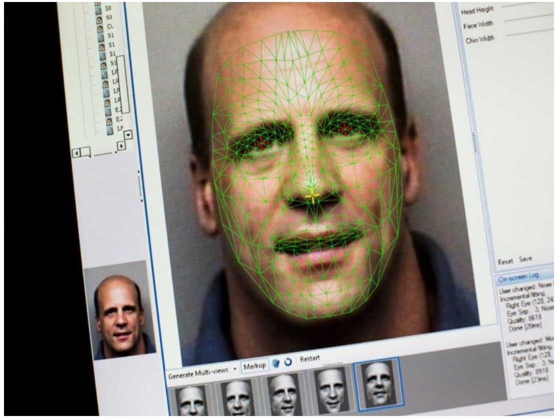
EGR 190 Final Project

FACIAL RECOGNITION

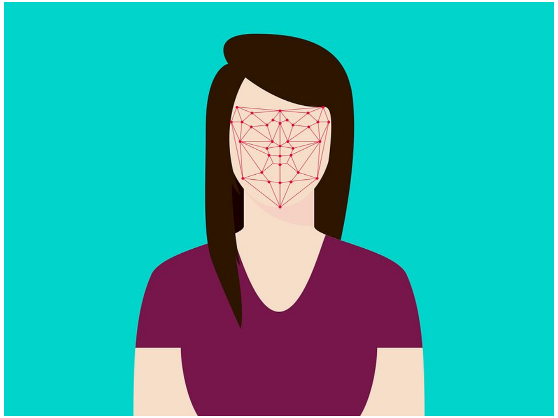
Applications

- Customized advertisements
- Unlocking personal devices
- Social media tagging and recommendations
- Purchasing goods
- Pollutant regulation



Legal Implications

- Maintaining anonymity
- Recognition error amongst race
- Permission to add faces to database



How It's Done

- Traditionally:
  - Algorithms
  - Biometrics
- Our approach:
  - Principal Components Analysis

The Task at Hand

Given:

- 400, 92x112 images
- 10 images of each of the 40 faces

What We Want:

- Which are not faces?
- Which is face 1? Face 2? Face 3?....

PCA:

- How many components are necessary to describe the data?

Classification:

- Feeding simplified data from PCA into a classifier of our choosing
- Explore which classification technique is best

Analysis:

- Did we effectively identify the 40 different faces?
- Were we able to identify if it was not a face?

PCA Method 1: Eigendecomposition of Covariance

1. Rasterize

Each 92x112 image becomes a single column with 10,304 entries

2. Format

Combine each rasterized image to create a 10,304 by 400 matrix

3. Find Covariance

Calculate the covariance of the data: produces a 10,304 by 10,304 matrix

4. Eigendecomposition

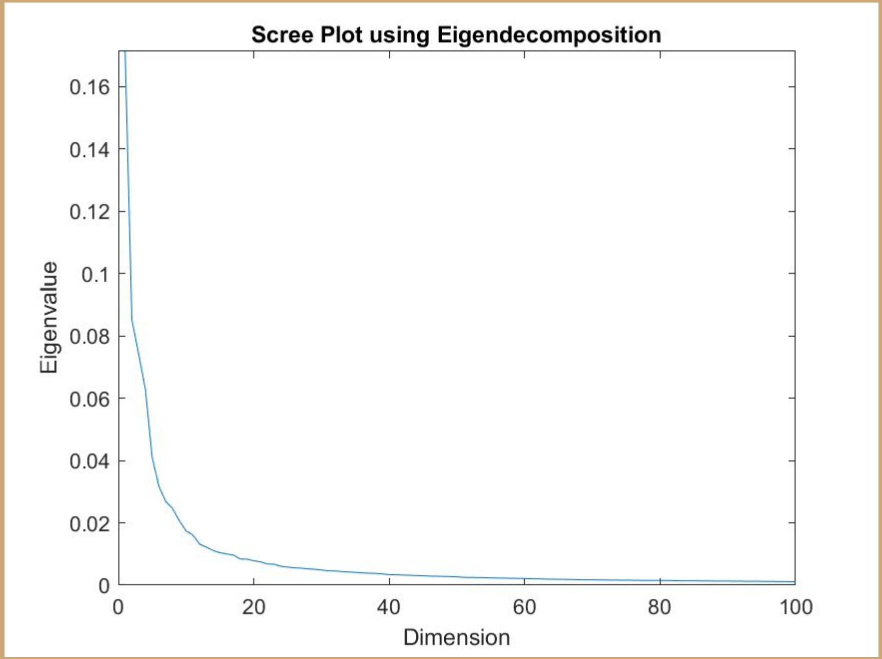
Calculate the 10,304 eigenvectors and their associated eigenvalues

5. Find Significant Eigenvectors

Plot eigenvalue magnitude against the eigenvector number to find "knee"

PCA Results (1/3)

When using Covariance Matrix Eigendecomposition



“Knee” at 20 eigenvectors

However, covariance eigendecomposition proved computationally inefficient…

… so we tested out two other techniques!

PCA Method 2: Single Variable Decomposition (SVD)

1. Rasterize

Each 92x112 image becomes a single column with 10,304 entries

(just like before)

2. Format

Combine each rasterized image to create a 10,304 by 400 matrix (call this matrix X)

(just like before)

3. SVD(X) = U*L*V transpose

SVD of original data X provides the right singular vectors V

V forms basis for PCA

4. Find Significant Eigenvectors

Plot eigenvalue weight against eigenvector number to find "knee"

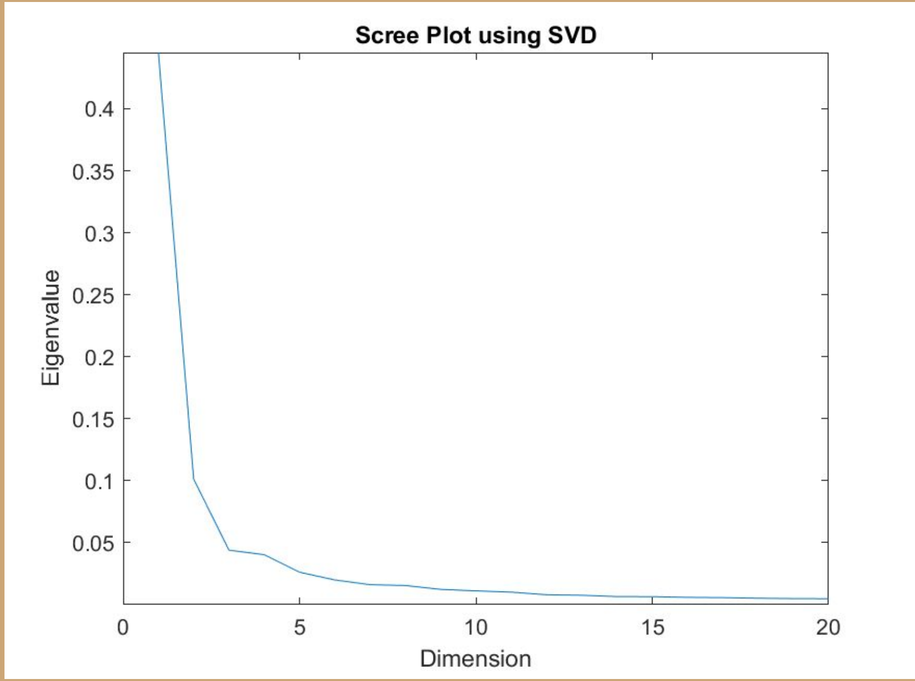
(just like before)

5. Compare to Other Methods

Computationally more efficient, but different "knee"

PCA Results (2/3)

When using SVD on X



"Knee" at 5 eigenvectors

PCA Method 3: Gram Matrix Eigendecomposition

1. Rasterize

Each 92x112 image becomes a single column with 10,304 entries

(just like before)

2. Format

Combine each rasterized image to create a 10,304 by 400 matrix (call this matrix X)

(just like before)

3. Calculate

Eigendecomposition of the Gram Matrix ($X*X^T$) returns:

- Eigenvectors U
- Eigenvalues λ

The product U*λ returns projections onto principal components

4. Find Significant Eigenvectors

Plot eigenvalue weight against principal component number to find "knee"

(just like before)

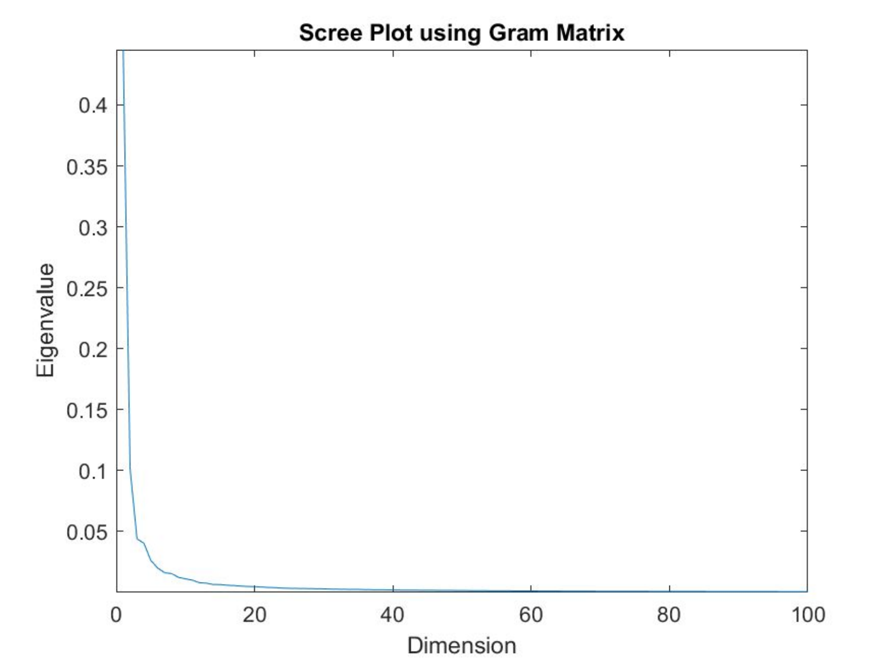
5. Compare to Other Methods

Computationally more efficient, but very different results

(similar to SVD results)

PCA Results (3/3)

When using Gram Matrix Eigendecomposition



"Knee" at 10 eigenvectors

What We Found From Performing PCA

20 Principal Components

All PCA knee plots showed that 20 principal components are *more than enough* to represent the data

But we want even less

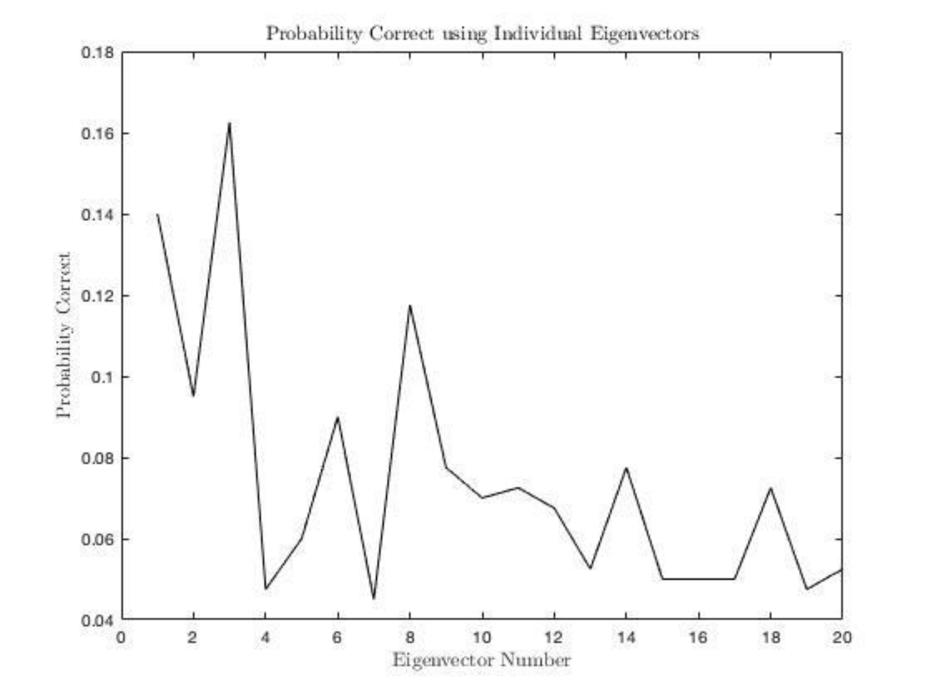
- 20 principal components accurately represent the faces
- We can find a subset of these 20 components that:
  - performs almost as well
  - decreases runtime and data required

The game plan:

- Run KNN with the data projected onto first 20 individual principal components
  - Hold K = 5
  - 10-fold CV
- Find which individual principal components provided highest classification accuracy

Individual Principal Component Accuracy

Determined by KNN with K = 5, 10-fold cross-validation



- First 10 eigenvectors are better predictors than last 10
- Components 4 and 7 do not produce high accuracy
- Want a more concrete answer:
  - Next step: group principal components to find highest accuracy

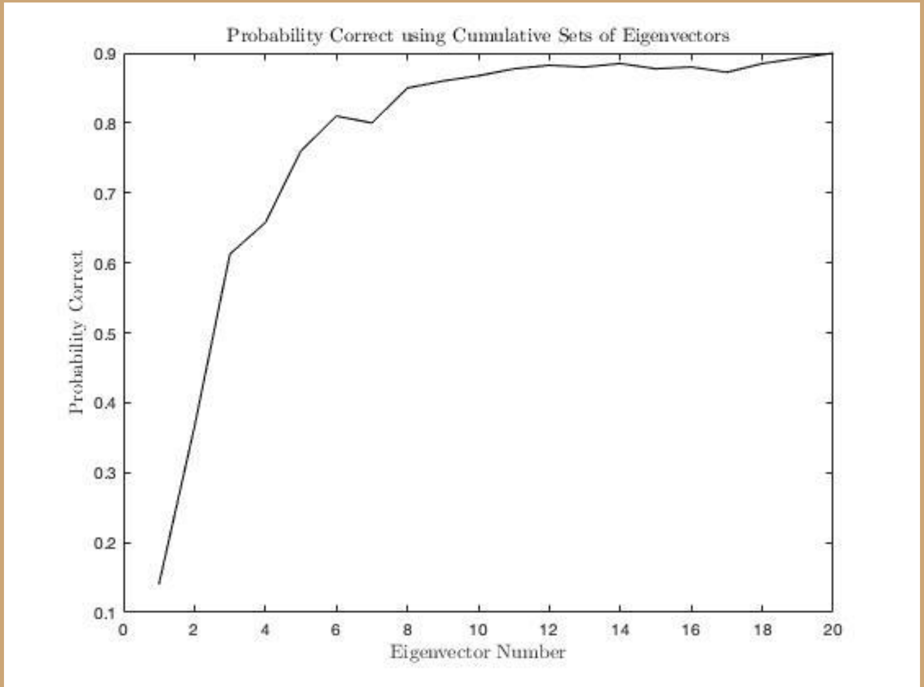
Next, we found the classification accuracy of subsets of principal components

I.e. (component 1), (component 1, 2), (component 1, 2, 3),...

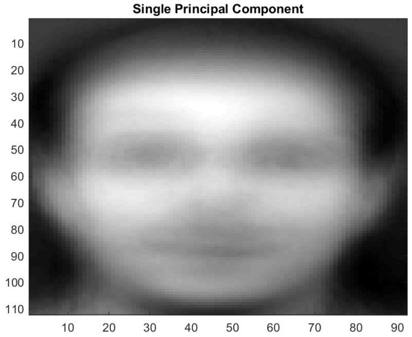
Reminder: At this step in the process, we have performed PCA but want to minimize number of principal components (eigenvectors) used while maintaining a high level of accuracy in facial recognition (as measured by classification methods).

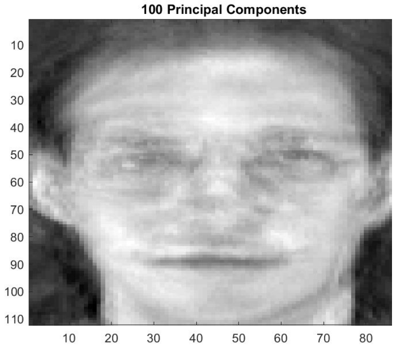
Results

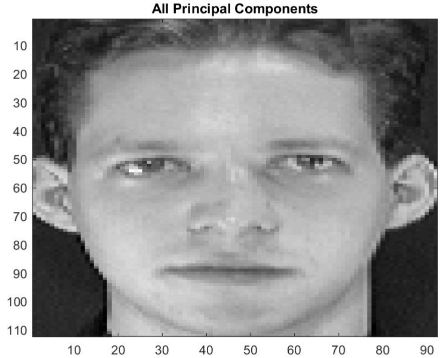
Accuracy of different groups of principal components



Note: the first 8 principal components adequately classify the faces (located at the "knee" of the plot)

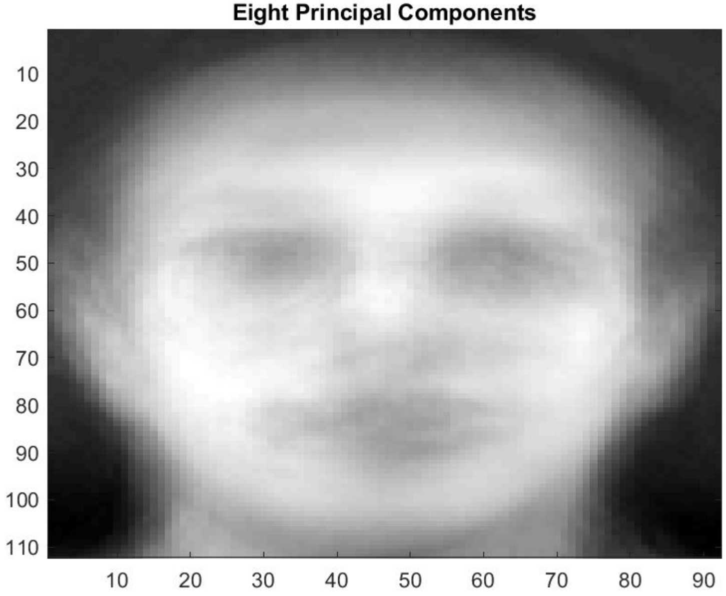






Examples of a Projected Face

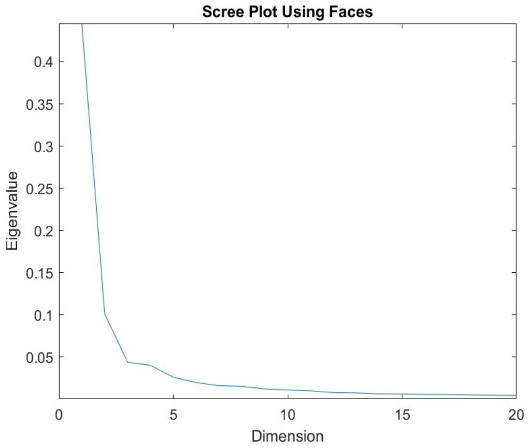
Using 1 (top picture), 100 (bottom left), and all (bottom right) principal components

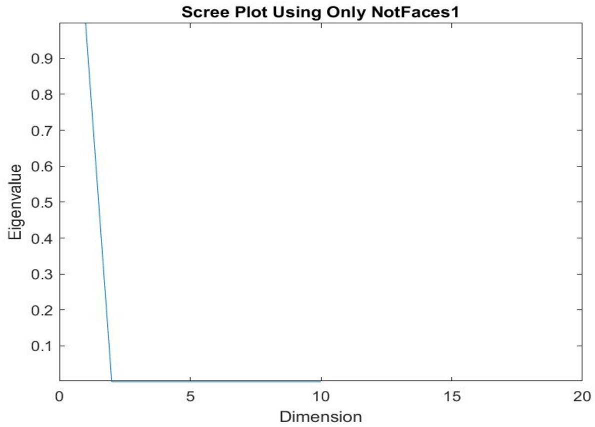


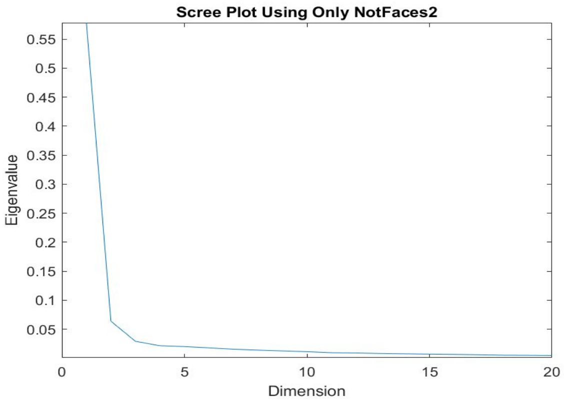
An Adequate Representation

8 principal components allows us to accurately distinguish between faces while minimizing data!

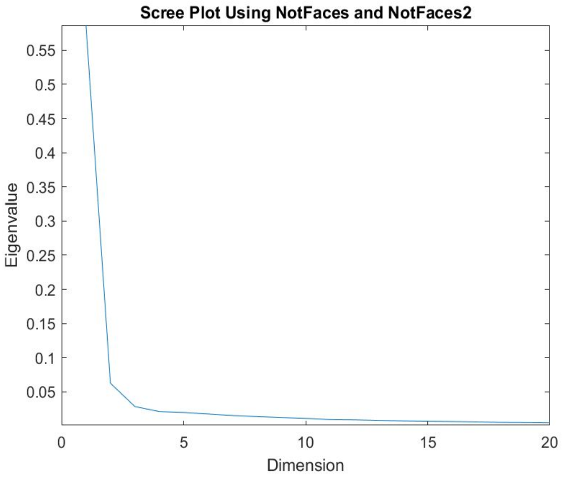
PCA: Scree Plot of Individual Data Sets

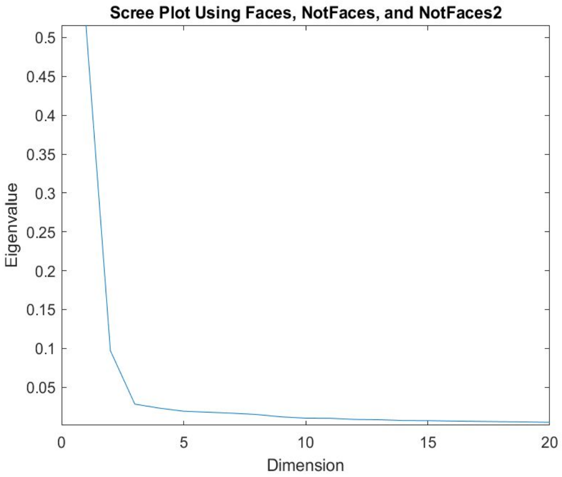


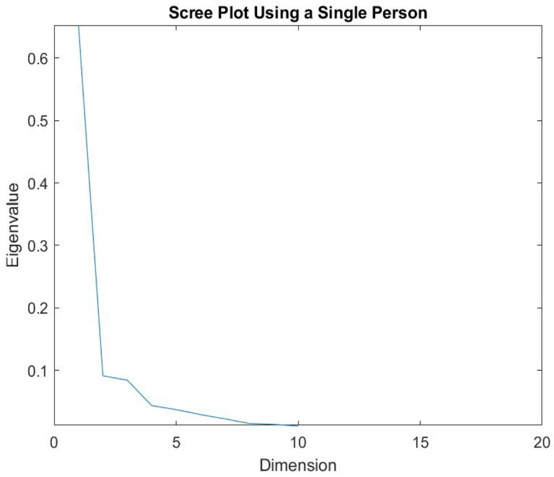




PCA: Scree Plots of Select Combinations of Data Sets







Now that we've found the efficient number of principal components that accurately portray each face..

... we can find the best way to classify the faces using KNN, LR, and LDA!

Roadmap of Performing KNN Classification

1. Create a Truth Vector

- 400 row data set (1 row per face)
  - First 10 rows = 1 for face 1
  - Second 10 rows = 2 for face 2
  - etc.

2. Visualize reduced dataset

- Used t-distributed stochastic neighbor embedding (TSNE)
- Projects points into 2-D space to visualize clustering of images

3. Find most important components

- Test different combinations of the 8 components to maximize correct classifications
- Held K = 5

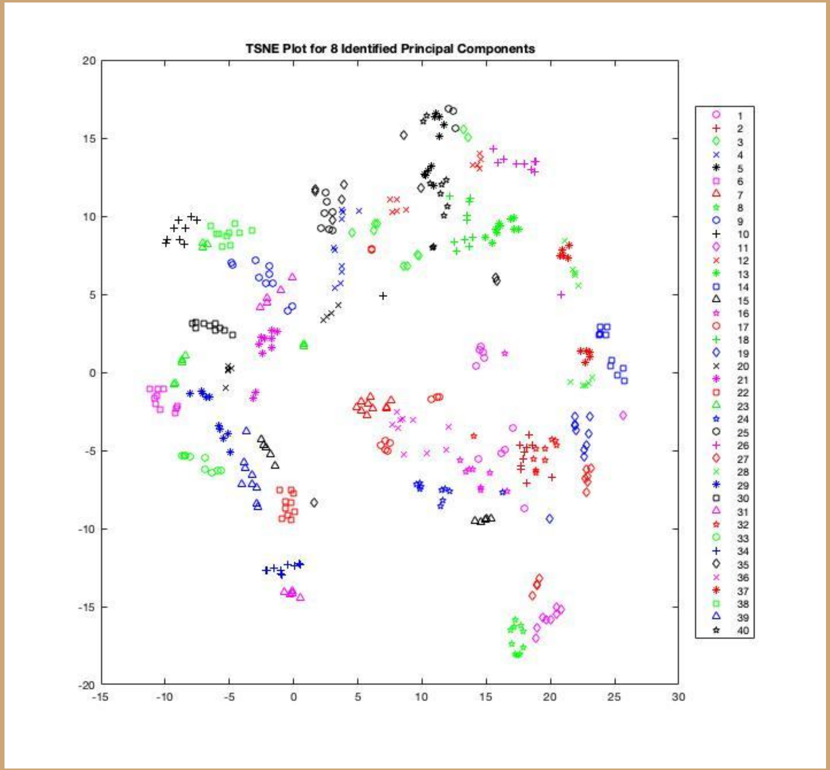
4. Find optimal value of K

- Find probability of correct classification for several values of K
- Select K giving the best results

5. Analyze results

Visualizing the Reduced Data Set

Used t-distributed stochastic neighbor embedding (TSNE)



TSNE plot

Finding the most important principal components

8 principal components were found to best classify the faces, but is that the case for KNN?

*recall K was held at K=5 for this experiment

# in Group	Components	% Correct
2	3, 8	.45
3	1, 3, 8	.635
4	1, 2, 3, 6 OR 1, 3, 5, 8	.7225
5	1, 2, 3, 6, 8	.85
6	1, 2, 3, 5, 6, 8	.8525
7	1, 2, 3, 4, 5, 6, 8	.8775
8	1, 2, 3, 4, 5, 6, 7, 8	.85

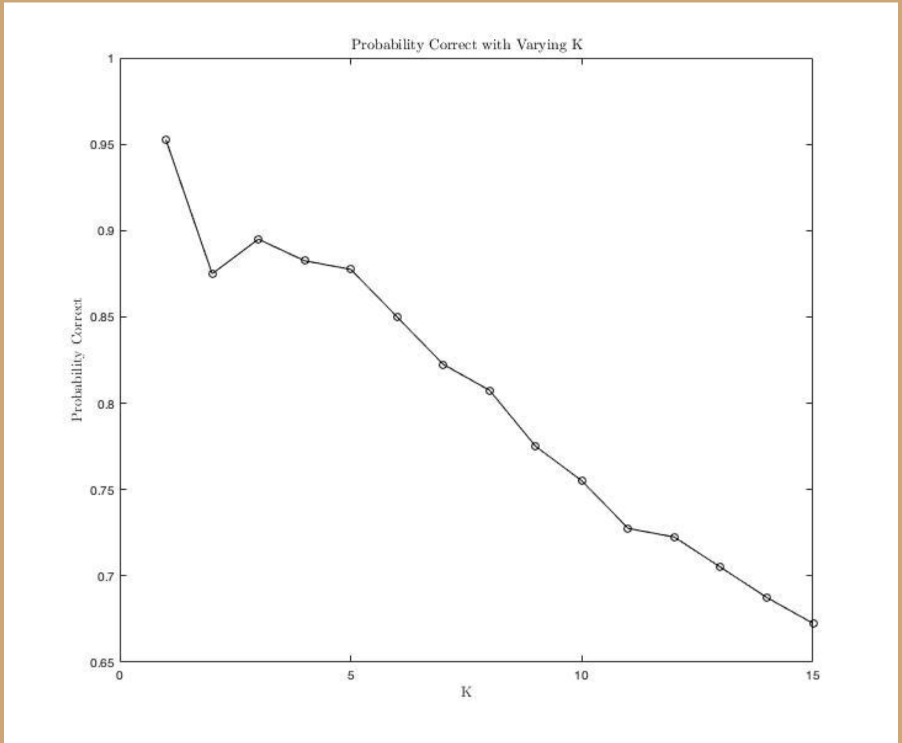
The 7th principal component was actually found to *hinder* accuracy in KNN

Finding the best number of K

Using the 7 principal components that produced the highest accuracy

Chose to evaluate K=1 through K=15

K=1 provides trivial accuracy



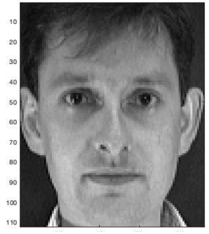
K=3 provided best accuracy

Results of KNN Classification

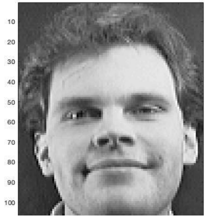
Subjects with low probability of correct classification include 1, 23, 35, 40:

Subject	% Correct	Subject	% Correct	Subject	% Correct	Subject	% Correct
1	0.6	**11**	1	**21**	1	**31**	0.9
2	1	**12**	1	**22**	1	**32**	0.8
3	0.9	**13**	1	**23**	0.6	**33**	1
4	1	**14**	1	**24**	0.9	**34**	1
5	1	**15**	1	**25**	0.7	**35**	0.2
6	1	**16**	0.8	**26**	0.9	**36**	0.9
7	1	**17**	0.8	**27**	1	**37**	1
8	1	**18**	0.8	**28**	0.9	**38**	1
9	0.9	**19**	0.9	**29**	1	**39**	1
10	0.8	**20**	1	**30**	1	**40**	0.5

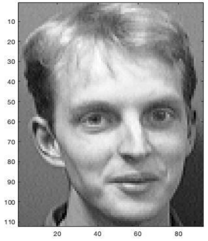
Subject 1



Subject 23



Subject 35



Subject 40

Although KNN produced accurate results, we wanted to see if binary classification techniques yielded accurate results as well

A different kind of classification:

Not by person, but by characteristic (ex: glasses, facial hair,...)

Roadmap of Performing Binary Classification

1. Create Truth Vectors

- Gender
- Facial hair
- Glasses

2. Sample with replacement

- Balance classes to address bias
- 10-run 2-fold cross-validation with identified principal components
- Performed for each of the 3 characteristics

3. Find most important components

- Test different combinations of 8 principal components to maximize AUC

4. Plot the ROC curve

- Calculate the AUC

5. Analyze results

Results of Logistic Regression Classification

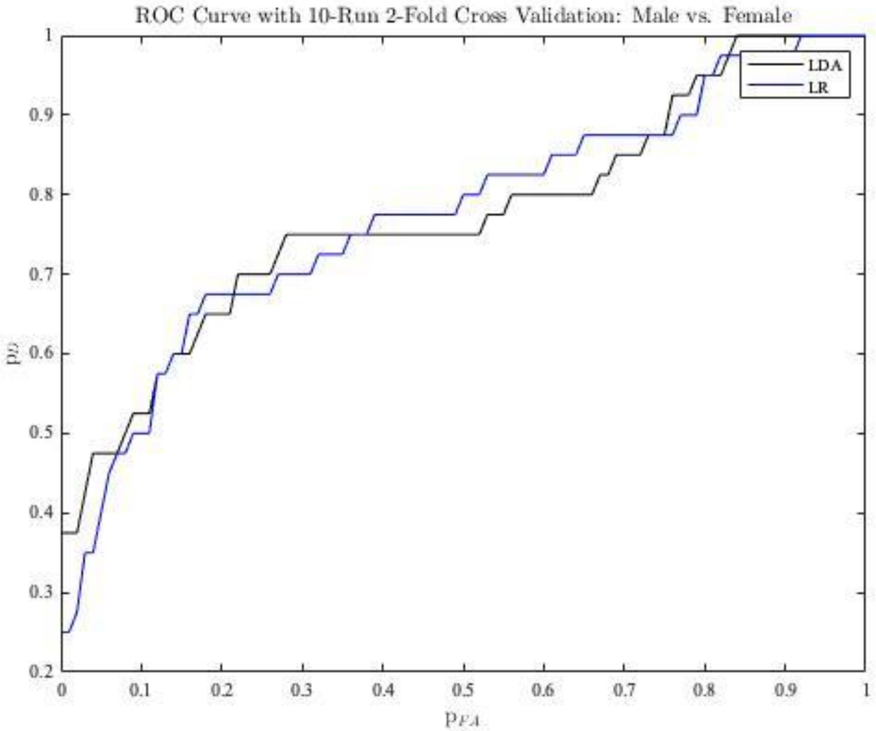
For Female vs. Male:			For Facial hair vs. No Facial hair:			For Glasses vs. No glasses:		
# in Group	Components	% Correct	# in Group	Components	% Correct	# in Group	Components	% Correct
1	4	0.7421	1	8	0.7355	1	1	0.6807
2	4,5	0.7946	2	1,8	0.8001	2	1,5	0.7154
3	4,5,6	0.8024	3	1,2,8	0.8483	3	1,2,5	0.7258
4	2,4,5,6	0.8055	4	1,2,5,8	0.8628	4	1,2,4,5	0.7319
5	2,4,5,6,8	0.8057	5	1,2,5,6,8	0.8745	5	1,2,3,4,5	0.7308
6	2,3,4,5,6,8	0.8086	6	1,2,4,5,6,8	0.8769	6	1,2,3,4,5,7	0.7264
7	1,2,3,4,5,6,8	0.8068	7	1,2,4,5,6,7,8	0.8776	7	1,2,3,4,5,6,7	0.7184
8	1,2,3,4,5,6,7,8	0.7981	8	1,2,3,4,5,6,7,8	0.8730	8	1,2,3,4,5,6,7,8	0.7100

Results of Linear Discriminant Analysis Classification

For Female vs. Male:			For facial hair vs. No facial hair:			For Glasses vs. No glasses:		
# in Group	Components	% Correct	# in Group	Components	% Correct	# in Group	Components	% Correct
1	4	0.7440	1	8	0.7363	1	1	0.6810
2	4,5	0.7879	2	6,8	0.7593	2	1,5	0.6893
3	4,5,6	0.7884	3	2,6,8	0.7920	3	1,2,5	0.6960
4	4,5,6,8	0.7901	4	2,5,6,8	0.8054	4	1,2,4,5	0.6997
5	3,4,5,6,8	0.7856	5	2,5,5,6,8	0.8119	5	1,2,3,4,5	0.7010
6	3,4,5,6,7,8	0.7811	6	2,3,4,5,6,8	0.8136	6	1,2,3,4,5,8	0.6999
7	2,3,4,5,6,7,8	0.7755	7	2,3,4,5,6,7,8	0.8143	7	1,2,3,4,5,7,8	0.6987
8	1,2,3,4,5,6,7,8	0.7552	8	1,2,3,4,5,6,7,8	0.7992	8	1,2,3,4,5,6,7,8	0.6973

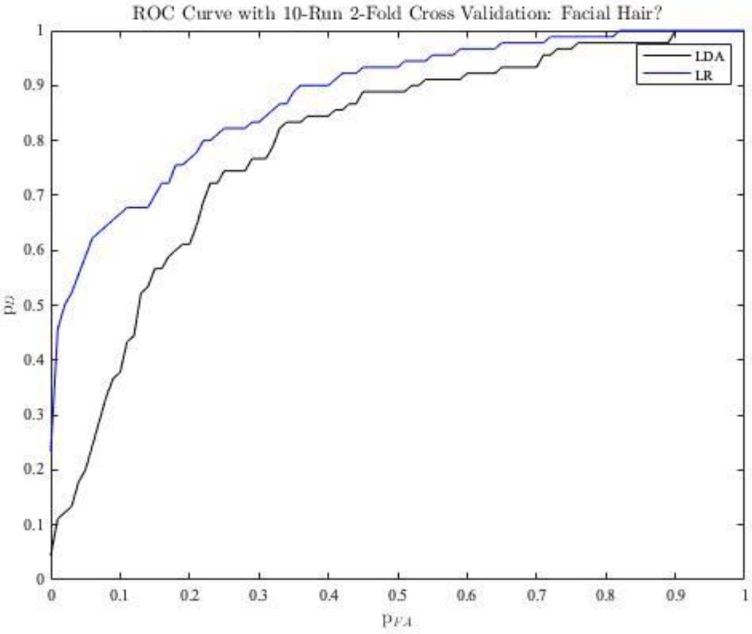
ROC Curves For Male vs Female

	LR AUC
Male vs. Female	0.7726
	LDA AUC
Male vs. Female	0.7401



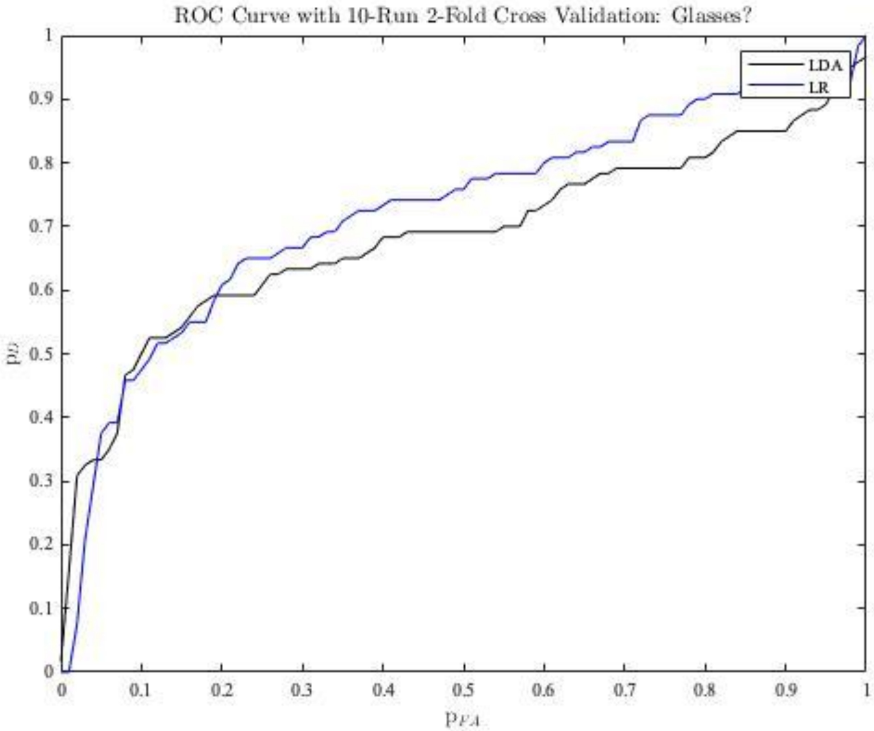
ROC Curves For facial hair vs No facial hair

	LR AUC
Facial hair?	0.7237
	LDA AUC
Facial hair?	0.6854



ROC Curves For Glasses vs No Glasses

	LR AUC
Glasses?	0.8846
	LDA AUC
Glasses?	0.8103



Logistic Regression displays higher accuracy

- LDA conditions are not met (no Gaussian distribution, mean/covariance more difficult to estimate)
- LR classification delivers highly promising results for all characteristics except “no glasses”

Classification by Subject	Gender	Facial Hair	Glasses
% Underrepresented (1) Classified Correctly	75	77.78	75
% Overrepresented (0) Classified Correctly	86.11	83.8	57.14

Results

Principal Components Analysis

a. Eigendecomposition of covariance
    i. Knee at 20
b. Single Variable Decomposition (SVD)
    i. Knee at 5
c. Gram Matrix Eigendecomposition
    i. Knee at 10

Classification

a. KNN
    i. 94.75% accuracy
    ii. On <2% of data (8 PC)
b. Logistic Regression (LR)
    i. Female: 75%
    ii. Male: 86.11%
    iii. Facial hair: 77.78%
    iv. No facial hair: 83.8%
    v. Glasses: 75%
    vi. No glasses: 57.14%
c. Linear Discriminant Analysis (LDA)
    i. Less accurate than LR due to assumptions

Bibliography

“Facial Recognition System.” Wikipedia, Wikimedia Foundation, 4 Dec. 2018, en.wikipedia.org/wiki/Facial_recognition_system.

Frankle, Clare Garvie Jonathan. “Facial-Recognition Software Might Have a Racial Bias Problem.” *The Atlantic*, Atlantic Media Company, 7 Apr. 2016, www.theatlantic.com/technology/archive/2016/04/the-underlying-bias-of-facial-recognition-systems/476991/.

Sobel, Ben. “Facial Recognition Technology Is Everywhere. It May Not Be Legal.” *The Washington Post*, WP Company, 11 June 2015, www.washingtonpost.com/news/the-switch/wp/2015/06/11/facial-recognition-technology-is-everywhere-it-may-not-be-legal/?utm_term=.431642b4cb40.

S.L. Tantum, “Eigenfaces Guidelines,” Duke University, November 2018.

Thurston, George and Spengler, John.“A Quantitative Assessment of Source Contributions to Inhalable Particulate Matter Pollution in Metropolitan Boston.” *NeuroImage*, Academic Press, 14 May 2003, www.sciencedirect.com/science/article/pii/0004698185901325.