Pattern Recognition Coursework 1

Paul Streli - ps4715
Imperial College London
CID: 01103106

Karoly Horvath - kth15
Imperial College London
CID: 01088730

Abstract

Face recognition is the problem of identifying a human, based on a picture of their face. In this coursework various face-recognition algorithms are investigated, including Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) and PCA-LDA Ensemble in different variations. The recognition accuracies of the models are compared and advantages and disadvantages are determined.

1. Introduction

A data set of 520 face images (46x56 pixels) is given, consisting of 10 pictures per person. The faces have already been normalized for scale, orientation and translation, therefore these transformations are out of the scope of the experiments.

2. Q1 - Training and Testing Sets

The data set was split into training and testing sets. In this report the testing set is used for validation purposes, i.e. for evaluation and comparison of different methods. 80% of the images (416) are used for training and the remaining 20% (104 images) for testing, following the conventional 80-20 rule. [1]

It is essential to put equal amounts of images of the different people in the training set to be able to tackle the face-recognition problem as a balanced problem. If a particular identity had more images in the training set, the examined model's decision would be biased in favor of that person. Therefore, 8 out of the 10 images of each person were randomly picked for the training set.

3. Q1 - Principal Component Analysis

3.1. Mean Image

For Principal Component Analysis (PCA), first the *mean image* of the training set was computed by taking the average of the image vectors. It is shown on Figure 1, which clearly depicts a distorted face. The mean image shows the characteristics that all of the pictures share, taking away the individual components.

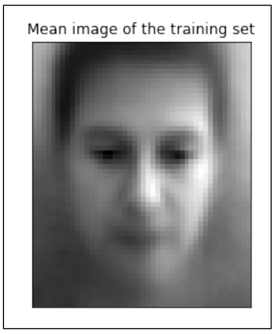

Figure 1. The mean image

3.2. Eigenvalues and Eigenvectors

After normalizing the training set by subtracting the mean image from all of the samples, the *covariance matrix* S was computed using the equation $S = \frac{1}{N}AA^T$, where A is the matrix composed from the training images in its columns. The rank, eigenvalues and corresponding eigenvectors of S (2576x2576) were inspected. Knowing that the rank of the unbiased covariance matrix is *n-1* (n unique face images with the mean face subtracted), the rank of S is expected to be $416 - 1 = 415$ and thus $2576 - 415 = 2161$ eigenvalues should be zero. The calculations resulted in $rank(S) = 415$ and - disregarding the computational rounding errors - it was confirmed that the number of non-zero eigenvalues is 415, ranging between 961191.8 and 96.8 - all in line with the expectations. Based on Figure 2, which shows the magnitude of the eigenvalues, it was decided that it should be sufficient to use the largest 50 eigenvalues for face detection, as the rest are comparably small in size and therefore do not significantly improve the recognition accuracy. Note that the chance of overfitting increases with the number of eigenfaces chosen as features. The recognition success rates using different number of PCA bases will be explored in Section 3.5.

Appendix 1 contains the images of the 3 eigenvectors (eigenfaces) that belong to the three largest eigenvalues. Physically, each eigenvector indicates a direction and the corresponding eigenvalues indicate how much variance there is in the data along that eigenvector (Principal Component). In the PCA analyses, it is desired to find the principal components that can capture the most amount of data variation, hence the eigenvectors corresponding to the largest eigenvalues are used. An eigenvector with zero eigenvalue explains none of the data variance. Therefore, any face vector in the training set can be built up as a linear combination of the 415 eigenfaces corresponding to non-zero eigenvalues and the mean face.

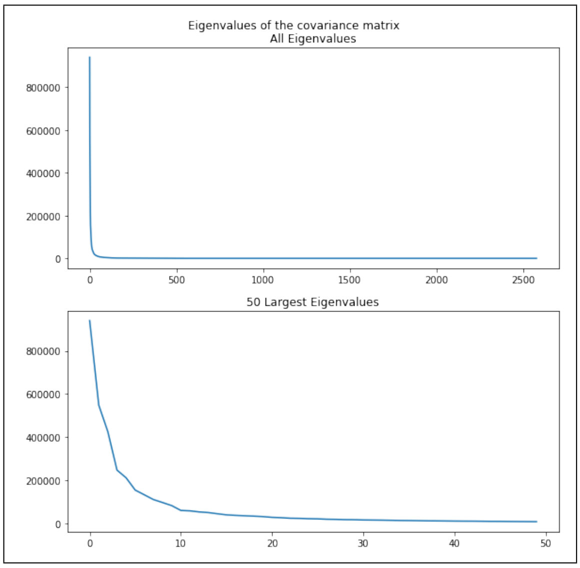

Figure 2. The size of the eigenvalues of the covariance matrix

3.3. Low Dimensional Computation

It is possible to use an alternative, less computationally expensive method for calculating the same values. S' is found from the equation $S' = \frac{1}{N}A^TA$ and the rank, eigenvalues and eigenvectors of S' are calculated. Note that $rank(A) = rank(A^T) = rank(A^TA) = rank(AA^T)$, therefore $rank(S')$ should also be 415. The new covariance matrix is of size 416x416 and the eigenvalues of it are found to be the same as of S. The eigenvectors v_i of S' that correspond to the non-zero eigenvalues relate to the eigenvectors u_i of S that correspond to the same eigenvalues through the equation $u_i = Av_i$. Indeed, the rank of S' was found to be 415.

An advantage of using this method it that it is almost *100* times faster to get to the same results. It took *5.26* seconds to calculate them for matrix S and *0.0538* seconds to get the same results for matrix S'. Another advantage is that this calculation uses *202.59 MiB* less memory; using the *linalg.eig()* function of *numpy*. However if the number of data points is larger than the dimensions, it is advisable to use the original method, because then that computation will result in a smaller covariance matrix. In this report, from this point on, the low-dimensional computation will be used.

3.4. Face Reconstruction

The training set was projected to the space spanned by the eigenvectors corresponding to the M_{pca} largest eigenvalues. Then the images were reconstructed to the original high-dimensional space and the Euclidean distances between the reconstructed and the original pictures were calculated. Figure 3 shows that the reconstruction error becomes smaller when more PCA bases are used. Projecting the image to a smaller dimensional space is a lossy transformation, so some information will be unreconstructable, i.e. the reconstructed picture will be blurry. It is important to note that the reconstruction error for projected images of the training set is zero when all eigenvectors are used. This is expected, as it has already been mentioned, they span the whole training set together with the mean vector. From Figure 4 it is possible to qualitatively see these properties. It is noticeable that the reconstruction quality gets worse and worse when a lower dimensional projection space is used.

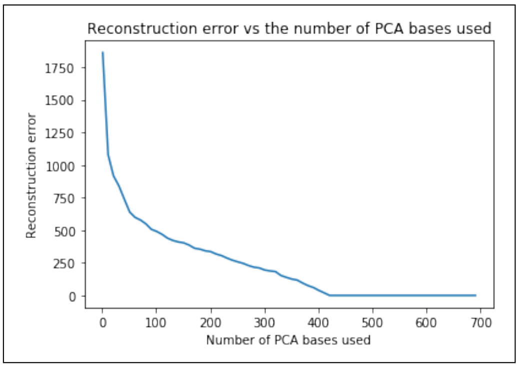

Figure 3. The reconstruction error versus the number of PCA bases used

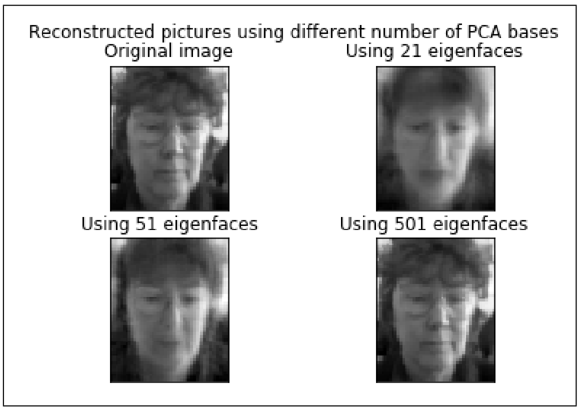

Figure 4. The reconstructed images

3.5. Face Recognition

Two PCA face-recognition methods were explored; the Nearest Neighbour (NN) classifier and an alternative method based on minimizing reconstruction errors. Both methods were inspected for different number of PCA bases M_{pca}.

The Nearest Neighbour classification method measures the Euclidean distance between each (projected and normalized) testing image and the projected training images, in the M_{pca} dimensional space. The testing pictures are classified as the person on the closest training image.

In the alternative method, first the training samples are grouped according to the person on the pictures. Since the training set consists of samples from 52 people, the number of groups is 52, each of them consisting of 8 images. For all of the groups, the covariance matrix of the samples is computed. Then the best M_{pca} eigenvalues and eigenvectors of each group's covariance matrix are used to project each of the testing images. They are then reconstructed straight away to their original dimension. The group's identity where the reconstruction error is the smallest is assigned to be the person on the testing image. It is important to note, that in all of the groups M_{pca}, the number of PCA bases, is the same, otherwise comparing the reconstruction errors would not be meaningful. As there are 8 samples in all of the groups, $M_{pca} <= 7$ for the alternative method, because 7 is the maximum number of non-zero eigenvalues of the covariance matrices for each group. The recognition success rates, computational times and memories can be seen in Figure 5 for the NN method and in Figure 13 in Appendix 2 for the alternative method. It can be seen that as expected, all three metrics increase as the number of PCA bases increases, which is desired for the recognition success rate, but adverse for the other two measures.

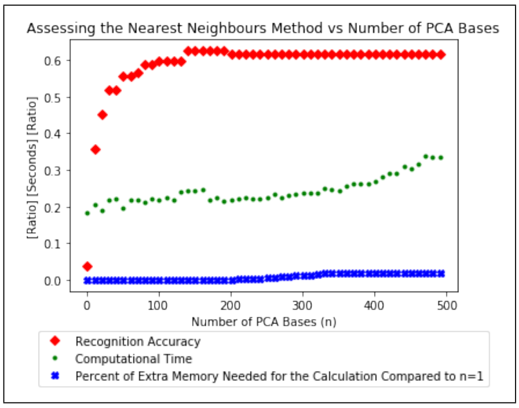

Figure 5. The assessment of the NN method versus the number of PCA bases

Example success and failure cases and the confusion matrix can be found in Figure 6 and 7 respectively for the NN method and in Figure 14 and Figure 15 respectively in Appendix 3 for the alternative method. The best recognition accuracy using PCA with NN was 0.644 ($M_{pca} = 171$). The alternative method performed even better on the test data achieving an accuracy of 0.721 ($M_{pca} = 7$). However, it required on average twice as much computational time.

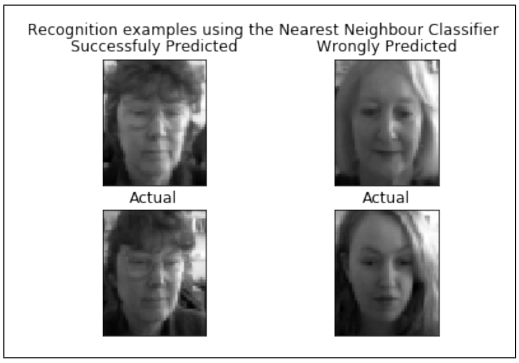

Figure 6. Successful and failed classifications using the NN method

4. Q3 - PCA-LDA

PCA is an unsupervised algorithm, which means that it does not take into account the class labels of the training data to compute the projection space. This is one of its biggest drawbacks for its use for classification and leads to the expectation that a supervised method might yield better results for the face-recognition problem. In this section, the Linear Discriminant Analysis (LDA) will be applied in combination with PCA. The same data partition will be used as the one described in Section 2 to balance the number of images in the sets.

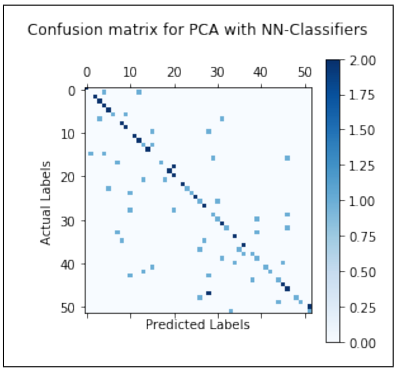

Figure 7. Confusion matrix of the NN method

LDA is a supervised method that tries to find a low-dimensional projection space that separates the data points according to their classes (discriminatively). Since the face-recognition task requires differentiation between multiple classes, the objective function for this problem, also known as the Multiple Discriminant Analysis (MDA), is given by Equation (7) in Appendix 4. The objective function maximizes the distance between the mean class vectors, while keeping the inter-class variance of the data points as small as possible.

The solution, $\mathbf{W_{lda}}$ is given by the normalized eigenvectors corresponding to the largest M_{lda} number of eigenvalues of the matrix $\mathbf{S_W^{-1}S_B}$ as derived in the lecture notes, for non-singular $\mathbf{S_W}$.

4.1. Scatter Matrices

First, the between-class scatter matrix $\mathbf{S_B}$ was computed following Equation (8) in Appendix 4 and its rank was inspected to be 51 (number of classes - 1). This is expected, as it is the product of a matrix with rank 51 (with vectors $\mathbf{m_i} - \mathbf{m}$ in its rows) and its transpose. The vectors $\mathbf{m_i} - \mathbf{m}$ represent the average deviations of each class from the global mean face and an example of them is shown on Figure 16 in Appendix 4. Thus, the between-class scatter matrix $\mathbf{S_B}$ can be interpreted as the covariance matrix of the class means.

Next, the within-class scatter matrix $\mathbf{S_W}$ was computed according to Equation (9) in Appendix 4. The rank of the within-class matrix is found to be 364 = number of unique pictures in the training set (N) - number of classes (c). This was also expected, since $\mathbf{S_W}$ is the sum of 52 matrices each of which has rank = 7 (number of pictures in the class - 1 (since $\mathbf{m_i}$ is subtracted)) abiding the rule rank(A+B)$\leq$ rank(A) + rank(B).

4.2. PCA Dimensionality Reduction

At this point it becomes apparent that the 2576x2576 matrix $\mathbf{S_W}$ with rank 364 is singular and therefore not invertible. To circumvent this problem, PCA is utilized to find a lower-dimensional projection space that keeps as much information of the original data set as possible. The new optimal projection matrix is given by:

$$\mathbf{W_{opt}^T} = \mathbf{W_{lda}^T W_{pca}^T} \tag{1}$$

where $\mathbf{W_{pca}}$ and $\mathbf{W_{lda}}$ are given by the new equations shown in Appendix 5.

$\mathbf{W_{lda}}$ is then built from the eigenvectors corresponding to the largest M_{lda} number of eigenvalues of matrix $(\mathbf{W_{pca}^T S_W W_{pca}})^{-1}(\mathbf{W_{pca}^T S_B W_{pca}})$.

To compute $\mathbf{W_{pca}}$, the total scatter matrix $\mathbf{S_T}$ needs to be found first using Equations (13) and (14) in Appendix 5. It is necessary to choose M_{pca} that defines how many eigenvectors of $\mathbf{S_T}$ will be selected to construct the $\mathbf{W_{pca}}$ matrix. For $\mathbf{W_{pca}^T S_W W_{pca}}$ to be invertible, M_{pca} needs to be smaller or equal to $N - c$, which follows from identities rank(AB) $\leq \min(\text{rank}(A), \text{rank}(B))$and rank($\mathbf{S_W}$) = N-c.W$_{pca}$ will ensure that after the projection the between-class and within-class scatter are maximized (see Equation (14)) and therefore the relative spacing between the individual data points remains as close to the input matrix as possible.

4.3. The optimal PCA-LDA Projection Matrix

To find $\mathbf{W_{opt}}$, the eigenvectors and eigenvalues of $\mathbf{Q} = (\mathbf{W_{pca}^T S_W W_{pca}})^{-1}(\mathbf{W_{pca}^T S_B W_{pca}})$ are computed. Since rank($\mathbf{S_B}$) $= c-1$, rank($\mathbf{Q}$) $\leq c-1$ since $\mathbf{S_B}$ has the smallest rank in the product of matrices. Indeed, 51 non-zero positive eigenvalues were found. This fact also limits our second hyper-parameter M_{lda}, the number of eigenvectors with the largest eigenvalues that are selected as columns for $\mathbf{W_{lda}}$, to be smaller than the number of classes c.

Finally, $\mathbf{W_{opt}}$ is computed according to Equation (1). $\mathbf{W_{opt}}$ has a rank equal to M_{lda} and is of size 2576xM_{lda}. Its eigenvectors, also known as fisherfaces capture discriminative features that are common within certain classes. Figure 18 in Appendix 6 shows the largest eigenvalues of the matrix $\mathbf{Q}$ and Figure 19 shows the three fisherfaces corresponding to the three largest eigenvalues.

4.4. Recognition Accuracies

To apply the model, both the training and the testing sets were projected to the new subspace. For classification, the previously described Nearest Neighbours (1-NN) method was used (see Section 3.5). Recognition accuracies were recorded using different M_{lda} and M_{pca} values (Appendix 8). Figures 8 and 22 show that the recognition accuracy in general increases with increasing M_{lda} and M_{pca}. Similarly to the results in PCA, after a certain number of eigenfaces and fisherfaces, the recognition accuracy reaches a plateau where deviations are only minor, since most of the information is already captured by previous projection directions. If M_{pca} gets too large, recognition accuracy might even start to suffer as the matrix for calculating $\mathbf{W_{lda}}$ $\mathbf{Q}$ becomes more susceptible to noise and unstable [2]. It is also interesting to mention that the NN Classifier performs especially bad for small M_{lda} (< 10) and when M_{lda} is similar in size to M_{pca}. In this case, most of the dimensionality reduction is done by PCA and the positive separating effect of LDA has only a small influence (i.e. LDA will only rotate the given directions to maximize its objective function without any further dimensionality reduction). The best approach is therefore to reduce dimensions with PCA up to the point where the inverse can be taken and the least amount of information is lost (generally $M_{pca} = N - c$ if there is no indication of overfitting). Further feature selection should then be done with a supervised method like LDA. The highest recognition accuracy (0.894) was achieved for $M_{lda} = 51$ and $M_{pca} = 190$. These parameter values will also be used in the further analysis.

The confusion matrix on Figure 9 indicates that most of the test images were correctly classified. Figures 21 in Appendix 9 and Figure 10 show examples for success and failure cases respectively. On Figure 10 the faces that were misclassified look very similar or share certain significant characteristics and even for a human, it would be hard to tell them apart. Overall, a significant improvement to the simple PCA method was observed, which shows the strong advantages of supervised learning for face recognition.

Further elaboration of the results, including reconstruction and the methods used can be found in Appendix 9. Result from more variations of PCA-LDA are also reported in Appendix 9.

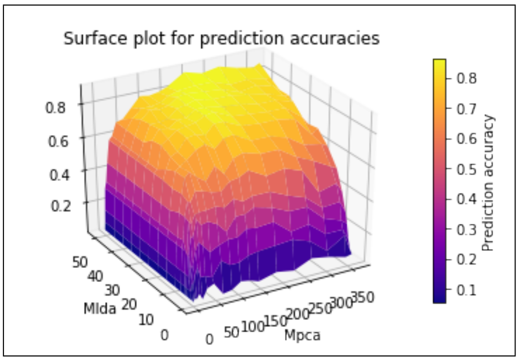

Figure 8. Surface plot for recognition accuracies for different M_{lda} and M_{pca}

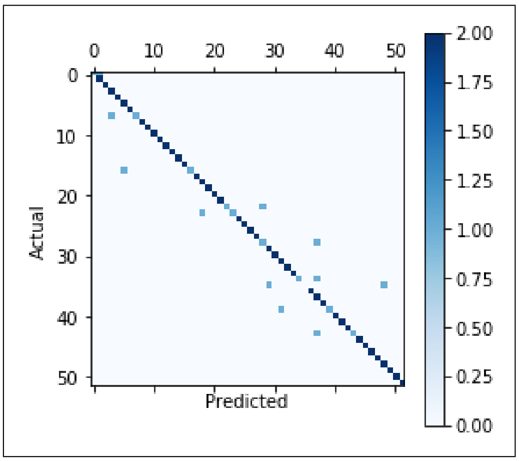

Figure 9. Confusion matrix for PCA-LDA method with 1-NN-Classifier and $M_{lda} = 51$ and $M_{pca} = 190$

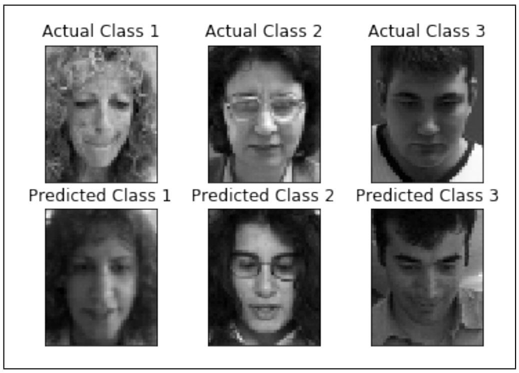

Figure 10. Example failure cases for PCA-LDA face recognition

5. Q3 - PCA-LDA Ensemble

While satisfactory results were obtained with the PCA-LDA face recognition method in the previous section, it is expected that the final hypothesis will perform worse on unseen test data as the hyper-parameters were optimized for and validated on the given test data set. Ensemble learning is a way to overcome this problem as it improves generalization and robustness to noise. For this approach, several weaker models that all randomly differ from each other are constructed. The test data is then fed to all classifiers and the final prediction is obtained by comparing the different outputs according to a selected fusion scheme. In this way, noisy model contributions and overfitted predictions will have a smaller influence on the outcome. The selected fusion rules are:

- **Majority voting** : Each model votes for its preferred prediction. The label with the most votes is assigned to the test point.
- **Maximum** : The label is set following the prediction with the highest confidence of all ensemble models (prediction confidence = Number of nearest neighbours from the same class / K (from K-NN)).
- **Average** : Average the prediction confidences over all ensemble models and take the label with the highest average prediction confidence.

In the following constructed committee machines, the hyper-parameters of the individual models were optimized over a random selection of values. While [2] could show that a random parameter selection for an ensemble of models still leads to similar results as for the optimized single PCA-LDA approach, it was decided to spend extra computation on this task to yield the best possible performance as training time was still acceptable for a smaller set of validation parameters.

5.1. Randomization on Data Samples

The first architecture consists of three individual models which use the classic PCA-LCA method but are trained on bagged training data (i.e. for each model eight training pictures are uniformly sampled from each class with replacement). In this way, each model is expected to be trained on 63.2% of unique data and thus performing more confidently on certain test points than on others. Using the 5-NN Classifier (the test label is decided upon a vote by the five closest neighbours), the individual models of the ensemble have an individual prediction accuracy ranging from 0.769 to 0.827. After fusing the individual outputs together, the recognition accuracy rises to 0.856 (maximum fusion), 0.837 (majority voting) and 0.846 (average fusion). This proves our expectation that the ensemble performs better than the individual models alone since it will decide based on the more confident predictions. Using seven bagged models in our ensemble led to a further improvement in performance (highest recognition accuracy: 0.894 (average fusion)) showing that an increase in the number of base models in the committee machine can lead to a further improvement in recognition accuracy.

Alternatively, it is also possible to bootstrap the individual models by taking all unique training images of a number of randomly picked classes only. This decreases the total number of training samples in each set and leads to the fact that some classes are not considered in each model. However, this might be beneficial as there are more classes than samples in each class. The resulting individual hypotheses will still be able to discriminate - even though in a non-optimal way - between the other classes that were not used for training, as certain discriminative characteristics are shared between persons. Using seven bootstrapped models with each making use of only 30 training classes, similar results were observed to the previous architecture. The individual models varied in their prediction accuracy between 0.721 to 0.808, while the complete architecture scored an highest recognition accuracy of 0.8846 (average fusion). While the obtained accuracy is similar to the basic PCA-LDA method, it is expected that the final hypothesis will generalize better to unseen data.

5.2. Randomization in the Feature Space

Next, randomization in the feature space was explored. In this architecture, the individual models differ in the eigenvectors that are selected for the construction of the $\mathbf{W_{pca}}$ matrix. $M0$ determines the number of eigenvectors of $\mathbf{S_T}$ with largest eigenvalues that are chosen. The remaining $M1$ dimensions of $\mathbf{W_{pca}}$ are randomly sampled from the other $N-1-M0$ eigenvectors of $\mathbf{S_T}$.

The models are all trained on the complete training set and are optimized on a random set of M_{lda} values. The ratio $M0 : M_{pca}$ is related to the randomness parameter ρ that was introduced in lectures (it follows that $M1 = M_{pca} - M0$). $M0 : M_{pca} = 1$ means that the individual models are highly correlated. $M0 : M_{pca} = 0$ means that the models are chosen with a high level of randomness and are therefore lowly correlated. In lectures, it was shown that a committee machine with highly decorrelated models is expected to perform on average better than an individual model under equal conditions.

In Section 4.4 it was already discussed that if the dimensions of $\mathbf{W_{pca}}$ are too large, recognition accuracy may start to suffer. On the other hand, selecting only the M_{pca} eigenvectors with largest eigenvalue for $\mathbf{W_{pca}}$ may cut out eigenvectors with smaller eigenvalues that store addi-

tional discriminative information that might be useful for recognition [2]. The ensemble approach thus helps to resolve this problem by creating several PCA-LDA models that each use a sensible number of M_{pca} with different eigenvectors. Thus, decreasing the chance that certain information is completely lost in the training process.
The baseline architecture consists of five models with $M_{pca} = 210$ and $M0 : M_{pca} = 0.5$. For this model, a recognition accuracy of 0.923 (average) was achieved. Decreasing randomness in our model ($M0 : M_{pca} = 0.7$) led to a worse success rate (0.885 (max fusion)). The same happened when $M0 : M_{pca}$ was decreased to 0.1 (0.856 (majority)). The optimal value for M_{pca} seems to be around 210 for this architecture. If $M_{pca} = 80$ recognition accuracy decreases (0.837 (majority)) indicating that too much information was already lost through PCA. Increasing M_{pca} to 350 also leads to a slight decline in accuracy (0.914% (average)) which confirms that there is not much more helpful discriminative information in the additional eigenvectors.

With the correct settings, the committee machine with randomization in the feature subspace performs better than the basic PCA-LDA model and is expected to generalize better.

5.3. Hierarchy

The two previous ensemble approaches can also be put in hierarchy. For this, seven bagged training data sets (i.e. bag eight samples from each class) are created and each one of them is trained on a randomized subspace. Following the observations from the previous sections, the optimal hyper-parameters were chosen to be $M0 : M_{pca} = 0.5$, $M_{pca} = 210 (< \frac{2}{3}N - c)$ and the number of base models was set to seven. The algorithm also randomly optimizes M_{lda} for each individual model. In this case, the committee machine actually performed worse than the basic PCA-LDA method. The highest recognition accuracy of 0.865 was achieved using the averaging fusion scheme. From this, the conclusion was drawn that the hierarchy approach will be discarded for further architectures.

5.4. Combined Committee Machine

Finally, several different architectures combining the randomization of subspace and data were explored. The most successful architecture was built up of of four bagged data set models and seven randomized subspace models ($M0 : M_{pca} = 0.5$ and $M_{pca} = 21$). The recognition accuracy was equal to 0.923 (average) (see the confusion matrix for this result in Figure 23 in Appendix 11). From the experiments it also became apparent that the average fusion scheme leads to the best overall prediction accuracies for most of the committee machines and seems to be working well for the 5-NN face recognition task.

An additional increase in recognition accuracy could be achieved when using the nearest centroid classification method (closest projected class mean determines the label) instead of 5-NN with the same architecture. In this case only the majority voting fusion scheme could be used. Only six out of the 104 test pictures were wrongly classified (recognition accuracy: 0.9423) (for confusion matrix see Figure 11). Due to its construction which makes use of randomization in data and features, this hypothesis is also expected to generalize well to unseen data. Using SVM instead of NN or NC was also investigated with majority voting for the classification part. However, this led to unsatisfactory results (recognition accuracy of 0.154) that pointed to overfitting and the decision to leave research into this topic as future work.

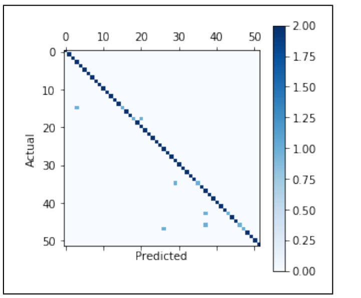

Figure 11. Confusion matrix for the final hypothesis (Committee machine using randomization on data and subspace with NC-Classifier and Majority Voting Fusion Scheme)

6. Q2 - Generative and Discriminative Learning

In Section 3, a generative method (PCA); and in Section 4 a discriminative method (LDA) were explored. In Section 4, PCA was used as a tool for dimensionality reduction, but the overall model lost its generative characteristics. In this section, a new model will be mathematically formulated and a solution will be derived. This model combines the generative nature of PCA with the discriminative advantages of LDA. To achieve this, α was defined, a constant that controls the balance between PCA (the generative characteristics) and LDA (the discriminative characteristics).

$$\mathbf{W}_{\mathbf{opt}_{\mathbf{PCA-LDA}}} = \arg\max_{\mathbf{W}} \frac{|\mathbf{W}^{\mathbf{T}}[(1-\alpha)\mathbf{S}_{\mathbf{B}} + \alpha\mathbf{S}_{\mathbf{T}}]\mathbf{W}|}{|\mathbf{W}^{\mathbf{T}}[(1-\alpha)\mathbf{S}_{\mathbf{W}} + \alpha\mathbf{I}]\mathbf{W}|} \tag{2}$$

Equation (2) shows the proposed objective function for this problem. This reduces to the objective function of PCA when $\alpha = 1$ (3) and to the objective function of LDA when $\alpha = 0$ (4). The solution can be found very similarly to the solution of the LDA Problem. Appendix 12 shows the derivation using the Lagrange multiplier formulation and the generalized eigenvalue-eigenvector analysis to solve for W_{opt}.

$$\mathbf{W}_{\mathbf{opt}_{\mathbf{pca}}} = \arg\max_{\mathbf{W}} |\mathbf{W}^{\mathbf{T}}\mathbf{S}_{\mathbf{T}}\mathbf{W}| \tag{3}$$

$$\mathbf{W}_{\mathbf{opt}_{\mathbf{lda}}} = \arg\max_{\mathbf{W}} \frac{|\mathbf{W}^{\mathbf{T}}\mathbf{S}_{\mathbf{B}}\mathbf{W}|}{|\mathbf{W}^{\mathbf{T}}\mathbf{S}_{\mathbf{W}}\mathbf{W}|} \tag{4}$$

$$[(1-\alpha)\mathbf{S}_{\mathbf{W}} + \alpha\mathbf{I}]^{-1}[(1-\alpha)\mathbf{S}_{\mathbf{B}} + \alpha\mathbf{S}_{\mathbf{T}}]\mathbf{W}_{\mathbf{opt}} = \mathbf{A}\mathbf{W}_{\mathbf{opt}} \tag{5}$$

The solution, W_{opt} can be found from Equation (4) and therefore it is composed of the eigenvectors of the matrix in Equation (5).

$$[(1-\alpha)\mathbf{S}_{\mathbf{W}} + \alpha\mathbf{I}]^{-1}[(1-\alpha)\mathbf{S}_{\mathbf{B}} + \alpha\mathbf{S}_{\mathbf{T}}] \tag{6}$$

Appendix 1

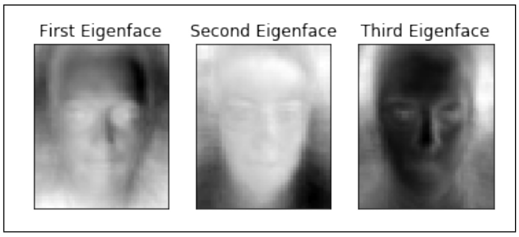

Figure 12. Visualization of the three best Eigenfaces

Appendix 2

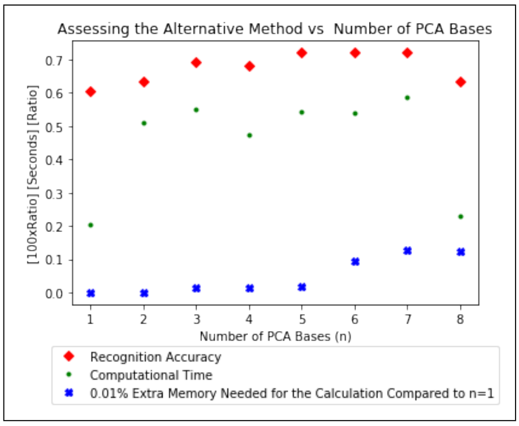

Figure 13. The assessment of the alternative method versus the number of PCA bases

Appendix 3

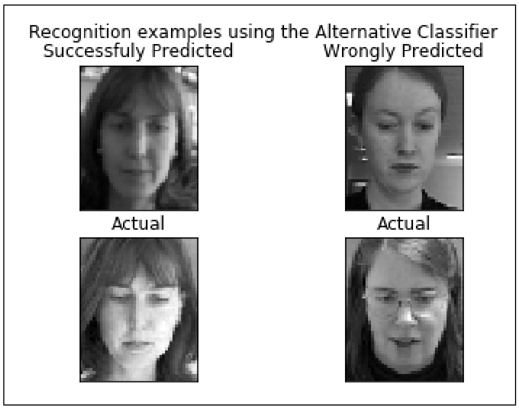

Figure 14. Successful and failed classifications using the alternative method

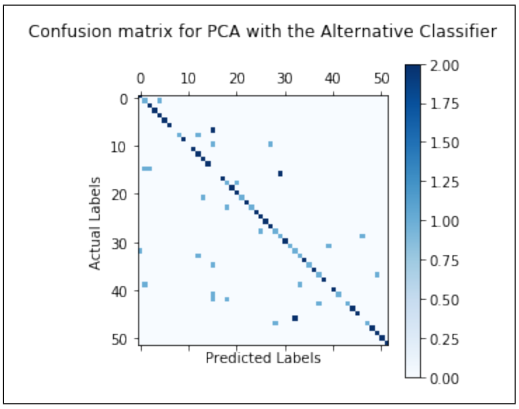

Figure 15. Confusion matrix of the alternative method

Appendix 4

$$\mathbf{W_{lda}} = \arg\max_{\mathbf{W}} \frac{|\mathbf{W^T S_B W}|}{|\mathbf{W^T S_W W}|} \tag{7}$$

where S_W and S_B are the *within-class* and *between-class* scatter matrices as defined as (9) and (8) respectively.

$$\mathbf{S_B} = \sum_{i=1}^{c} (\mathbf{m_i} - \mathbf{m})(\mathbf{m_i} - \mathbf{m})^T \tag{8}$$

$\mathbf{S_B}$ is computed as the product of a matrix with $\mathbf{m_i} - \mathbf{m}$ in its columns and its transpose. $\mathbf{m}$ is the global mean of the training set and $\mathbf{m_i}$ is the class mean of class i. An example of $\mathbf{m_i} - \mathbf{m}$ is shown on Figure 16.

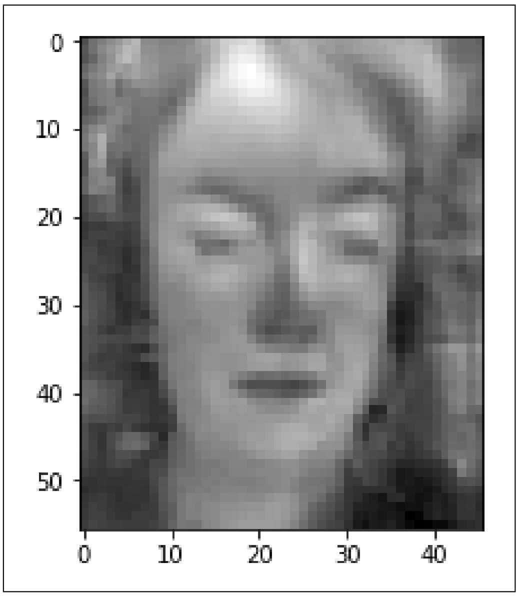

Figure 16. Visualization of a vector $\mathbf{m_i} - \mathbf{m}$

$$\mathbf{S_W} = \sum_{i=1}^{c} \mathbf{S_i} \tag{9}$$

$$\mathbf{S_i} = \sum_{\mathbf{x} \in c_i} (\mathbf{x} - \mathbf{m_i})(\mathbf{x} - \mathbf{m_i})^T \tag{10}$$

The within-class scatter matrix $\mathbf{S_W}$ is computed as the sum of all intra-class covariance matrices. The intra-class covariance matrices $\mathbf{S_i}$ are computed according to Equation (10). The calculation includes subtraction of the corresponding class mean vectors $\mathbf{m_i}$ from each picture $\mathbf{x}$. An example of the subtracted vector is shown on Figure 17. Equation (9) assumes that there is an uniform number of data points in each class which is satisfied by our test-training split. If this was not the case the formula could be corrected by weighting the matrices $\mathbf{S_i}$ with the reciprocal of the number of samples in the class. Otherwise, $\mathbf{S_W}$ would be biased towards minimizing the projected inter-class variances of the classes with more data points.

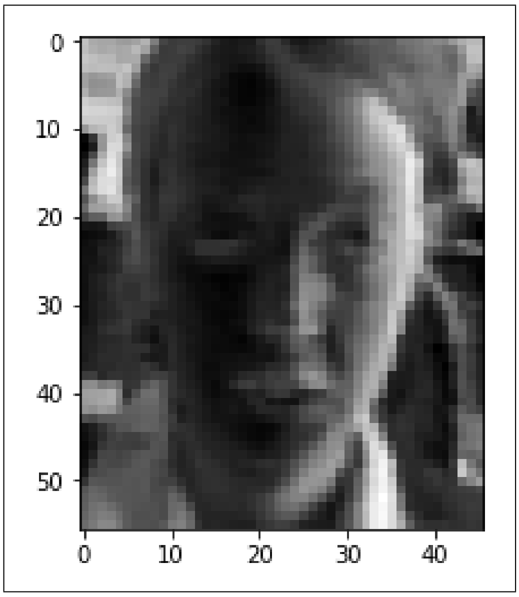

Figure 17. Visualization of the deviation of a picture from its class mean: $\mathbf{x} - \mathbf{mi}$

Appendix 5

$$\mathbf{W_{opt}^T} = \mathbf{W_{lda}^T W_{pca}^T} \tag{11}$$

where

$$\mathbf{W_{lda}} = \arg\max_{\mathbf{W}} \frac{|\mathbf{W^T W_{pca}^T S_B W_{pca} W}|}{|\mathbf{W^T W_{pca}^T S_W W_{pca} W}|} \tag{12}$$

$$\mathbf{W_{pca}} = \arg\max_{\mathbf{W}} |\mathbf{W^T S_T W}| \tag{13}$$

$$\mathbf{S_T} = \sum_n (\mathbf{x_n} - \mathbf{m})(\mathbf{x_n} - \mathbf{m})^T = \mathbf{S_B} + \mathbf{S_W} \tag{14}$$

Appendix 6

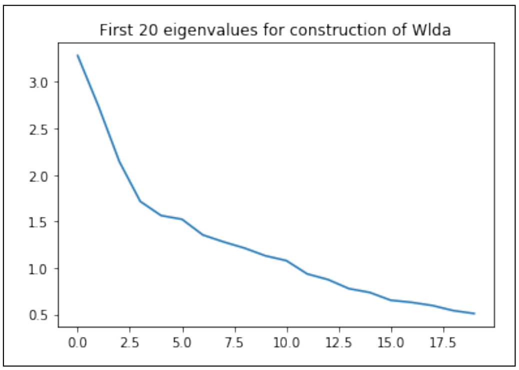

Figure 18. Largest 20 eigenvalues of $\mathbf{Q}$ with $\mathbf{M_{pca}} = 190$

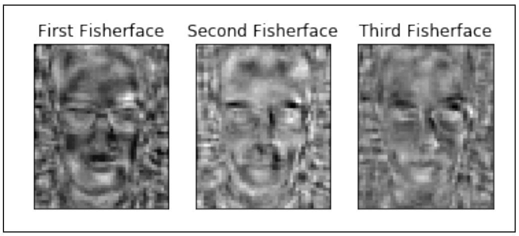

Figure 19. Visualization of the first three fisherfaces ($\mathbf{M_{lda}} = 51$ and $\mathbf{M_{pca}} = 190$)

Appendix 7

The LDA method also comes with drawbacks. Since it is a discriminative (i.e. not generative) method the subspace projection discards information that is common to all training points and thus does not help for data separation. That is why, the images cannot be reconstructed without loss. The reconstructed picture in Figure 20 is of much lower quality than in Section 3.4.

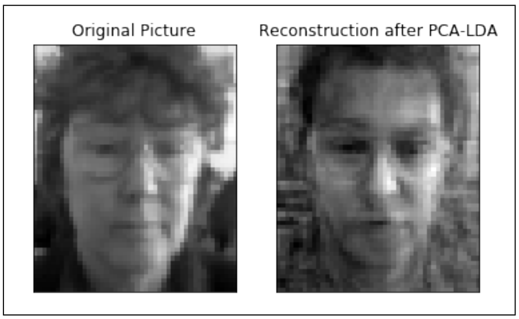

Figure 20. Reconstruction after PCA-LDA projection with 1-NN-Classifier and $\mathbf{M_{lda}} = 51$ and $\mathbf{M_{pca}} = 190$

Appendix 8

In order to find the optimal values for $\mathbf{M_{lda}}$ and $\mathbf{M_{pca}}$, the recognition accuracies for all possible combinations of $\mathbf{M_{pca}} \in [1, 2, 3, 5, 8, 12, 18, 25, 35, 50, 70, 90, 110, 130, 150, 190, 230, 270, 310, 350, 364]$ and $\mathbf{M_{lda}} \in [1, 2, 3, 4, 5, 7, 10, 15, 20, 25, 30, 35, 40, 45, 51]$

were computed (in case $\mathbf{M_{lda}} > \mathbf{M_{pca}}$ we set $\mathbf{M_{lda}} = \mathbf{M_{pca}}$).

Appendix 9

Using the PCA-LDA method with the 1-NN-Classifier and $\mathbf{M_{lda}} = 51$ and $\mathbf{M_{pca}} = 190$, the recognition accuracy is 0.8942. From the confusion matrix it becomes apparent that most of the test images were correctly classified.. Looking at the cases where the person was wrongly classified, it becomes apparent that the faces on the pictures indeed look very similar or share certain significant characteristics (e.g. hair, glasses and face shape). In certain cases, it is expected that even a real person would have difficulties for the given cases. For the person with label 36 (failure case example 3) both test images were wrongly classified. This might be explainable due to the large intra-class deviation between the different pictures of the training set, making it quite difficult to find characteristic class features.

Overall, the results of the PCA-LDA face recognition method are very promising. We obtained a significant improvement to the simple PCA methods (highest recognition accuracy of alternative PCA method: 0.683 - see section 3.5). This shows the strong advantages of supervised learning approaches for face recognition.

For comparison and to find out whether it is possible to do better on the final classification part, several other classifier were used for determining the test labels after the projection to the PCA-LDA subspace (see Table 1). In all cases, the best recognition accuracies for all possible combinations of $\mathbf{M_{pca}}$ and $\mathbf{M_{lda}}$ are shown (i.e. test set was used for validation). The nearest centroid classifier, where the closest projected class mean determines the label, performed slightly better then the rest. This might be due to the fact that it is less prone to noise and outliers. 3-NN and 5-NN, where the test label is decided upon a vote by the three or five closest neighbours, showed similar performance to the baseline 1-NN classifier.

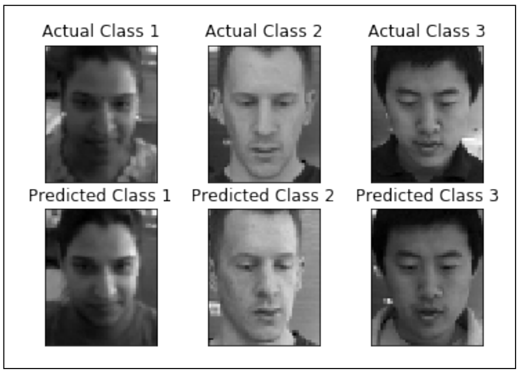

Figure 21. Example succuess cases for PCA-LDA face recognition

Classifier method	$\mathbf{M_{pca}}$	$\mathbf{M_{lda}}$	Recog. acc.
1-NN	190	51	0.894
3-NN	350	51	0.884
5-NN	350	51	0.894
Nearest centroid classifier	230	51	0.923

Table 1. Recognition accuracies for different classifier methods in the PCA-LDA subspace and given optimal hyper-parameters

Appendix 10

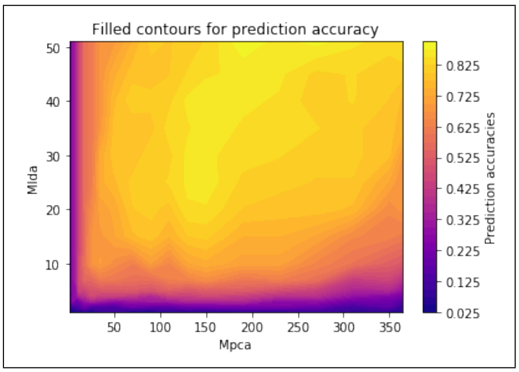

Figure 22. Filled contours for recognition accuracies for different $\mathbf{M_{lda}}$ and $\mathbf{M_{pca}}$

Appendix 11

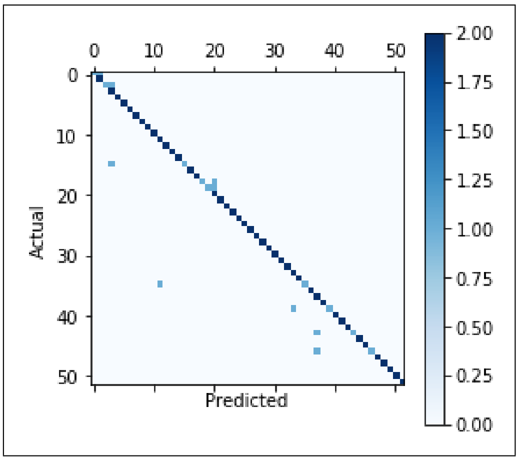

Figure 23. Successful and failed classifications using the alternative method

Appendix 12

$$\mathbf{W_{opt_{PCA-LDA}}} = \arg\max_{\mathbf{W}} \frac{|\mathbf{W^T}[(1-\alpha)\mathbf{S_B} + \alpha\mathbf{S_T}]\mathbf{W}|}{|\mathbf{W^T}[(1-\alpha)\mathbf{S_W} + \alpha\mathbf{I}]\mathbf{W}|} \tag{15}$$

When α = 0: this reduces to the objective function of LDA:

$$\mathbf{W_{opt_{lda}}} = \arg\max_{\mathbf{W}} \frac{|\mathbf{W^T S_B W}|}{|\mathbf{W^T S_W W}|} \tag{16}$$

And when $\alpha = 1$, as the denominator is the identity matrix and the optimization problem simplifies to the PCA optimization problem:

$$\mathbf{W_{opt_{pca}}} = \arg\max_{\mathbf{W}} |\mathbf{W^T S_T W}| \tag{17}$$

In these equations the within-class scatter matrix $\mathbf{S_W}$, the between-class scatter matrix $\mathbf{S_B}$ and the total scatter matrix $\mathbf{S_T}$ are as defined in Equations (9), (8) and (14) respectively.

Maximizing the ratio in (15) is equivalent to maximizing the numerator while keeping the denominator constant, i.e. take

$$\max_{\mathbf{W}} \mathbf{W^T}[(1-\alpha)\mathbf{S_B} + \alpha\mathbf{S_T}]\mathbf{W} \tag{18}$$

subject to:

$$\mathbf{W^T}[(1-\alpha)\mathbf{S_W} + \alpha\mathbf{I}]\mathbf{W} = k \tag{19}$$

(k = constant)

This can be accomplished using Lagrange multipliers as:

$$L = \mathbf{W^T}[(1-\alpha)\mathbf{S_B}+\alpha\mathbf{S_T}]\mathbf{W}+\lambda(k-\mathbf{W^T}[(1-\alpha)\mathbf{S_W}+\alpha\mathbf{I}]\mathbf{W} \tag{20}$$

and maximizing L with respect to both $\mathbf{W}$ and λ After simplifying this equation and setting the gradient of L with respect to $\mathbf{W}$ to zero:

$$2([(1-\alpha)\mathbf{S_B}+\alpha\mathbf{S_T}]-\lambda[(1-\alpha)\mathbf{S_W}+\alpha\mathbf{I}])\mathbf{W} = 0 \tag{21}$$

This equation is equivalent to:

$$[(1-\alpha)\mathbf{S_B} + \alpha\mathbf{S_T}]\mathbf{W} = \lambda[(1-\alpha)\mathbf{S_W} + \alpha\mathbf{I}]\mathbf{W} \tag{22}$$

This is a generalized eigenvalue-eigenvector problem. The solution is easy when the matrix $(1-\alpha)\mathbf{S_W} + \alpha\mathbf{I}$ is non-singular (analysis when it is singular is out of the scope of this report). The solution is:

$$[(1-\alpha)\mathbf{S_W}+\alpha\mathbf{I}]^{-1}[(1-\alpha)\mathbf{S_B}+\alpha\mathbf{S_T}]\mathbf{W_{opt}} = \mathbf{AW_{opt}} \tag{23}$$

where W and λ are the eigenvalues and eigenvectors of matrix:

$$[(1-\alpha)\mathbf{S_W} + \alpha\mathbf{I}]^{-1}[(1-\alpha)\mathbf{S_B} + \alpha\mathbf{S_T}] \tag{24}$$

References

[1] Anon. Pareto principle, 2018. Wikipedia, Available from: https://en.wikipedia.org/wiki/Pareto_principle, Accessed 1st November 2018.

[2] X. Wang and X. Tang. Random sampling for subspace face recognition, Oct 2006.