# ISyE 6740 – Summer 2023
# Project Report

**Team Member Names:** Max Diamond (mdiamond32@gatech.edu)

**Project Title:** COVID Faces

## Problem Statement

Throughout the duration of the COVID-19 pandemic, people in the United States were instructed by the CDC to wear face masks to reduce the spread of the Coronavirus disease. While adherence to these guidelines became the topics of heated debates, I want to focus on one shortcoming brought about by wearing a mask. When wearing a face mask, Face ID on the Apple iPhone would often fail to work. This left iPhone owners in a bit of an uncomfortable position. Do they peel off their mask for a split second to unlock their iPhone, exposing their nose and mouth to the COVID-abundant world? Or do they inconveniently wait a few seconds to be prompted to manually enter their passcode?

Admittedly, this might seem like a First World problem, and an insignificant one at that. To put things into perspective, I'll try to quantify the inconvenience of having to enter a passcode. Americans pick up their phones an average of 96 times per day [6]. If Face ID is active and recognizes a face, an iPhone unlocks almost instantaneously. If Face ID is active, but unable to recognize a face, it take about 3 seconds to be prompted with the "Enter Passcode" screen. It takes an additional 1 second or so to enter the simplest version of a Passcode containing 4 digits (I measured these times myself). That totals to over 6 minutes of time spent each day simply trying to unlock an iPhone when Face ID isn't working.

People were able to quickly find a workaround to get Face ID to recognize them while wearing a face mask. One week after the CDC recommended Americans wear face masks, an article published by Business Insider suggested people cover part of their face with a mask or cloth and redo their Face ID setup [3]. While this method required a bit of trial and error, it largely served as a solution for using Face ID with a face mask. In March of 2022, almost 2 years after the onset of the COVID-19 pandemic, Apple rolled out iOS 15.4 with the "Face ID Mask Unlock" feature, allowing users to set up Face ID while wearing a mask [1].

I want to explore how and why the solution of covering part of your face was sufficient to get Face ID working while wearing a mask. I would also like to explore whether similar results hold for using Face ID while wearing tinted sunglasses. Lastly, I want to explore how blocking different areas of the face impacts the effectiveness of Face ID.

## Research Motivation

Facial recognition using eigenfaces was a novel approach introduced by Matthew Turk and Alex Pentland in 1991. The eigenface algorithm makes use of principal component analysis to generate a set of "eigenfaces". Each eigenface should capture some variation among the faces in the training set, with the top eigenface capturing the most variation. Each face in the training set can be represented as a linear combination of the eigenfaces. To perform facial recognition, a test face is projected onto the eigenface space, yielding a vector of weights which linearly combines the eigenfaces. Each "face class" (i.e. a distinct subject in the training data) has a unique weight vector that describes a linear combination of eigenfaces to represent a given subject. To classify a test face, its projection weight vector is compared with the weight vector for each face class. The pair with the lowest dissimilarity score is assigned as the prediction for the test face [8].

Several studies have been done since to replicate Turk and Pentland's results. Notably, Turk and Pentland were able to achieve a classification rate of 96% when using infinite eigenfaces. In other words, using more principal components for classification tended to yield better results [8]. A 2012 paper titled *Face Recognition Using Eigenface Approach* noted that, when using a training set of 190 images, only the first 20 or so eigenfaces were significant. The eigenvalues associated with the remaining 170 eigenfaces were very close to zero. The paper noted that using 5 eigenfaces yielded a 77.5% recognition rate with Euclidean distance as the dissimilarity metric. The recognition rate for 10 eigenfaces was 92.5%, and it was 97.5% for both 20 and 190 eigenfaces, which is a better rate than Turk and Pentland were able to achieve [7]. In another paper, titled *A Face Recognition System Based on Eigenfaces Method*, noted similar results. When using 10 eigenfaces to represent a total of 3040 images, a success rate of 94.74% was achieved [4]. In this paper, I will compare my classification accuracy with that achieved in past research. I will also seek to extend these studies by exploring how facial obstructions impact the eigenface algorithm's ability to accurately classify human faces.

## Data Source

The data source for this project is a subset of the Yale Faces dataset [2][5]. It contains 15 subjects, each with 11 pictures of their faces, totaling to 165 images. The data is gathered from Kaggle.

## Methodology

The eigenface algorithm uses the entire set of images, along with principal component analysis, to create eigenfaces. If there are a total of 100 images in the training set, then a maximum of 100 eigenfaces can be created to represent the "eigenface" space. This approach works well for certain facial recognition use cases. For example, think of a facial recognition device outside of a classified saferoom. The device should only grant access to a set of cleared individuals. In this way, the training set would be the faces of all cleared individuals and perhaps a random assortment of non-cleared individuals. When an individual scans their face, it will be projected onto the eigenface space, which is defined by the top n eigenfaces. There will be a weight vector associated with this projection which describes the linear combination of the eigenfaces used to represent the individual's face. If this weight vector is similar enough to that of one of the expected face classes (i.e. one of the cleared individuals), access will be granted. If, however, the weight vector is not similar enough, access will not be granted. Access should always be denied to non-cleared individual seeking to gain access to the saferoom.

On the contrary, Face ID on an iPhone should only ever grant access to one person. While I don't know the internals to Apple's Face ID technology, it seems likely that the set of training faces should *only* belong to the iPhone owner. This is different from the typical eigenface algorithm, since the training set is isolated to only a single person rather than an assortment of different people. As such, instead of using all 165 faces from the Yale Faces dataset to create a set of eigenfaces, I will only use a maximum of 11 faces for each subject to create their individualized eigenface space. This methodology assumes that Face ID will only ever be trained on the face of the iPhone owner, and never consider any other faces when creating the eigenfaces.

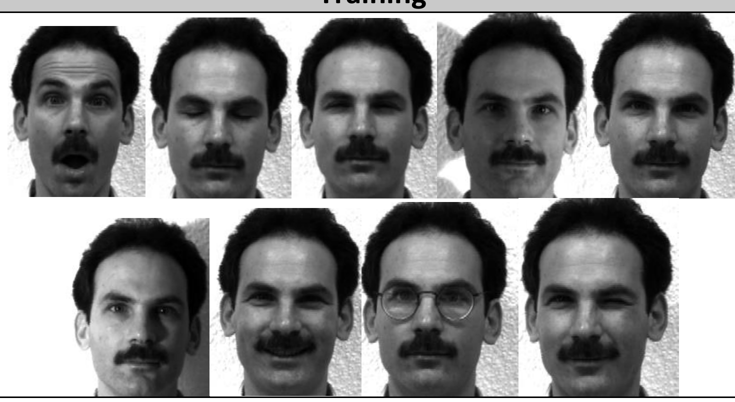
Before jumping into the eigenface algorithm application, I had to clean the data. I started by using the uncleaned Yale Faces data to create a set of top eigenfaces for each subject. Below is a sample of faces from the uncleaned data.

*Three sample faces from the Yale Faces dataset*

Note that about 50% of the width in each picture is occupied by the background. While the background is often white, there are several images for each subject that have shadows. I noticed that the top few eigenfaces often had a darker background than the rest. They also often had a distinct shadow behind the face. I didn't want the background color or shadows included in the eigenfaces for risk of classifying faces based on background color or shadow similarity. As such, I opted to crop each of the images so that only the faces were present with minimal background. The cropped versions of the images above are shown in the next figure.

*Three sample faces from the Yale Faces dataset after cropping*

Next, I split the data into a training set, a baseline set and a test set by subject. The training set contained 9 faces, and the baseline and test sets each contained 1 face. Below is a sample split for one of the subjects in the data.

| Training | Baseline | Test |
|---|---|---|
|  |  |  |

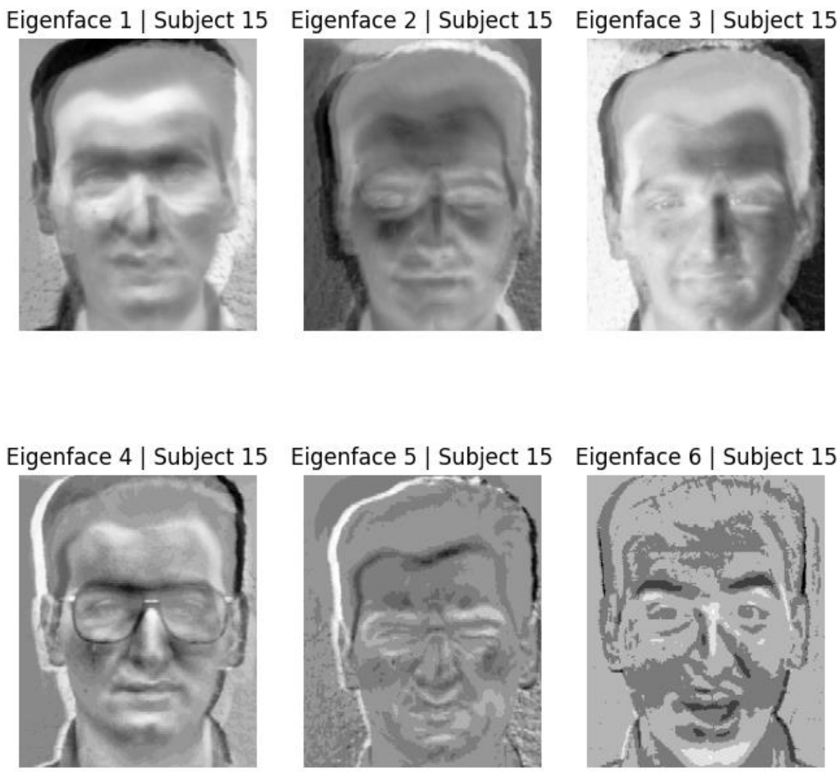
*Training/baseline/test set splits for subject 2 in the Yale Faces dataset*

The training set was used to create the top 6 eigenfaces for each subject. The baseline picture was used to generate the "ground truth" weights for each subject. In other words, the baseline face was projected into the eigenface space to determine the linear combination of the eigenfaces that represented that face in the eigenface space. The weights of each eigenface were assigned to a weight vector, $\boldsymbol{w}$, that would be used for classification.

The test face was projected onto the eigenface space for *each* subject to create a weight vector, $\boldsymbol{w}_k$, where *k* represents the eigenface space for subject k. Then, a series of dissimilarity metrics were calculated using the Euclidean distance between $\boldsymbol{w}$ and $\boldsymbol{w}_k$ for each *k*. If $\|\boldsymbol{w} - \boldsymbol{w}_k\|$ was smaller for a test face projected into its own eigenface space than when projected into a different subject's eigenface space, that test face is considered accurately classified. For example, if $\boldsymbol{w}$ represents the "ground truth" weight vector of subject 1, then $\|\boldsymbol{w} - \boldsymbol{w}_1\|$ represents the Euclidean distance between subject 1's ground truth weight vector and the weight vector assigned to subject 1's test face project into subject 1's eigenface space. Then, $\|\boldsymbol{w} - \boldsymbol{w}_2\|$ represents the Euclidean distance between the subject 1's ground truth weight vector and the weight vector assigned to subject 1's test face projected into subject 2's eigenface space. If the former is less than the latter, then subject 1's test face is better represented by the eigenfaces in subject 1's eigenface space. Alternatively, if the former is greater than the latter, then subject 1's test face is better represented by the eigenfaces in subject 2's eigenface space. The former instance considers subject 1's test face correctly classified, while the latter considers it incorrectly classified.

## Evaluation

After splitting the data, I began implementing the eigenface algorithm. I started by creating the top 6 eigenfaces for each subject using that subject's training set. Below is an example of the eigenfaces for subject 15.




*Top 6 eigenfaces for subject 15*

Next, I calculated the classification accuracy for each subject. I'll use subject 15 to illustrate. I started by taking subject 15's test face and subtracting the "average face" of subject 15's 9 training faces. Then, I projected this "centered" face onto subject 15's eigenface space. I found the Euclidean distance between this weight vector and subject 15's ground truth weight vector (i.e. $||\boldsymbol{w} - \boldsymbol{w}_{15}||$). Then, I took subject 15's test face and subtracted the "average face" used to create subject 1's eigenfaces. I subtracted the average of subject 1's training faces from subject 15's test face, then projected it onto subject 1's eigenface space to get $\boldsymbol{w}_1$. If $||\boldsymbol{w} - \boldsymbol{w}_{15}|| < ||\boldsymbol{w} - \boldsymbol{w}_1||$, I considered subject 15's face correctly classified, otherwise it was incorrectly classified. I repeated this process for subject's 2-14 until I had a total # of correctly classified subject 15 faces. In doing so, I am checking if each subject's test face can be properly classified using its own eigenface space rather than that of another subject. For reference, the classification accuracy for subject 15 was 93%. I then repeated this process for the remaining subjects. This served as a baseline level of classification accuracy. This baseline would be used later to compare with the classification accuracy when various obstructions cover the faces (ex. face masks or sunglasses).

After generating a base level of classification accuracy, I started to explore the research question at hand – how does the presence of a facemask impact the ability of the eigenface algorithm to recognize faces? First, I had to add facemasks to each of the subject test faces. Below is an example of some test faces with facemasks, emulated by black boxes covering the nose and mouth.

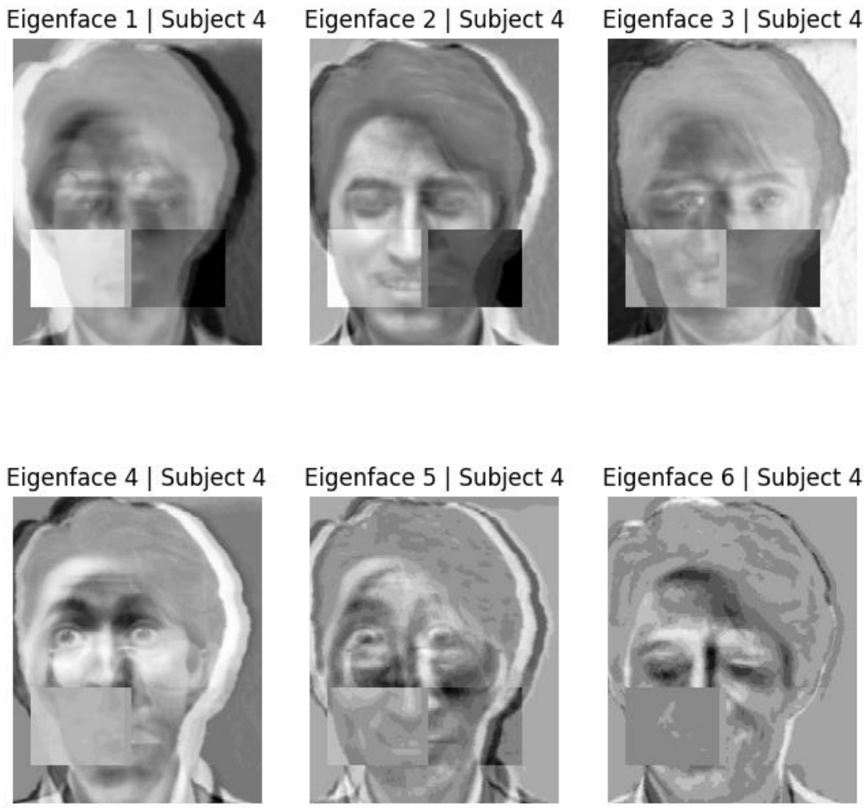
*A sample of 3 faces with "facemasks"*

Next, I projected each face-masked test face onto the eigenface space for each subject to see how well the base eigenfaces can identify the face-masked faces. This trial emulates the scenario where an iPhone user tries to unlock their iPhone while wearing a facemask. I anticipated that the classification accuracy would drop for the face-masked faces.

Then, I generated a new set of eigenfaces using the technique suggested by the Business Insider article, where a user covers part of their face with a folded facemask and retrains Face ID. To emulate this, I randomly covered each face in the training set with a "half-mask" – that is, a facemask that randomly covers the left or right side of the face. Below is a sample of half-masked faces from the training set.

*A sample of 3 half-masked faces from the half-masked eigenface training set*

As demonstrated by the third subject above, the half-masks aren't perfect. Based on the cropping and position of the subjects' heads, the half-masks may end up covering more or less than half of the bottom of the face. On average, the half-masks do a good job of obstructing a quarter of each subject's face. Below are the 6 eigenfaces that represent the half-masked eigenface space for subject 4.

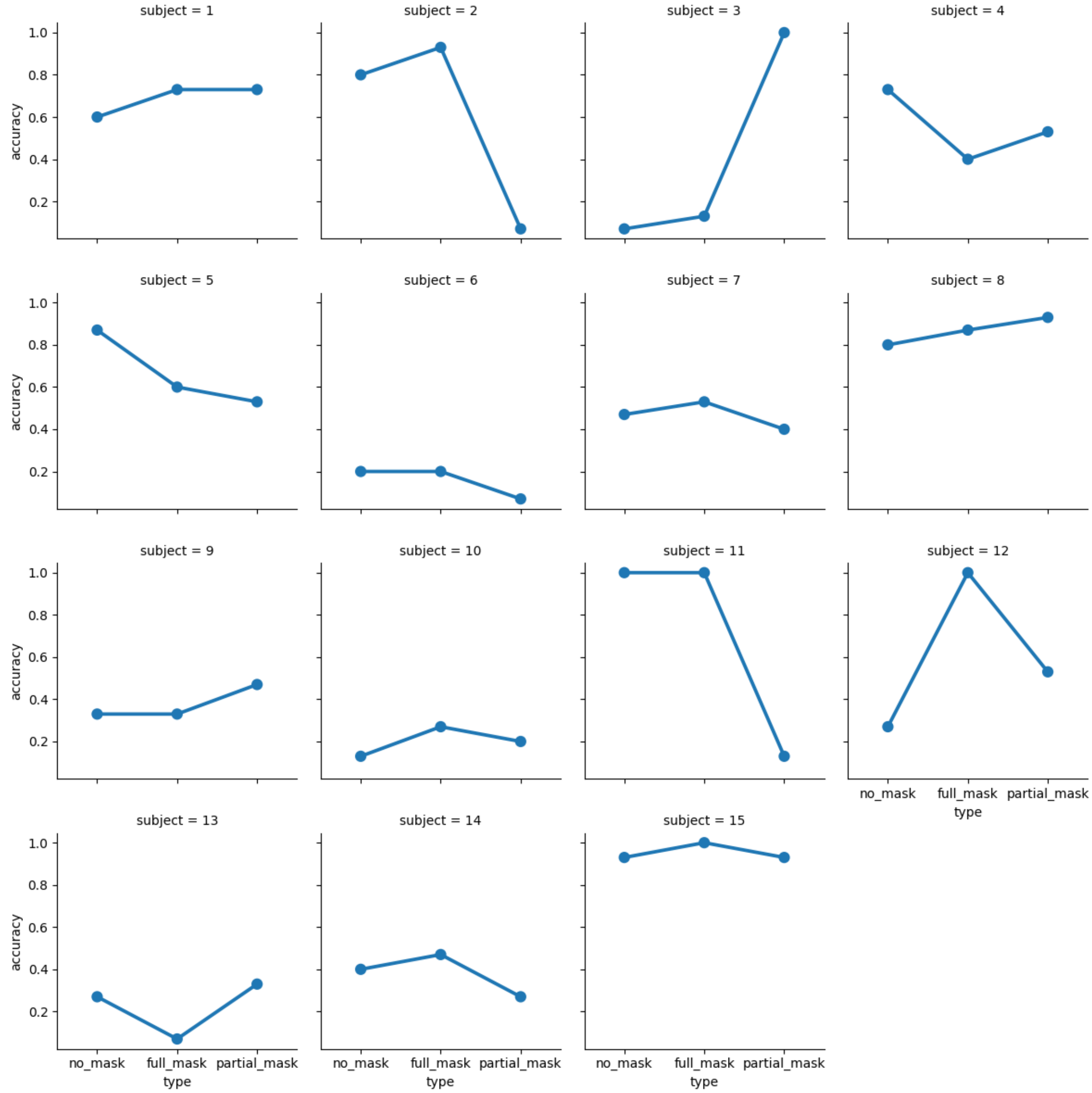



*Top 6 half-masked eigenfaces for subject 4*

Note the presence of the 2 boxes over each eigenface. These serve to obscure some of the features of each eigenface, notably the nose and the mouth. I anticipated the nose and mouth to be key features for classifying faces, which is why I figured the fully-masked test faces would be more difficult to classify with the base eigenfaces. The half-masked eigenfaces, however, have noses and mouths that are already partially obscured. In theory, this should place more prominence on other features, such as the eyes, for classifying faces.

I gathered a third set of classification accuracies using the same process as before. The average face used in this case was the average face for each subject's half-mask training set rather than the average face for each subject's unmasked training set. Finally, the accuracies from all three trials were combined in a series of line charts, shown below. Recall that the three experiments are as follows:

1. Unmasked eigenfaces used to classify unmasked test faces
2. Unmasked eigenfaces used to classify fully-masked test faces
3. Half-masked eigenfaces used to classify fully-masked test faces




*Classification accuracy of each subject's test face for 3 different scenarios*

Observe the top left plot, titled "subject = 1". The left-most point represents the classification accuracy of unmasked test faces using unmasked eigenfaces (labelled *no_mask* on the x-axis). The middle point represents the classification accuracy of masked test faces using unmasked eigenfaces (labelled *full_mask* on the x-axis). The right-most point represents the classification accuracy of masked test faces using half-masked eigenfaces (labelled *partial_mask* on the x-axis). Subject 1's unmasked test face was classified correctly about 60% of the time using the unmasked eigenfaces. Subject 1's masked test face was classified correctly about 70% of the time using both the unmasked eigenfaces and the half-masked eigenfaces. This indicates that subject 1's face was identified better when wearing a mask than without a mask, and that retraining the eigenfaces with half-masked faces did not improve that accuracy. The classification accuracy split for subject 2 is approximately 80% / 93% / 7%, indicating that subject 2's face was easier to identify while wearing a mask with unmasked eigenfaces, but nearly impossible to identify with half-masked eigenfaces while wearing a mask.

I expected that each line plot would be v-shaped. The unmasked face would be easy to identify with unmasked eigenfaces, but there would be a sharp drop-off in accuracy when trying to identify masked test faces. Then, the half-masked eigenfaces should have done a better job than the unmasked eigenfaces at classifying the masked test faces. This pattern only holds for subjects 4 and 13. Other subjects see a sharp rise or fall in classification accuracy when using the half-masked eigenfaces, such as subjects 2, 3 and 11. Most of the remaining faces see very little change in classification accuracy regardless of masking. Based on these results, it seems as if covering the face with a folded facemask and retraining Face ID should *not* improve its ability to recognize a face.

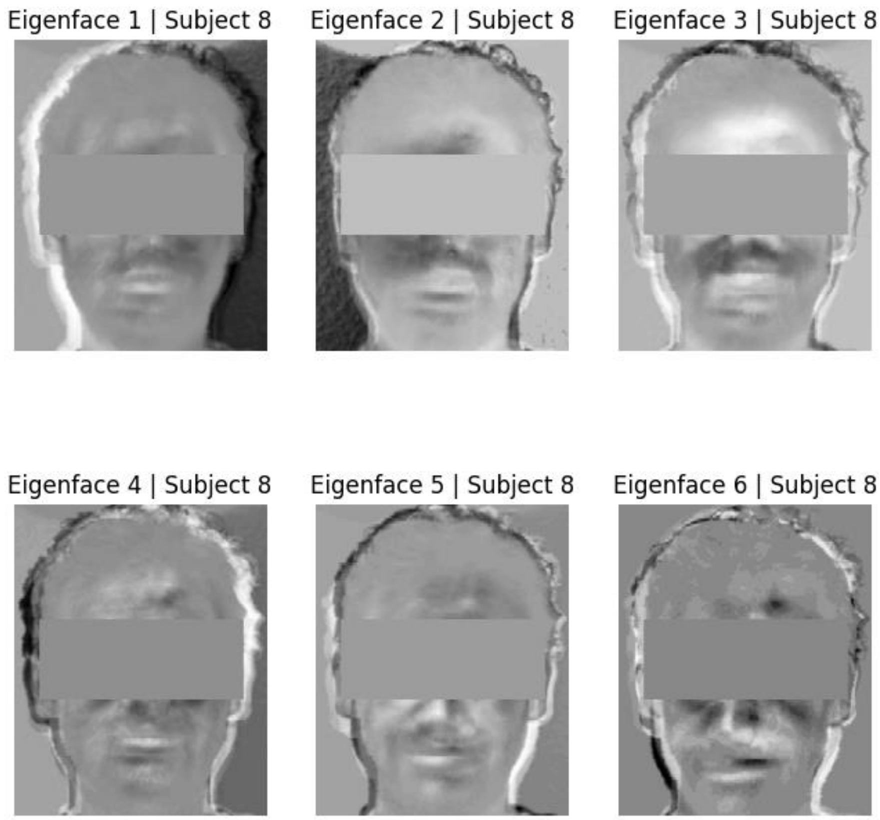
It's unfortunate that I was unable to validate the "hack" to get Face ID working while wearing a facemask, but the results also beg the question: are certain facial feature more important than others for facial recognition? Namely, are the eyes more important than the nose and/or the mouth? The results above seem to suggest so, since the facial recognition accuracy was largely unaffected before and after covering the test subjects' nose and mouth with facemasks. To explore whether other features carry more importance than the nose/mouth, I will perform the same analysis as used above, this time covering the subjects' eyes to emulate sunglasses. If the eyes are more important than the nose/mouth for facial recognition, I expect to see a large improvement in accuracy with "sunglasses" eigenfaces compared to the base eigenfaces when attempting to classify a test face wearing sunglasses.

To start, I drew black boxes over the subjects' eyes to emulate sunglasses similarly to how the facemasks were emulated. Below is a sample of some "sunglasses" faces from the training set.

*A sample of 3 training faces with "sunglasses" (very stylish!)*

Then, I used the sunglasses-faces from the test set to generate accuracy metrics using the base set of eigenfaces. I expected the accuracy to drop off when using the base eigenfaces to classify the sunglasses test faces.

Next, I created a new set of sunglasses-eigenfaces, where each subject's sunglasses-eigenface space is represented by the top 6 sunglasses-eigenfaces for that subject. Below is the set of 6 sunglasses-eigenfaces for subject 8.

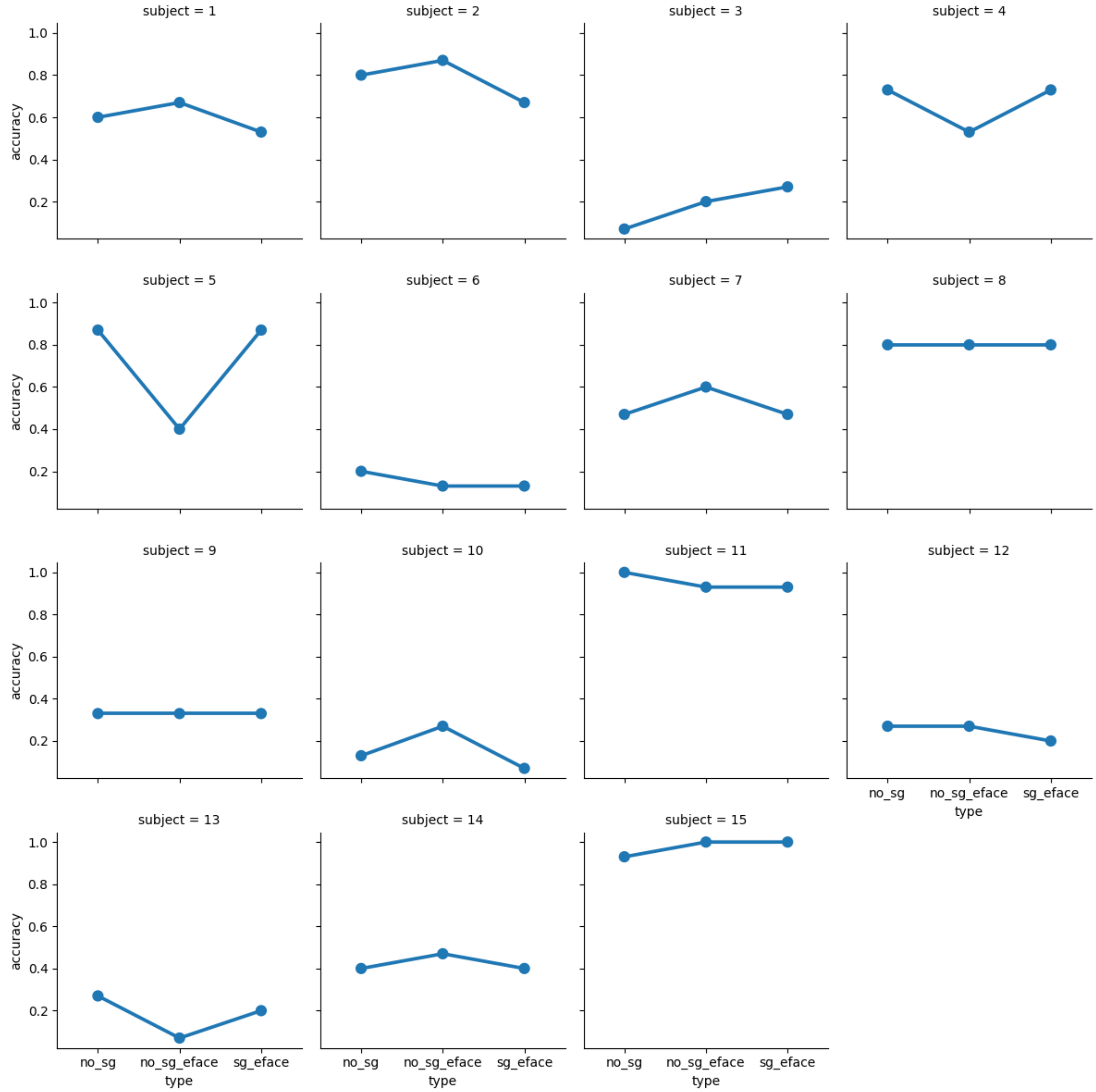



*Top 6 sunglasses eigenfaces for subject 8*

Note that, unlike with the face masks, I created the sunglasses eigenfaces using "full-sunglasses" training faces rather than "half-sunglasses" faces. Then, I used the sunglasses eigenfaces to classify the sunglasses test faces. In this way, I collected accuracy metrics for the following three scenarios:

1. No-sunglasses eigenfaces used to classify no-sunglasses test faces
2. No-sunglasses eigenfaces used to classify sunglasses test faces
3. Sunglasses eigenfaces used to classify sunglasses test faces

The results are represented below in a similar set of line plots.



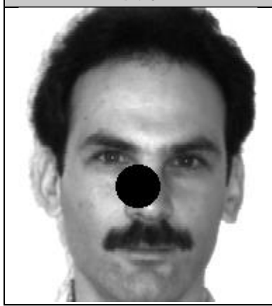

*Classification accuracy of each subject's test face for 3 different scenarios*

The left-most point in each chart represents the same as before – base eigenfaces used to classify the base test faces. The middle point represents base eigenfaces used to classify "sunglasses" test faces, and the right-most point represents sunglasses-eigenfaces used to classify "sunglasses" test faces.

Like before, the classification accuracy remains relatively constant for most subjects across all 3 scenarios. Subjects 4, 5 and 13 follow the v-shaped pattern I expected to see, indicating the sunglasses-eigenfaces do a better job at classifying the sunglasses test faces than the base eigenfaces. Unlike before, there are no subjects who experience a sharp rise or fall in classification accuracy. Since there is no clear pattern in classification accuracy across the different sunglasses scenario, it is difficult to say that the eyes are more important than the nose and/or mouth for the eigenface algorithm to correctly classify faces.

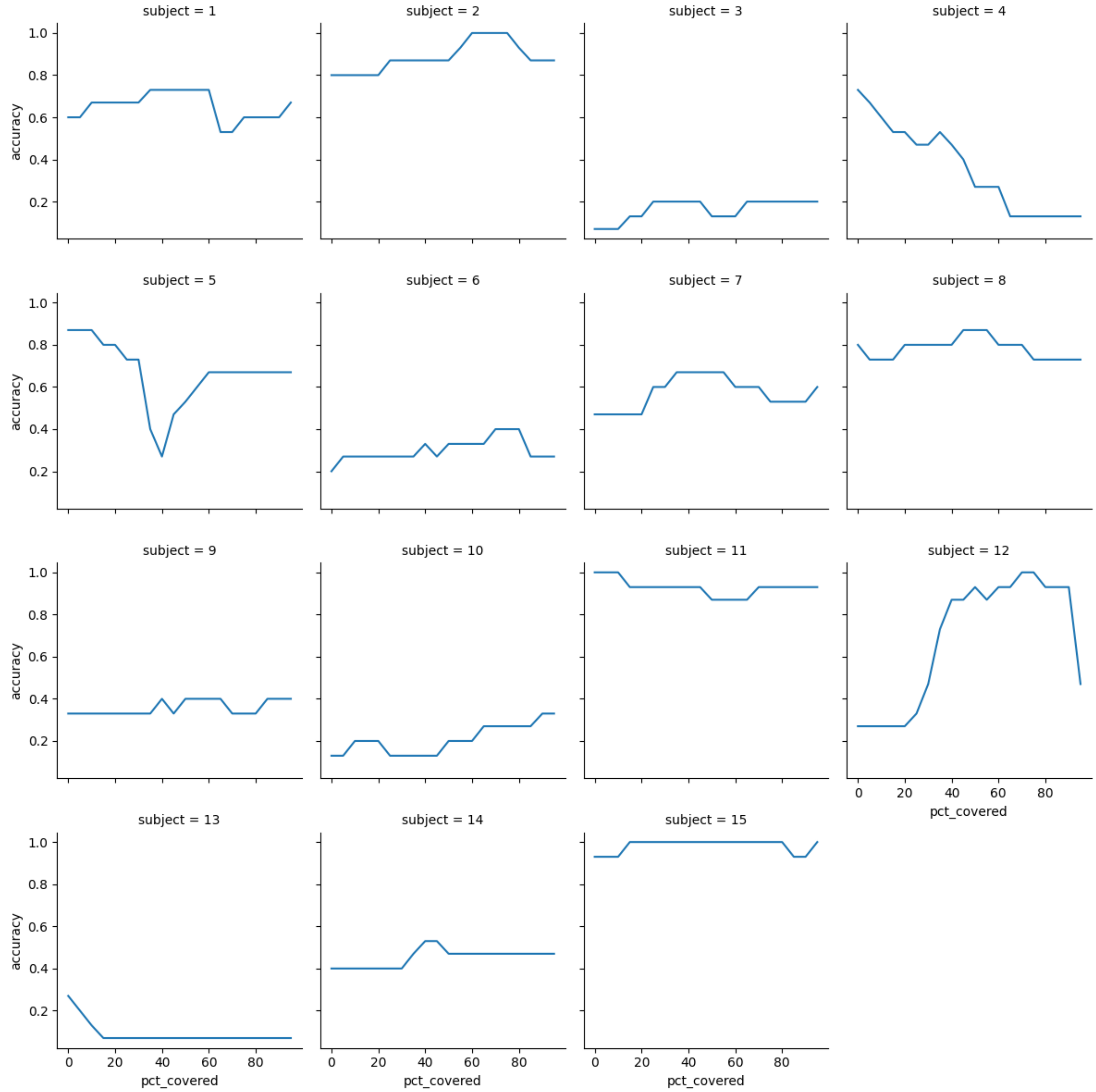
I wanted to perform one more set of experiments to see if I could tease out *any* change in classification accuracy from the eigenface algorithm. Covering the nose/face and the eyes of each subject seemed to have little impact on the classification accuracy. I wondered if there was some cutoff or some set of feature that, once covered, would dramatically alter the classification accuracy. As such, I carried out an experiment where I covered 5%, 10%, 15%, …, all the way to 95% of each subject's face and attempted to classify each "%-obstructed" face using the base eigenfaces. I anticipated that there would be an inflection point, perhaps around 50-60% facial obstruction when the nose, eyes and mouth would all be covered, where the classification accuracy would drastically drop off.

To carry out this experiment, I created 19 different sets of test faces, each corresponding to a different percentage of facial obstruction. Below is a sample of test faces with varying levels of obstruction.

| 5% | 25% | 50% | 75% | 95% |
|---|---|---|---|---|
|  |  |  |  |  |

*A sample of 5 test faces with varying % amounts of facial coverings*

Next, I found the classification accuracy of the obstructed test faces using each subject's base eigenface space. This was done to see how well an unobstructed eigenface could classify test faces with varying levels of obstruction. The classification accuracies for these experiments is shown below for each subject.




*Classification accuracy using the base eigenface spaces for varying levels of test face obstruction*

As with the face mask and sunglasses experiments, there doesn't appear to be a clear pattern followed by classification accuracy as the % of facial obstruction increases. I expected to see something similar to subject 4's plot, where the classification accuracy decreases as the facial obstruction increases. Instead, the classification accuracy for most test faces remains relatively constant across all levels of facial obstruction. This experiment fails to illuminate a cutoff point where the eigenface algorithm begins to classify faces much worse than before. It also does not indicate whether any individual feature or set of features is more important for classifying faces than others. If anything, this experiment suggests that perhaps the outline of the faces is more useful for classifying faces than anything else. An experiment that covers the outlines of the test faces with a black ring, for example, could be used to tease this information out. Additionally, perhaps a further exploration of eigenface feature importance could help determine which, if any, feature(s) are most important in performing facial recognition.

**Limitations**

There are a few limitations to both the eigenface algorithm for facial recognition and the experimentation approach I took. The eigenface method requires a very particular set of data – namely cropped, centered and forward-facing images of faces. On the contrary, Face ID requires a user to move their head up/down and side to side to ensure it captures the full facial profile. As such, the validity of this analysis, in part, hinges on the assumption that people will always be looking straight at their iPhones while unlocking them with Face ID.

Secondly, the way I emulate facial obstructions has its downfalls. There are some faces where the facemasks don't cover the nose, for example. In other instances, the half-masks end up covering more or less than 50% of the lower half of the subjects' faces. I attempted to cover the subjects' faces uniformly, but inconsistency is still present based on some of the more unique facial positions and orientations.

A final limitation is the custom algorithm I implement, whereby I create an eigenface space for each subject and use that to try to classify all subject faces. This differs from the classical eigenface approach, whereby all training faces are used to create a single eigenface space, and that single space would be used to classify the test faces for every subject. Perhaps my custom algorithm needs tuning.

**Conclusion**

In this paper, I wanted to use the eigenface algorithm to understand a problem many people encountered during the onset of the COVID-19 pandemic – the inability to use Face ID to unlock an iPhone while wearing a facemask. I started with a single research question in mind – does the trick of partially covering the face and retraining Face ID on the iPhone allow it to work while the owner is wearing a full face mask?

To answer this question, I trained 2 different set of eigenfaces – one on unobstructed faces and another on faces partially covered by a half-mask (i.e. a folded over face mask or cloth). I projected the test face for a subject onto two different eigenface spaces – one belonging to that subject, and one belonging to a different subject. If the projection weight vector for the test face projected onto its own eigenface space was less than that for the test face projected onto the other eigenface space, the test face was considered accurately classified. I found that there was no noticeable improvement in the half-mask eigenfaces' ability to classify masked test faces than that of the unmasked eigenfaces.

I performed a similar experiment with sunglasses, where I trained a set of sunglasses eigenfaces to see if they could classify faces wearing sunglasses better than the unobstructed eigenfaces. The results of this experiment were like the last, in that there was no noticeable improvement in the sunglass-eigenfaces' ability to classify faces.

Lastly, I performed an experiment to see if there was a percentage of facial obstruction where the eigenface algorithm's classification accuracy drastically dropped off. I created several test sets, where each test set's faces were obstructed by 5, 10, … all the way to 95%. There was no such cutoff point where the facial recognition accuracy changed dramatically.

The paper titled *Face Recognition Using Eigenface Approach* suggests that the classification accuracy noticeably improves as the number of eigenfaces used to represent the eigenface space increases. Given the nature of this problem, I was constrained to a maximum of 9 eigenfaces per eigenface space, of which I elected to use 6. Perhaps the results of this analysis would be different if I opted to use the full set of 9 eigenfaces, or if I oversampled the faces to permit the use of even more eigenfaces. Additional techniques, such as using a single eigenface space to classify all faces, could be used to explore how results may vary.

While these experiments were inspired by the COVID-19 pandemic, their future applicability are much more extensive. This study can serve as a baseline for exploration into how facial obstructions impact individuals' ability to interact with technology. Some groups of interest include healthcare professionals, individuals with visual impairments, or even people who wish to wear masks in crowded public areas. The eigenface algorithm, and facial recognition in general, provides a powerful technique for enforcing security. Despite its prominence in our daily lives, the field is still ripe for exploration and improvement.

# References


1. Apple. “About Face ID advanced technology.” Apple Support, 2022, https://support.apple.com/en-us/HT212788.

2. Belitskaya, Olga. Yale Face Database. Kaggle, 2017, https://www.kaggle.com/datasets/olgabelitskaya/yale-face-database.

3. Business Insider. “iPhone Face ID not working with mask? Try this workaround trick.” Business Insider, 7 Apr. 2020, https://www.businessinsider.com/iphone-face-id-not-working-with-mask-try-workaround-trick-2020-4.

4. Çarıkçı, M.üge & Özen, Figen. (2012). A Face Recognition System Based on Eigenfaces Method. Procedia Technology. 1. 118–123. 10.1016/j.protcy.2012.02.023.

5. Georghiades, A.S. and Belhumeur, P.N. and Kriegman, D.J., “From Few to Many: Illumination Cone Models for Face Recognition under Lighting and Pose”. IEEE Trans. Pattern Anal. Mach. Intelligence, 2001

6. PR Newswire. “Americans Check Their Phones 96 Times a Day.” PR Newswire, 21 Nov. 2019, https://www.prnewswire.com/news-releases/americans-check-their-phones-96-times-a-day-300962643.html.

7. Slavković, Marijeta & Jevtić, Dubravka. (2012). Face Recognition Using Eigenface Approach*. Serbian Journal of electrical engineering. 9. 121-130. 10.2298/SJEE1201121S.

8. Turk, Matthew. “Face Recognition Using Eigenfaces.” Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference, 1991, pp. 586-591.