Project 1
Group #1

Introduction

A. Understanding Biometrics

The study of biometrics has been an ongoing topic that continues to expand as the importance of more secure applications are discussed. From government offices to simply identifying oneself as a citizen, biometric recognition is critical in many applications to ensure a secure biometric system. Biometrics is the science of setting up the identity of an individual based on physical characteristics, behavioral characteristics, or both of that person either in a fully automated or semi-automated manner. When considering the aspects of biometric recognition, there are knowledge-based, or token-based attributes to authenticating a user. Each of these has their own positives and negatives, but they are the basis for determining how a user is to be authenticated by a biometric system. Biometric recognition is important for ensuring a reliable and natural solution to recognizing a person in a system. The person who presents their biometric identifier to a biometric system to be recognized is called a user of the system. The user must be present at the time of the authentication, which also helps to prevent imposters from accessing the system. Biometrics can also establish whether a user is already known within the system or not. The biometric system itself measures one or more physical or behavioral characteristics, such as face, fingerprint, iris, voice, signature, gait, palmprint, retina, or DNA. Each of these pieces of information can be used to verify someone's identity.

B. The Biometric System

The biometric system exists to identify a user based on the physical characteristics, behavioral characteristics, or both. There are two main phases in the process of the system, enrollment and recognition. In the enrollment phase, biometric data from the user seeking to enroll themselves in the system is obtained and stored within a database along with their identity. During the recognition phase, the user becomes the query instead of simply enrolling. To authenticate themselves with the system, biometric data is re-acquired from the individual and compared against the data stored in the database to determine the user's identity. This part is referred to as matching. The decision determined from the matching process will inform the user if they are authenticated with the system or not. In general, the biometric systems consists or a sensor, feature extractor, database, and a matcher.

C. Face Recognition

Face is the most commonly used biometric traits used in the biometric research area. A human face exposes a great deal of information for perceivers. An individual's mood, attentiveness, and intention are seen at face, and it also serves to identify the person. There are additional means to identify a person than face. For instance, gait, body shape, and voice may all help in identifying persons when facial information is not available.

Face recognition involves the matching between the structural coding and previously stored data. The face recognition helps in deciding whether the initial matching is sufficient and close to accurate recognition, or it is merely a resemblance. Among research topics, face recognition is one of the active topics in the area of computer vision. It is because various face recognition techniques perform well in a controlled environment. However, these techniques suffer when variation is observed among factors such as illumination and pose. Therefore, research is on the way to increase the robustness of face recognition techniques by eliminating the effects of influencing factors.

Face recognition starts matching between detected face and face ID stored in a database. Between detected face and stored information, an algorithm works that converts face features into machine readable format.

Once the face is recognized, facial recognition algorithm executes to identify the certain points of the face i.e. spot between pupils. Then the algorithm uses the measured points and creates a template or pattern of a face. Then the newly created template or pattern is compared with others already stored in the database.

Principal component analysis (PCA) is one of these techniques that computes the reduced set of factors. The PCA technique serves as a linear transformation from the space of the original image. Furthermore, local binary patterns (LBP) is a crucial performing technique in the area of face recognition. Since a face is composed of micro-patterns and LBP is the most appropriate to analyze them. In combination with LBP, PCA is applied to reduce the size of the vector. From a large set of variables, PCA extracts the most important variables to examine the information exactly.

In comparison with the other popular biometric techniques, including iris, retina, and fingerprint recognition, face recognition has the potential to be used in surveillance security, digital entertainment, and forensics.

Methods

There are various matching methods for fingerprint authentication, all of which rely on some kind of algorithm for matching minutiae. The following are just a few:

K Nearest Neighbors, a classification algorithm that utilizes other ‘close’ examples in the data set (neighbors) to assign a classifier. The number, k, must be odd in order to prevent ties.

- Benefits: it’s simple and easy to implement. Allows for system updating with each new query (can add to the example pool with each correct classification). The more it is used, the better it becomes with classifying.

- Cost: Choosing a value of k that is too large or too small may result in inaccurate results due to the search exceeding the limitations of the example pool (too many neighbors selected in the given neighborhood). Could also be vulnerable to overfitting/underfitting.

There are other variations of the K Nearest Neighbors algorithm, such as the Condensed Nearest Neighbors algorithm. The CNN or Hart algorithm uses prototyping to condense and reduce the data set which helps with vulnerabilities to over/underfitting.

Normalized Cross Correlation (NCC) is a method that breaks a finger print down into smaller mosaic images of partial prints. It enhances the images with a thinning algorithm, before running a Phase-Only Correlation (POC) to find rotational and transformative differences between query and template. The query image is then superimposed over the template for comparison.

The purpose of a biometric matcher is to contrast the query features against the templates in order to generate matching scores. A matching score measures the similarity between a query and a stored template. The greater similarities between the query and template are, the higher a matching score is. A matcher can also measure the dissimilarity between features. This measurement is referred to as the distance score. Thus, if a score shows a small distance score, this indicates that two features have a high matching score.

The matcher module encapsulates a decision making module, were the match scores are used to authenticate a claimed identity or provide a ranking of the enrolled identities in order to identify an individual.
The fingerprint matcher performs template matching in a one-to-one comparison between a query and a claimed template, of a one-to-many comparisons between a query and all templates. The processes are used for the verification and identification of an individual.

Naive Bayes methods are a set of supervised and effective machine learning algorithms. It is a probability classifier that is based on the Bayes theorem where the probability of an event is measured by previous knowledge about items, elements, or characteristics that may lead to that event.

System Architectures

There are two system architectures used in this face recognition system which are brightness image enhancement and contrast image enhancement.

The most important factor for a face recognition system to recognize, verify or identify a person easily is using images that have a proper level of brightness and contrast. According to Olympus, contrast is the amount of color of grayscale differentiation that exists between various image features in both analog and digital images [2]. Images having a higher contrast level generally display a greater degree of color or grayscale variation than those of lower contrast. Image brightness (or luminous brightness) is a measure of intensity after the image has been acquired with a digital camera or digitized by an analog-to-digital converter [2].

In the given data set, there are images that were taken in the dark. Therefore, we used the image enhancement library PIL (short for Pillow (PIL Fork)). Out of four image enhancement classes (Sharpness, Color, Brightness and Contrast). Our team chose to enhance the brightness and contrast of the images to compare the performances of each of the enhancements to the original images. For both enhancement classes, they use a single common method containing enhance(*factor*) and it returns an enhanced image. The contrast class is used to adjust the level of contrast of an image. A factor of 0.0 gives a solid grey image. A factor of 1.0 gives the original image. Similarly, the brightness class can also be used to adjust the overall brightness for an image. A factor of 0.0 gives a black image. A factor of 1.0 also gives the original image [3]. In our image enhancement implementation
(enhance_images.py), the brightness factor is set 2.0. Figure 1 showed that the brightness was doubled in the comparison between the before and after images. Furthermore, the contrast factor is set to 0.5, it was reduced by half resulted in a blurry after image compares to the original photo, as shown in Figure 2.

The database we used for this project contains 5 main subjects. There are 114 images for subject 1, 97 images for subject 2, 98 images for subject 3, 92 images for subject 4 and 99 for subject 5. They were generated from tasks our team performed.

The process of enhancing the images: First step is to set different cases for each of the enhancement classes for performance comparison purposes, such as case 1 is for the brightness enhancement and case 2 is for contrasting the images. Secondly, the code will go through each of the data folder (each team member's face data which totals 500 images among all the folders); it will also run the PIL library to enhance the images based on the case selection and then make a new folder for each of the enhancement class for each of the team member such as Group1_FaceData_EnhancedBrightness and Group1_FaceData_EnhancedContrast.

The goal is to compare the performance of enhanced images to the performance of original images. In addition, analyze how enhancing the images can improve the results of the original images from the performed tasks.

Figure 1: Comparison of images before and after brightness enhancement.

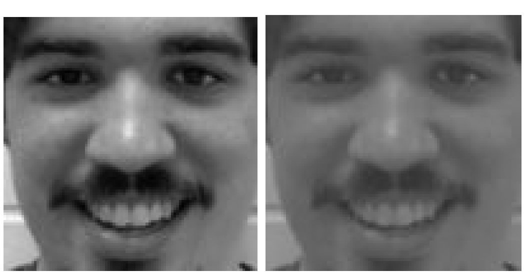

Figure 2: Comparison of images before and after contrast enhancement.

Results

The results were determined by first identifying the performance of the original images by extracting their features with either local binary pattern (LBP) or principal component analysis (PCA). Once the features were obtained, a matcher was used to get all the genuine and imposter scores. These scores were used to generate the score distributions, ROC curves, and DET curves.

To define the architectures used, the original images were compared with enhanced versions of those images to record any differences in performance that existed between the score distributions, ROC curves, and DET curves for both PCA and LBP, as well as with the KNN and Naive Bayes (NB) matchers. The two primary additional architectures were to brighten the images and add sharpness to the images. The KNN matcher was classified with 50 neighbors and manhattan distance as the metric.

A. Original Results

The original images were tested by first storing the images and their labels into two numpy arrays. The images were passed through either LBP and PCA to get the features of the images. Next, a matcher for KNN or Naive Bayes was used to determine the number of genuine and imposter scores. Afterwards, the performance results could be determined on the score distribution, ROC curve, and DET curve. The results will be numbered from (1 - 4), indicating the test performed to be used for comparison with the enhanced images.

1. When running LBP with the KNN matcher on the images the following figures show the results.

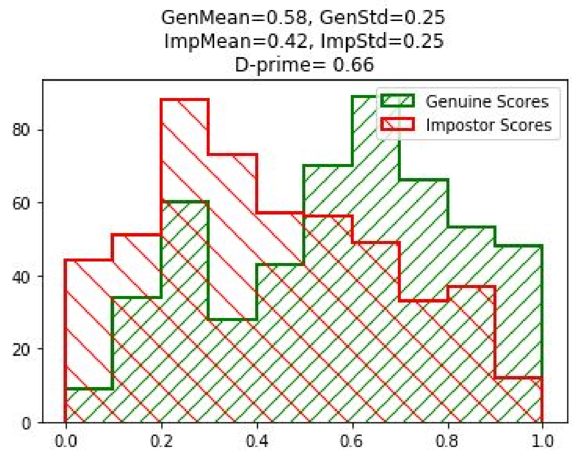

Figure 3: Score distribution of original images using LBP and KNN matcher.

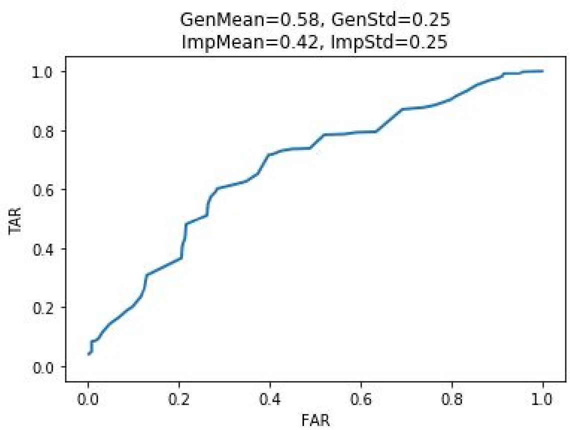

Figure 4: ROC curve of original images using LBP and KNN matcher.

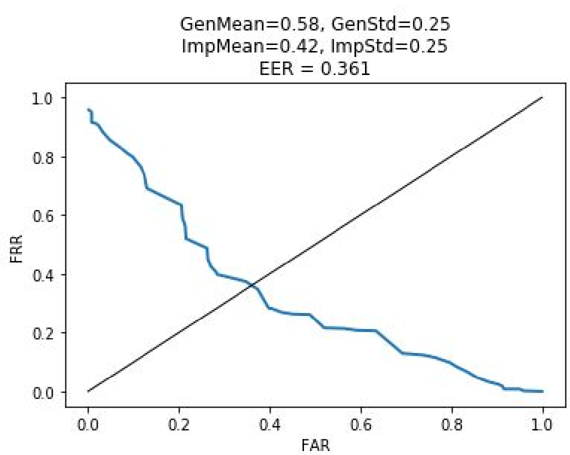

Figure 5: DET curve of original images using LBP and KNN matcher.

2. When running LBP with the NB matcher on the images the following figures show the results.

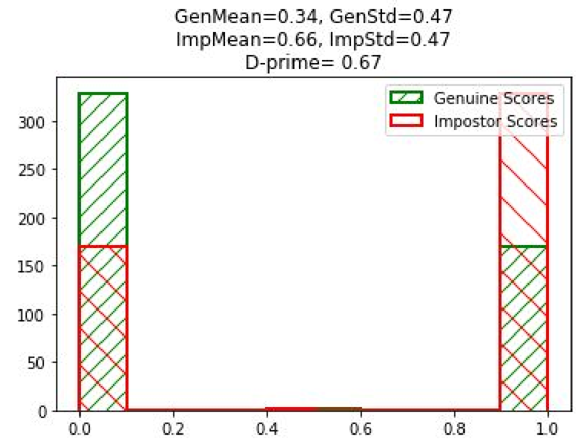

Figure 6: Score distribution of original images using LBP and NB matcher.

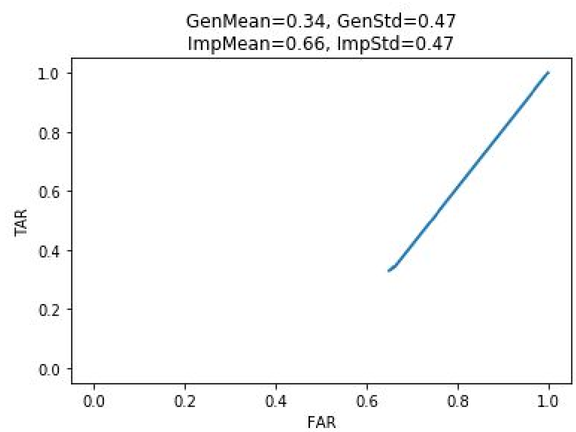

Figure 7: Score distribution of original images using LBP and NB matcher.

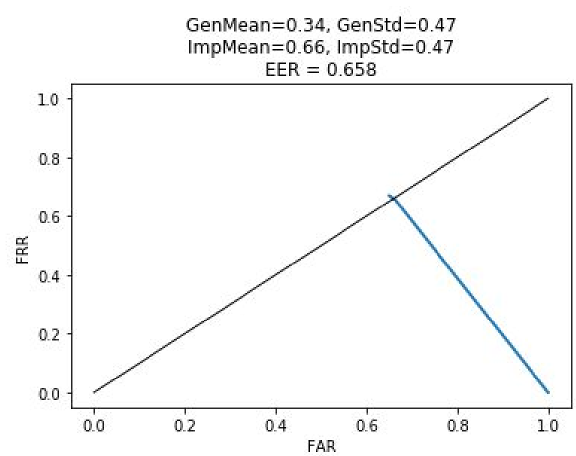

Figure 8: Score distribution of original images using LBP and KNN matcher.

3. When running PCA with the KNN matcher on the images the following figures show the results.

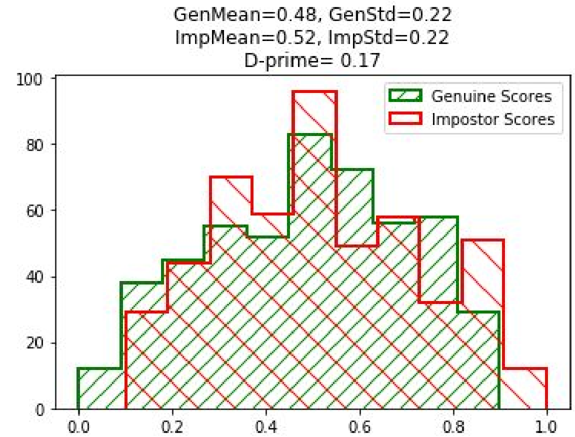

Figure 9: Score distribution of original images using PCA and KNN matcher.

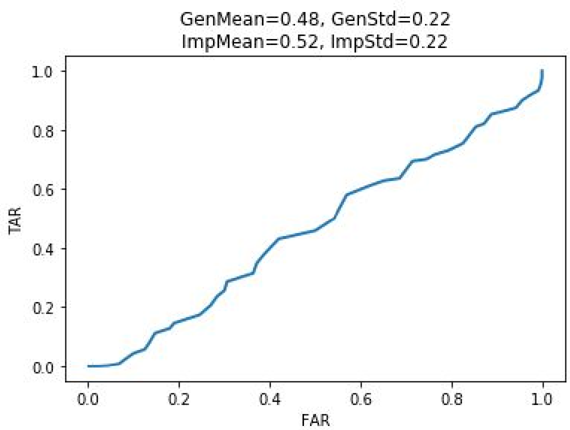

Figure 10: ROC curve of original images using PCA and KNN matcher.

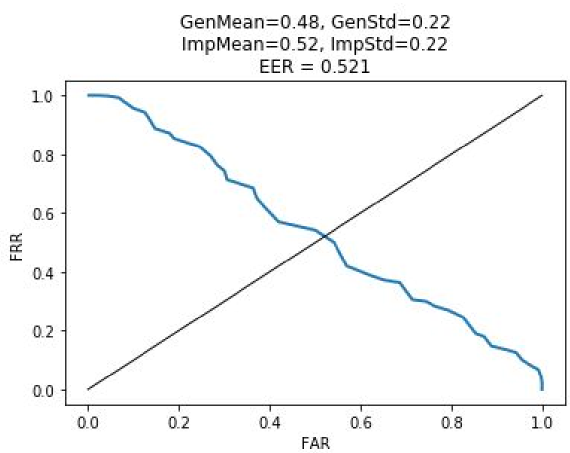

Figure 11: DET curve of original images using PCA and KNN matcher.

4. When running PCA with the NB matcher on the images the following figures show the results.

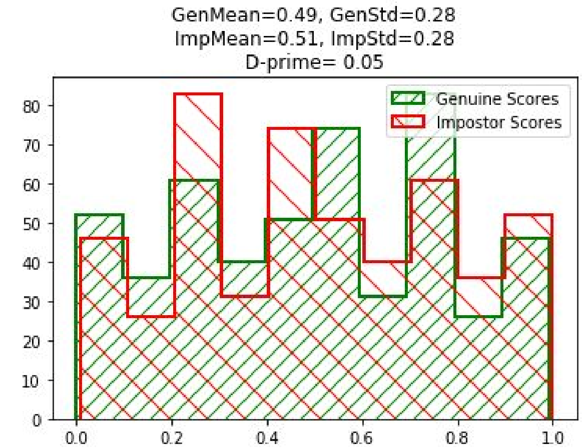

Figure 12: Score distribution of original images using PCA and NB matcher.

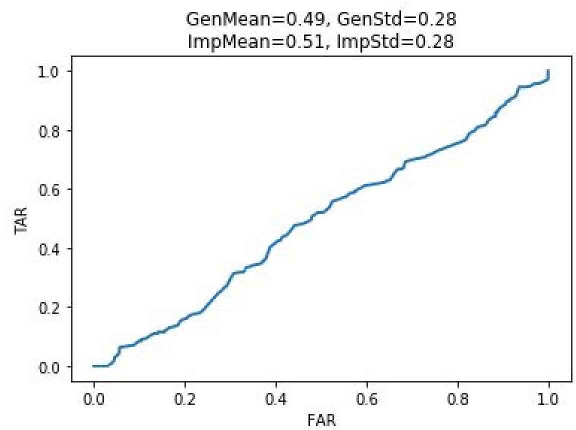

Figure 13: ROC curve of original images using PCA and NB matcher.

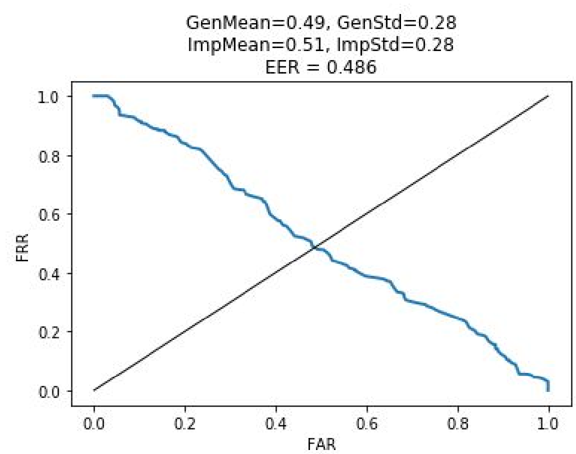

Figure 14: DET curve of original images using PCA and NB matcher.

Now, the comparison will be drawn with the two additional architectures, both being a form of image enhancement.

B. Architecture 1 - Brightness Enhancement

The first image enhancement was brightening the images by doubling the brightness value on the image. It is possible to do this, because images can be enhanced with Python PIL [1].

1. When running LBP with the KNN matcher on the images the following figures show the results.

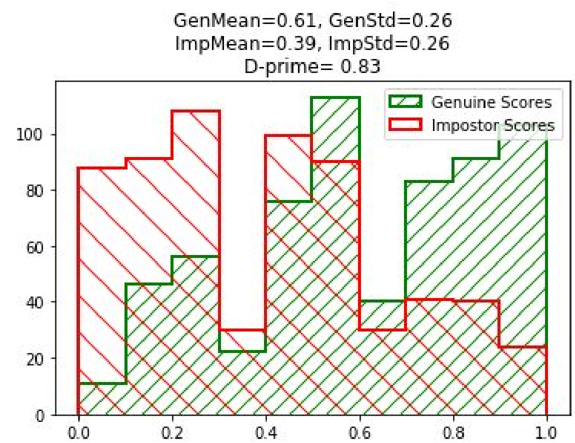

Figure 15: Score distribution of brightened images using LBP and KNN matcher.

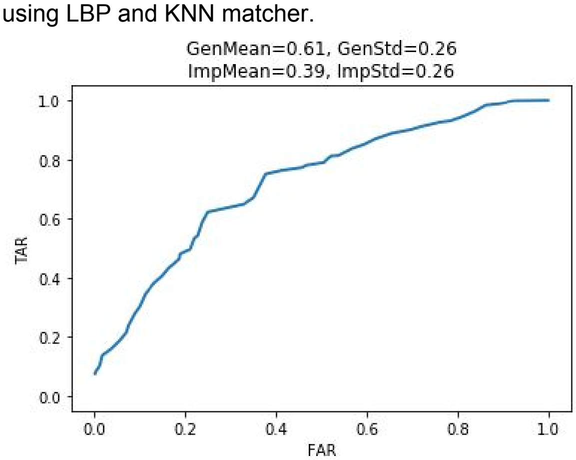

Figure 16: ROC curve of brightened images using LBP and KNN matcher.

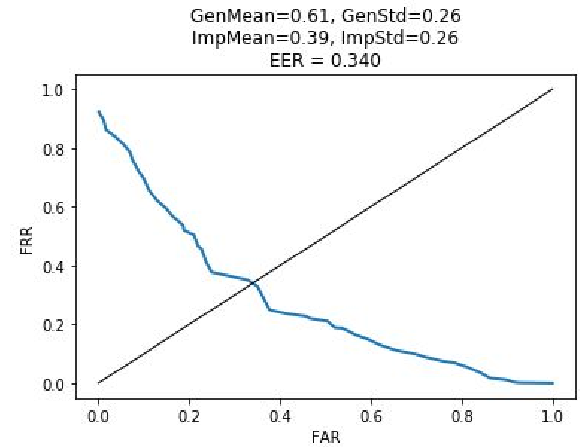

Figure 17: DET curve of brightened images using LBP and KNN matcher.

2. When running LBP with the NB matcher on the images the following figures show the results.

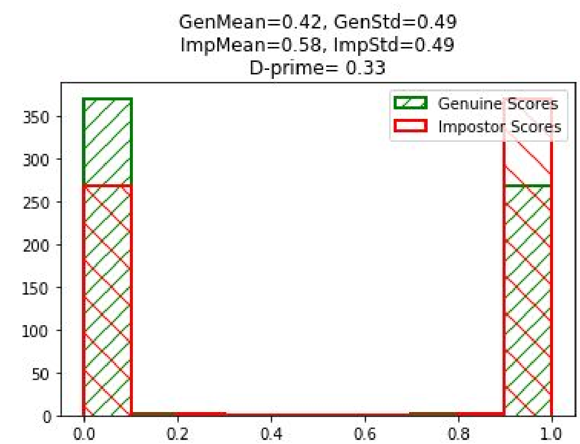

Figure 18: Score distribution curve of brightened images using LBP and NB matcher.

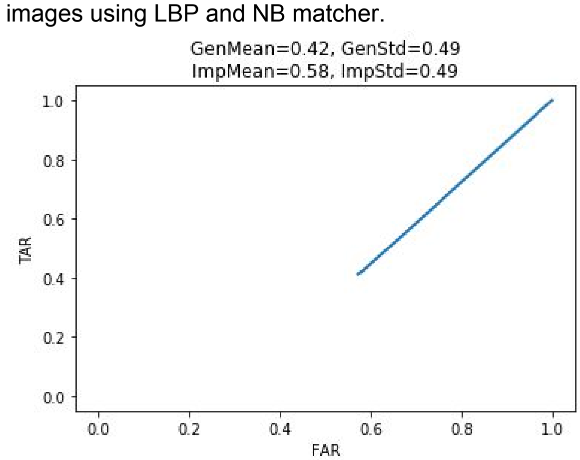

Figure 19: ROC curve of brightened images using LBP and NB matcher.

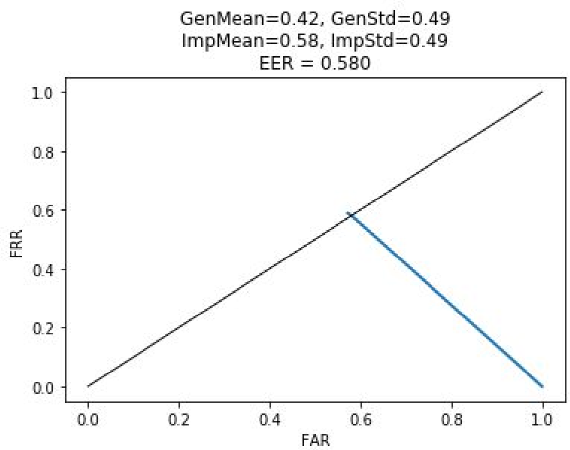

Figure 20: DET curve of brightened images using LBP and NB matcher.

3. When running PCA with the KNN matcher on the images the following figures show the results.

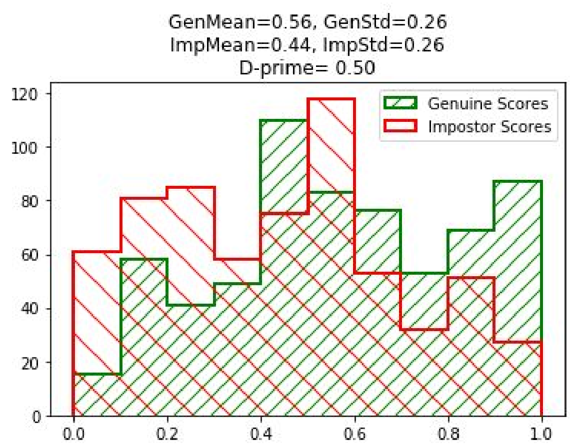

Figure 21: Score distribution of brightened images using PCA and KNN matcher.

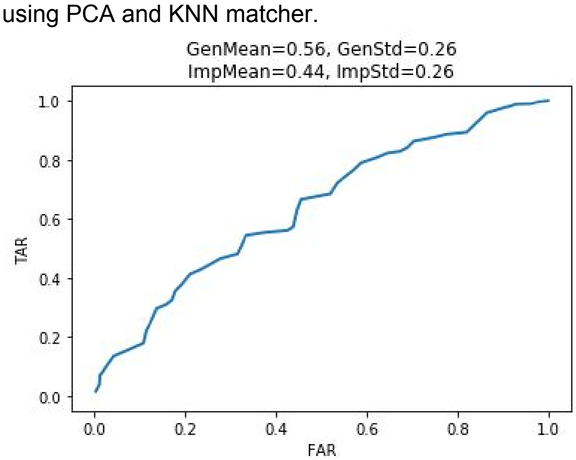

Figure 22: ROC curve of brightened images using PCA and KNN matcher.

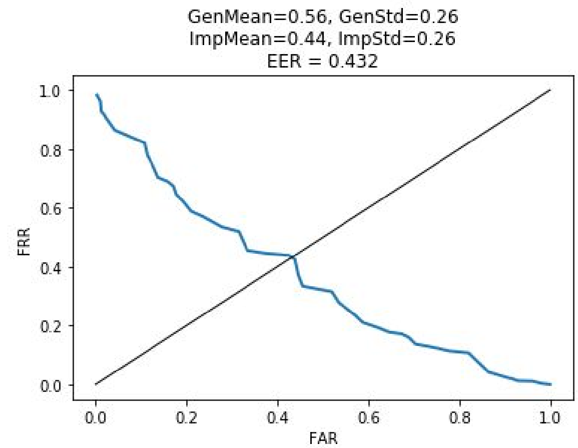

Figure 23: DET curve of brightened images using PCA and KNN matcher.

4. When running PCA with the NB matcher on the images the following figures show the results.

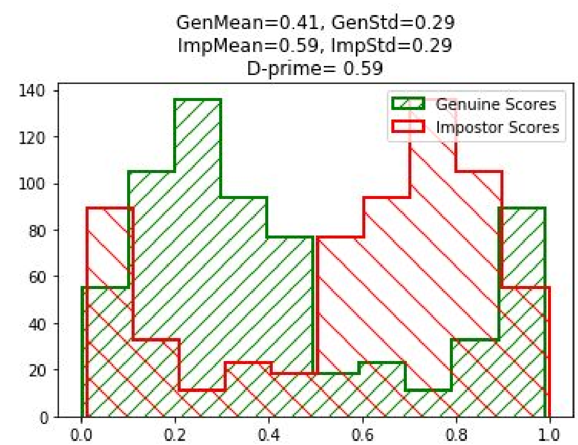

Figure 24: Score distribution of brightened images using PCA and NB matcher.

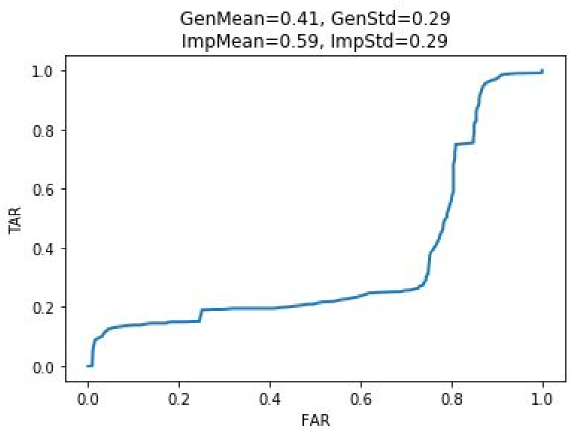

Figure 25: ROC curve of brightened images using PCA and NB matcher.

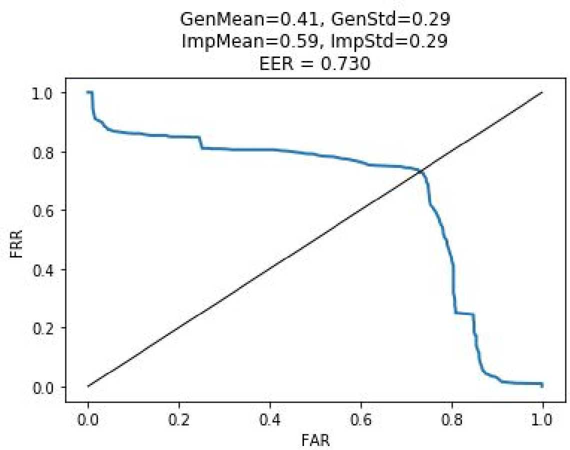

Figure 26: DET curve of brightened images using PCA and NB matcher.

	Original Image	Brightened Image
Test 1 (LBP w/ KNN)	d-prime = **0.660** EER = **0.361**	d-prime = **0.830** EER = **0.340**
Test 2 (LBP w/ NB)	d-prime = **0.670** EER = **0.658**	d-prime = **0.330** EER = **0.580**
Test 3 (PCA w/ KNN)	d-prime = **0.170** EER = **0.521**	d-prime = **0.500** EER = **0.432**
Test 4 (PCA w/ NB)	d-prime = **0.050** EER = **0.486**	d-prime = **0.590** EER = **0.730**

Table 1: Performance results of brightened images versus original images.

Using the results from Table 1, for test 1, the original images obtained a d-prime value of 0.660 on the score distribution and an EER of 0.361, while the brightened images obtained a d-prime value of 0.830 and an EER of 0.340. This shows that for LBP with the KNN matcher, the brightened images had a better performance than the original images.

For test 2, the original images obtained a d-prime value of 0.670 on the score distribution and an EER of 0.658, while the brightened images obtained a d-prime value of 0.330 and EER of 0.580. This shows that for LBP with the NB matcher, the original images had a better performance than the brightened images.

For test 3, the original images obtained a d-prime value of 0.170 on the score distribution and an EER of 0.521, while the brightened images obtained a d-prime value of 0.500 and EER of 0.432. This shows that for PCA with the KNN matcher, the brightened images had a better performance than the original images.

For test 4, the original images obtained a d-prime value of 0.050 on the score distribution and an EER of 0.486, while the brightened images obtained a d-prime value of 0.59 and an EER of 0.730. This shows that for PCA with the NB matcher, the brightened images had a better performance than the original images.

Overall, by brightening the images, a better performance can be expected.

C. Architecture 2 - Contrast Enhancement

The second image enhancement was contrasting the images by reducing the contrast value by 0.5.

1. When running LBP with the KNN matcher on the images the following figures show the results.

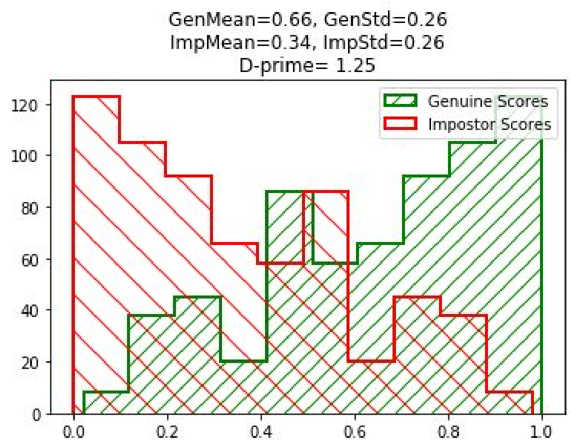

Figure 27: Score distribution of contrasted images using LBP and KNN matcher.

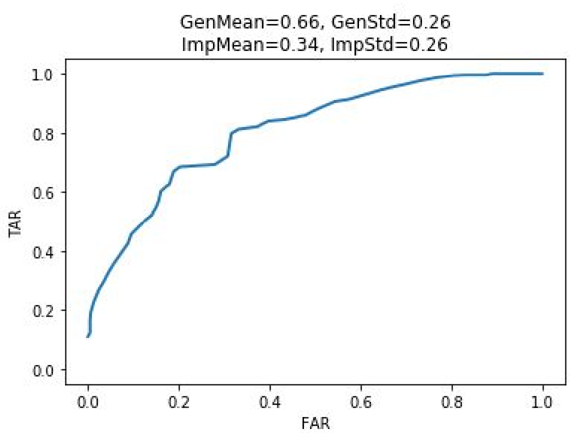

Figure 28: ROC curve of contrasted images using LBP and KNN matcher.

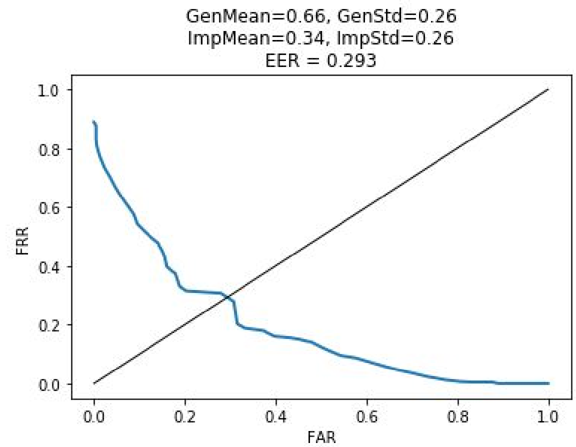

Figure 29: DET curve of contrasted images using LBP and KNN matcher.

2. When running LBP with the NB matcher on the images the following figures show the results.

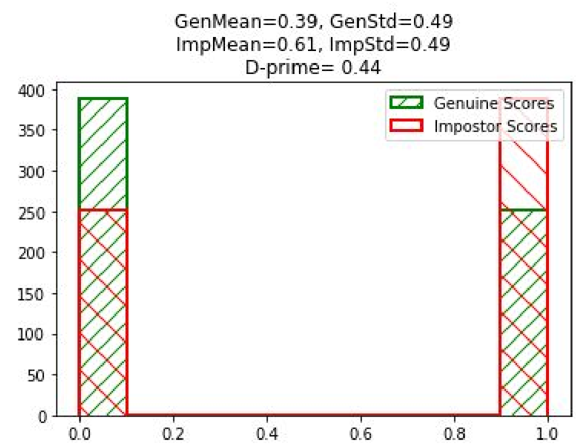

Figure 30: Score Distribution of contrasted images using LBP and NB matcher.

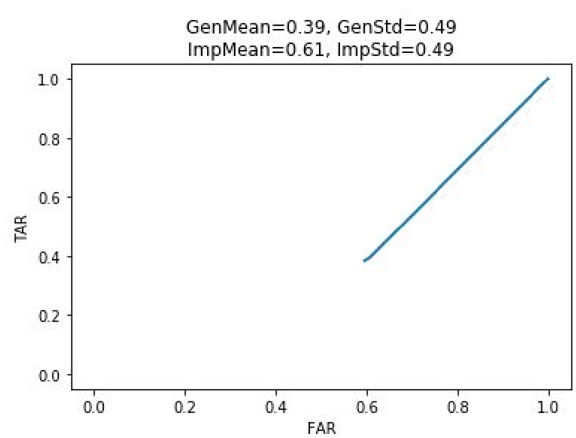

Figure 31: ROC curve of contrasted images using LBP and NB matcher.

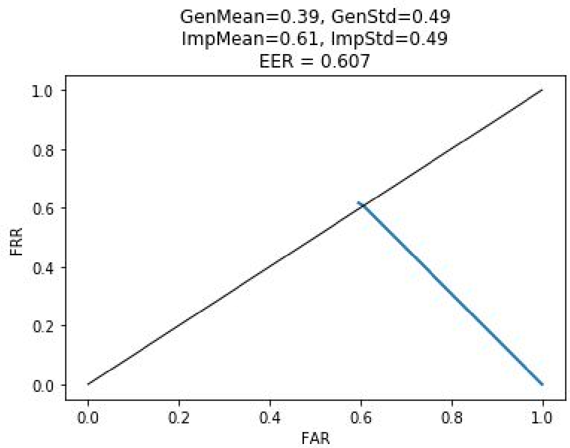

Figure 32: DET curve of contrasted images using LBP and NB matcher.

3. When running PCA with the KNN matcher on the images the following figures show the results.

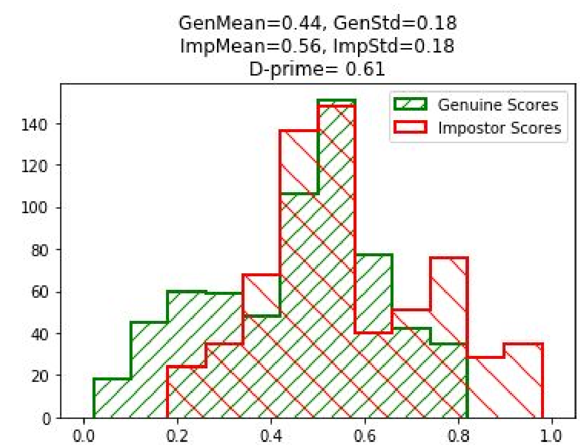

Figure 33: Score distribution of contrasted images using PCA and KNN matcher.

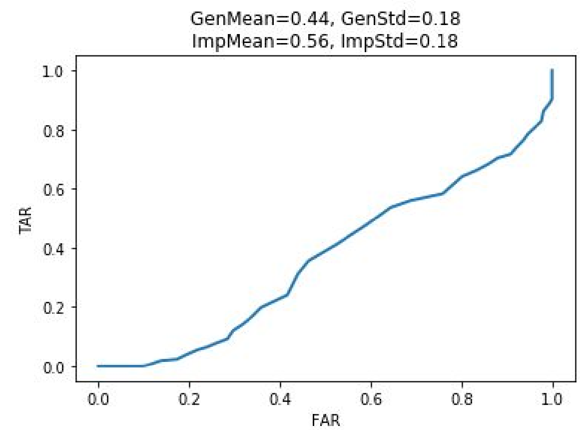

Figure 34: ROC curve of contrasted images using PCA and KNN matcher.

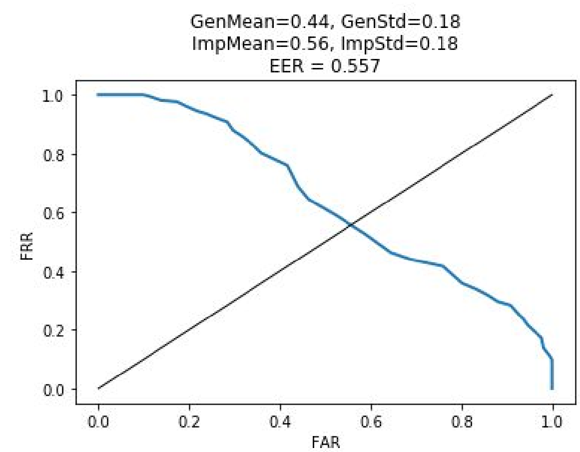

Figure 35: DET curve of contrasted images using PCA and KNN matcher.

4. When running PCA with the NB matcher on the images the following figures show the results.

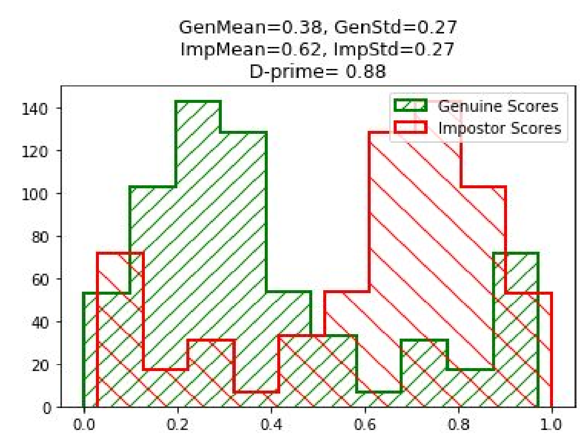

Figure 36: Score distribution of contrasted images using PCA and NB matcher.

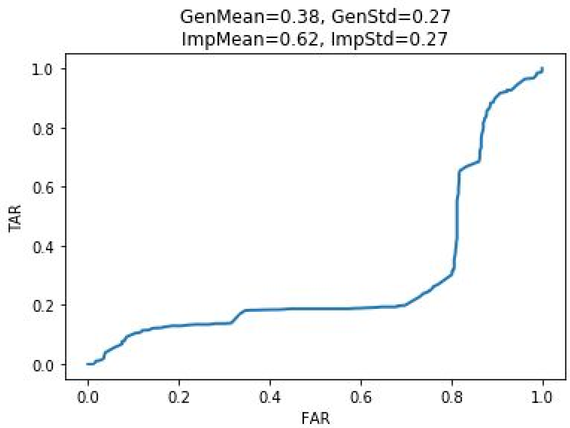

Figure 37: ROC curve of contrasted images using PCA and NB matcher.

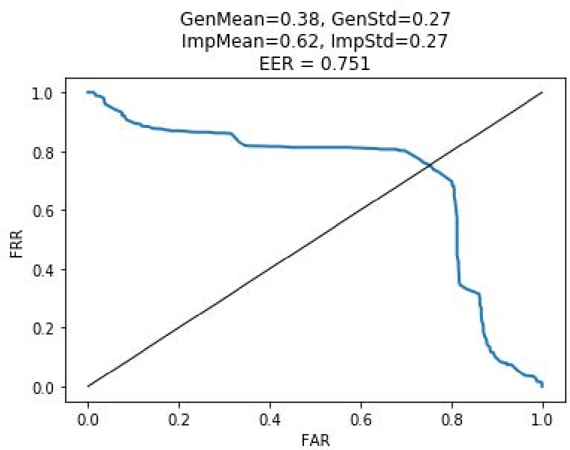

Figure 38: DET curve of contrasted images using PCA and NB matcher.

	Original Image	Contrasted Image
Test 1 (LBP w/ KNN)	d-prime = **0.660** EER = **0.361**	d-prime = **1.25** EER = **0.293**
Test 2 (LBP w/ PCA)	d-prime = **0.670** EER = **0.658**	d-prime = **0.440** EER = **0.607**
Test 3 (PCA w/ KNN)	d-prime = **0.170** EER = **0.521**	d-prime = **0.610** EER = **0.557**
Test 4 (PCA w/ NB)	d-prime = **0.050** EER = **0.486**	d-prime = **0.880** EER = **0.751**

Table 2: Performance results of contrasted images versus original images.

Using the results from Table 2, for test 1, the original images obtained a d-prime value of 0.660 on the score distribution and an EER of 0.361, while the contrasted images obtained a d-prime value of 1.25 and an EER of 0.293. This shows that for LBP with the KNN matcher, the contrasted images had a better performance than the original images.

For test 2, the original images obtained a d-prime value of 0.670 on the score distribution and an EER of 0.658, while the contrasted images obtained a d-prime value of 0.440 and an EER of 0.607. This shows that for LBP with the NB matcher, the original images had a better performance than the contrasted images.

For test 3, the original images obtained a d-prime value of 0.170 on the score distribution and an EER of 0.521, while the contrasted images obtained a d-prime value of 0.610 and an EER of 0.557. This shows that for PCA with the KNN matcher, the contrasted images had a better performance than the original images.

For test 4, the original images obtained a d-prime value of 0.050 on the score distribution and an EER of 0.486, while the contrasted images obtained a d-prime value of 0.880 and an EER of 0.751. This shows that for PCA with the NB matcher, the contrasted images had a better performance than the original images.

The images that were hardest to classify were the images that had been processed as Failure to Capture (FTC). These images had the greatest impact on the data in regards to performance.

Conclusions

In conclusion, the study of Biometrics is the science of determining the identity of an individual based on physical characteristics, behavioral characteristics, or both. The biometric system exists to identify a user based on the physical characteristics, behavioral characteristics, or both. Within the biometric system, enrollment and recognition are two main phases in the process of the system.

Face recognition is a process that involves matching between the structural coding and previously stored data of the face presented to the biometric system. There are various methods by which facial extraction is performed to retrieve the features of the face. Acquiring these features can allow for generating the genuine and imposter scores from matching classifiers to generate performance metrics.

The two primary architectures used in the experimentation were brightening the original images and contrasting the original images. From the tests performed, we learned that each of these enhancements proved to have better performance than the original counterpart in terms of d-prime from the score distribution and the EER from the DET curve. One consistent result that was in favor of the original images were from test 2. Running LBP with the NB matcher on the images always had better performance with the original images than the enhanced ones. We also learned that it is important to plan a consistent strategy for testing each of these performance metrics. Being that each image can have its features extracted using LBP or PCA, then run through either the KNN or NB matcher, it was important to organize the code to ensure we recorded the desired results accurately. Moreover, we learned that in terms of extracting features, LBP with either the KNN or NB matcher had a better performance than PCA with either the KNN or NB matchers.

As it relates to the system, it was easy to operate with the original images to get the images from the folders and extract the features using PCA and LBP. Furthermore, running the matchers and determining the performance were not difficult. However, it was hard for the system to enhance the images and then process them in the same manner as with the original images. With the performance being overall better, it took the system much longer to process the images to extract features and run the matchers to see their performance.

In the future, there are many modifications we would like to make to the system to see how they affect the performance of the system. We could determine the performance of specific tasks instead of using the entire set of images. It is likely that removing the FTC images will improve the performance in the score distributions, ROC curves, and DET curves. We would also be interested in expanding our image set with different facial images to determine how that will affect the performance of the system. Other tests we could attempt are sharpening the images to see the performance compared to the original images. Combining every enhancement into the images would provide unique understanding of the changes in performance, as well. There is also the idea of incorporating score fusion into the project by taking the mean of the genuine and imposter scores for both matchers to determine how that affects our performance results.

The limitations to the approach we decided to do, is that by using the entire image set, we could expect the results to be less defined instead of using a specific task's images.

References

[1] Petercour. (1970, July 15). Enhance image with Python PIL. Retrieved from https://dev.to/petercour/enhance-image-with-python-pil-222e.

[2] Brightness and Contrast in Digital Images. (n.d.). Retrieved from http://olympus.magnet.fsu.edu/primer/java/olympusmicd/digitalimaging/contrast/index.html.

[3] ImageEnhanceModule¶. (n.d.).Retrieved from https://pillow.readthedocs.io/en/3.0.x/reference/ImageEnhance.html.