PCA (Eigenface) and PCA-LDA (Fisherface) Ensemble for Face Recognition

Eren Kopuz
CID: 01095555
ek2315@imperial.ac.uk

Rajan Patel
CID: 01062228
rnp15@imperial.ac.uk

Abstract

We present an exploration into face recognition, using generative unsupervised learning techniques such as Principal Component Analysis (PCA), discriminative supervised learning approaches such as Linear Discriminant Analysis (LDA) and ensemble methods employing randomization in the training and feature spaces. We endeavour to find an optimal approach that maximizes the recognition accuracy whilst taking into consideration computational complexity.[1]

1. Dataset Characteristics

The data-set contains $N = 520$ raster-scanned face images, each with pixel dimensions of $D = 46x56 = 2576$, that represent $L = 52$ unique faces. Re-scaling and reorientation of images, translations of facial landmarks (i.e using eye locations) or any further facial standardization's are not required, as every image in the data set has the same dimensions and a consistent central alignment.

2. Preprocessing

Each image is flattened and represented mathematically in the form of a column-vector denoted as x_i, where $x_i \in \mathbb{R}^D$ and i ∈ [1, 520]. The data-set is split into a training set X_{train} and a testing test X_{test}. In order to ensure equal representation of each unique face (class label), the division of which images go to X_{train} and X_{test} is applied in a class-wise manner rather than a crude split of the whole data-set that ignores class labels. Otherwise, classifiers trained on an unbalanced X_{train} are biased against under-represented class labels. Following an exploration of the data-set and validation of preliminary models a ratio of X_{train}: X_{test} = 4:1 was chosen. The cardinalities of X_{train} and X_{test} are denoted as N_{train} = 416 and N_{test} = 104 respectively.

3. Eigenfaces

3.1. Principal Component Analysis

Every x_i of X_{train} is projected onto a feature space $\mathcal{F}$ of dimension $\mathcal{M} << D$. The principal components that span $\mathcal{F}$ ('Eigenfaces', Figure 12) would ideally capture the significant variations among face images to maximize the projected data variance.[2] The mean face (Figure 11) $\bar{x} = \frac{1}{N_{train}} \sum_{i=1}^{N_{train}} x_i$ is subtracted from x_i such that $\phi_i = x_i - \bar{x}$. We define the orthonormal matrix A = $[\phi_1, \phi_2, ..., \phi_{N_{train}}] \in \mathbb{R}^{DxN_{train}}$ where the columns are the normalised image vectors of X_{train}.

3.2. Naive PCA (High Dimensional Computation)

3.2.1 Theoretical Expectations

To construct $\mathcal{F}$, the co-variance matrix $S_n = \frac{1}{N_{train}} AA^T \in \mathbb{R}^{DxD}$ is computed. We can then calculate the best $\mathcal{M}$ eigenvectors, $u_1, u_2, ..., u_{\mathcal{M}}$ that correspond to the $\mathcal{M}$ largest eigenvalues, $\lambda_1, \lambda_2, ..., \lambda_{\mathcal{M}}$ such that $S_n u_i = \lambda u_i$. S_n is a real and symmetric matrix: $S_n = AA^T = (AA^T)^T = S_n^T$, thus its eigenvalues are real and can be ordered.

$$rank(S_n) = rank(AA^T) = rank(A^TA) = rank(A) \quad (1)$$

From (1), $rank(S_n) = rank(A)$ and $A \in \mathbb{R}^{DxN_{train}}$:

$$rank(S_n) \leq min\ (D, N_{train}) = N_{train} = 416$$

More precisely [1], rank(A) = rank(X_{train}) - 1[3]. Hence, we anticipate the number of non-zero eigenvalues to be $\leq N_{train} - 1 = 415$.

3.2.2 Experimental Results & Verification

The eigendecomposition of S_n, surprisingly yielded D non-zero eigenvalues. However from closer observation, $D - (N_{train} - 1)$ of these eigenvalues have magnitudes of the order 10^{-12} or smaller[4], thus are considered with virtual magnitudes of 0. Therefore, we obtain $N_{train} - 1$ real non-zero eigenvalues, which substantiate the theoretical claims made in section 3.2.1.

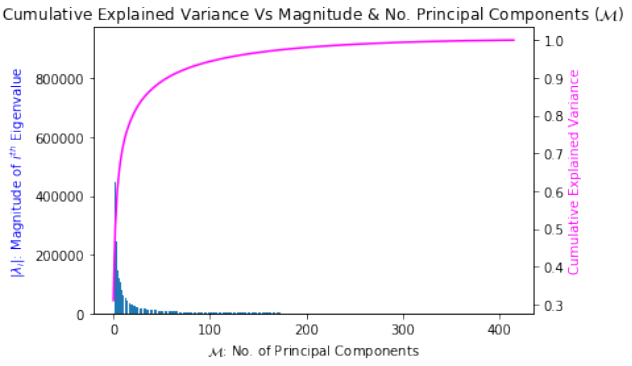

Figure 1. Magnitudes of Eigenvalues in descending order & Cumulative Explained Variance Vs Number of Principal Components

[1] Working code for all experiments is documented at: **https://github.com/RajanPatel97/EE4-68-Pattern-Recognition-CW1**

[2] These eigenvectors do not necessarily correspond to distinct facial features such as the eyes, nose, and mouth.

[3] Since X_{train} is normalized thus A is centered

[4] These minuscule magnitudes originate from computational precision errors in the numerical algorithm used by NumPy Python library.

X_{train} is inherently sparse since $N_{train} << D$, thus $D - (N_{train} - 1)$ eigenvectors whose corresponding eigenvalues have zero magnitude are omitted, as they do not contribute to face reconstruction. Figure 1 reveals that reasonably accurate reconstructions are obtainable with an even smaller subset of eigenvectors $< N_{train} - 1$, as some eigenvalues have a significantly greater magnitude than others.

To maximize the projected data variance and have $M < N_{train} - 1$, $\mathcal{M}$ = 123 principal components are chosen such that their cumulative explained variance[3] is 95% (Figure 1), a suitable trade off with regards to computational complexity. Consequently, the sum of the corresponding $\mathcal{M}$ eigenvalues are 95% of the sum of all $\mathcal{D}$ eigenvalues, which constitutes to a 5% reconstruction error (3).

3.3. Efficient PCA (Low Dimensional Computation)

3.3.1 Theoretical Expectations

We propose $S_e = \frac{1}{N_{train}} A^T A \in \mathbb{R}^{N_{train} x N_{train}}$. The eigenvalues and eigenvectors of S_e are denoted as λ and **v** respectively. The equivalence of eigenvalues: $\lambda(S_e) = \lambda(S_n)$, and the trans-formative relationship of eigenvectors[2]: **u** = A**v**, between S_e and S_n show that the eigendecomposition of S_e can find the same non-zero $N_{train} - 1$ eigenvalues as S_n.

3.3.2 Experimental Results & Verification

Following the eigendecomposition of S_e, the summation of the absolute difference over the largest $N_{train} - 1$ eigenvalues between S_e and S_n yielded: $\sum_{i=1}^{N_{train}-1} |\lambda_i(S_n) - \lambda_i(S_e)| = 0$. This proves that the largest $N_{train} - 1$ eigenvalues between S_e and S_n are numerically equivalent. By multiplying A with each of the **v** eigenvectors of S_e, we verify the relationship **u** = A**v** does indeed hold true in the numerical computation.

3.4. Naive PCA vs Efficient PCA

The eigen-decomposition of a matrix using the NumPy Python library has a time complexity of $O(n)^3$ in a worst case scenario. For Naive PCA, we note that S_n has dimensions DxD, which is relatively large thus is computationally expensive. A significant amount of this computation is redundant and increases execution time as $D - (N_{train} - 1)$ eigenvectors are not going to be used for reconstruction. The advantage of using S_e (Efficient PCA) is that it is a much smaller matrix of dimensions $N_{train} x N_{train}$, such that its eigendecompositon produces only the required $N_{train} - 1$ linearly independent eigenvalue-eigenvector pairs required for full facial reconstruction in $\mathcal{F}$.

The results in Table 1 reveal that the faster computation of Efficient PCA is a much more pragmatic approach.

PCA	Execution Time/s
Naive (S_n)	22.40
Efficient (S_e)	0.32

Table 1. Execution time for the eigendecompositon of S_n and S_e

4. Application of Eigenfaces

For the experiments conducted in this section and beyond[5], Efficient PCA is adopted as the technique of choice for the reasons given in Section 3.4.

4.1. Face Image Reconstruction

4.1.1 Reconstruction Algorithm

In PCA, images are projected to the compressed 'face space' $\mathcal{F}$ using the transformation $\mathcal{X} \rightarrow \mathcal{F}$. We define the projection matrix $\mathcal{W}_n = [a_{n1}, a_{n2}, ..., a_{nM}]$, where $a_{ni} = \phi_n{}^T \mathbf{u}_i$ and n $\in [1, N_{train}]$, such that each normalized face image ϕ_n can be represented by its projections in $\mathcal{F}$.

We examine the reconstruction of an image by computing the inverse transformation $\mathcal{F} \rightarrow \mathcal{X}$. The reconstruction of the n^{th} compressed image in $\mathcal{F}$ can be represented as:

$$\widetilde{x_n} = \bar{x} + \sum_{i=1}^{\mathcal{M}} a_{ni} \boldsymbol{u}_i \qquad \widetilde{x_n} \in \mathbb{R}^{\mathcal{D}} \tag{2}$$

Equation (2) demonstrates that each face in X_{train} can be reconstructed from a linear combination of the best $\mathcal{M}$ mutually orthornormal eigenvectors that span $\mathcal{F}$.

The data compression that PCA performs is lossy, thus the quality of reconstruction is dependent on the choice of $\mathcal{M}$. We utilize the distortion measure $\mathcal{J}$ as a way to record the reconstruction error:

$$\mathcal{J} = \frac{1}{N_{train}} \sum_{i=1}^{N_{train}} ||x_i - \widetilde{x_i}||^2 = \sum_{i=\mathcal{M}+1}^{\mathcal{D}} \lambda_i \approx \sum_{i=\mathcal{M}+1}^{N_{train}} \lambda_i \tag{3}$$

where λ_i's are sorted in descending order of magnitude. It is important to note that if $\mathcal{M} = N_{train} - 1$, then $\mathcal{J} \approx 0$, since $\sum_{i=N_{train}}^{D} \lambda_i \approx 0$.

4.1.2 Results & Observations

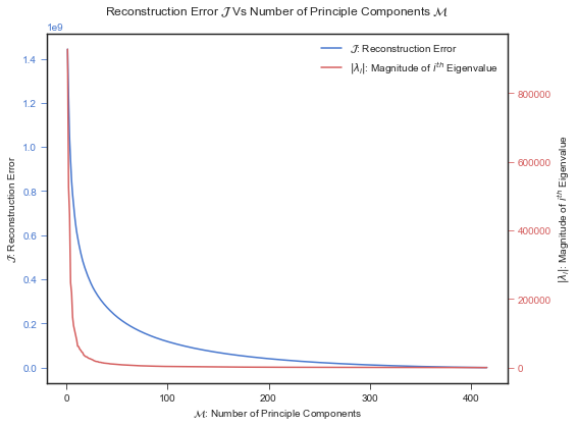

Figure 2. Reconstruction Error $\mathcal{J}$ Vs Number of Principal Components $\mathcal{M}$

Figure 2 illustrates the inverse proportional relationship between $\mathcal{J}$ and $\mathcal{M}$, as well as the impact that the magnitude

[5]Only when relevant and applicable.

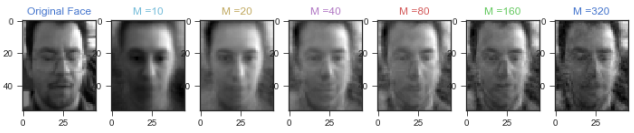

Figure 3. Reconstructed Faces Vs Number of Principal Components $\mathcal{M}$

of each λ_i has on $\mathcal{J}$. This reaffirms statement (3) and quantitatively justifies our reasoning for the choice of $\mathcal{M}$ = 123 in section 3.2.2.

From Figures 3, 25 and 26, we qualitatively assess that the visual accuracy of the reconstructed image with respect to the original image improves as $\mathcal{M}$ increases.

4.2. Face Recognition

4.2.1 Nearest Neighbours (NN) Classification

For NN classification, we proceed having fitted X_{train} with PCA such that the projection matrix for each normalized face ϕ_n is $\mathcal{W}_n$ for n $\in [1, N_{train}]$, as described in section 4.1.1. The recognition of a new face x_i from X_{test} is outlined in Algorithm 1.

Algorithm 1 PCA + k-Nearest Neighbours for k = 1

For x_i in X_{test}:
1: Normalize x_i: $\phi_i = x_i - \bar{x}$
2: Project ϕ_i to $\mathcal{F}$: $a_j = {\phi_i}^T \mathbf{u}_j$ ▷ j $\in [1, \mathcal{M}]$
3: $\mathcal{W}_i = [a_{i1}, a_{i2}, ..., a_{iM}]$ ▷ Projection Matrix of ϕ_i
4: Find: $min_n ||\mathcal{W}_i - \mathcal{W}_n||$ ▷ n $\in [1, N_{train}]$
5: x_i identified as x_n from X_{train}, denoted with label $\hat{y}_i$

NN classification is dependant on hyperparameters: $\mathcal{M}$ and *k* (number of nearest neighbours used for classification). For k > 1, we retain the k samples from X_{train} with minimum euclidean distance from x_i. Majority voting is used to determine $\hat{y}_i$.

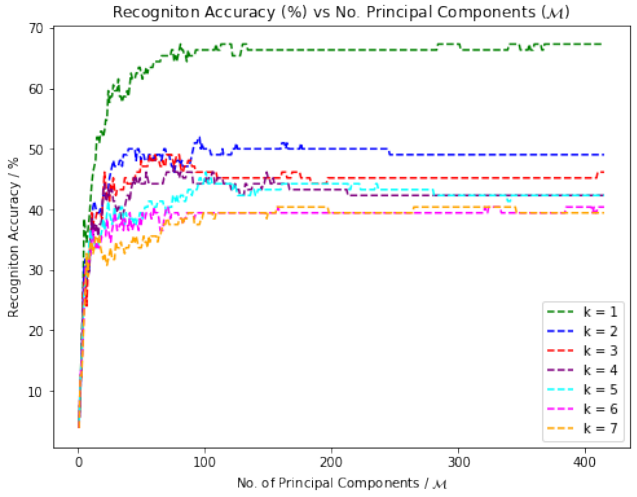

Figure 4. Recognition Accuracy vs No. Principal Components ($\mathcal{M}$) for $1 < k < 5$

The best k-NN classifier is when k = 1 (Figure 4), where accuracy decreases significantly for k > 1 due to the inherent sparsity of X_{train}, despite greater robustness for outliers. Regardless of the value of k, the recognition accuracy is highly coupled to $\mathcal{M}$, which supports theoretical arguments made in section 4.1.1 regarding the error introduced by PCA. Greater accuracy can be achieved through larger $\mathcal{M}$, however their is a computation and storage trade-off with respect to $O(n)$ execution time and $O(n)$ memory constraints as depicted in Figure 21 and Figure 22 respectively. For this implementation of PCA + k-NN, the optimal recognition accuracy achieved was 67.31% for k = 1 and $\mathcal{M}$ = 81. Examples of failed and successful image classifications along with the confusion matrix[6] for these optimal hyper-parameters are illustrated in Figures 14 and 19.

4.2.2 Alternative Method for Face Recognition

As an alternative to k-NN, we also implement a classification algorithm based on minimum reconstruction error. A test image x_i from X_{test} is reconstructed on distinct eigen-subspaces generated for each class, where the class with the minimum reconstruction error is assigned.

Algorithm 2 PCA + Alternative Method

For each class c_j: ▷ j $\in [1, L]$
1: Compute eigen-subspace $\mathcal{F}_j$
For each x_i in X_{test}:
2: Project x_i onto each subspace $\mathcal{F}_j$
3: Reconstruct to retrieve $\widetilde{x_{ij}}$
4: Find: $min_j ||x_i - \widetilde{x_{ij}}|$
5: x_i identified as such class j and assigned to label $\hat{y}_i$

For this algorithm, PCA is applied to each class independently. Due to the 4:1 split between X_{train} and X_{test}, there exists 8 unique instances of each of the 52 classes in X_{train}. This means that the number of non-zero eigenvalues, and hence the dimensionality of the transformed matrix, is limited to 7 for each eigen-subspace. Hence using a value greater than $\mathcal{M} = 7$ does not lead to an improvement in accuracy.

Through experimentation we observed the following trend (Figure 13). We see that accuracy flattens out for $\mathcal{M} \geq 6$, and that there exists a slight peak for $\mathcal{M} = 5$ with accuracy of 82.68%. We may be able to tie this down to that fact that this number leads to an ideal trade-off between the explained variance per class and greater generalizability of the subspaces.

As we can also see from the confusion matrix in Figure 15, the alternate method has provided greater accuracy as opposed to the k-NN classifier. This is expected as the subspaces for this method were learnt taking into account the class labels. Hence, we can infer that less discriminative information was lost in the process as opposed to performing PCA on the whole of X_{train} irrespective of class labels.

[6]Image classification examples and confusion matrices for all future experiments are available in Appendices IV & V

4.2.3 Example Success and Failure Cases and General Comments

Example success and failure cases for both algorithms can be seen in Figure 19. We choose to examine two images, one which both algorithms classified correctly and one which both classified incorrectly. Looking at the original and reconstructed versions of these images it is possible to draw some interesting conclusions. Comparing the two images it is easy to see the semblance between the original and reconstructed version of the correctly classified image. However, the same cannot be said for the incorrectly classified image. Even to the human eye, the inherent information in the image associating it with it's correct class is difficult to discern. We can conclude that as a result of the PCA process, much of the information necessary to differentiate the image from the others is missing, hence both algorithms fail in classifying it correctly.

5. Generative and Discriminative Subspace Learning

We proceed by introducing a hybrid PCA-LDA model that learns the subspace for generative and discriminative features simultaneously[7]. We define the cost functions as inner products for PCA $H_1(e) = \langle e, Se \rangle$ and LDA $H_2(e) = \frac{\langle e, S_b e \rangle}{\langle e, S_w e \rangle + \epsilon}$, where S, S_b, S_w are scatter matrices.[7] $\epsilon > 0$ is a small number that prevents vanishing denominators and e is a unit vector that represents the reduced feature subspace.[8] For $0 \leq t \leq 1$ we define the joint cost function:

$$F_t(e) = \frac{1-t}{2} H_1(e) + \frac{t}{2} H_2(e) \tag{4}$$

which is a linear interpolation of $H_1(e)$ and $H_2(e)$. The primary objective is to find $e_t \in \mathbb{R}^{N_{train}}$ that maximizes $F_t(e)$: $e_t = \arg\min_{e_t} F_t(e)$. We formulate the Lagrangian function with the aim to maximize $F_t(e)$ under the constraint $||e||^2 = 1$:

$$L(e\lambda) = F_t(e) + \lambda(||e||^2 - 1) \tag{5}$$

$$\implies \frac{\partial L(e,\lambda)}{\partial e} = \frac{\partial F_t(e)}{\partial e} + \lambda \frac{\partial (||e||^2 - 1)}{\partial e} = 0 \tag{6}$$

This constrained optimization problem is turned into a unconstrained one:

$$\nabla F_t(e) = (1-t) S_e + \frac{t S_b e}{\langle e, S_w e \rangle + \epsilon} - \frac{t \langle e, S_b e \rangle S_w e}{(\langle e, S_w e \rangle + \epsilon)^2} \tag{7}$$

We aim for $\nabla F_t(e)$ to be parallel to e: $\nabla F_t = \lambda e$. Since S is positive semi-definite and from (7), then $\langle \nabla F_t(e), e \rangle \geq 0$ and $\lambda = \langle \nabla F_t(e), e \rangle$ such that λ is non-negative. If $\lambda > 0$ we define $T(e)$ which maps unit vectors to unit vectors:

$$T(e) = \frac{\nabla F_t(e)}{||\nabla F_t(e)||} \implies T(e) = \frac{\alpha e + \nabla F_t(e)}{||\alpha e + \nabla F_t(e)||} = e \tag{8}$$

For $\alpha > 0$ we add positive multiples of e to both numerator and denominator of (8) to prevent λ from disappearing.

A iterative method similar to constrained gradient descent is utilized to search for a fixed point, unit vector e, where the constrained gradient of $F_t(e)$ vanishes[9].

For matrix A, we arbitrarily initialize random unit vector e_t, then use update $e_{n+1} = T(e_n)$ until $e_{n+1} - e_n \approx 0$. We return the resultant reduced data matrix $\mathrm{A}^R = \mathrm{A}e_t$.

6. LDA Ensemble for Face Recognition

6.1. PCA-LDA

Having examined a theoretical model to learn generative and discriminative features simultaneously, we now move on to the more generally applicable model of Fisherfaces. We experimentally combine PCA with Linear Discriminant Analysis, to include both generative and discriminative features in the subspace which we use for classification. We note that whilst PCA projects over dimensions that maximize the overall data variance, LDA finds the dimensions which best separate the different classes. This is represented in Figure 5.

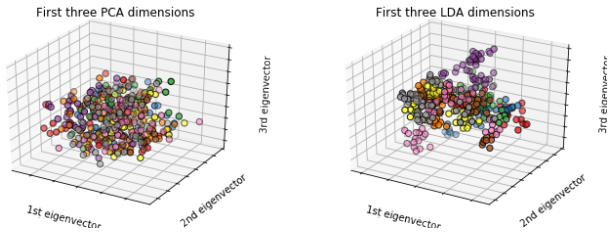

Figure 5. Class separation for PCA and LDA utilizing first 3 dimensions

It does this by maximizing the ratio between the between class scatter matrix S_b and within class scatter matrix S_w as in equation 9. This can be simplified to the problem of determining the set of generalized eigenvectors corresponding to the $\mathcal{M}_{lda}$ largest eigenvalues, as illustrated in equation 10. [6] [2] There are at most $L-1$ non zero eigenvalues, a limit set by the rank of matrix S_b.

$$W_{opt} = argmax_W \frac{|W^T S_b W|}{|W^T S_w W|} = [w_1, w_2, ...w_{\mathcal{M}lda}] \tag{9}$$

$$S_b w_i = \lambda_i S_w w_i \triangleright i \in [1, \mathcal{M}_{lda}] \tag{10}$$

Furthermore, for a straightforward solution to this problem we require that S_w be non-singular. This matrix has rank at most $N_{train} - L$, and in our case $N_{train} << D$ which will result in a singular matrix. To overcome this we perform PCA to reduce dimension, followed by LDA. The

[7] S - Scatter Matrix of X_{train}, S_b - Total Between Class Scatter, S_w - Within Class Scatter. Full mathematical definitions can be found in Appendix II.

[8] 1-dimensional subspace used to aide clarity in explanation

[9] Summarized algorithm can be found in Appendix II

$\mathcal{M}$ number of selected eigenvalues for each process will be denoted $\mathcal{M}_{pca}$ and $\mathcal{M}_{lda}$.

To experimentally find the optimal combination of these we loop through all possible combinations obeying the limits: $\mathcal{M}_{pca} \leq N_{train} - L$ to ensure S_w is non-singular and $\mathcal{M}_{lda} \leq L - 1$, the maximum number of non-zero eigenvalues for LDA, equivalent to the rank of S_b. The results are displayed in the 3D chart in Figure 6.

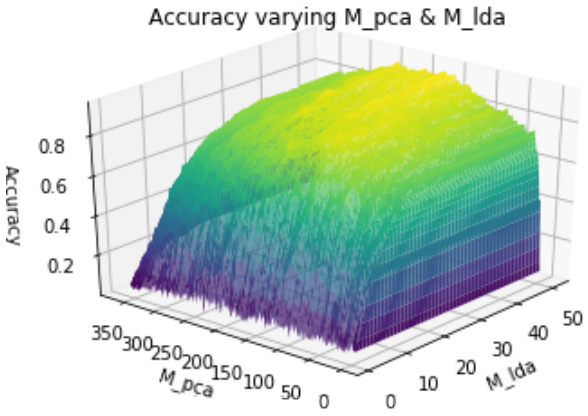

Figure 6. Accuracy (ratio) against varying $\mathcal{M}_{pca}$ and $\mathcal{M}_{lda}$

From the experimental results we see that higher $\mathcal{M}_{lda}$ values provide the best results. This is expected as it means we are capturing more of the discriminative data. What is more interesting is that the $\mathcal{M}_{pca}$ values appear to have an ideal range between 120-250. Anything below this would most likely destroy too much of the discriminative information in the data even before the LDA algorithm is applied, and larger values most likely lead to a higher risk of overfitting. In fact, experimentally $\mathcal{M}_{pca} = 147$ and $\mathcal{M}_{lda} = 46$ produced the best results with an accuracy of 94.23%. The confusion matrix for this example is provided in Figure 16. We see that the addition of discriminant information to the subspace has drastically improved our recognition accuracy, demonstrated by the greater collection of values around the central identity line. Example success and failure cases are displayed in Figure 20.

6.2. PCA-LDA Ensemble

Ensemble learning provides for a means to introduce randomness into our model. An ensemble model consists of a set of classifiers, each randomly different from each other. These decorrelated classifiers ensure greater generalization properties of the overall model. We will investigate the effect this will have on test accuracy for face recognition.

We implement ensemble learning with two main methods, Bootstrap Aggregating (Bagging, randomization on training samples) and randomization in the feature space.

6.2.1 Projection to Lower Dimension

Before working with ensemble techniques, we apply a first stage PCA to the dataset to reduce its dimensionality to $N_{train} - 1$ as this is the maximum number of non zero eigenvalues present in the data. This allows us to obtain a much smaller matrix much more suited to generating random subsets both in the sample and feature space. Note for the PCA and LDA processes in each of the ensemble models we retune to find the new ideal values for $\mathcal{M}_{pca}$ and $\mathcal{M}_{lda}$.

6.2.2 Fusion Rules

There exists a range of methods (sum, product, min, max, majority voting) to combine the decision functions of the set of classifiers and determine the output class of the ensemble model. However the first four of these require probability distributions for the decision functions of the individual classifiers, whereas it is possible to build a majority voting rule using just the output labels from the classifiers. [5] For the NN classifier we don't have useful information on the probability distributions of output functions and hence it is difficult to utilize any of the prior methods. It is possible to derive representative probabilities for k-NN and this method was trialed, however the produced results were below satisfactory. Hence, majority voting was used for the following sections as a more concrete method.

6.2.3 Randomization on Training Samples (Bagging)

In this type of ensemble learning we generate T bootstrap replicates $\{S_t\}_{t=1}^{T}$. Each of these replicates contains a subset of the images present in X_{train} selected with replacement. We then create T PCA-LDA-NN estimators. We define the randomness parameter $\rho = |\mathcal{T}_t|$, $t \in [1, T]$ as the number of selected images from the total set of images $|\mathcal{T}| = N_{train}$. A high value of ρ indicates high model correlation as each of the models is likely to contain similar samples as the others. A low value of ρ indicates more decorrelation between the models as each is fitted on a smaller randomized subset of the data which may be more varied.

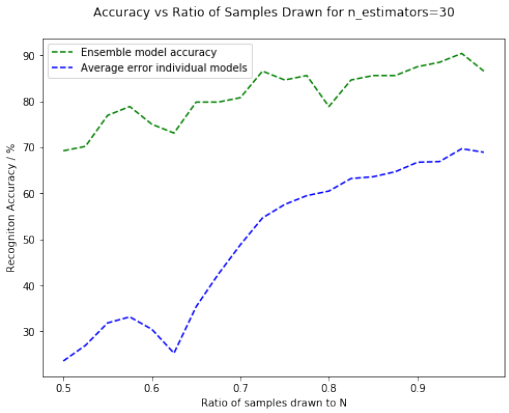

Figure 7. Accuracy of Committee Machine and Average Individual Models vs $\rho/|\mathcal{T}|$

In Figure 7, we examine the effects of changing this value, represented here as a ratio of selected samples over N_{train}, $\rho/|\mathcal{T}|$. We plot both the accuracies of the committee machine as well as the average accuracies of the indi-

vidual models for number of estimators = 30. We observe as expected, that as the ratio drops, accuracies of individual models drop drastically as the amount of training data available to them becomes very low. Due to the power of the ensemble model however, the overall model accuracy is always greater than that of individual models and stays safely above 70% for all cases. Interestingly, we see that the experimentally ideal value for ρ is around 0.95 of N_{train}(Figure 17). We can tie this down to the fact that we already have a very small data set, and the adverse effects on individual models of the very small training set (resulting from small values of ρ) overpower the benefits of ensemble randomization.[4] Furthermore, as X_{train} and X_{test} are inherently correlated, it can be expected that an increased training error may lead to an increased test error for this data-set.

6.2.4 Randomization in Feature Space

For random sampling in the feature space we instead pick a random subset of the $N_{train} - 1$ dimensions in the W matrix resulting from the initial PCA. Here we separate our selection into a fixed set of the largest $\mathcal{M}_0$ eigenfaces in W, and $\mathcal{M}_1$ dimensions randomly selected from the remaining $N_{train} - 1 - \mathcal{M}_0$ eigenfaces. Hence each model receives a random subset of the dimensions. Here we can define the randomness parameter ρ as $\mathcal{M}_0 + \mathcal{M}_1$, the subset of dimensions used. However it is also worth stating that a better measure of randomness could be $\mathcal{M}_1/(N_{train} - 1 - \mathcal{M}_0)$ to only take into account elements randomized between models.

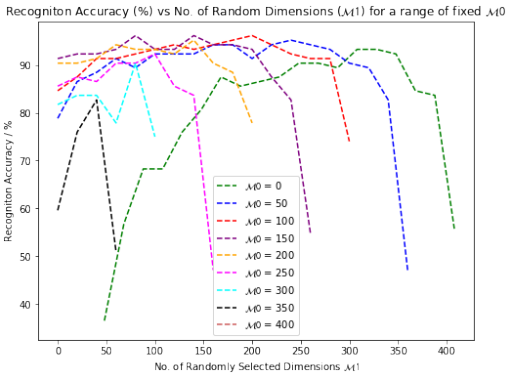

Figure 8. Recognition Accuracy vs $\mathcal{M}_0$ and $\mathcal{M}_1$

We examine experimentally the behaviour of accuracy for a range of $\mathcal{M}_0$ and $\mathcal{M}_1$ (Figure 8). The best case accuracy was 96.15% for $\mathcal{M}_0 = 100$ and $\mathcal{M}_1 = 200$ (Figure 18). We examine that for each level of $\mathcal{M}_0$, high values of $\mathcal{M}_1$ result in the S_w matrix becoming singular which results in a drop in accuracy. For low values of $\mathcal{M}_0$ we struggle to get good accuracies without significantly increasing $\mathcal{M}_1$. Mid range values of $\mathcal{M}_1$ appear to provide for the best generalization properties and results.

6.2.5 Varying Number of Estimators / Base Models

From Figure 9, conducted with bagging with $\rho/|\mathcal{T}| = 0.95$ and random subspace with $\mathcal{M}_0 = 100$ and $\mathcal{M}_1 = 200$, we can see that the minimum number of estimators to achieve satisfactory results appears to be around 8-10, however using more than 30 estimators appears to provide for the best range of accuracies. We also observe that random subspace appears to consistently provide better results than bagging. This can be explained through the fact that $N_{train} << D$. This means that we have much more room for observing the benefits of randomization over the larger set of dimensions, as compared to the relatively small set N_{train}.

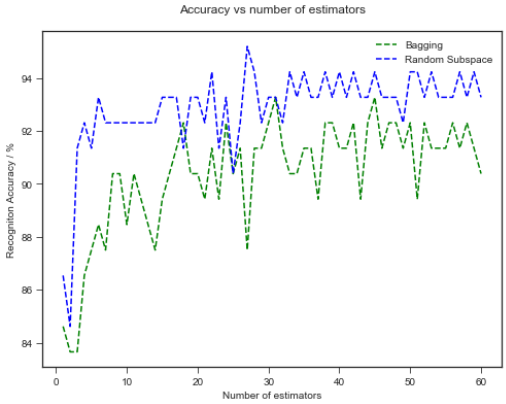

Figure 9. Accuracy vs number of estimators

7. Conclusion

As a result of our exploration into various image classification techniques we managed to increase our initial accuracy of 67.31% with PCA and k-NN to 96.15% with LDA-PCA ensemble with random subspace. We have seen that the major increase in this accuracy resulted from the inclusion of discriminant information by applying LDA to the data. Whats more we saw that ensemble techniques were not able to provide a significant increase in accuracy, with random subspace providing small bump of 1-2%. For our highest accuracy produced in the random subspace section we have the following 5 faces within X_{test} which the algorithm fails to classify (Figure 10). To further improve on accuracy future work may include the addition of more sophisticated classification algorithms to replace k-NN, an example to these may be multi-class SVM. With the integration of such methods we may move further towards 100% recognition accuracy.

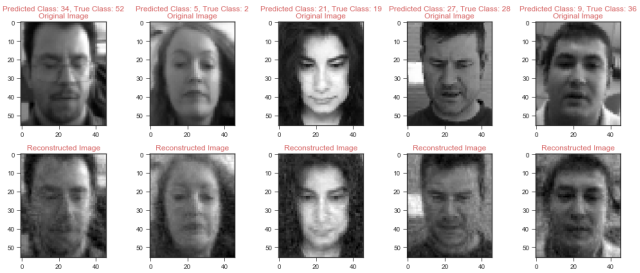

Figure 10. All failure cases for best case accuracy (PCA-LDA Ensemble with random subspace)

References

[1] P. Honeine. An eigenanalysis of data centering in machine learning, Jul 10, 2014.

[2] T.-K. Kim. *Low-Dimensional Computation of Eigenspace*, pages 14–15. Optimization in Pattern Recognition, Lecture 3, Pattern Recognition. 2018.

[3] U. Lorenzo-Seva. How to report the percentage of explained common variance in exploratory factor analysis. pages 2–6. Department of Psychology, Universitat Rovira i Virgili, Tarragona, 2003.

[4] D. Opitz. 2.1 bagging classifiers. 1999.

[5] V. Smolyakov. Ensemble learning to improve machine learning results, 2017.

[6] N. Vasconcelos. Pca and lda, 2003.

[7] N. Zhao, W. Mio, and X. Liu. A hybrid pca-lda model for dimension reduction. pages 2184–2190. IEEE, Jul 2011.

Appendix I - Mean Face and Eigenfaces

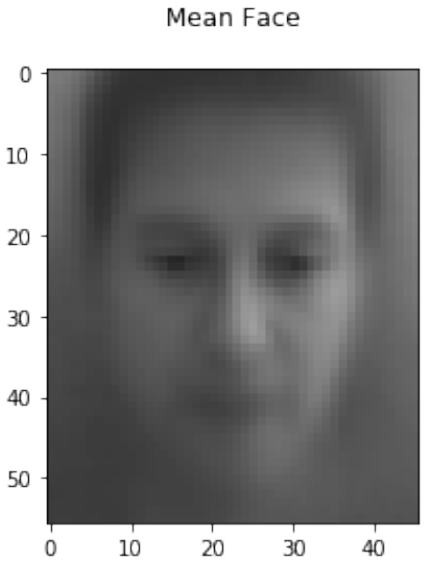

Figure 11. Mean Face of X_{train}

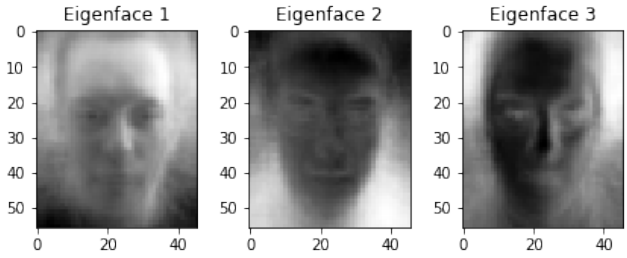

Figure 12. First 3 Eigenfaces of X_{train}

Appendix II - Hybrid PCA-LDA model

Definitions

We redefine a feature vector from X_{train} as $x_{ij} \in \mathbb{R}^D$ where $1 \leq i \leq L$ amd $1 \leq j \leq C_i$ where $C_i = 8$ is the number of faces within each class of X_{train}.

$\bar{x}$ remains the mean of X_{train}, but we also define the mean of the i^{th} class of X_{train} as:

$$\bar{x_i} = \frac{1}{C_i} \sum_{j=1}^{C_i} x_{ij} \tag{11}$$

We can now define the scatter matrices S, S_i, S_b, S_w.

Scatter Matrix of X_{train}:

$$S = \sum_{i=1}^{L} \sum_{j=1}^{C_i} (x_{ij} - \bar{x})(x_{ij} - \bar{x})^T \tag{12}$$

Scatter Matrix of i^{th} Class:

$$S_i = \sum_{j=1}^{C_i} (x_{ij} - \bar{x_i})(x_{ij} - \bar{x_i})^T \tag{13}$$

Within Class Scatter Matrix:

$$S_w = \sum_{i=1}^{L} S_i \tag{14}$$

Total Between Class Scatter Matrix:

$$S_b = \sum_{i=1}^{L} C_i (\bar{x_i} - \bar{x})(\bar{x_i} - \bar{x})^T \tag{15}$$

Algorithm

Algorithm 3 Hybrid PCA-LDA

Input: Orthonormal feature matrix A
Output: Reduced feature matrix A^R
begin:
1: Arbitrarily initialize random unit vector $e_t \in \mathbb{R}^{N_{train}}$
2: For $\alpha > 0$, **iterate**: $e_{t+1} = T(e_t)$
3: **until**: $\Delta e = e_{t+1} - e_t \approx 0$ ▷ where $t_{ideal} = t + 1$
4: **return**: $A^R = Ae_{t_{ideal}}$
end

Appendix III - Recognition Accuracy vs $\mathcal{M}$ for Alternative Method

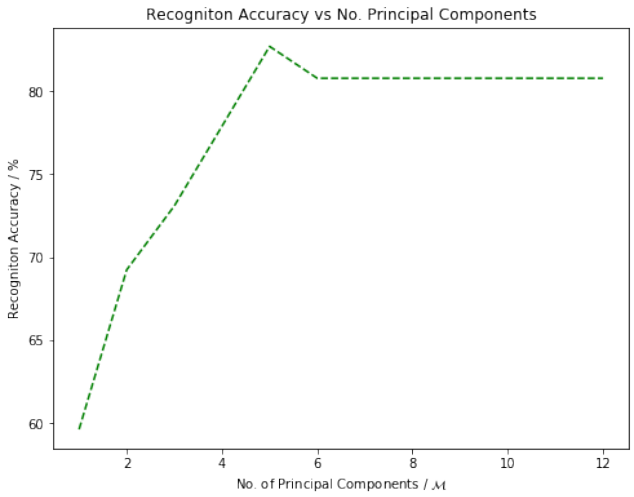

Figure 13. Recogniton Accuracy vs No. Principal Components ($\mathcal{M}$) for Alternative Method

Appendix IV - Confusion Matrices

Appendix IV.I

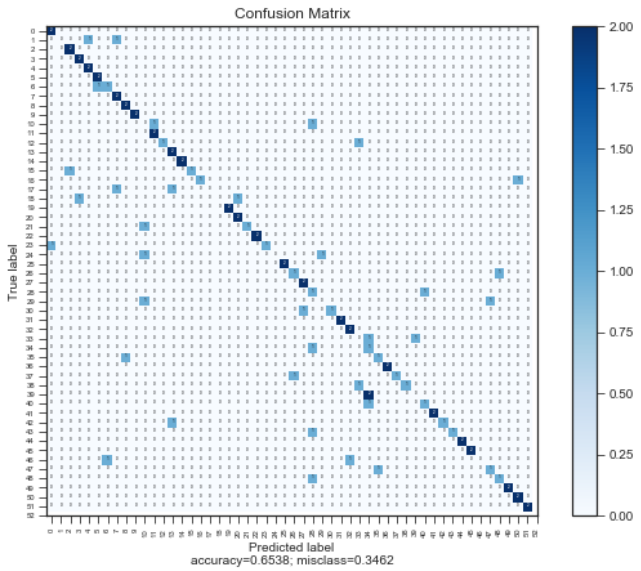

Figure 14. Confusion Matrix for k-NN with k = 1 and $\mathcal{M}$ = 81

Appendix IV.II

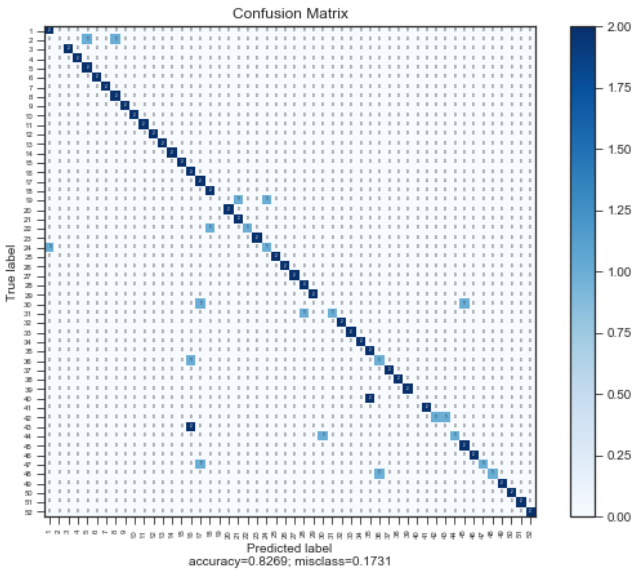

Figure 15. Confusion Matrix for Alternate Method with $\mathcal{M} = 5$

Appendix IV.III

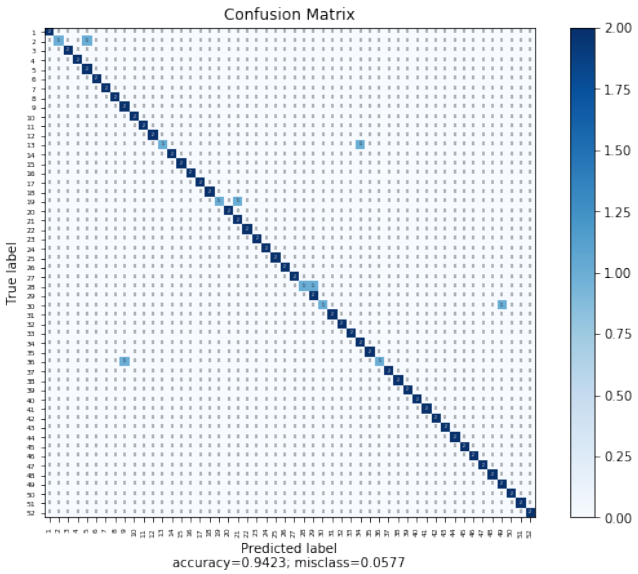

Figure 16. Confusion Matrix for PCA-LDA with NN classifier for $\mathcal{M}_{pca} = 147$ and $\mathcal{M}_{lda} = 46$

Appendix IV.IV

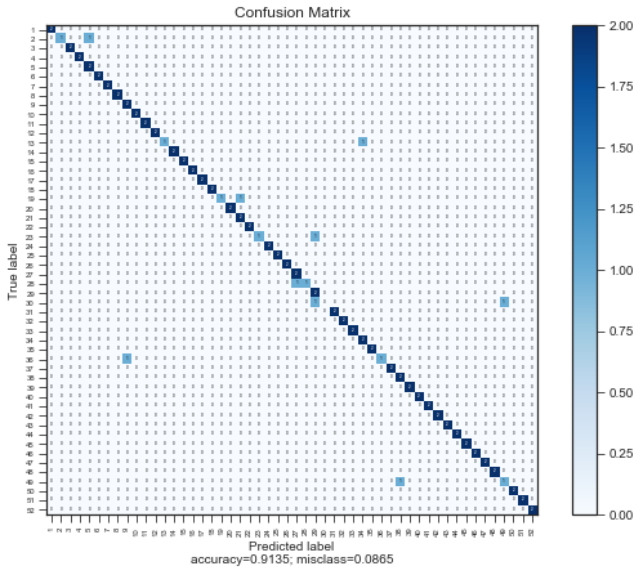

Figure 17. Confusion Matrix for PCA-LDA Ensemble with Bagging, $\rho/|\mathcal{T}| = 0.95$

Appendix IV.V

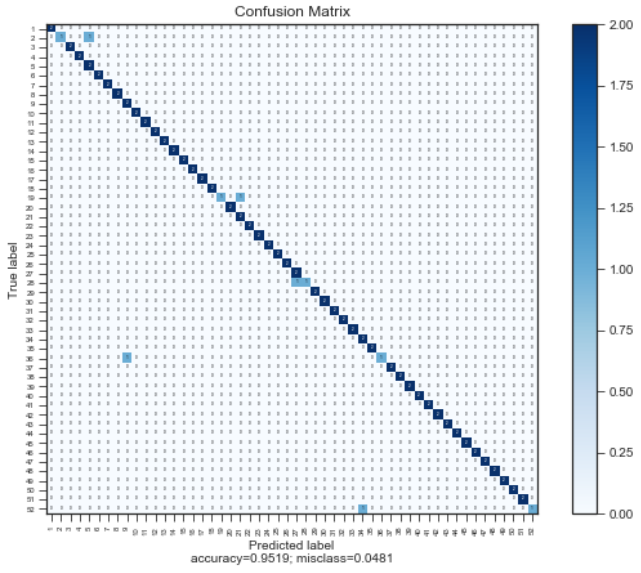

Figure 18. Confusion Matrix for PCA-LDA Ensemble with Random Subspace $\mathcal{M}_0 = 100$ and $\mathcal{M}_1 = 200$

Appendix V - Example Success and Failure Cases

Appendix V.I

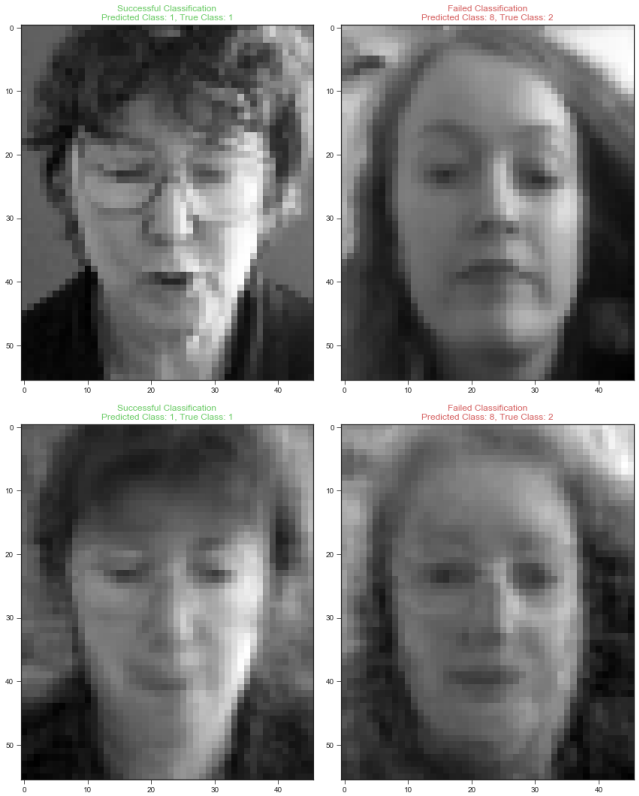

Figure 19. Example Success and Failure Cases for k-NN and Alternate Method (Above: original image, Below: reconstructed image)

Appendix V.II

Figure 20. Example Success and Failure Cases for PCA-LDA

Appendix VI - Time, Memory Usage

Appendix VI.I

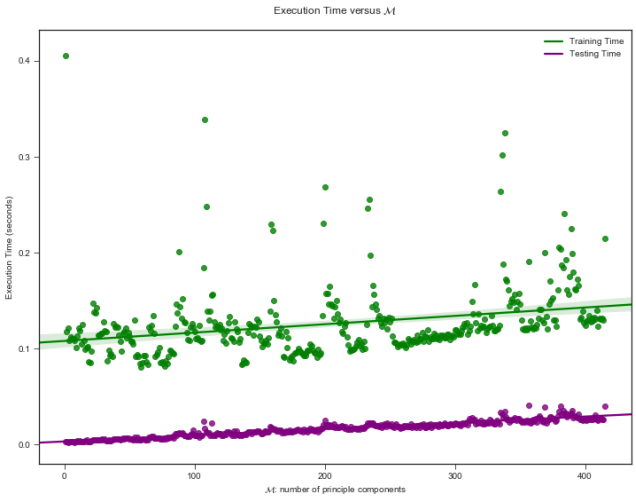

Figure 21. Execution Time versus $\mathcal{M}$ for k-NN

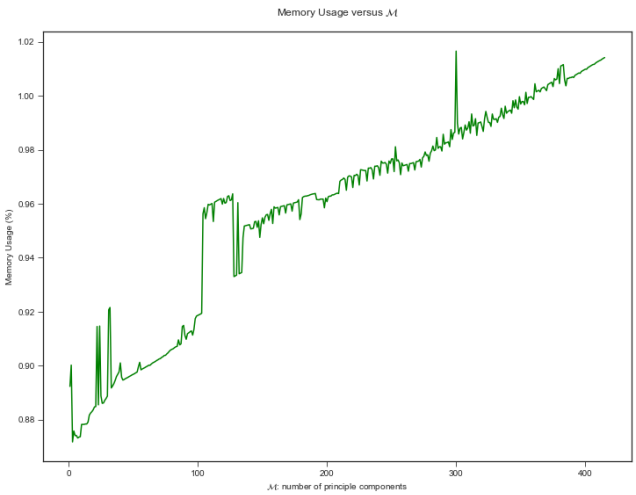

Figure 22. Memory Usage versus $\mathcal{M}$ for k-NN

Appendix VI.II

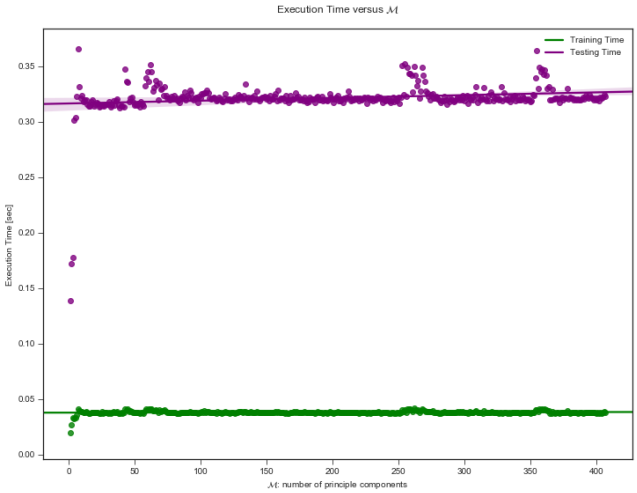

Figure 23. Execution Time versus $\mathcal{M}$ for Alternative Method

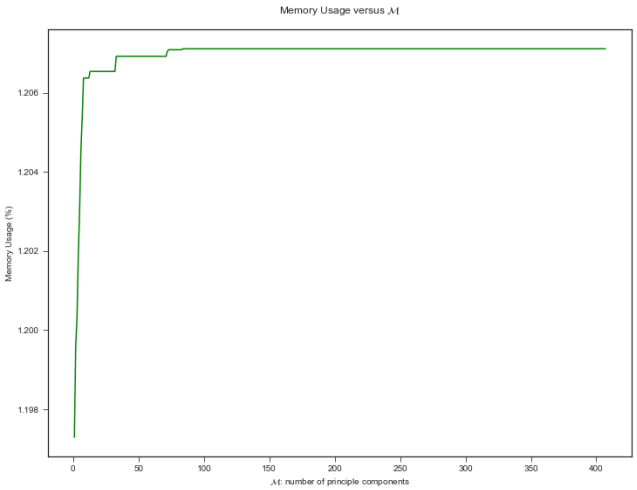

Figure 24. Memory Usage versus $\mathcal{M}$ for Alternative Method

Appendix VII - Face Reconstruction vs $\mathcal{M}$

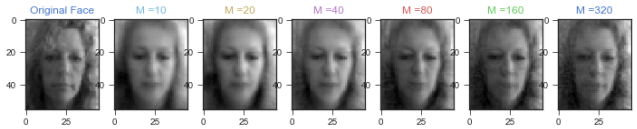

Figure 25. Face Reconstruction versus $\mathcal{M}$ for class ID 8

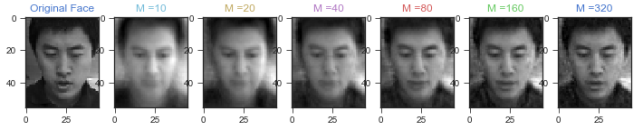

Figure 26. Face Reconstruction versus $\mathcal{M}$ for class ID 51