Midterm Project

Oscar Martinez

Florida State University
Department of Statistics
Instructor: Dr. Wei Wu

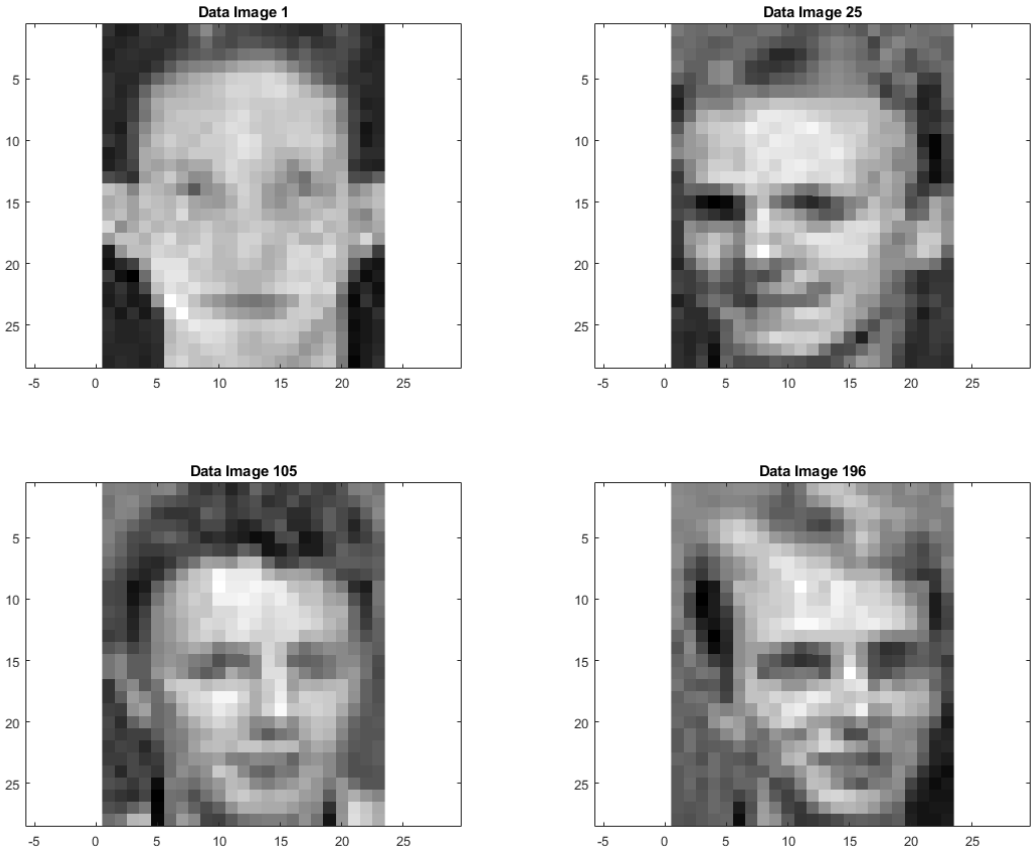
October 24, 2019

1. Problem Statement

Identification and classification of objects in images is a task that is arguably as old as images themselves. Identification and classification of these objects has traditionally been done by humans, however this method requires both time and workers and still allows for mistakes which are costly and difficult catch. The problem of identification is essentially this: "given an image (or multiple images) of a particular scene, the goal is to detect and identify objects of interest in that image(s)." [1] Video surveillance, monitoring of dyed cell cultures, examination of medical imaging, geospatial mapping, and other areas would benefit from computer-aided image identification.

In this project, we seek to identify and classify images which consist of people's facial profiles. The figures in section 1.(a) illustrate examples of the images we will be dealing with:

(a) Image Examples



The goal of the problem is to match each image with the individual whose facial profile is the basis of the image. The rest of the paper is organized as follows: Section 2. details the methodology of the analysis, Section 3. presents the code used to perform this analysis,

[1] Florida State University. Course: STA 5106. Midterm Prompt. Fall 2019

Section 4. presents the results in terms of accuracy of the analysis as well as provides examples of when our methodology was correct versus incorrect, and Section 5. concludes.

2. Methodology and Approach

(a) Background

Due to the inherent high-dimensionality of images, image analysis can be costly in terms of computational power needed. Thus, it is often beneficial to reduce their dimensionality before proceeding with the analysis. This reduction presents a trade-off between efficiency and accuracy; higher dimensionality is generally more accurate but also more computationally expensive (more inefficient). Several ways to reduce dimensionality while maintaining accuracy have been developed and implemented such as principal component analysis (PCA) and linear discriminant analysis (LDA) among other methods.

In this paper, we analyze the trade-offs between dimensionality and accuracy for PCA as compared to a simple projection for dimension reduction. In essence, PCA is used to transform data $\mathbf{X}$ with corresponding dimension N into a subspace $\mathbf{Y}$ with corresponding dimension K, where K is generally less than N, while keeping as much as possible of the variation in the data. It does so by identifying a new set of uncorrelated variables called the principal components (PC) which are ordered such that each subsequent PC has less variation than the previous component.[2] Thus, high-dimension data can be represented by lower-dimension data while maintaining the majority of variation. This allows for similarities and differences among the data to be more efficiently analyzed in comparison to using a simple projection of the original data onto a lower subspace.

(b) Data Structure

In this project, there are two sets of data, Y_{train} and Y_{test} of which Y_{train} have implicitly already been identified for the desired object. That is, that the person whose facial profile composes the image is known. In contrast, Y_{train} is implicitly unknown. The use of implicit is because the two sets of data have the same structure and the persons whose facial profiles compose the images are already known, however the project proceeds as if we didn't know this information for Y_{test} and only makes use of this information when evaluating the accuracy of the two methods.

The sets of data Y_{train} and Y_{test} are both 644×200 matrices with each image corresponding to a column of the matrix. Each image has resolution (size) of 28×23 and the first five columns (images) of the sets of data correspond to the same person, the second set of five columns (images) to the second person, and so on for a total of forty different individuals. Thus, there are a total of $n_1 = 5$ different images for each of $n_2 = 40$ different individuals with each image being of size $s_1 \times s_2 = 28 \times 23 = 644$.

[2] Jolliffe, Ian. Principal component analysis. Springer Berlin Heidelberg, 2011.

(c) Feature Extraction

The images of Y_{train} are training images and are projected onto their principal K-dimensional subspace. The test images in Y_{test} are also projected onto their K-dimensional subspace and are then compared with the training images. For comparison, a simple projection of both sets of data is used and compared.

i.) PCA Algorithm

The PCA algorithm employed in this analysis is as follows:[3]

1. Find the sample covariance matrix $C \in \mathbb{R}^{(s_1 \times s_2) \times (n_1 \times n_2)}$ of the elements of Y_{train}.
2. Compute the singular value decomposition (SVD) of C to obtain the orthogonal matrix $U \in \mathbb{R}^{(s_1 \times s_2) \times (n_1 \times n_2)}$.
3. Let U_1 be the first K columns of U so that $U_1 \in \mathbb{R}^{(s_1 \times s_2) \times K}$.
4. Each image in Y_{train} can now be reduced to a K-dimensional vector using the projection $Y_1 = U_1^T Y_{train}$ so that $Y_1 \in \mathbb{R}^{(s_1 \times s_2) \times K}$.

ii.) Simple Projection

The simple projection algorithm is similar to the PCA algorithm except that U_1 is just the first K columns of the $(s_1 \times s_2)$ identity matrix.

(d) Classification

Taking one image (column) I from Y_{test}[4]

1. Compute the projection of I using $I_1 = U_1^T I$ so that $I_1 \in \mathbb{R}^{K \times 1}$.
2. Compute the distance between I_1 and each column of $Y_1 = U_1^T Y_{train}$ using the 2-norm.
3. Find the nearest neighbor by finding the label of the column that has the smallest distance to I_1.
4. Gauge accuracy by determining whether the label in the previous step is correct and finding the fraction of correct labels to total labels.

The above algorithms were for $K = 1, K = 2, ..., K = 40$.

[3] Florida Statue University. Course: STA 5106. Lecture 05. Fall 2019
[4] Florida State University. Course: STA 5106. Midterm Prompt. Fall 2019

3. MATLAB Programs

The code was compiled in Matlab version R2018b. The code, as it was compiled is given below:

(a) Code:

```
%Midterm OM, Begin:
clc
clear
%Diary
dfile ='MATLAB_Output_OM.txt';
if exist(dfile, 'file') ; delete(dfile); end
diary(dfile)
diary on
%Introduction
fprintf('
   ————————————————————————————————————————————————————————————————————————\n'
   );
fprintf('\t Oscar Martinez \t Midterm \t STA 5106\n');
fprintf('
   ————————————————————————————————————————————————————————————————————————\n'
   );
%Load Data
load mid_train.mat %Loads the 644 x 200 double Ytrain
load mid_test.mat %Load the test data, Y_test, 644 x 200
%The images are arranged in a vector of size (s1 x s2) x (n1 x n2)
%s1 = 28, s2 = 23, n1 = 40, n2 = 5
%First 5 columns are of person 1, send 5 (cols 6—10) are of person 2 etc.
    X = Ytrain'; %200 x 644
    [m,n] = size(X); %m=200, n=644
 %Visualization of the images
 for t = 1:m
    figure(1);
    I = reshape(X(t,:),28,23); %Reshape column i of X into a 28x23 matrix
    imagesc(I); %Takes the reshapen column/matrix and converts it to an
       image
    colormap(gray); %The image's color scheme is grayscale
    axis equal; %Equal Margins
    title(['Data Image ', num2str(t)]); %Title
    %pause;
 end
%PCA
C = cov(X); %C is the 644 x 644, Var—Cov Matrix of X
```

```
[U,S,V]=svd(C); %SVD decomposition w/ U = V being our matrix of interest
IDM=eye(644); %644 x 644 IDentity Matrix
%Preallocating For Speed
K=40; %Principal components
d_PCA=zeros(m,m); %PCA distance
d_Proj=zeros(m,m); %Projection distance
nn_PCA=zeros(1,m); %PCA nearest neighbor
nn_Proj=zeros(1,m); %Projection Nearest Neighbor
FPCA=zeros(1,m); %PCA Accuracy Logical
FProj=zeros(1,m); %Projection accuracy logical
LPCA=zeros(m,K); %Aggregate PCA logical
LProj=zeros(m,K); %Aggregate Proj logical
AccPCA=zeros(1,K); %PCA accuracy
AccProj=zeros(1,K);  %Projection accuracy
Diff=zeros(1,K); %Difference in accuracy

for k=1:K %X is 200 x 644, Ytrain/test is 644 x 200
    U1=U(:,1:k); %PCA Proj Matrix, (s1 x s2) x k; 644 x k
    U2=IDM(:,1:k); %Simple Projection Matrix, (s1 x s2) x k; 644 x k
    Y1=U1'*X'; %PCA Projection, k x 200
    Y2=U2'*X'; %Simple Projection, k x 200
%---Classification---
%Form feature Vector
for i=1:m %m=200 images
    I_Test = Ytest(:,i); %Select an image (644 x 1 col vect) from test data
    I_PCA = U1'*I_Test; %PCA Feature, k x 1
    I_Proj = U2'*I_Test; %Projection Feature, k x 1
%Nearest Neighbor
    for j = 1:m %Compute distances for each image i in test and all m images
        of train
      d_PCA(i,j) = norm(I_PCA-Y1(:,j)); %Distance of i from PCA projection j
         , 200 x 200
      d_Proj(i,j) = norm(I_Proj-Y2(:,j)); %Distance of i from simple
         projection j, 200 x 200
    end
%Distances and Neighbors
    [dis_PCA, dis_indPCA] = sort(d_PCA(i,:)); %Sorting PCA distances in
       ascending order with index
    [dis_Proj, dis_indProj] = sort(d_Proj(i,:)); %Sorting SP distances in
       ascending order with index
    nn_PCA(i) = dis_indPCA(1); %Column of Y1 with closest distance to I_PCA
    nn_Proj(i) = dis_indProj(1); %Column of Y1 with closest distance to
       I_PCA
%Accuracy
    FPCA(i) = ceil((nn_PCA(i)/5))==ceil(i/5); %Logical Accuracy PCA
```

```
    FProj(i) = ceil((nn_Proj(i)/5))==ceil(i/5); %Logical Accuracy SP
end
    LPCA(:,k) = FPCA; %Aggregate Logical Correct PCA Matrix
    LProj(:,k) = FProj; %Aggregate Logical Correct Proj Matrix
    AccPCA(k)=sum(FPCA)/m; %ACC = #Correct/Total
    AccProj(k)=sum(FProj)/m; %ACC = #Correct/Total
    Diff(k)=abs(AccPCA(k)-AccProj(k)); %difference in accuracy
%Print Results
    fprintf('k = %2.0f \t PCA Acc = %2.2f%% \t SP Acc = %2.2f%% \t Diff =
       %2.2f%% \n', k, AccPCA(k)*100, AccProj(k)*100, Diff(k)*100);
end
%Accuracy Plot
    figure(2);
    t=1:K;
    plot(t,AccPCA*100, 'bo-', t,AccProj*100,'ro-', 'LineWidth',1);
    grid on;
    title('Accuracy (%)');
    legend('PCA Projection', 'Simple Projection','Location', 'southeast');
    axis([1 40 0 100]);
    xlabel('k');
    ylabel('F(k)');
%Visual Examples
    PCA_Right = find(LPCA(:,k)); %PCA Right ID
    PCA_Wrong = find(1-LPCA(:,k)); %PCA Wrong ID
    SP_Right = find(LProj(:,k)); %SP Right ID
    SP_Sup = find((1-LPCA(:,k)).*(LProj(:,k))); %SP Right ID & PCA Wrong ID
    EX = {PCA_Right PCA_Wrong SP_Sup}; %Examples Matrix Reference
    for j=1:3
        [a,b]=size(eval('EX{j}'));
        for i=1:a
            figure(j+2); %Figures 3-5
        %Test Image
            subplot(1,3,1);
            I = reshape(Ytest(:,eval('EX{j}(i)')),28,23); %Col/Img i in
               Ytest
            imagesc(I); %Takes the reshapen column/matrix and converts it to
                an image
            colormap(gray); %The image's color scheme is grayscale
            axis equal; %Equal Margins
            title(['Test: ', num2str(eval('EX{j}(i)'))],'fontsize', 18); %
               Title
        %PCA Image
            subplot(1,3,2);
            I = reshape(Ytrain(:,nn_PCA(eval('EX{j}(i)'))),28,23); %Col/Img
               in Ytrain w/ nn corresponding to Col/Img i in Ytest
```

```
            imagesc(I);
            colormap(gray);
            axis equal;
            title(['PCA NN:' num2str(nn_PCA(eval('EX{j}(i)')))], 'fontsize',
                18);
        %SP Image
            subplot(1,3,3);
            I = reshape(Ytrain(:,nn_Proj(eval('EX{j}(i)'))),28,23); %Col/Img
                in Ytrain w/ nn_SP corresponding to Col/Img i in Ytest
            imagesc(I);
            colormap(gray);
            axis equal;
            title(['SP NN:' num2str(nn_Proj(eval('EX{j}(i)')))], 'fontsize',
                18);
            %pause
        end
    end
diary off
```

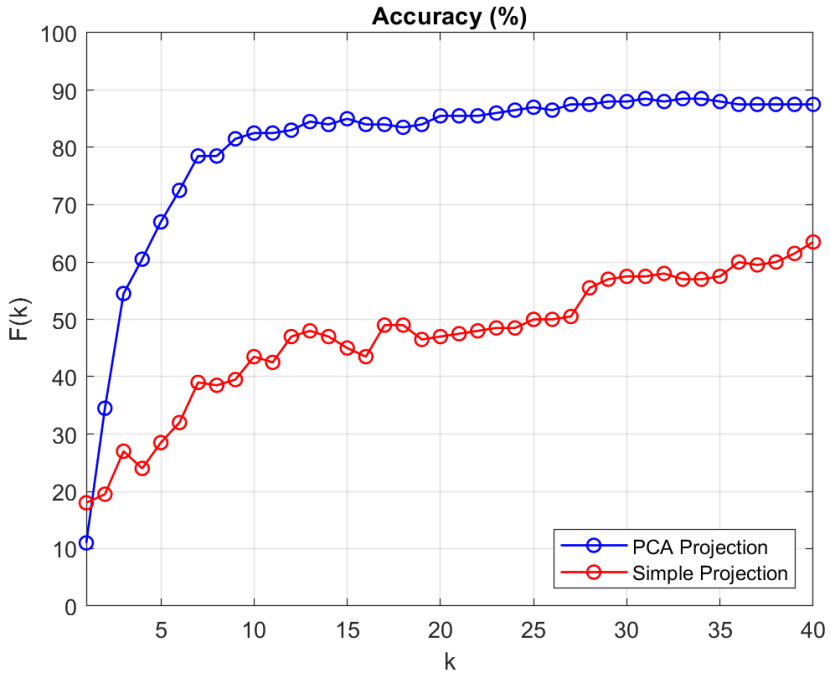
4. Results

The accuracy is given by the MATLAB output of the table below and graphed by the image above it. While for $K = 1$, the simple projection procedure was more accurate, for any $K \geq 2$, the PCA procedure was more accurate. While the simple projection seemed to increase linearly, the PCA procedure increased sharply until about $K = 9$ and then increased only gradually after that until apparently converging at an accuracy rate of 87.5%. In contrast, the simple projection reached only 63.5% accuracy with $K = 40$.

The graph's horizontal axis denotes the dimension, K, and the vertical axis denotes the accuracy. On this axis, $1 = 100\%$ accuracy, $0.5 = 50\%$ accuracy and so on. the table and graph help illustrate the gain in accuracy that is brought on by PCA as compared to simple projections for lower-dimensional analysis.

(a) Figures:

i.) Accuracy

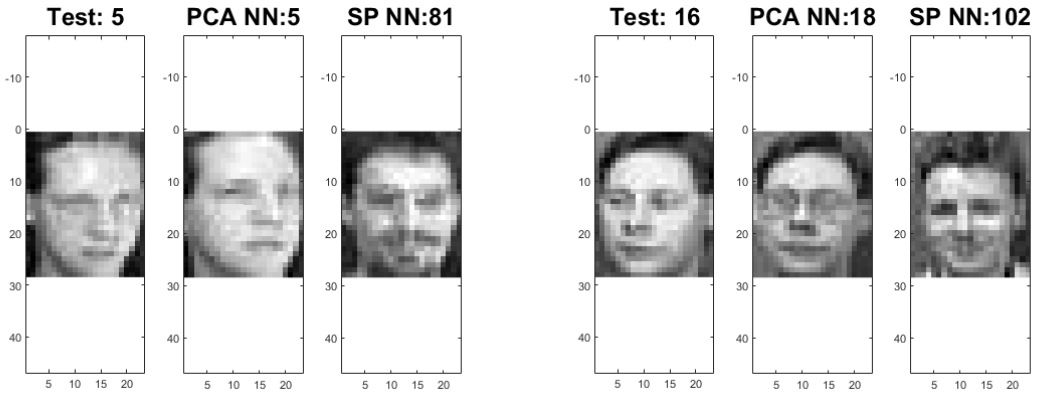


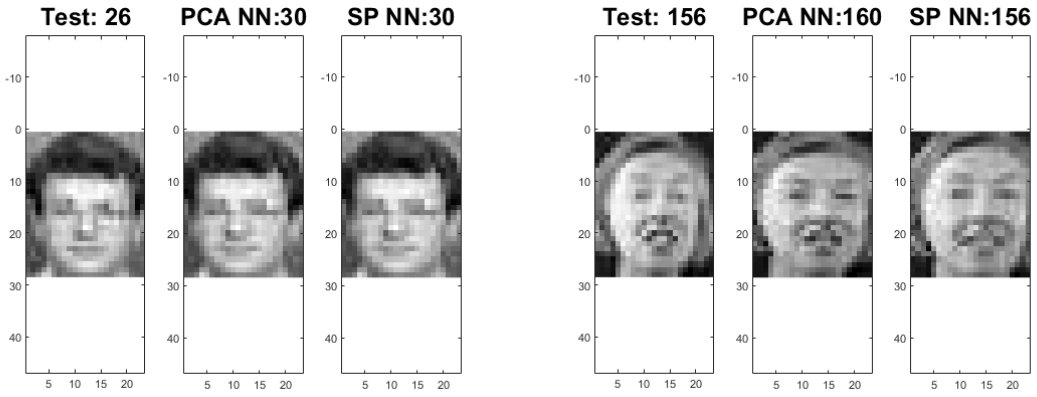

```
------------------------------------------------------------------------
         Oscar Martinez          Midterm          STA 5106
------------------------------------------------------------------------
k =  1          PCA Acc = 11.00%          SP Acc = 18.00%          Diff = 7.00%
k =  2          PCA Acc = 34.50%          SP Acc = 19.50%          Diff = 15.00%
k =  3          PCA Acc = 54.50%          SP Acc = 27.00%          Diff = 27.50%
k =  4          PCA Acc = 60.50%          SP Acc = 24.00%          Diff = 36.50%
k =  5          PCA Acc = 67.00%          SP Acc = 28.50%          Diff = 38.50%
k =  6          PCA Acc = 72.50%          SP Acc = 32.00%          Diff = 40.50%
k =  7          PCA Acc = 78.50%          SP Acc = 39.00%          Diff = 39.50%
k =  8          PCA Acc = 78.50%          SP Acc = 38.50%          Diff = 40.00%
k =  9          PCA Acc = 81.50%          SP Acc = 39.50%          Diff = 42.00%
k = 10          PCA Acc = 82.50%          SP Acc = 43.50%          Diff = 39.00%
k = 11          PCA Acc = 82.50%          SP Acc = 42.50%          Diff = 40.00%
k = 12          PCA Acc = 83.00%          SP Acc = 47.00%          Diff = 36.00%
k = 13          PCA Acc = 84.50%          SP Acc = 48.00%          Diff = 36.50%
k = 14          PCA Acc = 84.00%          SP Acc = 47.00%          Diff = 37.00%
k = 15          PCA Acc = 85.00%          SP Acc = 45.00%          Diff = 40.00%
k = 16          PCA Acc = 84.00%          SP Acc = 43.50%          Diff = 40.50%
k = 17          PCA Acc = 84.00%          SP Acc = 49.00%          Diff = 35.00%
k = 18          PCA Acc = 83.50%          SP Acc = 49.00%          Diff = 34.50%
k = 19          PCA Acc = 84.00%          SP Acc = 46.50%          Diff = 37.50%
k = 20          PCA Acc = 85.50%          SP Acc = 47.00%          Diff = 38.50%
k = 21          PCA Acc = 85.50%          SP Acc = 47.50%          Diff = 38.00%
k = 22          PCA Acc = 85.50%          SP Acc = 48.00%          Diff = 37.50%
k = 23          PCA Acc = 86.00%          SP Acc = 48.50%          Diff = 37.50%
k = 24          PCA Acc = 86.50%          SP Acc = 48.50%          Diff = 38.00%
```

```
k = 25          PCA Acc = 87.00%          SP Acc = 50.00%          Diff = 37.00%
k = 26          PCA Acc = 86.50%          SP Acc = 50.00%          Diff = 36.50%
k = 27          PCA Acc = 87.50%          SP Acc = 50.50%          Diff = 37.00%
k = 28          PCA Acc = 87.50%          SP Acc = 55.50%          Diff = 32.00%
k = 29          PCA Acc = 88.00%          SP Acc = 57.00%          Diff = 31.00%
k = 30          PCA Acc = 88.00%          SP Acc = 57.50%          Diff = 30.50%
k = 31          PCA Acc = 88.50%          SP Acc = 57.50%          Diff = 31.00%
k = 32          PCA Acc = 88.00%          SP Acc = 58.00%          Diff = 30.00%
k = 33          PCA Acc = 88.50%          SP Acc = 57.00%          Diff = 31.50%
k = 34          PCA Acc = 88.50%          SP Acc = 57.00%          Diff = 31.50%
k = 35          PCA Acc = 88.00%          SP Acc = 57.50%          Diff = 30.50%
k = 36          PCA Acc = 87.50%          SP Acc = 60.00%          Diff = 27.50%
k = 37          PCA Acc = 87.50%          SP Acc = 59.50%          Diff = 28.00%
k = 38          PCA Acc = 87.50%          SP Acc = 60.00%          Diff = 27.50%
k = 39          PCA Acc = 87.50%          SP Acc = 61.50%          Diff = 26.00%
k = 40          PCA Acc = 87.50%          SP Acc = 63.50%          Diff = 24.00%
```

ii.) Correct PCA Examples

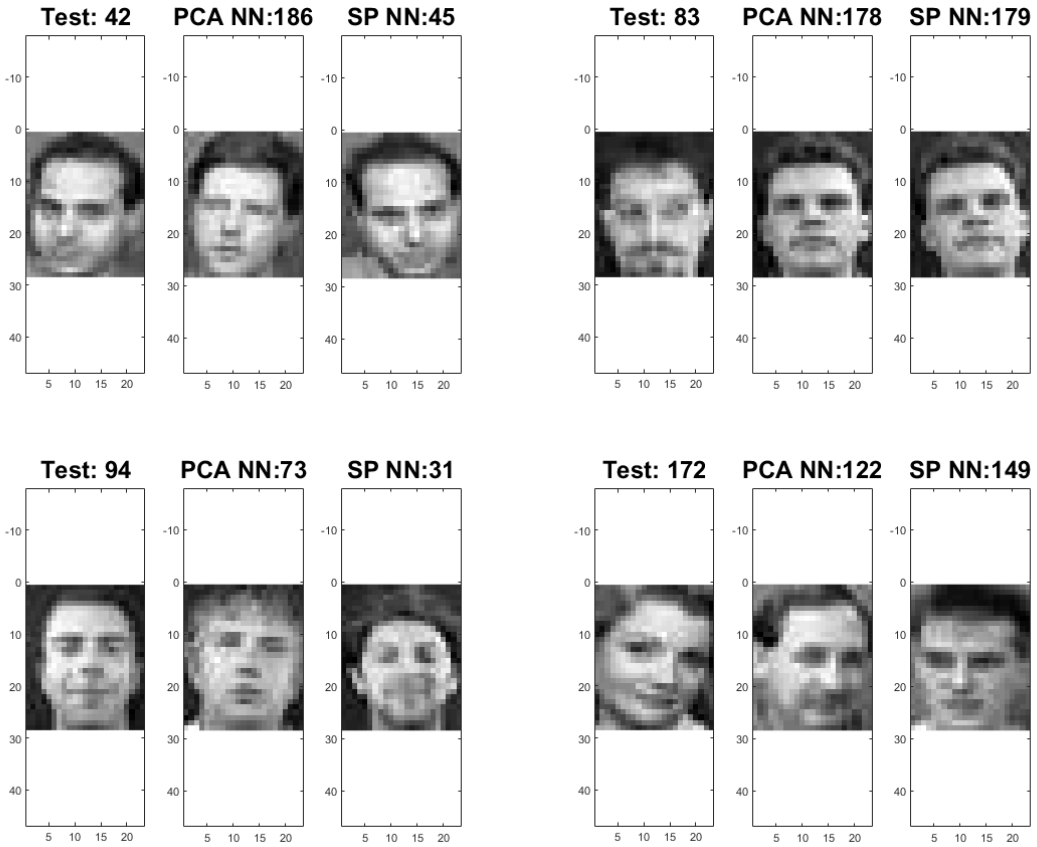
To illustrate the power of PCA, below are some examples of the images for which PCA was correct. For these images, PCA was correct and simple projection may have been correct or may have been wrong. The technique was correct if the image identified by the techniques belonged to the individual whose facial profile was presented in the test image. For example, if $Test = 7$ a correct identification would have been $6, 7, 8, 9,$ *or* 10. For each figure, which is composed of three images, the full image in Y_{test} is shown furthest to the left, the middle image is the image identified by PCA to be the closest, or most similar to the test image, and the right-most image is the image chosen by the simple projection technique to be the most similar. The numbers next to PCA or SP denote which image was identified as being the "closest".





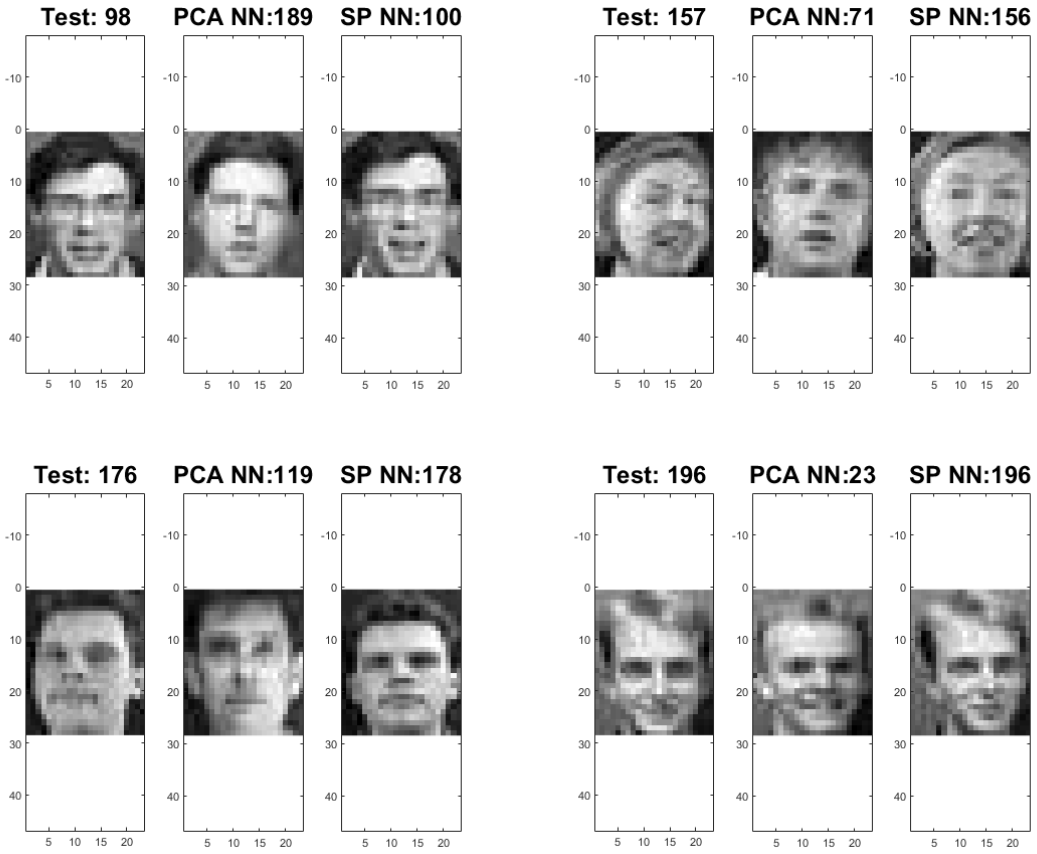
iii.) Incorrect PCA Examples

These are examples where PCA was wrong and simple projection may be right or may be wrong. Facial hair as well as face orientation seem to be difficult for PCA to match. Despite this difficulty, there is a definite similarity between the image chosen by PCA and the test image.



iv.) Examples Where Simple Projection was Correct but PCA Was Not

Lastly, here are examples where PCA was wrong and simple projection was correct.



5. Conclusion

Both projections are surprisingly accurate given that over 60% accuracy was attained by $40/644 \approx 0.0621 = 6.21\%$ of the dimension of the data. For any K greater than 1, PCA outperformed the simple projection. For lower values of K, PCA seemed to have greater variation regarding accuracy compared to the simple projection. As K increased, the variation in accuracy for PCA appeared to be lower than the corresponding variation in accuracy for the simple projection. The difference in accuracy between PCA and the simple projection reached its maximum at $K = 9$, with a difference of 42% accuracy. Afterwards, the difference steadily decreased but did not equal zero for any of the values of K used in this analysis. Conclusively, PCA offers notable gains in efficiency without sacrificing too much accuracy.