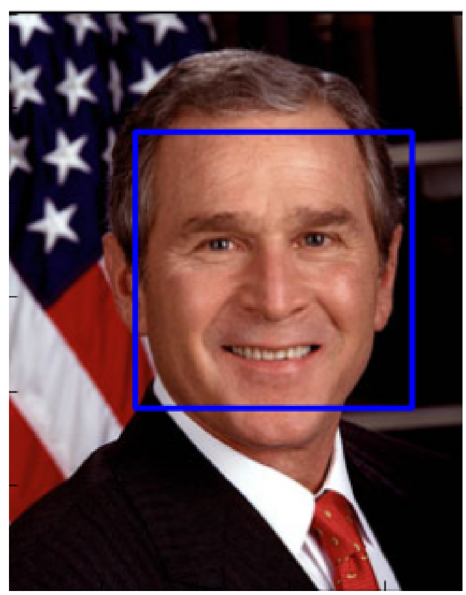
The dataset used in this paper is a preprocessed excerpt of the "Labeled Faces in the Wild", aka LFW. It contains 1288 samples covering 7 classes. For the same dataset, t-distributed Stochastic Neighbor Embedding[7], Non negative matrix factorization [8], Independent component analysis using a fast ICA algorithm [9], Locally Linear Embedding [10] , manifold Multi dimensional scaling [11], Spectral embedding have been used for nonlinear dimensionality reduction [12].Accuracies (if possible) for different methods and specific details are mentioned below.

A. Results for Detection

Viola Jones detector which is used for face detection performs much better and can detect faces in real time. It can detect faces irrespective of their scale and position. A result is shown here by using the image of George Bush. This is not cherry picked but just a random image on the Internet.

Input Image:

Output Image after detection is:



B. Results for Recognition

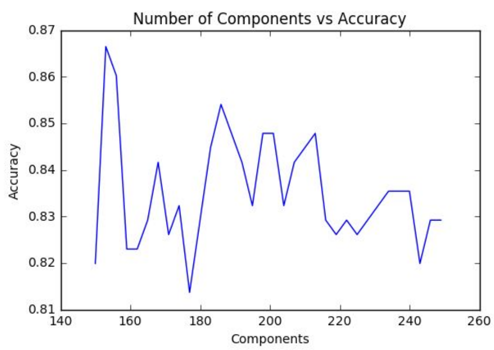
1) Convolutional Neural Network: We used the CNN architecture mentioned in Table 1 for face recognition. The accuracy for face recognition is 88.509%.Here we used one

Architecture for Face Recognition		
S. No.	Layer(type)	Sizes
1	convolution2d_1 (Convolution2D)	(8,3, 3)
2	maxpooling2d_1 (Maxpooling2D)	(2, 2)
3	dense_1 (Dense)	(40)
4	dense_2 (Dense)	(7)

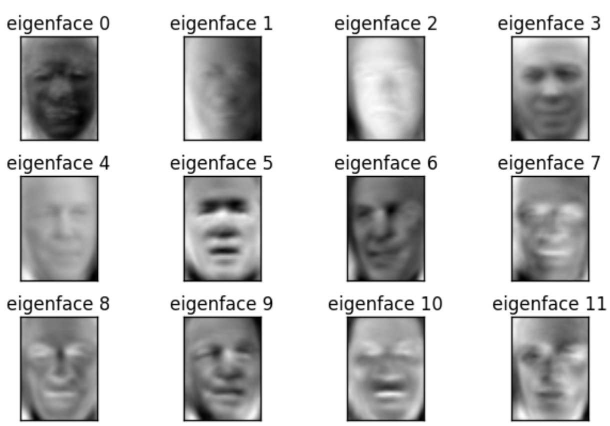
TABLE I. CNN ARCHITECTURE

convolutional layer, one max polling layer followed by 2 fully connected layer's.We get the highest accuracy for CNN when compared to all other methods.

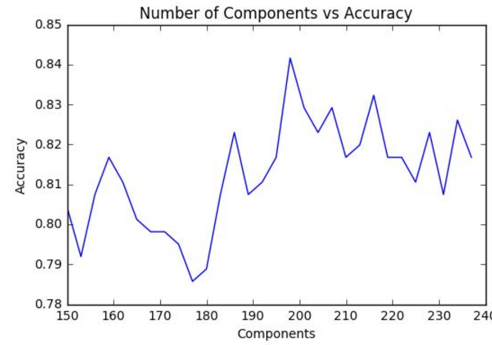
2) Results for PCA: The highest accuracy for PCA using top 153 Eigen value corresponding Eigen vectors.The accuracy is 86.6%.The Eigen vectors on which it is trying to project are more intuitive that highest Eigen Vector component nearly corresponds to average of all the faces and other maximum ones correspond to some more angles and details about the face.



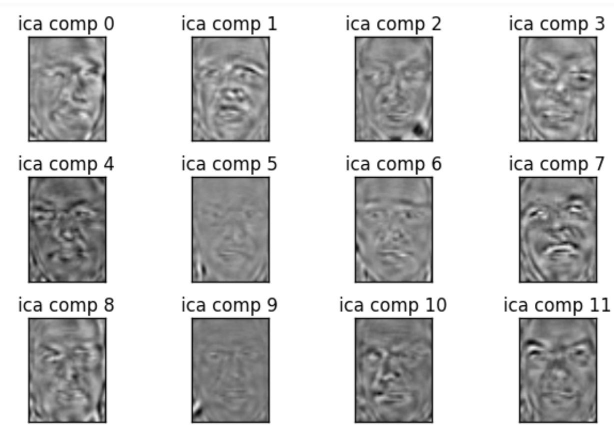
The top pca components corresponding to maximum 12 eigen value's are shown here below. Here we can make up the face just from the eigen face components.The eigen face's have even different "reactions"(expression for face) if seen closely.



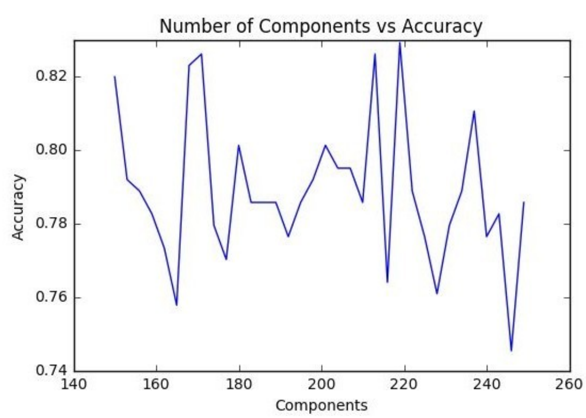
3) Results for ICA: The highest accuracy for ICA can be achieved using 198 component's for projecting onto.Accuracy is 84.16%.This is comparable to PCA percentage but a less than PCA and a more than NMF.



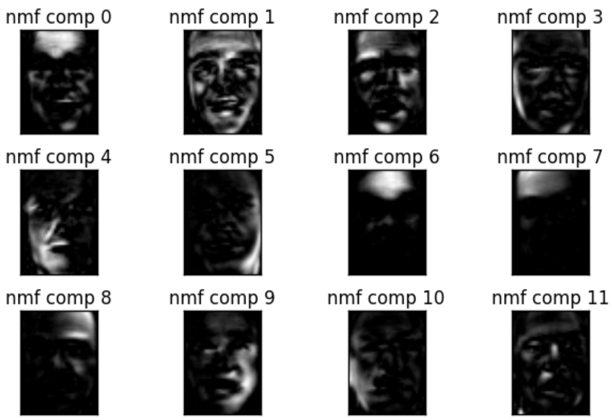
The top 12 ica components are shown here below.The Image does not look as much intuitive as PCA eigen faces.But the structure and relative position of nose, eye's, eye brow's, lips etc; is still maintained.Cheek and some regions does not look much smooth as we find in face's in general.The expression of the faces is even not much clear.



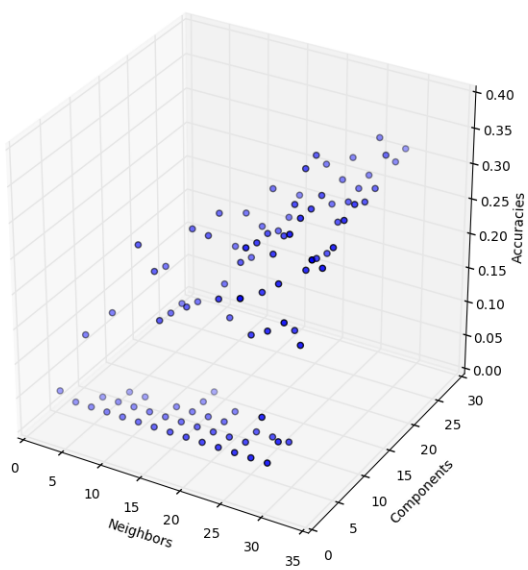
4) Results for NMF: The highest accuracy for NMF can achieved with projecting onto 219 components and the accuracy is 82.9%.This is comparable to PCA and ICA but lesser than both of them.



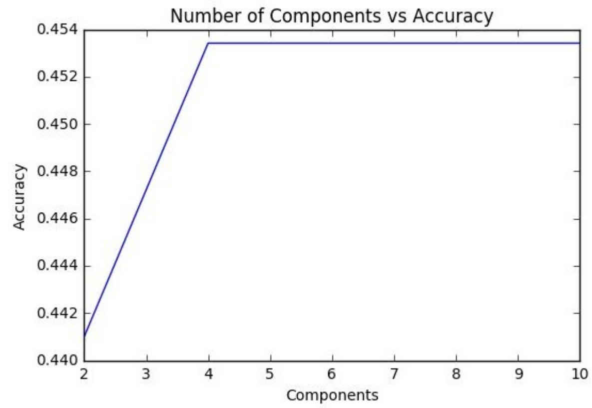
12 NMF components are shown here below.The component's look like they are interested in different part's of the face.and lighting condition's from different angles from components 0,4,5,6,7,8,9, and 10.But more or less all the component's look like faces.



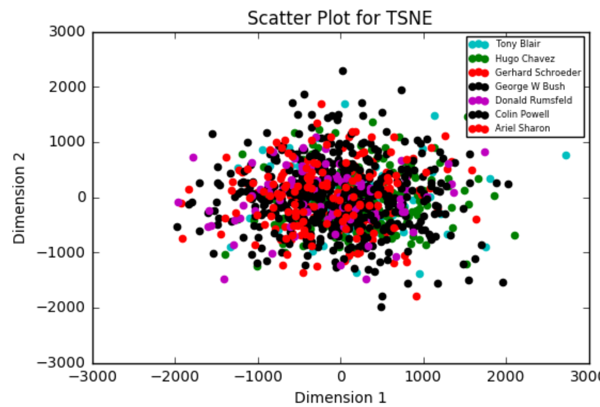
5) Results for LLE: The Highest accuracy for LLE is with number of components as 26 and number of neighbour's as 27.The accuracy is 33.85%.It is not better than PCA, ICA and NMF. Even if the accuracy is less than other non linear dimensionality reduction's like TSNE, Spectral Embedding e.t.c; but we think it is better than them because other non linear dimensionality reduction techniques are always predicting same class disregard of their input.



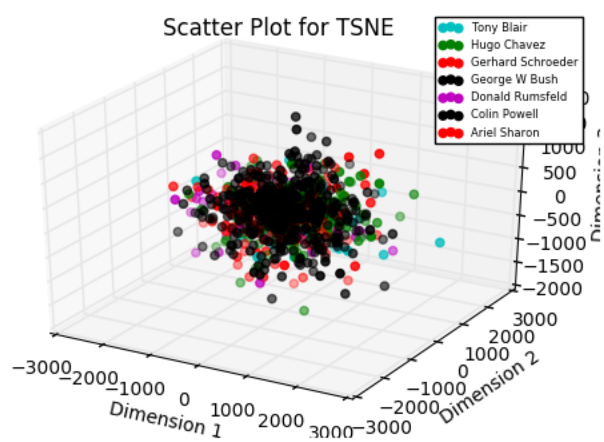
6) Results for TSNE: The highest accuracy for TSNE is 45% is almost constant with the increase in dimension's to which it is being projected onto.The class output is same disregard of the input. The scatter plot after projecting onto 2 and 3 dimensions even shows that it is tough to separate it in that space.



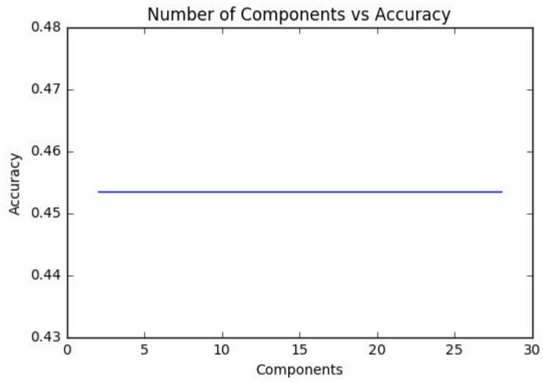
TSNE after projecting it onto 2 dimensions:



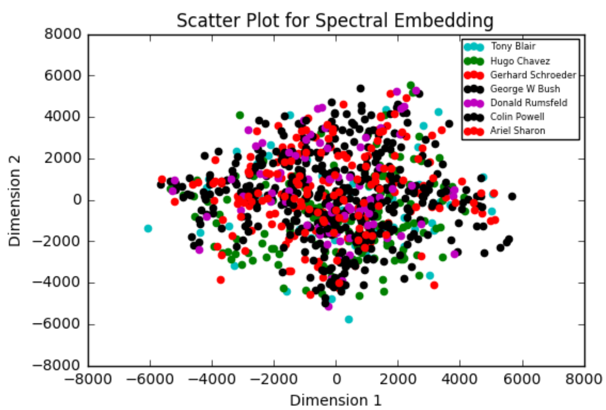
TSNE after projecting it onto 3 dimensions:



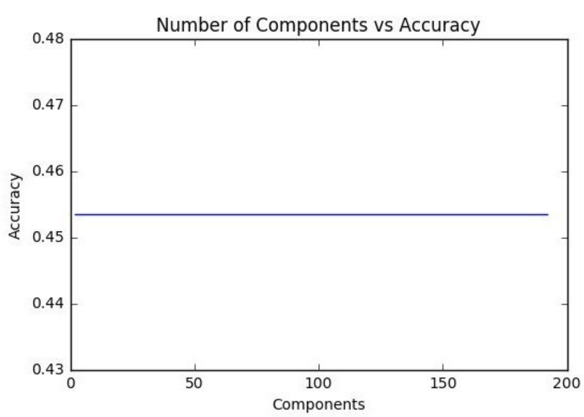
7) Results for Spectral Embedding: The highest accuracy for Spectral Embedding is 45% is almost constant with the increase in number of component's.The class output is same disregard of the input. The scatter plot after projecting onto 2 dimensions even shows that it is tough to separate it in that space.



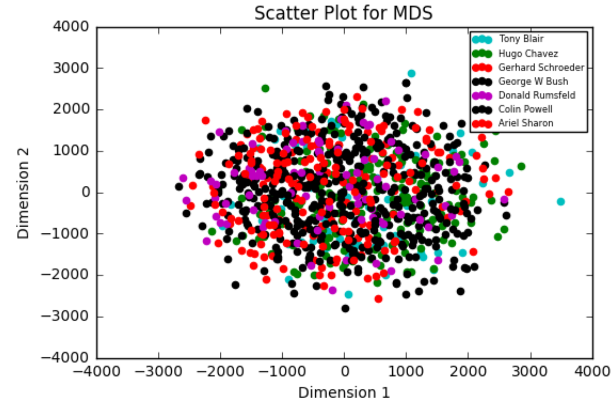
Spectral Embedding after projecting it onto 2 dimensions:



8) Results for MDS: The highest accuracy for Multi Dimensional Scaling is 45% is almost constant with the increase in number of component's.The class output is same irrespective of the input. The scatter plot after projecting onto 2 dimensions even shows that it is tough to separate it in that space.



MDS after projecting it onto 2 dimensions:



VII. Conclusion

Other than NMF, PCA and ICA all other dimensionality reduction techniques did not work well for Face Recognition and took much time for computation of lower dimensional representation, these techniques took into account mainly the distance between them in the high dimensional space and tried to get proportional distance in the lower dimensional space.This may not be suitable for images because a pixel shift along horizontally or vertically leads to a large change as seen from a larger dimensional perspective.Even a small head rotation can lead to a larger change in terms of the distance and correlation's between the images.But CNN's on the other hand can learn these spatial correlation's much better as compared to non linear dimensionality reduction techniques.In this paper we showed different non linear dimensionality reduction techniques followed by SVM classifier and CNN further compared accuracies for face recognition.Even if dimensionality reduction followed by work's well for many cases CNN based approach tend to perform better for Images.

References

[1] Davies, Ellis, and Shepherd (eds.), Perceiving and Remembering Faces, Academic Press, London, 1981

[2] Brunelli, R., and Poggio, T. Face recognition through geometrical features. In European Conference on Computer Vision (ECCV) (1992), pp. 792800

[3] Kanade, T. Picture processing system by computer complex and recognition of human faces. PhD thesis, Kyoto University, November 1973

[4] Turk, M., and Pentland, A. Eigenfaces for recognition. Journal of Cognitive Neuroscience 3 (1991), 7186

[5] Belhumeur, P. N., Hespanha, J., and Kriegman, D. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence 19, 7 (1997), 711720

[6] L. Sirovich and M. Kirby, Low-dimensional procedure for the characterization of human faces, J. Opt. Soc. Am. A, Vol. 4, No. 3, March 1987, 519-524

[7] van der Maaten, L.J.P. t-Distributed Stochastic Neighbor Embedding

[8] Cichocki, Andrzej, and P. H. A. N. Anh-Huy. Fast local algorithms for large scale nonnegative matrix and tensor factorizations. IEICE transactions on fundamentals of electronics, communications and computer sciences 92.3: 708-721, 2009.

[9] A. Hyvarinen and E. Oja, Independent Component Analysis: Algorithms and Applications, Neural Networks, 13(4-5), 2000, pp. 411-430

[10] Roweis, S. & Saul, L. Nonlinear dimensionality reduction by locally linear embedding. Science 290:2323 (2000).

[11] Modern Multidimensional Scaling - Theory and Applications Borg, I.; Groenen P. Springer Series in Statistics (1997)

[12] On Spectral Clustering: Analysis and an algorithm, 2011 Andrew Y. Ng, Michael I. Jordan, Yair Weiss

[13] Ashwin Bhandare, Maithili Bhide, Pranav Gokhale and Rohan Chandavarkar Applications of Convolutional Neural Networks. International Journal of Computer Science and Information Technologies, 2016.