CS479 - Programming Assignment 3: Classifying faces and dimensionality reduction through Principal Component Analysis

Tim Kwist and Shane Melton

Due April 11, 2016 - Submitted April 11, 2016

Contents

1 Technical Discussion

In this project, our goal is to recognize and classify faces using M. Turk and A. Pentland's "Eigenfaces for Recognition" methodology, while utilizing Principle Component Analysis (PCA) to reduce the dimensionality of our input variables. At a high level, PCA simply reduces the number of dimensions by finding the k-most eigenvalues that retain X% of the data portrayed by the original N variables (k ¡¡ N). In this experiment, we'll be using the eigenvalues that retain 80%, 90%, and 95% of the original data.

The training set and test set used consist of images from the FERET face database. Each image contains a face in frontal pose, with all faces normalized in regards to orientation, position, and size. Additionally, they have been masked to include only the face region. The training set contains 1204 images from 867 subjects, while the test set contains 1196 images from 866 subjects (There is only one subject in the training set who is not in the test set). Additionally, there are two sets of each of these sets: one of low resolution (16 x 20) and one of high resolution (48 x 60).

For each image used for training and testing purposes, the image's name includes a 5 digit identity number, with identity numbers in the training set correlating to the same faces of the identities in the test set.

1.1 Steps for carrying out Principal Component Analysis for face recognition

1. Obtain face images I_1, I_2, ... , I_M (training faces) - all centered and of the same size
2. For all M images, represent the individual image $\mathbf{I_i}$ from NxN matrix into an N^2x1 vector $\mathbf{\Gamma_i}$, and form the N^2xM matrix $\mathbf{\Gamma}$
3. Compute the average face vector $\mathbf{\Psi}$:
$$\mathbf{\Psi} = \frac{1}{M}\sum_{1}^{M}(\mathbf{\Gamma_i}) \tag{1}$$
4. Center each face around 0 by subtracting the mean face:
$$\mathbf{\Phi_i} = \mathbf{\Gamma_i} - \mathbf{\Psi} \tag{2}$$
5. Using matrix $\mathbf{A} = [\mathbf{\Phi_1\Phi_2...\Phi_M}]$, find the M best eigenvectors, $\mathbf{u}$, and M best eigenvalues λ by computing $\mathbf{A^T A}$:
$$\mathbf{A^T A v} = \mu\mathbf{v} \tag{3}$$
$$\begin{aligned} \mathbf{u} &= \mathbf{Av} \\ \lambda &= \mu \end{aligned} \tag{4}$$
Important: normalize eigenvectors of $\mathbf{u}$ such that $\|\mathbf{u_i}\| = 1$
6. Keep only K eigenvectors (corresponding to the K largest eigenvalues)

1.2 Eigenfaces for Regonition

1. Find the K best eigenvectors using PCA (described above) that represent X% of the data
$$\frac{\sum_1^K \lambda_\mathbf{i}}{\sum_1^M \lambda_\mathbf{i}} >= X\% \tag{5}$$
2. Given unknown image $\mathbf{\Gamma}$
   (a) Normalize image
$$\mathbf{\Phi} = \mathbf{Gamma} - \mathbf{Psi} \tag{6}$$

(b) Project onto face space

$$\hat{\mathbf{\Phi}} = \sum_{i=1}^{K}(\mathbf{w_i u_i}) + \mathbf{\Psi} \quad (\mathbf{w_i} = \mathbf{u_i^T \Phi}) \quad (\|\mathbf{u_i}\| = 1) \tag{7}$$

(c) Compute distance in face space

$$e_d = \|\mathbf{\Phi} - \hat{\mathbf{\Phi}}\| \tag{8}$$

(d) If e_d ¡ T_d (where T_d is some threshold), then Γ is a face

(e) If Γ is a face, compute the distance from the projected Γ to a projection of each training face, and classify based on the smallest distance.

$$classification = min_i(\|\widehat{\mathbf{\Phi_{training-image-i}}} - \widehat{\mathbf{\Phi_{unknown-image}}}\|) \tag{9}$$

1.3 Procedure for experiment - Part A

1. Follow the steps outlined in the first section regarding PCA for face recognition using the high resolution training set

2. Display the following:

   (a) The average face

   (b) The eigenfaces corresponding to the top 10 largest eigenvalues

   (c) The eigenfaces corresponding to the top 10 smallest eigenvalues

3. Choose the top eigenvectors preserving 80% of the information. Follow the steps outlined in the second section, "Eigenfaces for Recognition", ignoring the threshold requirement and skipping directly to classifying each face in the high resolution test set. Choose the top N (described below) training faces having the smallest distance from the new face to-be-classified. If the identity of the test image is within the N training faces selected, consider this image correctly matched; otherwise, consider this image incorrectly matched.

   After going through all test images, count the number of correctly matched images and divide by the total number of images in the test set to obtain the identification accuracy. Draw a Cumulative Match Characteristic (CMC) curve, using N (from 1 to 50) as the x coordinate and the identification accuracy for that given N as the y coordinate.

4. Using N=1, show 3 projected images which are correctly matched, along with the corresponding best matched training samples.

5. Using N=1, show 3 query images which are incorrectly matched, along with the corresponding mismatched training samples

6. Repeat steps 3 - 5 by keeping the top eigenvectors corresponding to 90% and 95% of the information in the data, plotting the CMC curves on the same graph for comparison purposes.

1.4 Procedure for experiment - Part B

1. Remove all images of the first 50 subjects from the training set.

2. Find the eigenvectors corresponding to 95% of the information in the data.

3. Using the original test set (ie, no subjects removed), compute the distance from face space (e_k for each image and accept each image as either a match or not a match based on its relation to a threshold (T) (ie, accept only if e_k ¡ T). Plot the False Positive (# false positives / # negatives) vs True Positive (# true positives / # positives) rate against various thresholds.

1.5 Procedure for experiment - Part C

1. Repeat Part A and B using the low resolution images instead of the high resolution images, for training and testing.

2 Results

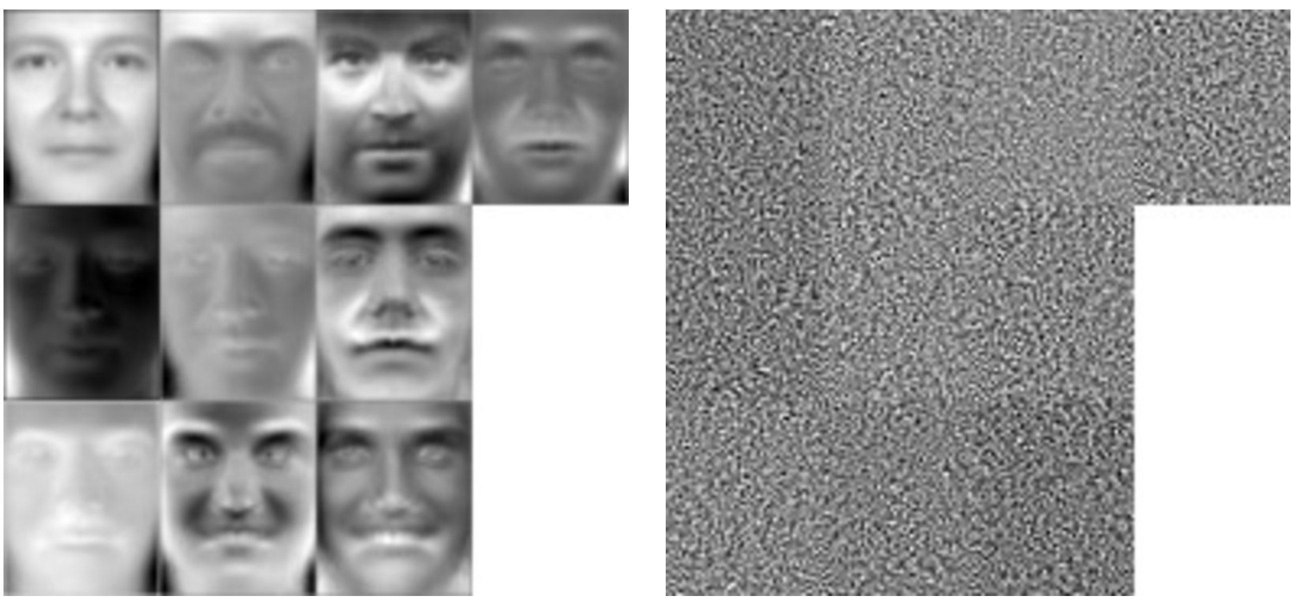
After studying and becoming acquainted with the technical details and running a few preliminary tests to make sure all our functions and algorithms were functioning properly we began experimenting with various thresholds and data preservation rates when performing dimensionality reduction.

2.1 Part A

2.1.1 Part A.I

Here is the average face for the high quality images (60 x 48), the top 10 eigenfaces, and the bottom 10 eigenfaces.

Figure 1: The computed average face for part A.I



(a) The top 10 eigenfaces

(b) The bottom 10 eigenfaces

Figure 2: The top (a) and bottom (b) eigenfaces for part A.I

2.1.2 Part A.II

Here is our cumulative match characteristic curve generated using N values from 1 to 50. We can see that by preserving more information with a greater number of eigenvectors we are able to improve the overall performance of classification significantly. This is likely due to having more information, or features, to compare each face too which improves performance.

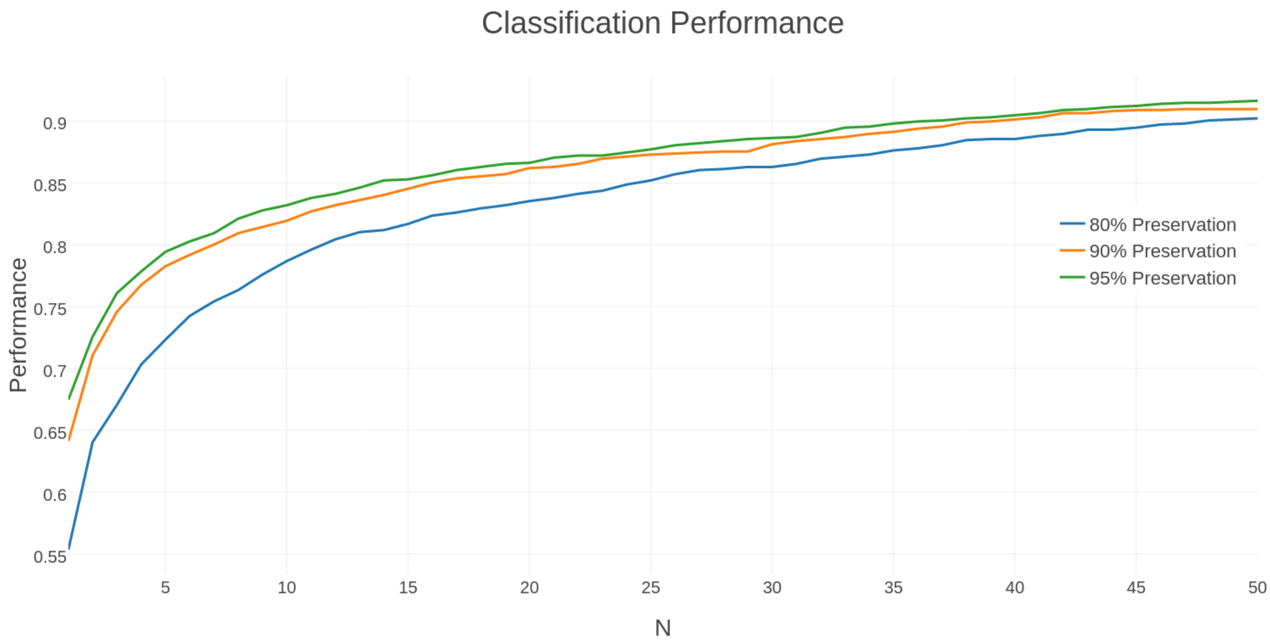
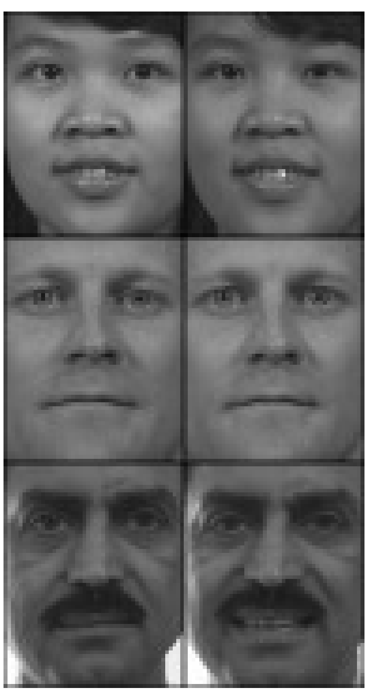


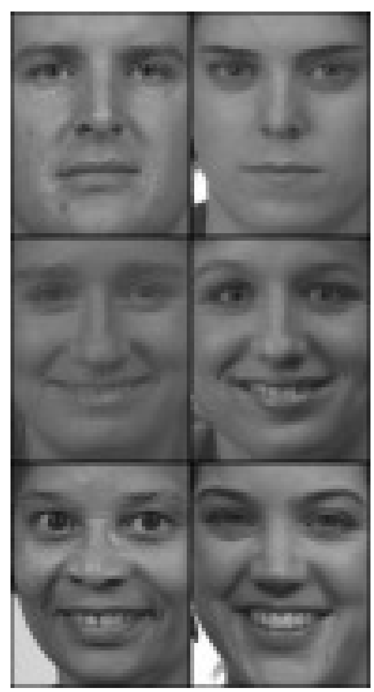
Figure 3: Cumulative match characteristic curve for part A.II.

2.1.3 Part A.III & A.IV

Below you can see 3 correctly matched query images (and their corresponding training sample) and 3 incorrectly matched query images (and their corresponding training sample) for each percentage of retained information. **Note: in the images below the query image is on the *right* and the training image is on the *left*.**



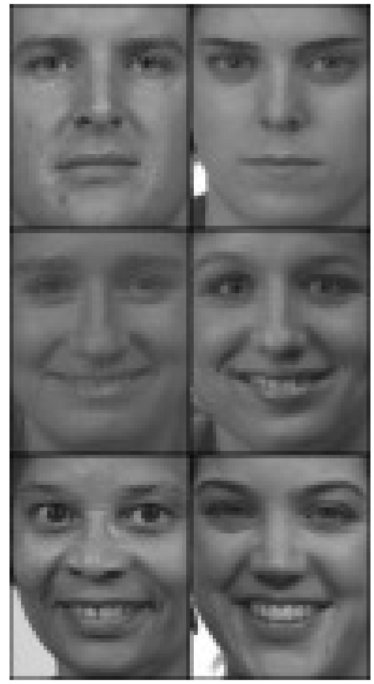
(a) Correctly matched query images (80%)



(b) Incorrectly matched query images (80%)

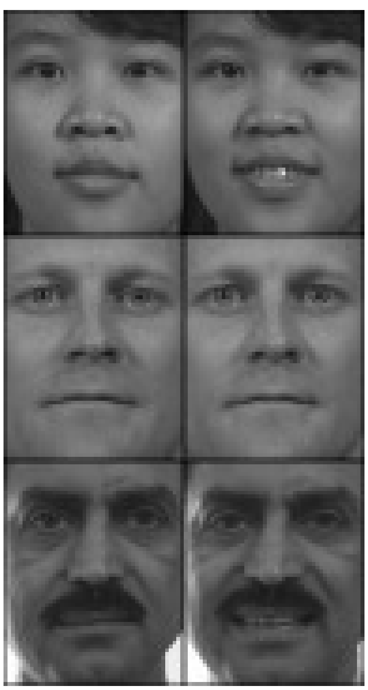
Figure 4: The correctly (a) and incorrectly (b) eigenfaces with 80% information preservation. (training on left, query on right)

(a) Correctly matched query images (90%)



(b) Incorrectly matched query images (90%)

Figure 5: The correctly (a) and incorrectly (b) eigenfaces with 90% information preservation. (training on left, query on right)



(a) Correctly matched query images (95%)

(b) Incorrectly matched query images (95%)

Figure 6: The correctly (a) and incorrectly (b) eigenfaces with 95% information preservation. (training on left, query on right)

2.2 Part B

Below you will find our ROC curve for part B, where we removed 50 subjects from the training set and used the eigenvectors corresponding to 95% of the information and classified subjects as intruders or non-intruders. We adjusted the threshold to produce the following ROC curve.

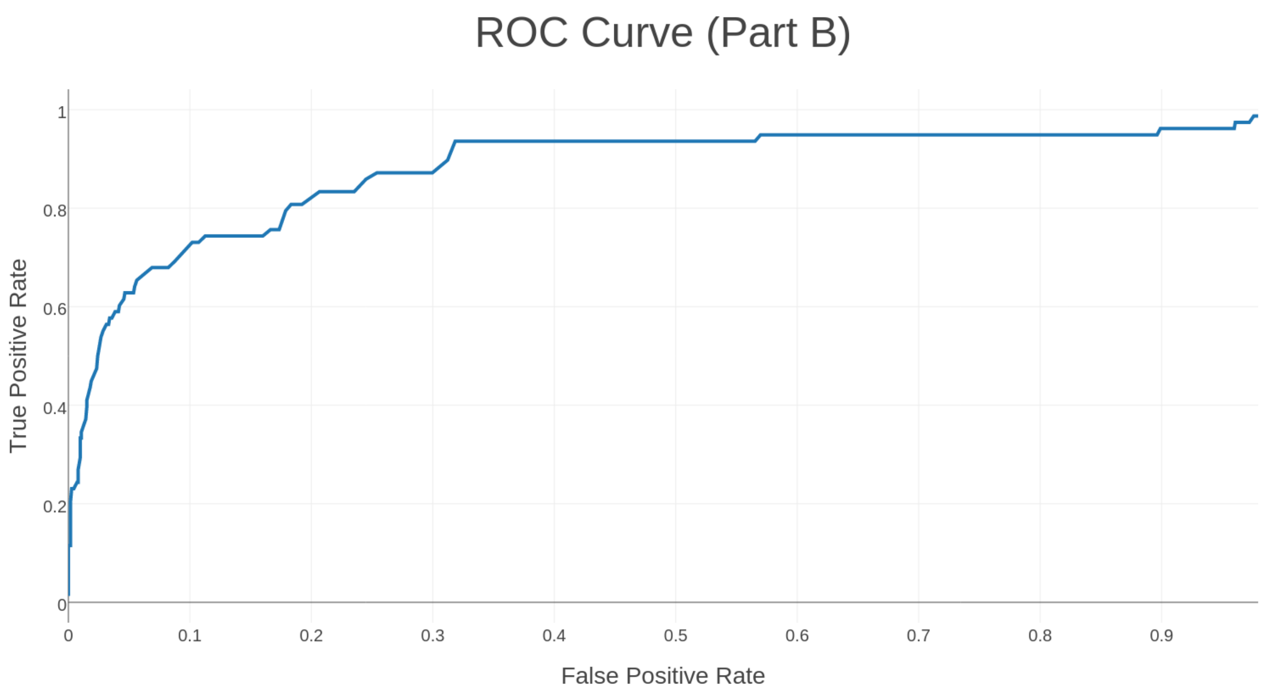


Figure 7: ROC curve generated by varying a threshold for classification.

2.3 Part C (Graduate / Extra Credit Problem)

2.3.1 Part C.I

Here is the average face for the high quality images (60 x 48), the top 10 eigenfaces, and the bottom 10 eigenfaces.

Figure 8: The computed average face for part C.I

(a) The top 10 eigenfaces

(b) The bottom 10 eigenfaces

Figure 9: The top (a) and bottom (b) eigenfaces for part C.I

2.3.2 Part C.II

Here is our cumulative match characteristic curve generated using N values from 1 to 50. In this instance we are using the low quality images to compare our results with the high quality images.

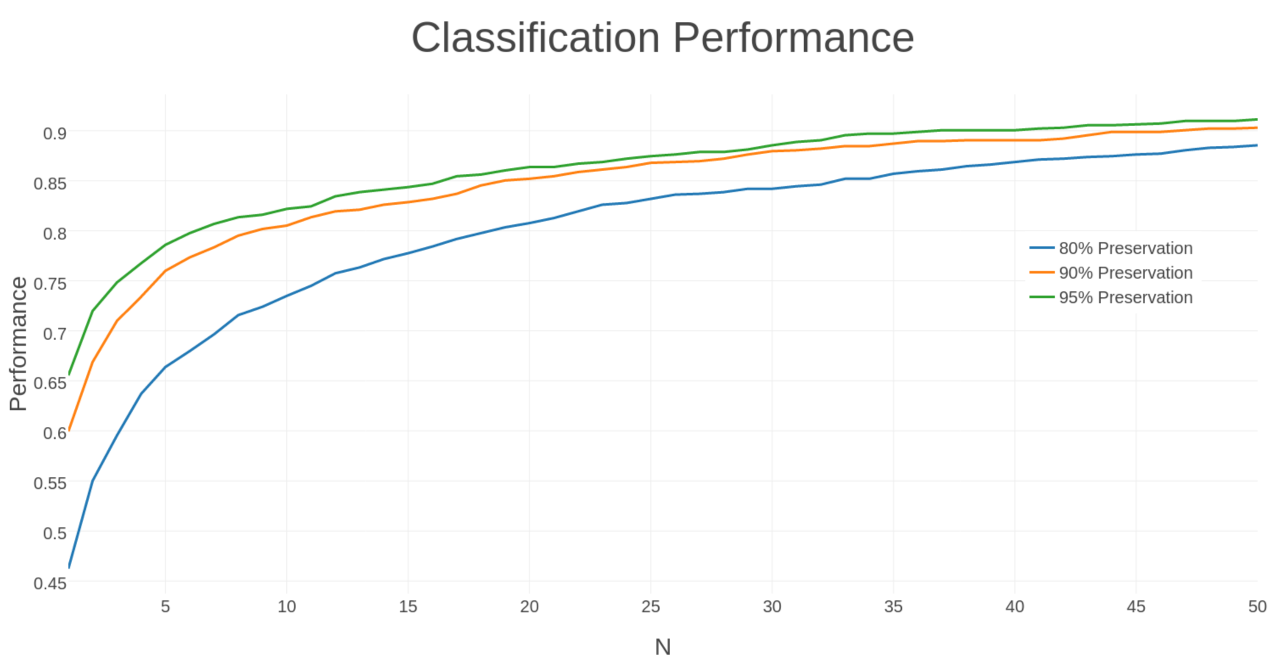


Figure 10: Cumulative match characteristic curve for part C.II.

2.3.3 Part C.III & C.IV

Below you can see 3 correctly matched query images (and their corresponding training sample) and 3 incorrectly matched query images (and their corresponding training sample). **Note: in the images below the query image is on the *right* and the training image is on the *left*.**

(a) Correctly matched query images (80%)

(b) Incorrectly matched query images (80%)

Figure 11: The correctly (a) and incorrectly (b) eigenfaces with 80% information preservation. (training on left, query on right)

(a) Correctly matched query images (90%)

(b) Incorrectly matched query images (90%)

Figure 12: The correctly (a) and incorrectly (b) eigenfaces with 90% information preservation. (training on left, query on right)

(a) Correctly matched query images (95%)

(b) Incorrectly matched query images (95%)

Figure 13: The correctly (a) and incorrectly (b) eigenfaces with 95% information preservation. (training on left, query on right)

2.4 Part D (Graduate / Extra Credit Problem)

Below you will find our ROC curve for part D (using the low resolution images), where we removed 50 subjects from the training set and used the eigenvectors corresponding to 95% of the information and classified subjects as intruders or non-intruders. We adjusted the threshold to produce the following ROC curve.

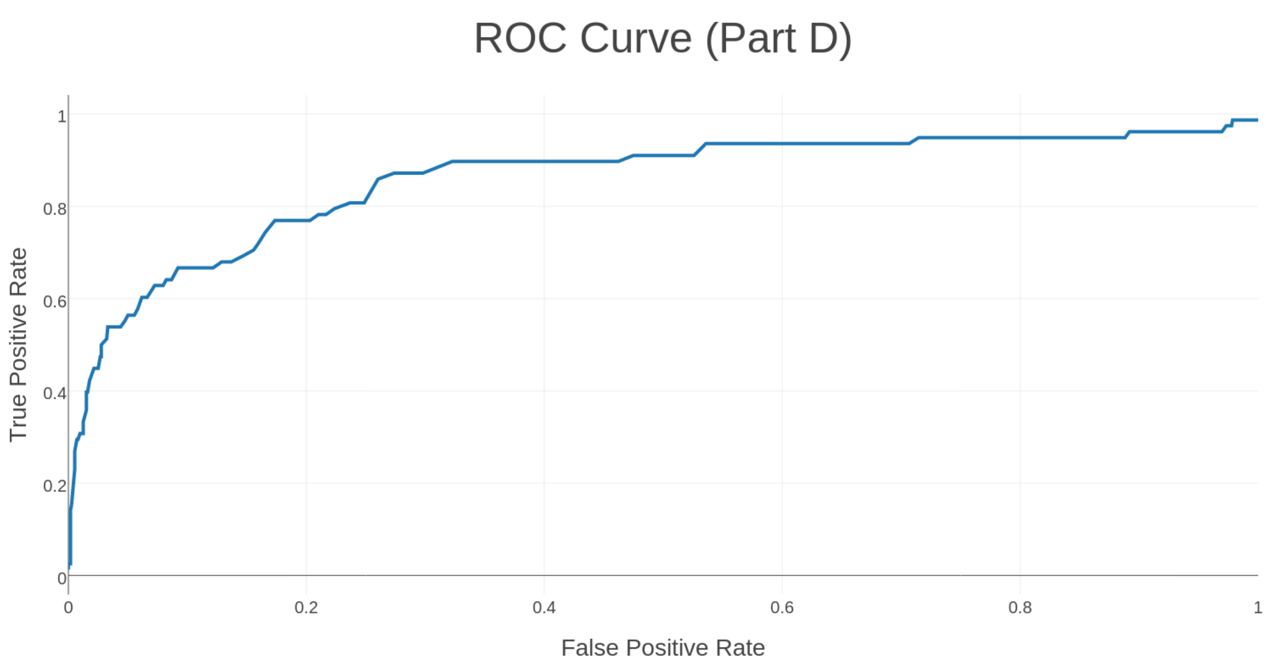


Figure 14: ROC curve generated by varying a threshold for classification with low resolution images.

2.5 Part E (Graduate / Extra Credit Problem)

When comparing our results between high resolution images and low resolutions images we do not see too much variation between their identification performances. At most, we see the high resolution images performing just slightly better than the low resolution. However, this slight performance increase comes at the cost of a much more intensive and time consuming computation cost for the high resolution images. This small performance increase likely stems from each high quality image having more data, which translates to more eigenfaces when reduced using PCA, resulting in a better performance of classification / identification.

3 Division of Work

Shane:

- Program classifier and classifier threshold methods
- Generate and compile data and charts
- Results write-up
- Organize main files
- Organize data output and results

Tim:

- Technical write-up
- Working proto-type of image read/write data
- Program base functions for compute error distance, project onto eigenspace, and calculate / write to file eigenvectors and values
- Program documentation + Readme
- Concatenate images together

Both (Essentially pair programming):

- Debugging of image threshold and classification
- Developing overall main files
- General discussion of project requirements and planning of project

4 Program Listings

Source will be sent by email and can be found on Github at the following URL:
`https://github.com/timkwist/CS479/tree/master/PA3-Eigenfaces`

Eigen library can be found at the following URL:
`http://eigen.tuxfamily.org/index.php?title=Main_Page`

Box-Muller Transformation C++ Code:
`ftp://ftp.taygeta.com/pub/c/boxmuller.c`

Image Mainpulation Classes:
`http://www.cse.unr.edu/~bebis/CS302/`

Matthew Turk and Alex Pentland's "Eigenfaces for Recognition":
`http://www.cse.unr.edu/~bebis/CS479/Readings/TurkEigenfaces.pdf`