Development of a Specialized Facial Recognition Technique for Individuals Wearing COVID-19 Facial PPE

Alison Davis
Engineering and Computer Science
Syracuse University
Syracuse, USA
adavis@syracuse.edu

Ricardo Alonzo Ugalde
Engineering and Computer Science
Syracuse University
Syracuse, USA
raalonzo@syr.edu

***Abstract*—Biometric technologies are already seeing major impacts as a result of the COVID-19 pandemic. In addition to an increasing preference for *contactless* biometrics such as facial recognition (FR) technologies, FR itself is also experiencing new challenges as the problem is presented of how to perform FR on people wearing COVID-19 facial PPE. An evaluation of the performance impacts of COVID-19 face masks on the well-known holistic method, Eigenfaces, shows that training the Eigenfaces algorithm with masked data and performing recognition on unmasked images sees higher performance metrics than training with unmasked data and performing recognition on masked images. This is attributed to the algorithm depending upon features that can no longer be seen. Thus, a modified Eigenfaces method is developed which builds the model with cropped portions of the training images. Training with unmasked data and performing recognition on masked images sees a more than 10% increase in accuracy using the mask-modified Eigenface model.**

I. Introduction

The changes to daily life brought about by the COVID-19 pandemic have been felt around the world. Since the declaration of COVID-19 as a global pandemic in March of 2020, it has become increasingly evident that many of these changes will not be reverting to their pre-COVID state in the near future—if ever. One of the most prevalent changes to daily life is the wearing of masks or face coverings over the nose and mouth in public environments. In some locations across the country, the uncontrolled spread of the virus has forced local governments to recommend that residents living with high risk individuals even wear protective equipment inside their own homes [9]. This change in behavior has the potential to lead to major impacts on FR technologies.

Covering of the lower half of the face represents a significant loss of data for FR algorithms to work with. The National Institute of Standards and Technology (NIST) launched an investigation into 89 commercially used FR algorithms which discovered error rates anywhere from 5% to 50% for matching people with and without COVID-19 masks. The tests involved digitally inserting mask shapes onto approximately 6 million photos that had previously been studied using the Face Recognition Vendor Test (FRVT). The mask shapes inserted varied in shape, degree of coverage, and color [10].

Regardless of the variations in masks, all of the algorithms showed decreased accuracy when the facial coverings were applied to the images. On the better end of the spectrum, algorithms that had previously seen 0.3% false negative rates for facial identification now saw about a 5% failure rate. However, many of the other previously functional algorithms were seen to fail 20% to 50% of their test cases [10].

Several factors also contributed to the reduced accuracy of algorithms such as the amount of nose coverage provided by the mask, shape of the mask, and color of the mask. Most of the algorithms performed worse on rounded masks than any other shape tested. Black masks resulted in less accuracy than surgical blue masks. It's worth noting, however, that the variation in colors and pattern was limited. No colors beyond blue and black were tested in this study [10]. This loss of accuracy builds upon the previous discovery of inaccuracy in FR due to race and gender [5].

FR systems have already seen an increased use in businesses and public places due to COVID-19 precautionary requirements, albeit with some COVID-19 alterations. Previously popular biometric systems that require contact, such as fingerprint scanners, are beginning to be replaced with FR [8]. As these systems become increasingly popular and with no definitive end to the pandemic in sight, removing COVID-19 PPE for FR systems to work makes using them more inconvenient, unsafe, and in some cases, impossible.

II. Background

Automation of FR technologies has has experienced a tremendous growth in sophistication since its initial inception in the early 1960s. The first reported semi-automated face recognition system classified faces using marks entered on photographs by hand and used normalized distances and ratios among points such as eye corners, mouth corners, nose tip, and chin tip. Other fully-automated variations on this feature-based method followed using features such as shade of hair, length of ears, and lip thickness [12].

By the late 1980s several fully automated algorithms had been developed using a variety of techniques. Several feature-based approaches that extract features from an object, create a model, and achieve recognition by matching feature sets began to be widely tested and implemented. As algorithms continued to improve, more focus was given to the issues of noisy, unclear, and missing features [12].

Recognition of faces, much like the recognition of other objects using computer vision algorithms, is influenced by a variety of outside factors such as shape, reflectance, pose, occlusion, and illumination [12]. Although the problem of how to recognize people wearing COVID-19 Personal Protective Equipment (PPE) is a new hurdle for FR technologies, the problem of partial facial occlusion is not. Investigations have been made into the ability to perform 3D geometry-based FR of faces that are partially occluded by external objects. These studies revealed a dramatic decrease in performance when faces are covered with eyeglasses, hair, hands, scarves or hats: all common real-world scenarios [1].

The loss of performance found in the algorithms can be attributed to the decrease in discriminative information of the face available for processing while subjects are wearing PPE. In particular, the eye region has been shown to be the most discriminative region of the face. Fortunately for recognition in COVID PPE, this is the area of the face that can still be analyzed. Previous research into this area involves both methods for *occlusion detection* as well as *restoration* [1].

One technique for working with occluded facial data is to use radial facial curves originating from the nose tip for shape analysis. This technique is less viable for the scenario of masks due to the occlusion of the nose tip itself. Another simplistic technique that has been applied is to adaptively select the non-occluded facial regions for FR instead of using the entire facial surface. This requires strategies for identifying the validity of each region. This method was able to achieve a 97.87% recognition rate (RR) for mouth occlusion [1].

Another technique that has generated useful results is Iterative Closest Normal Point (ICNP). ICNP is computationally less expensive than other FR techniques, and it has actually generated reasonably accurate test results. ICNP works by first determining sample data points on a subject's face then denoting said points as the "closet normal" points. Once all of the data points are aligned across a data set of many faces, the algorithm can predictably and effectively apply a discriminant method of analysis. [7].

Alternatively, another attempted approach is a variant of the Iterative Closest Point (ICP) algorithm for 3D FR. This method involved discarding the occluded areas by removing the worst *n%* of points: yielding a maximum point-to-point square error. This method showed its relative effectiveness with an 85.78% overall RR [1].

Once occlusion is detected there are only two options: *restoral* or *removal* of the occluded areas. Incorrect restoral has been seen to decrease identification rates by a significant margin. Although some techniques have found traction with the restoration method, several other groups to research this area have opted for the removal method due to its simplicity and to avoid reducing accuracy [1].

Dagnes, Marcolin, Nonis, et al. proposed an automatic 3D Occlusion Detection and Restoration system that detects, localizes, and classifies occlusions. One of the cornerstones of this study is the use of the symmetrical property of the face in the restoration process [1]. Unfortunately, in the case of masks, it is known that the portion of the face that will always be covered will not be symmetrical to the part that is exposed.

The eigenfaces method, based on Principal Component Analysis (PCA), was developed to consider the face holistically. Its goal was to consider the face *between the features*. Feature-based approaches simplify the problem down to just feature-points, discarding some potentially useful data that is not captured by those specific points. Depending too much on singular features leads to a lack of robustness for an algorithm, especially when occlusion comes into play. Eigenfaces originated to create an FR system that was allowed to determine what features are most important to encode for future recognition at run-time, rather than complying to a prescribed set of features [12].

III. Eigenfaces Overview

The eigenfaces algorithm spawns from the linear algebra method of calculating eigenvalues and eigenvectors.

A. Eigenanalysis Summary

An eigenvalue (λ) is a scalar associated with the eigenvector (X). An eigenvector is a vector scaled by a linear transformation that is a property of a matrix. When an eigenvector is acted upon by a matrix (**A**), the vector is altered only in magnitude, not direction [11].

$$AX = \lambda X \tag{1}$$

Equation 1 can be manipulated with an Identity matrix and rearranged to create Equation 2:

$$(A - \lambda I)X = 0 \tag{2}$$

where **I** is an $n \times n$ Identity matrix. This homogeneous equation can be solved for λ in the event that **X** is non-zero using the determinant as shown in Equation 3:

$$det(A - \lambda I) = 0 \tag{3}$$

Solving Equation 3 results in the characteristic equation: an n-degree polynomial equation that when solved yields the eigenvalues. When A is a $n \times n$ matrix, n roots may be found when solving the characteristic equation. Therefore, there is a maximum of n distinct eigenvalues for **A** satisfying the following Equation 4:

$$AXi = \lambda Xi \qquad s.t. \quad i = 1, 2, 3...n \tag{4}$$

Each distinct eigenvalue is associated with a linearly independent vector that represents a *unique direction*. Thus, the eigenvectors can span at most an n-dimensional Euclidean space.

Specific to the case of eigenfaces method, the approach is comprised of two sub-steps: *initialization* and *recognition*.

B. Initialization

The steps for the initialization process are as follows:

1) Acquire a set of training images.
2) Calculate the highest eigenfaces from the training set such that M images define the face space.
3) Calculate the M-dimensional weight for the individuals "known" by the model by projecting the images of the individual's face onto the *face space*.

The face space is the sum of the eigenfaces. Ideally these steps should not have to be repeated before each recognition is performed but rather a model can be trained once, maintained, and used repeatedly for recognition.

C. Recognition

The steps for recognition then follow:

1) Accept an input sample and calculate the corresponding set of weights from the M eigenfaces by projecting the image onto each eigenface.
2) Identify if the image is a face at all using Viola-Jones.
3) Given the test sample is evaluated as a face, classify if the face is known or unknown by comparing the weight pattern to those of the known faces. A face is only predicted as known if the closest known face is within a certain threshold limit by distance from the test sample weight vector.
4) *(Optional):* To continue adapting over time if the same unknown face is identified several times then it should be "learned" as a new known face.

D. Calculating Eigenfaces

Eigenfaces operates on the premise that all face images can be described as a weighted linear combination of a series of basis faces. The steps of this type of analysis is used to create a lower-dimensional representation of the members of a set of high-dimensional objects: such as a large training set of face images. Let $\Gamma_1, \Gamma_2, ..., \Gamma_m$ represent the set of training images where each image is I(x,y). Each image needs to be converted into a set of vectors. The new full-size matrix is of size $m \times p$, where m is the number of training images and $p = x \times y$.

Now the *mean face* can be found by Equation 5:

$$\psi = \frac{1}{m}\sum_{i=1}^{m}\Gamma_i \tag{5}$$

Then the mean-subtracted face (value found by subtracting the mean from each face) is represented by Equation 6. This normalizes all of the images by the mean face.

$$\phi_i = \Gamma_i - \psi \qquad s.t. \quad i = 1, 2, 3...m \tag{6}$$

Now the mean-subtracted matrix vector becomes:

$$A = [\phi_1, \phi_2, ..., \phi_m] \tag{7}$$

where matrix **A** is of size $m \times p$. Now it is important to reduce the mean-subtracted matrix vector using the following matrix transformations:

$$C_{mn} = A_{mp} \times A_{pm}^T \tag{8}$$

given **C** is the covariance matrix and $\mathbf{A}^T$ is the transpose matrix of **A**. This step is critical for reducing the dimensional space of the model, otherwise it would make for a more computationally intensive alogrithm. At this point the eigenvalues (λ_m) and eigenvectors ($\mathbf{V}_{mm}$) can be found as described in Section III-A. Now apply the eigenvectors matrix ($\mathbf{V}_{mm}$) and adjusted matrix ψ_m which identify linear combinations of the training set images to form the eigenfaces (U_k) using Equation :

$$U_k = \sum_{n=1}^{m}\phi_n V_{kn}, \qquad s.t. \quad k = 1, 2, 3...m \tag{9}$$

Now instead of m eigenfaces, there are m' eigenfaces such that $m' < m$. Each image now has a unique face vector:

$$W_k = U_k^T(\Gamma - \psi), \qquad s.t. \quad k = 1, 2, 3...m' \tag{10}$$

The mean subtracted vector is of size $p \times 1$ and eigenfaces is $U_{pm'}$. The weights form the feature vector (Ω) where

$$\Omega^T = [w_1, w_2, ..., w_{m'}] \tag{11}$$

After the creation of this weight vector, also known as the feature vector, the only task left to perform recognition is classification. Classification is done using distance. For the version of eigenfaces used in this study the distance can be considered as:

$$e_r = min||\Omega - \Omega_i|| \tag{12}$$

What is shown in Equation (12) is the calculation of the distance between the test probe's feature vector to each of the weight vectors associated with each individual it was trained on. The probe image is matched to the weight vector of one of the training individuals that it is closest to by distance. If this distance is less than a certain threshold (θ), such that $e_r < \theta$ then it is said that the test probe is *recognized* as that individual. If $e_r > \theta$ then the prediction is that the individual in the probe image does not belong to the database.

Determining the threshold (θ) is done depending upon the use case of the algorithm and impacts the performance metrics. The distances may be calculated using different distance formulas but is most commonly calculated as the Euclidean distance [11].

IV. Significance

The COVID-19 pandemic marks a potential turning point for biometric technologies. The first major change is that the amount of facial data easily accessible for FR is reduced due to occlusion by masks. Secondly, FR technologies are also predicted to see a major increase as a result of the pandemic due to its ability to perform *contactless* identification [8]. Thus, it is important for FR systems to work without the removal of masks. Additionally, it has been seen that pre-existing FR algorithms that do not account for masks have seen a unanimous degredation of performance as a result of COVID PPE [10]. Thus, the question arises: *is it possible to introduce adjustments to a pre-existing FR algorithm to improve recognition in COVID-19 facial PPE?*

V. Experimental Design

Following a survey of the possible algorithms to improve upon, it was decided to focus in on principal component analysis in the form of the Eigenfaces algorithm. This algorithms has served as a cornerstone for FR technologies for many years as a holistic approach that handles occlusion better than most feature-based algorithms. In addition to being widely accepted, it is also well-documented with existing libraries for OpenCV. Improving upon the ability of PCA or specifically the Eigenfaces algorithm with regards to recognition in masks would be a valuable biometric contribution moving into the post-COVID era. While the goal of the experiments is to generate significant positive results, any outcome of the tests performed is of significance.

A. Assumptions

The focus of this experiment is to hone in specifically on improving FR for those wearing COVID masks. Although there are many nuances and difficulties involved in FR, the effects of neither pose nor lighting in masks will be considered in this study outside of what is already provided by the algorithm. Only full frontal pose will be used for the training data. Lighting of images will try to be kept as constant as possible. Finally, it is also assumed that the algorithm itself does not have to perform the detection of the occlusion, but rather a flag can be set to inform the algorithm to use the modified version for masks.

A variety of technologies that fill the need for PPE protection have already been been developed and released such a Amazon's *Amazon Rekognition* or Vehant Technologie's *PPEye*. These technologies play an important role in ensuring the safety of individuals when presented with the need to gather in public places. The goal of this study however, is not to develop or improve upon existing technologies that detect face masks, but rather enhanced recognition of individuals wearing face masks following detection of said face masks.

VI. Sources of Data

The first source of training used was from the online database of faces: *NIST Special Database 32*. Samples of this data have were digitally altered to insert masks onto the images as shown from Figure 1 to Figure 2.

Fig. 1: Unmasked Sample

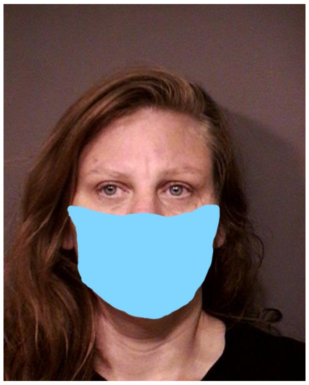

Fig. 2: Masked Sample

However, manually inserting the masks proved to be too labor intensive to generate enough training and test images to perform the scale of analysis desired in the Experiments outlined in Section VII. Thus, a new source of data was found which provided 10,000 samples of faces along with a companion set of the same face images with masks digitally inserted as is shown in Figure 3 and Figure 4.

Fig. 3: Unmasked Sample

Fig. 4: Masked Sample

VII. Algorithm Modifications

The first goal of the study is just to quantify the loss of accuracy the Eigenfaces algorithm experiences when presented with a data set of individuals wearing face masks. In most cases, the face mask will be digititally inserted into an existing database of images that fit the following critiera: the masks will either be blue or black, the subject in the mask must be facing the camera directly, the size and lighiting of the images must be kept consistent. Modifications will be made to the algorithm/program to try to enhance the ability to recognize faces in masks. The following are planned experiments:

A. Experiment A: Evaluate Eigenfaces ability to perform recognition in masks using non-masked training data

After running non-masked training data for the Eigenfaces algorithm, try to perform recognition on the faces with masks virtually added. Evaluate using statistics for accuracy, false match rate (FMR), false non-match rate (FNMR), precision, and recall. This test case will involve training the algorithm with 500 non-masked facial images such as the one shown in Figure 3, and then test using 1000 samples with masks digitally inserted as shown in Figure 4. The model will have been trained for 500 of the test samples which should be recognized and the other 500 test samples will not have been trained and should be rejected.

B. Experiment B: Evaluate Eigenfaces ability to perform recognition in masks using masked training data

In the second experiment the Eigenfaces model will be trained with masked data then it will be used to perform and evaluate recognition on unmasked faces. The experimental results will be evaluated using statistics for accuracy, FMR, FNMR, precision, and recall. This test case will involve training the algorithm with 500 masked facial images with masks digitally inserted such as the one shown in Figure 4, and then test using 1000 unmasked samples as shown in Figure 3. The model will have been trained for 500 of the test samples which should be recognized and the other 500 test samples will not have been trained and should be rejected.

C. Experiment C: Perform a modification to the algorithm to enhance performance

Finally, the algorithm/program will be modified to try to improve its ability to perform recognition of masked faces. Using the information gained from analyzing results of Experiment A and Experiment B, the Eigenfaces algorithm will be modified to try to enhance recognition performance for the case of having to recognize masked faces. Further clarification of Experiment C is made in the Analysis of Experiments A & B in Subsection IX-C.

VIII. Code Selection

The coding tools used for this experiment were OpenCV and Visual Studio. Apart from the script for prepping the CSV files for importing data, which is written in python, all coding and analysis was done in C++. The coding foundation for development was taken from an online project for the eigenfaces solution provided by OpenCV https://docs.opencv.org [2].

Prior to choosing the final code base to use as a starting point, a survey of existing C++ code options pertaining to Eigenfaces was performed. The output of one of the considered algorithms from https://learnopencv.com is shown in Figure 5 which included a user interface with slider bars for each of the principal components [6]. In this figure it is also possible to lightly see the effects of a glasses occlusion around the eyes. This arose by adjusting the eighth principal component slider. However, this code base was ultimately eliminated as a potential starting point due to its lack of built-in recognition. Since eigenfaces as an FR algorithm already exists it did not make sense to spend time re-implementing the recognition feature just to start with this code base.

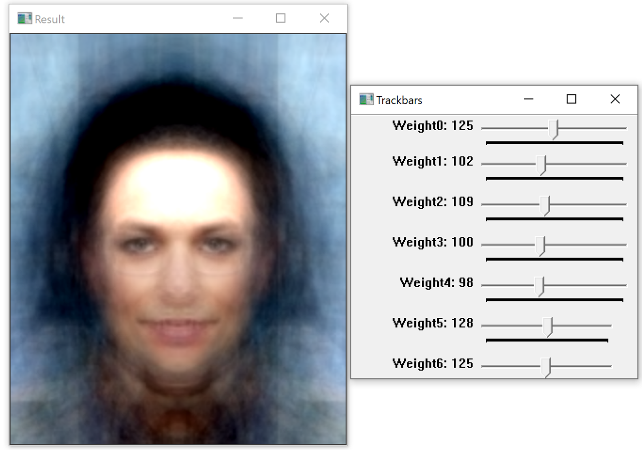

Fig. 5: Standard Eigenfaces Solution

However, for learning purposes, experimentation with this code was done using the manually edited data from the *NIST* database images as shown in Figure 6.

Fig. 6: Eigenfaces Algorithm with Masked Training Faces

This first eigenfaces code base that was run was a useful learning tool used to gain a better understanding of principal components, and how changing the weights of these linearly uncorrelated vectors impacts the image. However, due to lack of recognition capabilities this code was not used for anything further. Next, a brief investigation of how PCA Anlysis was also done using pre-made OpenCV code from https://docs.opencv.org [3]. This analysis was not specialized for faces like Eigenfaces, but rather its usually implemented for the analysis any arbitrary object. The PCA algorithm was run on some of the sample data made for this test is shown in Figures 7 and 8.

Although this algorithm produces interesting results that were unique depending on whether or not the sketched mask was present, it was not desired to work with PCA directly when it has already been specialized to faces using eigenfaces.

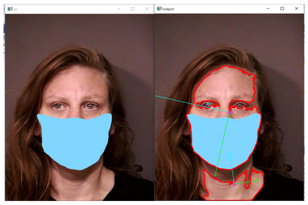

Fig. 7: Normal Mask

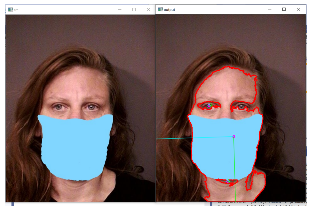

Fig. 8: Enlarged Mask

The final eigenfaces code base that was considered and ultimately used is capable of performing FR by matching a test sample to the trained eigenface that it is closest to by ”distance” in the vector space. In the unedited version of this code base, the final sample in the dataset given to this version of the eigenfaces code is treated as the *“test sample”* and used to perform recognition following the training of the model. This code base from https://docs.opencv.org yields outputs as shown in Figure 9. This is the code base that was modified and ultimately used for Experiments A, B, and C.

Fig. 9: Second Eigenfaces Algorithm with Recognition Capabilities

IX. Results

Before producing experimental results with the Eigenfaces C++ algorithm provided by OpenCV it was first necessary to modify the program to enable multiple test samples to undergo recognition for a single training model. After adding this ability, the desired experimental results were produced and documented for *Experiment A* and *Experiment B*.

A. Experiment A

Experiment A was performed according to the description in Section V. First, an Eigenfaces model was trained with 500 unique *unmasked* face images such as the one shown in Figure 3. Then, 1000 test samples of face images with masks digitally inserted into the image were run through the trained model. The first 500 images matched the trained faces and *should* be matched. The second 500 were not trained and *should* have been identified as not belonging to the model. The Eigenfaces model can provide a confidence value for its prediction in the form of a distance. This distance represents the euclidean distance of the test sample’s weighted feature vector from the weighted trained data vector that it is paired to. Table I shows the evaluation of the results taken from the experiment as the distance threshold is varied from 11,000 to 5,000 at increments of 500. The table shows metrics for the true positive (TP), false positive (FP), true negative (TN), false negative (FN), accuracy (Acc.), false match rate (FMR), false rejection rate (FRR), precision (Prec.) and recall (Rec.).

Experiment A: Unmask Train, Mask Test									
Dist.	**TP**	**FP**	**TN**	**FN**	**Acc.**	**FMR**	**FRR**	**Prec.**	**Rec.**
11000	498	433	67	2	0.57	0.87	0.004	0.535	0.99
10500	498	384	116	2	0.61	0.77	0.004	0.56	0.99
10000	489	312	188	11	0.68	0.62	0.022	0.61	0.98
9500	478	210	290	22	0.77	0.42	0.044	0.70	0.96
9000	457	117	383	43	0.84	0.23	0.086	0.78	0.91
8500	418	50	450	82	0.87	0.10	0.164	0.89	0.84
8000	386	10	490	114	0.88	0.02	0.228	0.97	0.77
7500	323	1	499	177	0.82	0.0	0.354	0.99	0.65
7000	249	0	500	251	0.75	0.0	0.502	1.0	0.50
6500	170	0	500	330	0.67	0.0	0.660	1.0	0.34
6000	99	0	500	401	0.60	0.0	0.802	1.0	0.20
5500	55	0	500	445	0.56	0.0	0.890	1.0	0.11
5000	16	0	500	484	0.52	0.0	0.968	1.0	0.03

Table I: Analysis of Experiment A results by varying distance threshold

The maximum accuracy seen in this table is approximately 88% at a threshold of 8000. Plotting of the ROC curve corresponding to the data in Table I can be seen as the blue line in Figure 10. Although the accuracy is highest at this point, it can be noted that the FRR is considerably higher than the FMR meaning that the algorithm is leaning towards rejecting more faces than is ideal.

B. Experiment B

Experiment B was performed according to the description in Section V. In this experiment, the Eigenface model was trained with 500 *masked* face images such as the one shown in Figure 4. The model was then tested with 1000 sample images *without masks* such as Figure 3. Once again, the first 500 matched the trained faces apart from the difference in mask and should have been able to have been matched. The second 500 were not trained and should have been identified as not belonging to the model. The results were generated with the distance confidence value for each prediction. Table II shows the evaluation of the results taken from the experiment as the distance threshold is varied from 11,000 to 5,000 at increments of 500.

The maximum accuracy seen in this table is approximately 97% at a threshold of 7000. Plotting of the ROC curve corresponding to the data in Table II can be seen as the orange line in Figure 10. At the highest accuracy threshold it can be noted that the FRR is slightly lower than the FMR but both are very low which corresponds to the precision and recall both being very high.

Experiment B: Mask Train, Unmask Test									
Dist.	**TP**	**FP**	**TN**	**FN**	**Acc.**	**FMR**	**FRR**	**Prec.**	**Rec.**
11000	500	486	14	0	0.51	0.97	0.0	0.51	1.0
10500	500	458	42	0	0.54	0.92	0.0	0.52	1.0
10000	500	422	78	0	0.58	0.84	0.0	0.54	1.0
9500	499	376	124	1	0.62	0.75	0.0	0.57	0.99
9000	499	316	184	1	0.68	0.63	0.0	0.61	0.99
8500	498	216	284	2	0.78	0.43	0.0	0.70	0.99
8000	497	124	376	3	0.87	0.25	0.01	0.80	0.99
7500	496	60	440	4	0.94	0.12	0.01	0.89	0.99
7000	489	20	480	11	0.97	0.04	0.02	0.96	0.98
6500	480	7	493	20	0.97	0.14	0.04	0.98	0.96
6000	456	2	498	44	0.95	0.01	0.09	0.99	0.91
5500	420	0	500	80	0.92	0.0	0.16	1.0	0.84
5000	346	0	500	154	0.85	0.0	0.30	1.0	0.69

Table II: Analysis of Experiment B results by varying distance threshold

C. Analysis of Experiments A & B

The performance of the models from Experiment A, and especially Experiment B, are almost impossibly good. Their performance relative to one another can be visualized by their ROC curves in the following Figure 10:

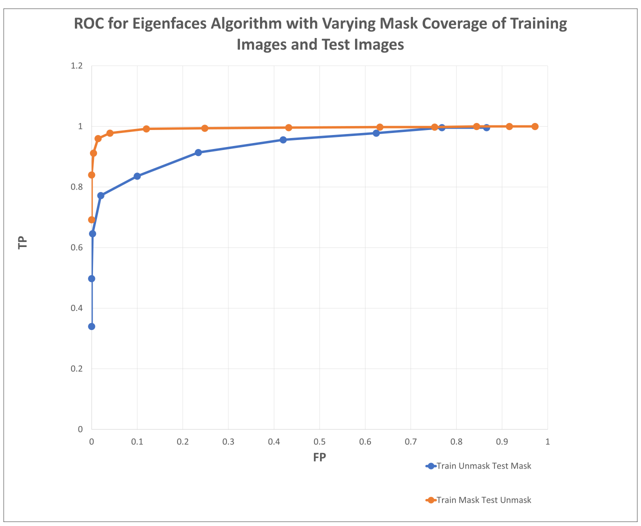

Fig. 10: ROC Curves for Eigenfaces Experiment A and Experiment B

The nearly impossibly good performance of these models is attributed to data limitations. Since the images are *identical* apart from the additions of the digitally inserted masks, it is unsurprising that these models perform almost too well. In reality, the data would never be this perfect: the images would not all be identical, faces would move, expressions would change, lighting would differ, etc. All of these factors will cause a considerable and realistic degradation in performance. Despite data limitations, the results are still able to illustrate the concept that this holistic algorithm performs better being trained with less facial data and identifying samples with more than it does being trained with more data than it will have in the test samples.

D. Design of Experiment C

The purpose of Experiment C was to perform an enhancement to the Eigenfaces algorithm to help it better perform on masked faces. Examination of the results from Experiments A & B in Figure 10 clearly shows that training the Eigenfaces algorithm with masked data then testing with unmasked data is not as accurate or robust as training the algorithm with the faces already masked then identifying the same face unmasked.

Following experiments A and B, the hypothesis was made that reducing the portion of the images that is used for training the model to only include the parts of the face that will be available both with and without the mask would improve accuracy in the case of Experiment A. This theory is further supported by prior research into facial occlusion that showed removal of occluded areas of the face from the recognition process improved performance in many cases.

To test this hypothesis, the code was updated to always train two models: one for identification in masks and one for identification without. A flag was added in the code for which algorithm model to pass the data through. This flag simulates the detection of the mask by a thrid-party software. The masked detector will be trained on facial images cropped as shown in Figure 11:

Fig. 11: Cropped Training Images

A possible training template/test sample combination is shown in Figure 12 and Figure 13 and a dimensioned depiction of the cropping rectangle that was used to create these images is detailed in Figure 14.

Using these dimensions, the previous training and test images can be cropped down to a new image that is still a continuous matrix but only includes areas of image data that will still be available when the face is masked. This experiment is performed with exactly the same training and test data fed to the code as in Experiment A: training data of 500 unmasked

Fig. 12: Training Sample

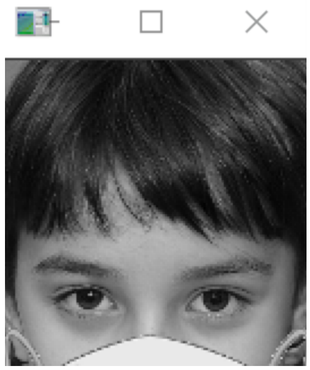

Fig. 13: Test Sample

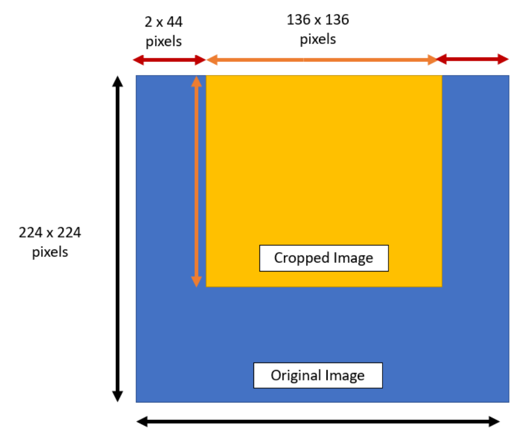

Fig. 14: Cropping Method

faces and test data of 1000 masked faces. The first 500 images of the test data have been trained for and the second 500 have not. The only difference for the user is setting the boolean to true for masked recognition instead of false. In this experiment the flag is now set that the data will have to be recognized as masked meaning that the mask-specific model that uses the cropped images is employed to perform the Eigenfaces FR.

E. Experiment C

Now when Eigenfaces is flagged for masked identification, it can use the model that trained itself on this 136×136 cropped set of images instead of the 224×224 set of images. The results seen from this test is shown in Table III.

Thus, it can be concluded from Table III that the maximum accuracy achieved by the mask-modified Eigenfaces algorithm was 99% at a threshold of 3000. For the exact same test data this improves upon the results of Experiment A by more than 10% by accuracy. Examining the other metrics shows that those were also only positively impacted by these modifications.

F. Analysis of Experiment C

It is logical that the threshold distance grows smaller from Experiment A to Experiment B and even smaller yet in Experiment C as there is less data and even less variety within the data to increase this distance. The performance of the model from Experiment C is added to the ROC curve from Figure 10 in Figure 15 as the green distribution of points.

Experiment C: Crop Unmasked Train, Crop Mask Test									
Dist.	**TP**	**FP**	**TN**	**FN**	**Acc.**	**FMR**	**FRR**	**Prec.**	**Rec.**
8000	500	500	0	0	0.50	1	0	0.50	1.0
7500	500	499	1	0	0.50	0.99	0	0.5	1.0
7000	500	499	1	0	0.501	0.998	0	0.5	1.0
6500	500	497	3	0	0.50	0.99	0.0	0.50	1.0
6000	500	490	10	0	0.51	0.98	0.0	0.51	1.0
5500	500	472	28	0	0.53	0.94	0.0	0.51	1.0
5000	500	418	82	0	0.58	0.84	0.0	0.55	1.0
4500	500	286	214	0	0.71	0.57	0.0	0.64	1.0
4000	500	109	391	0	0.89	0.22	0.0	0.82	1.0
3500	500	17	483	0	0.98	0.03	0.0	0.97	1.0
3000	498	0	500	2	0.99	0.0	0.01	1.0	0.99
2500	485	0	500	15	0.98	0.0	0.03	1.0	0.97
2000	431	0	500	69	0.93	0.0	0.14	1.0	0.86
1500	242	0	500	258	0.74	0.0	0.52	1.0	0.48
1000	45	0	500	455	0.55	0.0	0.91	1.0	0.09

Table III: Analysis of Experiment C results by varying distance threshold

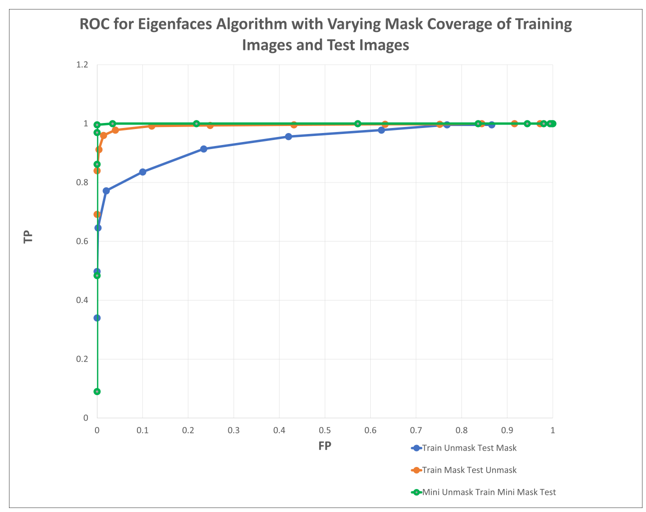

Fig. 15: ROC Curves for Eigenfaces Experiment A and Experiment B and Addition of Experiment C

The training and test data used in Experiment C is identical to that of Experiment A which is the blue distribution on Figure 15. Although these results are impossibly accurate due to the images being so similar—it is evident that the algorithmic modifications to how the model is trained and analysis is performed using cropped images made a difference in the Eigenfaces performance.

Although the data for these experiment is limited in variety and size, which is believed to be the cause of the results being as binary as they are, this study provides a foundation to the hypothesis that by knowing recognition in COVID masks may be necessary, a secondary model can be trained for holistic algorithms such as this for identification in COVID PPE. This mask-specific model can be trained for only the area of the face that will be available when a mask is being worn. Then using pre-existing technology such as Amazon Rekognition to detect masks, the mask-specialized model can be applied to try to improve FR performance.

X. Conclusion

Repercussions of the COVID-19 pandemic are widespread and continue to be felt in many areas of daily life. One major change of daily routine is the wearing of facial PPE to decrease the spread of the virus. This change in routine has a direct impact on the area of biometrics and FR Technologies.

As a result of COVID-19 it is predicted that not only will FR become more difficult due to occlusion by face masks, but it will also become more popular because of its ability to perform *contactless* identification.

This study shows that the holistic FR algorithm Eigenfaces performs better for FR when trained with masked data and used to recognize unmasked faces than it does when trained with unmasked data and used to recognize masked faces. Since masked data may not be as readily available for training, modifications were made to show that a second model could be trained using unmasked data but only trains using the areas of the face that will still be available when COVID facial PPE is present.

After training with this specialized model, a flag is set to inform the algorithm whether or not the test samples are wearing face masks. This causes Eigenfaces to perform recognition using the modified model. This model saw more than a 10% increase in maximum performance capabilities over the original algorithm using the same training and testing images as in Experiment A. The results of the modified algorithm also out-performed the results of Experiment B.

The results of this study are limited by lack of variation in the data. Further research should be done to further explore these results and add depth to the tests in terms of variety in images, lighting, expression, and even changes in mask color or coverage.

References

[1] A. K. H. Patil, M. N. AF. Abate, and L. S. R. Everson, "3D geometry-based face recognition in presence of eye and mouth occlusions," International Journal on Interactive Design and Manufacturing (IJIDeM), 01-May-2019. [Online]. Available: https://link.springer.com/article/10.1007/s12008-019-00582-7. [Accessed: 30-Jan-2021].

[2] CV, Open. "Text UnderScore Introduction." Text Underscore Introduction, Mar. 2021, docs.opencv.org/3.4/da/d60/tutorial_face.html.

[3] CV, Open. Text Underscore Introduction, Mar. 2021, docs.opencv.org/3.4/d1/dee/tutorial.html.

[4] H. Drira, B. Ben Amor, A. Srivastava, M. Daoudi and R. Slama, "3D Face Recognition under Expressions, Occlusions, and Pose Variations," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 9, pp. 2270-2283, Sept. 2013, doi: 10.1109/TPAMI.2013.48. [Accessed: 30-Jan-2021].

[5] K. Bowyer and M. King, "Why face recognition accuracy varies due to race," Biometric Technology Today, 16-Sep-2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0969476519301146. [Accessed: 30-Jan-2021].

[6] Mallick, Satya. "Eigenface Using OpenCV (C++/Python): Learn OpenCV." Learn OpenCV — OpenCV, PyTorch, Keras, Tensorflow Examples and Tutorials, 29 Jan. 2018, learnopencv.com/eigenface-using-opencv-c-python/.

[7] Mohammadzade, Hoda, and Dimitrios Hatzinakos. "Iterative Closest Normal Point for 3D Face Recognition." IEEE Xplore, May 2012, ieeexplore.ieee.org/document/6200285.

[8] R. Udas, "Post-lockdown: Will Facial Recognition Technology Replace Biometrics?" Express Computer, 2020. Available: https://libezproxy-syr-edu.libezproxy2.syr.edu/login?url=https://www-proquest-com.libezproxy2.syr.edu/trade-journals/post-lockdown-will-facial-recognition-technology/docview/2405330912/se-2?accountid=14214.

[9] "Some L.A. County residents should wear masks at home to slow virus spread, officials say," Los Angeles Times, 12-Jan-2021. [Online]. Available: https://www.latimes.com/california/story/2021-01-12/wear-covid-masks-at-home-los-angeles-county-officials-say. [Accessed: 30-Jan-2021].

[10] Tim Ring, "NIST finds flaws in facial checks on people with Covid masks," Biometric Technology Today, 16-Sep-2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0969476520301016. [Accessed: 30-Jan-2021].

[11] V. P. Kshirsagar, M. R. Baviskar and M. E. Gaikwad, "Face recognition using Eigenfaces," 2011 3rd International Conference on Computer Research and Development, Shanghai, China, 2011, pp. 302-306. doi: 10.1109/ICCRD.2011.5764137

[12] Wenyi Zhao, Rama Chellappa. "Face Processing: Advanced Modeling and Methods," Academic Press, 2006. [Online]. Available: http://search.ebscohost.com.libezproxy2.syr.edu/login.aspx?direct=true db=e000xnaAN=196100site=ehost-live [Accessed: 10-Feb-2021].