Face Recognition

Abstract:

Face Recognition is one of the long studied and researched problem in computer vision and Image processing area, despite that, it remains as one of the challenging problem of computer vision. The applications of facial recognition is mainly in security systems, cutting across industries of varying nature with huge financial implications and this explains the huge research efforts that has gone through to study the face recognition problem over past several decades.

Through this project, I'm trying to recognize the person in the given input image by using chain of image processing techniques that I learnt in the class. I'm generating a set of features from the training images, training the machine learning algorithm on these features and finally predicting the person on the test image using the predictive model that I built.

Related Work:

There has been a lot of research [1,2,3,4,5] have taken place and are currently in progress in the field of face recognition. All these research happen mainly in feature extraction of the image. These research methods differ in terms of feature generation, each of these research methods, formed their feature in a unique way. By studying these works, it's clear that the classification result obtained by applying various classification algorithms is directly depend on the way in which the features are built.

There are various different approaches that are being followed to recognize faces. Some of the most prominent ones based on geometric methods, template matching, Principal Component Analysis with Eigen faces, Linear Discriminant Analysis using Fisher Faces and Independent Component Analysis.

Geometric Methods:

The paper [1] discusses a geometry-based approach where the size, length, position and shape of facial features such as nose, mouth, eyebrows, eyes, chin, ears are considered while forming the feature along with shape of the face. This kind of approach works best when image is of low resolution and finer details in image is difficult to see.

Template Matching:

The paper [1] also discusses template-matching approach; this is similar to the template matching approach we studied in class. This is the most direct approach, where templates of each class is stored in database and test image is matched against the templates using cross correlation techniques, a vector is returned with correlation score for each template. The class of template with highest correlation score is returned as answer.

Principal Component Analysis:

This is one of the most prominent methods for face recognition. PCA is one of the most common approaches used for dimensionality reduction. "PCA uses orthogonal transformation to convert set of observations of possibly correlated variables into a set of linearly uncorrelated variables called principal components"[3] and these principal components in this context are called Eigen faces. In PCA approach [2] each pixel in a image is considered as a dimension. The concept here is that the variability in the output class can be explained using few features or dimensions. The transformation [2] produces Eigen faces such that the first Eigen face shows the most important features of the training image dataset, and each successive Eigen face vectors shows the next most important features of the image dataset and each of the Eigen faces generated are uncorrelated. The training set is projected on these Eigen faces so that they are transformed into this lower dimension space. Now the test image is projected on this Eigen faces and now projected test image is compared against each of projected training image class and the closest matched class is returned as answer.

Linear Discriminant Analysis:

This is the other most prominent face recognition technique. LDA[1] is similar to PCA as they both look for linear combination of features that explain the data[4], but the difference is that LDA preserves the linear separability [1] between classes while PCA does not. The linear combination of the components obtained using LDA is called fisher faces. LDA creates projections such that ratio of intra class distance and inter class distance is maximized. LDA is supervised model requiring explicit class labels tagged with each image in the training data.

Independent Component Analysis:

It originated from signal processing where it is used to decompose given signal into linear combination of two or more independent signals [5]. It is an enhancement of PCA where it produces an independent image representations rather than uncorrelated representations as in PCA. ICA [5] minimizes both second order and higher order dependencies in the input image, while the PCA uses second order statistics to produce uncorrelated image space.

Challenges Involved:

The face recognition is considered as hard problem due to some of the following reasons

Illumination Variation/ Exposure Problem [6]:

As an image is formed, the external factors like lighting have a huge impact on the appearance of image. This is amplified in cases, where the subject in the image is human due to the reflexive nature of human skin. Many well-known solutions for face recognition fails under varied lighting conditions.

Pose Changes [6]:

Many algorithms work perfectly, when the image has frontal view of the face and performance deteriorates as face angle changes in the image. This is mainly because certain features of the face become hidden in the captured image, as the angle of face to camera changes.

Facial Expression/ Facial Style Changes [6]:

Facial expressions affect the appearance of the face in a significant way, the facial features changes as the expression of the face changes. The presence of facial hair like beard and mustache would also affect the appearance of face.

Eye Glasses:

The wearing of eyeglasses affects the face recognition in significant way as eye remains as one of the prominent feature in face recognition and wearing eyeglasses would hinder the recognition specifically in top portion of the face.

Ethical Consideration:

The dataset I'm using requires the user of the dataset to cite the source and does not allow user to publish the dataset along with other dataset publically without prior permission. Since, I have cited the data source and I have no idea of publishing their dataset elsewhere, I believe I'm not violating any ethical principles.

Business Case:

The face recognition system developed could be used in security systems for verification and identification purposes. In facial verification system the subject can be verified by matching his current face with the previously recorded face. In identification system, the subject can be matched against set of already stored faces of different individuals, to find the subjects identity. This can be of immense use in high security areas like immigration, identifying a person in terror watch list or crime list etc.

Algorithms Used:

The papers [1,2,5,6] has used various algorithms like Principal Component Analysis, Linear Discriminant Analysis, Independent Component Analysis, Discrete Cosine Transform, Locality Preserving Projections, Gabor Wavelets and other neural network based model. Based on the algorithms used they were able to obtain different accuracies, directly dependent on the feature space they generated.

Other Issues:

The research regarding face recognition techniques was quite time consuming. There are no major issues related to project at this stage.

Dataset:

I'm using the Yale face database [1] with 165 GIF images of 15 subjects. Each subject has 11 different images taken under various lightning conditions, facial expressions and configurations. The configurations settings are center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink.

Figure 1: Normal face

Figure 2: Right Illumination

Figure 3: With Glasses

Figure 4: Sleepy

Plan to Solve:

- I'll first separate the training and test images from the image database.
- I'll detect the face portion from all the images using face detection algorithm like Viola Jones.
- I'll remove the noise in the image using filtering techniques like Gaussian filter.
- I'll use Principal Component Analysis for dimensionality reduction.
- I'll project the actual training and test images on the Eigen space created using PCA.
- I'll use the Euclidian distance, to classify the class of test image using the training image and test image projected in Eigen space.
- I'll return the predicted class of test image.

Experiment:

Data Cleaning and Preprocessing:

Data Split:

I randomly split the data into training and testing, using split criteria of 70:30 for training and testing respectively. I created a separate directory for storing training and test data. I Trained the model using this 70% training data and tested it using the remaining 30% test data.

Face Detection and Cropping:

I used cascade object detector from matlab, which uses Viola Jones algorithm under the hood for face detection. Once the classifier detects the face, I cropped the face out of the background image. This is essential because the background image could be part of feature space resulting in poor classification. Particularly hairstyle changes would affect the classification accuracy.

Figure 4: Normal Face

Figure 5: Detected Face

Filtering:

Gaussian filter of size 3*3 with standard deviation of 1 is used to smooth the image. This helped in reducing the high frequency noise in the image.

Figure 6: Normal Face

Figure 7: Smoothened Image

Exposure Enhancement:

The images in dataset were of varying exposure levels, which required exposure enhancement. I tried some of the exposure enhancement techniques learnt in the class like taking square root, histogram equalization and adaptive histogram equalization. Taking square root enhances the darker regions in the image.

Figure 8: Reduced Exposure

Figure 9: Exposure Enhanced

Vectorization:

Vectorization refers to transformation of matrix into column vector. Here each of the image matrix is transformed into column vector, as the Eigen faces algorithm which I'm using for building the model requires image as column vectors.

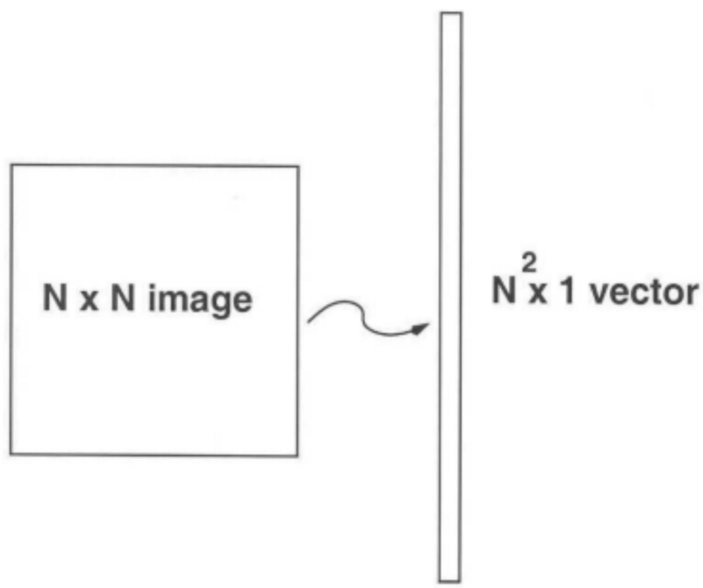

Figure 10: Vectorization (PCA Case Study, Johns Hopkins University)[11]

Algorithm:

I have implemented my face recognition system based on Eigen faces [2] method. The algorithm requires the following key steps to be followed.

Preliminary Requirements:

1. Requires M gray scale images $I_1, I_2 \ldots I_M$
2. Each of the image should be a square matrix of dimension N*N.

Assumptions:

1. The are K most significant Eigen faces which could be used approximate a face where K < M [2].

Algorithm [2]:

1. Vectorize the input images $I_1, I_2 \dots I_M$ of size N*N to column vectors $\Gamma_1, \Gamma_2 \dots \Gamma_M$ of dimension N^2*1.
2. Calculate the mean face vector $\Psi = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i$ [8].
3. Subtract the mean face from each of input face vector.
4. Calculate the co-variance matrix C=AA^T.
5. Calculate the Eigen values and Eigen Vectors from the co-variance matrix C.
6. Select only the Eigen vectors with large Eigen value and discard rest.
7. Now, project the mean centered faces on this Eigen space created using the retained Eigen vectors.

Face Classification [2]:

1. Vectorize the input test image to column vector as before.
2. Normalize the vectorized test image vector, by subtracting the mean face created during training, from it.
3. Project the normalized test image vector into Eigen space created during training.
4. Now any distance metric, say Euclidian distance can be used to find the distance between test image vector and training images projected in the Eigen space respectively.

5. Now, Whichever image in training set that gives the minimum distance among all other training images with test image is considered as match.

Discussion:

In step 3 of the algorithm, the mean face is subtracted from all other training face vectors. This is done to remove the features that are common among all the faces, which helps us in creating more unique features. Figure 11 is example of mean face created from the training set.

Figure 11: Mean Face

Assume the dimensions of A as $N^2 * M$, where M is the number of training images and $N^2 \gg M$. The step 4 mentioned in algorithm C=AA^T ,would create a resulting matrix of size $N^2 * N^2$. Calculating Eigen vectors u_i and Eigen values on this covariance matrix would be computationally intensive [11] and would require a lot computing resource and time. Therefore I replaced this step with the trick mentioned in [11], Where C is calculated using $A^T A$ which gives us the resulting matrix with dimensions M* M.

Now, The Eigen values and Eigen Vectors v_i can be calculated for this covariance matrix with reduced dimensions M*M. This trick is performed based on the assumption that original Eigen vectors u_i can be generated from Eigen vector v_i with reduced dimensionality using property $u_i = Av_i$ [11].

I observed that not all principal components are required to create the feature vector as the first few principal components/ Eigen Faces captures the maximum variance within the dataset and the rest of the remaining components captures less variance or captures noise in the dataset. Using all of the Eigen faces while forming feature vector could result in lot of cross-classification and results in poor accuracy. The variance captured by each principal component can be obtained by each Eigen vector's corresponding Eigen value. Figure 12 shows the plot of number of principal components against the Cumulative sum of variance capture by principal components in the training set.

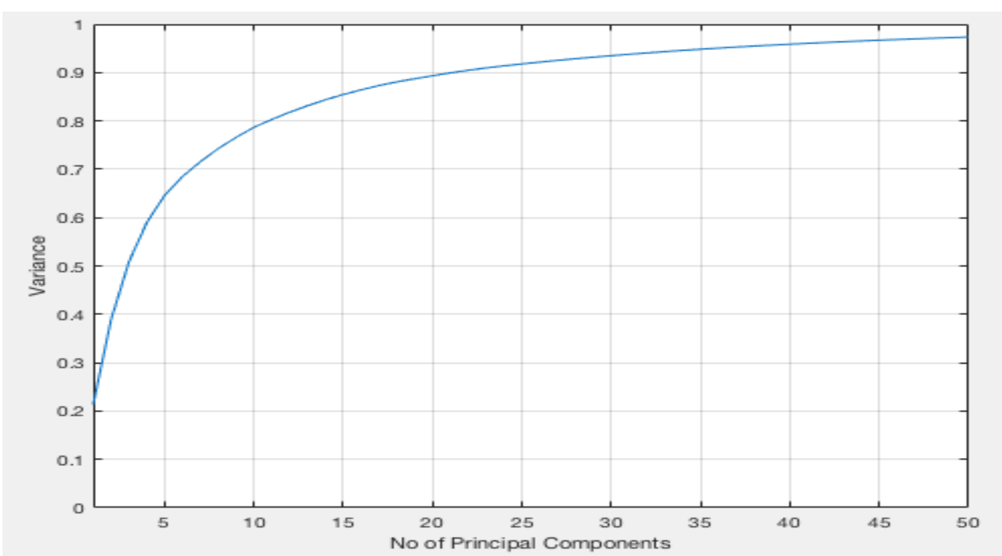

Figure 12: No of Principal Components VS Cumulative Variance

From the Figure 12, it can be seen that first 30 Principal Components captures about 90 percentage of variance in the data set.

Validation:

The accuracy of classification is used for evaluation and validation of the model. The accuracy that I got was about 73.33%, the model classified 44 out of 60 test images correctly. I observed that model was able to classify images with different expressions like sleepy face, happy face, surprised face etc. quite accurately. The model was even able to classify the faces with glasses correctly but the model failed to classify images with sharp lighting variations, which brought down the accuracy of the model.

I tried to adjust the illumination in the image by following techniques like adaptive histogram equalization followed by taking square root of the image, this improved by accuracy from earlier 70% to 71.67%. Then I only used adaptive histogram equalization instead of both which surprisingly increased by accuracy to 73.33%. Having followed these exposure enhancement techniques, still I could not improve my models sensitivity to lighting variations in the image.

Matlab Functions Used:

The below are the some of the important Matlab functions that I used in the project

step() – returns M by 4 matrix defining M bounding boxes containing detected objects [9]. Used for detecting faces in my case.

imcrop() – used for cropping the faces in my case.

imresize() – used for resizing the cropped faces in my case.

imfilter() – used for smoothing the image using gauss filter in my case.

reshape() – used for transforming 2d matrix to 1d vector in my case.

eig() – used to find Eigen values in increasing order along with Eigen Vectors.

cumsum() – used to find cumulative sum of variance while plotting Figure 12 in my case.

Conclusion:

Thus I went through various process of image processing chain like image cleaning, preprocessing, Feature extraction, model building and validation as part of this project. I also understood the limitations of PCA under various lighting conditions. Overall this project gave me a good start in understanding the face recognition problem, inspiring me to try more advanced models in the future.

References:

[1] R. Brunelli, T. Poggio, Face recognition: Features versus templates.IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 15(10),pp.1042-1052, 1998.

[2] M. A. Turk, A. P. Pentland, Media Lab, MIT, Cambridge, MA. Face recognition using eigenfaces. Proceedings of 1991 IEEE ComputerSociety Conference on Computer Vision and Pattern Recognition, pp.586-591, June 3-6, 1991.

[3] https://en.wikipedia.org/wiki/Principal_component_analysis accessed Oct 28th, 2016 at 9:00 PM EDT.

[4] https://en.wikipedia.org/wiki/Linear_discriminant_analysis accessed Oct 28th, 2016 at 9:38 PM EDT.

[5] Bruce A. Draper, Kyungim Baek, Marian Stewart Bartlett, and J. Ross Beveridge. 2003. Recognizing faces with PCA and ICA. *Comput. Vis. Image Underst.* 91, 1-2 (July 2003), 115-137.DOI=http://dx.doi.org/10.1016/S1077-3142(03)00077-8

[6] Hassaballah, M.; Aly, Saleh: 'Face recognition: challenges, achievements and future directions', IET Computer Vision, 2015, 9, (4), p. 614-626, DOI: 10.1049/iet-cvi.2014.0084

[7] http://www.vision.jhu.edu/teaching/vision08/Handouts/case_study_pca1.pdf accessed Nov 30th,2016 at 12:45 AM EDT.

[8] http://blog.manfredas.com/eigenfaces-tutorial/ accessed Nov 30th,2016 at 1:15 AM EDT.

[9] https://www.mathworks.com/help/vision/ref/vision.cascadeobjectdetector-class.html accessed Nov 30th,2016 at 9:15 AM EDT.