Enhancing Facial Recognition Using Advanced Deep Learning Techniques

Abstract

Facial recognition systems have gained considerable traction in various applications, including security, authentication, and human-computer interaction. This paper meticulously outlines the design and implementation of a facial recognition system that seamlessly integrates face detection and recognition. The system employs Haar cascades for precise face detection, followed by Eigenfaces and Principal Component Analysis (PCA), for dimensionality reduction and a Support Vector Machine (SVM) for robust classification. Extensive preprocessing techniques ensure quality inputs, and a comprehensive evaluation strategy guarantees reliable performance. The design's strength lies in its flexibility, scalability, and attention to detail, promising a significant contribution to the evolving field of facial recognition. Extensive testing on the system has been performed, a variety of dataset compositions, the inclusion or absence of face detection, and whether SMOTE resampling provides an advantage in the system have been considered and tested. The system shall be able to verify a person’s face against complete strangers to the system, or, put simply, “You” vs “Not You.”

**Keywords**: Facial Recognition, Face Detection, Haar Cascades, Eigenfaces, Principal Component Analysis (PCA), Support Vector Machine (SVM), Image Preprocessing, Dimensionality Reduction, Machine Learning, Classification, Synthetic Minority Oversampling Technique (SMOTE) Labeled Faces in The Wild (LFW)

# Introduction

Facial recognition has become a cornerstone technology in modern society, with applications ranging from security and surveillance to personalized user experiences. The intersection of machine learning, computer vision, and advanced algorithms allows for the development of systems that can recognize and interpret human faces with remarkable accuracy.

## Background

As facial recognition technology continues to advance, understanding its roots and evolutionary trajectory becomes vital. This section delves into the foundation of facial recognition, exploring both its technological underpinnings and the context in which it has emerged and thrived.

### Historical Context

Facial recognition, the computational and technological process of identifying or verifying an individual's identity using their face, has its roots in the mid-20th century, with the advent of computer science. Initially focusing on simple geometric measurements and patterns, the field has since advanced to utilize sophisticated machine learning and deep learning algorithms.

### Modern Applications

In today's rapidly advancing technological landscape, facial recognition has found extensive applications across various domains:

* **Security and Law Enforcement**: It aids authorities in criminal investigations, airport security checks, and border control.
* **User Authentication**: It provides a secure way for users to unlock devices or access restricted areas.
* **Personalized Marketing**: Businesses use this technology to offer customized experiences to customers by recognizing their preferences.
* **Healthcare**: Medical professionals are exploring facial recognition for patient identification and even potential diagnostic applications.

### Evolution of The Field

The growth of facial recognition has been heavily influenced by two key factors:

1. **Data Availability**: The proliferation of large, publicly available datasets has enabled researchers to train more complex models. In the case of this research, we have the Labeled Faces in the Wild dataset. A publicly available and easily obtainable dataset of labeled images of faces.
2. **Algorithm Advancements**: The shift from traditional image processing techniques to deep learning has revolutionized the field. Neural networks have been instrumental in automating feature extraction, contributing to the significant improvement in accuracy and efficiency [3].

## Objective and Scope

### Research Objectives

The main goal of this research is to design, implement, and evaluate a cutting-edge facial recognition system. The focus will be on integrating both traditional methods and advanced deep learning techniques to achieve a comprehensive understanding of facial features.

### Specific Goals:

1. **System Design**: To conceptualize a robust system that combines various algorithms for feature extraction and recognition.
2. **Algorithm Selection**: To explore different methodologies, including Eigenfaces, Convolutional Neural Networks (CNNs), and Support Vector Machines (SVMs), and select the most suitable combination.
3. **Algorithm Optimization**: Show comparison of results between different methods for preprocessing data, providing the algorithm features, and hyperparameter tuning, and make conclusions about the best configuration.
4. **Data Handling**: To gather, prepare, and augment a diverse dataset that represents various human faces with different expressions, angles, lighting conditions, and ethnic backgrounds.
5. **Training and Evaluation**: To implement an effective training strategy and conduct a thorough performance evaluation, encompassing aspects like accuracy, scalability, and real-time responsiveness.

### In Scope

The research will concentrate on creating a system that excels in face verification, where the objective is to confirm or deny the identity claim of an individual. This focus aligns with the vision articulated by Taigman et al. in DeepFace, striving to bridge the gap between machine and human-level performance in this task [1]. The research seeks to understand the limitations of the algorithm and how best to optimize the algorithm in the following ways:

1. **Impact of Data Composition:** In optimization testing, the composition of the data that will subsequently be used for training and testing will be controllable to some extent. The system has been tested with a variety of composition weights of the “You” vs “Not You.”
2. **Face Detection Usage:** This research tests whether inclusion of face detection during preprocessing makes a meaningful difference in performance. Face detection will be optionally toggled on and off in the algorithm. Whether the raw image or the image of the detected faces go through the system may have an impact on the PCA portion of the algorithm, thus impacting performance of the overall system.
3. **SMOTE Resampling:** Applying SMOTE Resampling to the system, data augmentation for the purpose of synthesizing more examples of the minority class in the data set, may have an impact on performance. This research aims to determine that impact.

### Out of Scope

The following items are out of scope for this research.

1. **Comparison of Algorithms:** The aforementioned “Algorithm Selection” will be based upon research into previous work in the field, this paper will not make assertions about which model performs better or are easier to work with in a practical sense.
2. I**mplementations Beyond Testing:** Regarding what is shown in this research,the trained and tested model will not be used for implementations beyond classifying the performance of the model. For example, face verification using a phone camera. This research is targeted at testing the model alone.

## Summary

The ever-growing field of facial recognition offers immense potential and poses unique challenges. This research aims to contribute meaningfully to the field by marrying traditional methods with modern deep learning techniques. In doing so, it hopes to provide insights into creating a system that is not only technically advanced but also cognizant of the broader societal and ethical implications. By thoroughly understanding the past work and meticulously designing the experiment, the research sets the stage for a promising exploration into the frontiers of facial recognition technology.

# Previous Work

## Traditional Approaches

Earlier approaches laid the groundwork for understanding facial features and their geometric and statistical relationships:

1. **Eigenfaces (Turk and Pentland)**: The Eigenfaces method introduced a way to represent faces using principal component analysis (PCA). By capturing the variance between facial images and representing them with a set of principal components or eigenfaces, this method allowed for efficient recognition. However, it struggled with variations in lighting and pose [4].
2. **Fisherfaces (Belhumeur et al.)**: Building on the Eigenfaces method, Fisherfaces utilized linear discriminant analysis (LDA) to maximize between-class variations while minimizing within-class variations. This enhanced discrimination between different faces, improving recognition accuracy [13].
3. **3D Models (Bowyer et al.)**: By employing 3D models, researchers attempted to address the limitations of 2D recognition, such as pose variations. This research led to more robust systems that could understand facial structures in three dimensions [14].

## Deep Learning Revolution

The introduction of deep learning methods has significantly advanced the field, leading to innovative solutions:

1. **DeepFace (Taigman et al.)**: DeepFace marked a breakthrough by using 3D face alignment and a nine-layer deep neural network. The 3D alignment corrected for pose, illumination, and expression variations, while the deep network learned a compact representation of faces. This approach dramatically reduced error rates in face verification [1].
2. **FaceNet (Schroff et al.)**: FaceNet extended facial recognition by directly learning a mapping from face images to a compact Euclidean space. Using a triplet loss function, FaceNet ensured that similar faces were closer in the embedded space, achieving impressive results on various benchmarks [2].
3. **Deep Face Alignment (Yang et al.)**: This study empirically evaluated several face alignment methods, showing that proper alignment can dramatically improve recognition accuracy. The authors demonstrated the effectiveness, in the form of an alignment sensitivity analysis, of both two-stage and cascaded methods for face alignment [6].
4. **Imagenet Classification (Krizhevsky et al.)**: The source introduces several innovations and techniques to improve the performance and efficiency of the CNN, such as using rectified linear units (ReLU) as activation functions, using dropout as a regularization method, using overlapping max-pooling layers, and using data augmentation to increase the size and diversity of the training set [8].
5. **Joint Identification-Verification (Sun et al.)**: This source is relevant for our project because it shows how deep learning can be used to learn effective face representations that are invariant to age, pose, expression, and illumination. The paper proposes a deep learning approach for face representation that uses both identification and verification signals as supervision. The paper introduces a novel network architecture that consists of two sub-networks: one for identification and one for verification. The two sub-networks share the same convolutional layers but have different fully connected layers [12].

## Other Relevant Works

1. **Gradient-based learning (LeCun et al.)**: The source provides a comprehensive overview of the gradient-based learning methods applied to document recognition, such as optical character recognition (OCR), handwritten digit recognition, and text categorization. This source also provides us with some useful references to other related works and datasets that we can explore further [7].

## Summary

The rich body of previous work in facial recognition spans a wide array of methods and considerations. From foundational geometric and statistical approaches to cutting-edge deep learning models, the evolution of the field reflects an ongoing dialogue between theory, technology, and ethics. This diverse landscape informs the current research, underscoring both the remarkable progress made and the opportunities for future innovation and refinement.

# Experiment Design

## Introduction

The field of facial recognition has witnessed tremendous advancements, yet building a system that is robust, efficient, and adaptable requires a careful orchestration of various components. This research is tailored to design a facial recognition system that intricately integrates machine learning models and image processing techniques. The system's architecture is founded on a deep analysis of current practices, challenges, and emerging trends in the field. The design is aimed at a balance between computational efficiency and the ability to handle real-world variations. The section that follows offers a detailed roadmap of the research design, exploring each element with precision and contextual relevance.

In the process of developing the scope of the research, CNNs, SVMs, as well as Eigenfaces and PCA were all considered. After researching extensively, it was decided that the more modern SVM should be combined with PCA. The features outputted by the PCA are then fed into the SVM.

## Face Detection Module

The Face Detection Module is a pivotal component of the facial recognition system, responsible for identifying and locating human faces within images. This section explores the various techniques and algorithms used in face detection, with an emphasis on the application of Haar cascades.

### Algorithm and Haar Cascades

Haar cascades are a critical element within the face detection module. This sub-section delves into the algorithms used for face detection and the specific application of Haar cascades, providing an understanding of how they function within the system.

#### Applying OpenCV's Haar Cascades:

Haar cascades are machine learning models trained to detect objects for which they have been trained, using simple features. Originating from Viola-Jones detection algorithm, they have become popular for their efficiency. OpenCV's pre-trained Haar cascades, such as haarcascade\_frontalface\_default.xml, have been applied to detect faces within images. It works by sliding a window across the image and applying a series of binary feature classifiers to assess the presence of a face.

#### Tuning and Parameters:

Parameters like the scaling factor, minimum size, and neighbors can be adjusted for optimal detection. They define how the detection window scales and how many neighboring candidate rectangles should be retained. This system was found to detect faces reasonably well, in the data it was used on, with the following parameters: scaleFactor=1.1, minNeighbors=5, minSize=(400, 400).

## Face Data Sets Module

Face datasets form the bedrock of any facial recognition system. By having a diverse and appropriately processed dataset, the system ensures that predictions made on new, unseen data are accurate and reliable. The provided code offers a clear insight into how images are sourced, processed, and later used in a facial recognition system.

### Preprocessing

Given the variability in images, preprocessing is crucial to ensuring that the data fed into the system is consistent and free of noise. The preprocessing pipeline involves several steps:

#### Resizing:

* To ensure uniform processing across all images, they are resized to a consistent dimension, specifically to 47×62 pixels. Uniformity in size ensures that the model receives inputs of the same shape during both training and prediction.
* The **preprocess\_image** function in the code takes care of resizing. Whether the input is a direct image path or a numpy array, the function ensures that the image is resized to the target dimensions.

#### Color Conversion:

* While color images contain a wealth of information, for facial recognition, grayscale often suffices. Grayscale images not only reduce computational complexity but also ensure that the focus remains on structural features rather than color variations.
* Within the **preprocess\_image** function, the image is converted to grayscale using the **convert('L')** method from the PIL library.

#### Noise Reduction:

* It is crucial to minimize any noise in the image to highlight the primary facial features. Techniques like Gaussian blurring can be applied to reduce random noise.
* While Gaussian blurring is not explicitly applied in the system, the face detection mechanism might implicitly handle some noise reduction. This is because the Haar or LBP cascades used in OpenCV's **CascadeClassifier** typically work better with clean and distinct features.

#### Normalization:

* Normalizing pixel values ensures that they lie in a specific range, usually [0, 1]. This step aids in the stability of neural networks and other algorithms sensitive to input scales.
* In the **preprocess\_image** function, after converting the image to an array, pixel values are normalized to the [0, 1] range.

#### Face Detection:

* The code leverages the **FaceDetector** class to identify faces within the images. This step ensures that only the relevant portions of an image (the face) are processed, eliminating any background or unrelated objects. If a FaceDetector is not provided to the class, the system will simply skip the face detection
* The detection mechanism also rotates the image through various angles (0, 90, 180, and 270 degrees) to ensure faces are detected regardless of their orientation in the image.
* Post detection, the face is further processed (resized, grayscaled, and normalized) before being added to the dataset.

### Dataset Structure and Extensions:

* The **FaceDataset** class fetches the LFW (Labeled Faces in the Wild) dataset. This dataset's images undergo the preprocessing pipeline to ensure they are suited for facial recognition tasks.
* The **ExtendedFaceDataset** extends the **FaceDataset** by introducing the capability to inject custom images (referred to as the "true" images). This allows for a balanced dataset where custom images can be added to achieve a desired percentage in the dataset.

In conclusion, the preprocessing and dataset structuring in the provided code aim to ensure that the facial recognition system works reliably across varied input images. Proper preprocessing ensures that only the most relevant and cleaned features are fed into the system, thus enhancing prediction accuracy.

### Reporting:

Knowing what the face detection module is doing from a more fine grained perspective required the usage of a report. It did not make sense to require the output of reports, as it would make the system dump a lot of data onto disk. The option was provided as a part of the Face Detection interface. By default this is turned off, but by passing generate\_report=True into an instantiation of the FaceDetector object, an HTML report can be outputted.

For clarity on the functionality of this component, here are a few examples highlighting the performance of the face detection module of the experiment. This reporting was used to fine tune the face detection module to a degree to obtain more target images. Images were coming in rotated, plus it was difficult to determine if the detector was successfully detecting faces.

Figure 1 is important to the verification of the face recog because it displays the robustness of the system against a rotated face.

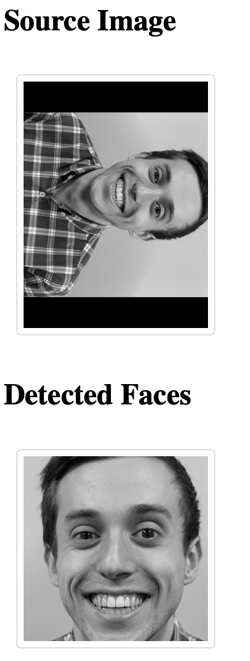


Figure : Rotated Image Example

Figure 2 displays the systems faults, as it does occasionally hit false positives on an image that is not a face. The system still displays robustness against a rotated face, however. It took tweaking, eventually settling on the parameters described in prior sections, to get the Face Detector down from at least one false positive in every image provided to 4 throughout the entire set of images containing the target face.

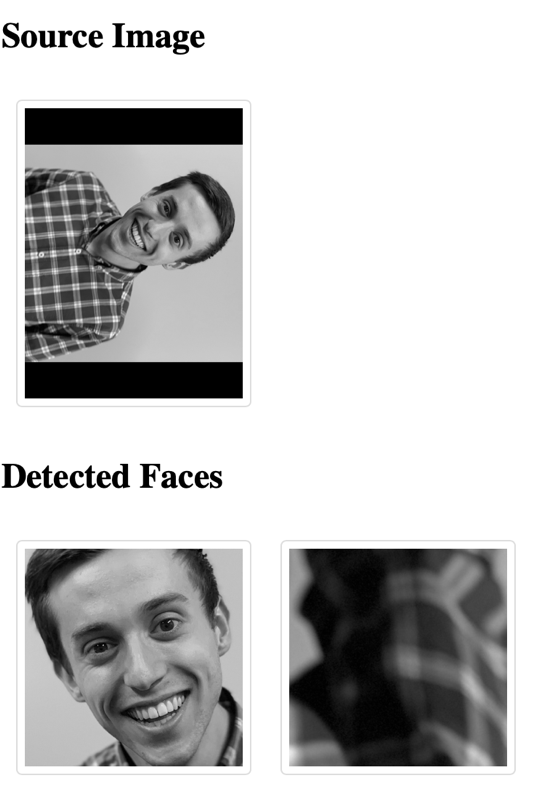


Figure : Rotated Image with False Positive

It is quite simple to detect a face on a plain background, but the system has also been tested in more unlikely conditions, Figure 3 emphasizes the capabilities of the face detector to catch a blurry face with an expression that is not necessarily common. Figure 4 does a bit of this too, but Figure 4 has other issues.

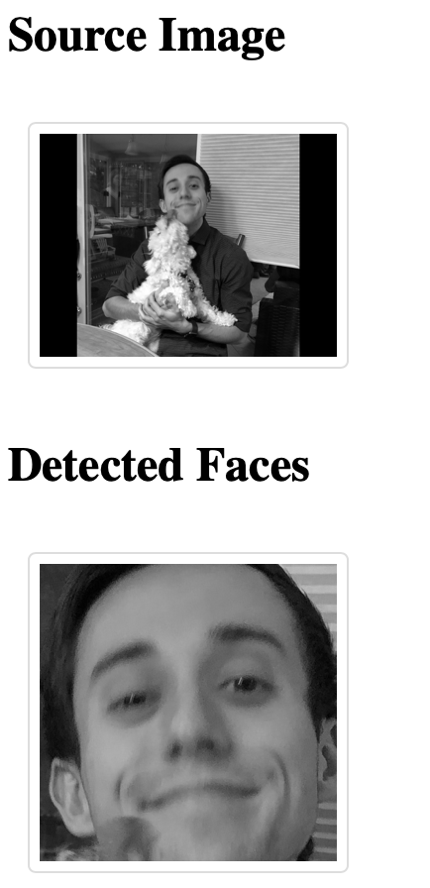


Figure : More Challenging, Slightly Blurry

Figure 4, caught a dog face. This dog face gets fed through the system on every Train/Test run of the face recognizer. Whether this face ends up in the training or testing data is presently unknown. This is another one of the 4 false positives, all of which are fed through the Face Recognizer, each as a part of training or testing data. Later the results will show that these false positives really did not impact performance that strongly.

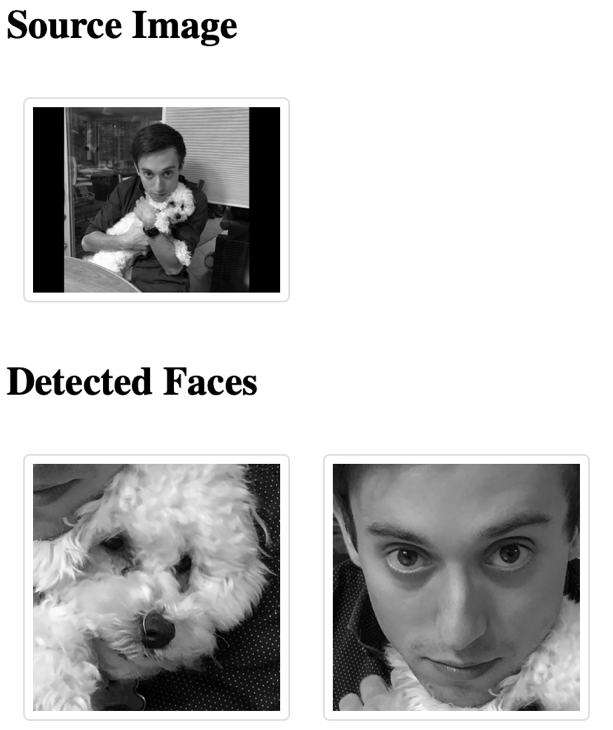


Figure : Dog Face Captured

## Face Recognition Module

The Face Recognition Module is where the previously detected faces are identified. This complex task requires a carefully orchestrated series of algorithms and techniques, discussed in detail below.

### Eigenfaces and Dimensionality Reduction

#### Introduction to Eigenfaces:

Eigenfaces represent a facial recognition method based on Principal Component Analysis (PCA). They provide a way to reduce the dimensionality of the image space, enabling effective computation and classification. This technique treats face recognition as a two-dimensional problem, where images are converted into a set of basis faces. The mathematics of PCA, its relevance to facial recognition, and its comparison with other dimensionality reduction techniques are explored here.

In the implementation of this system, eigenfaces are not directly handled by the programmer, rather the PCA implementation utilized by the programmer handles this behind the scenes.

#### PCA Implementation:

Implementing PCA involves creating a covariance matrix from the face vectors, followed by extracting eigenvectors and eigenvalues. These form the Eigenfaces when applied to original faces.

**Steps involved:**

* 1. **Standardize the data**: Before applying PCA, it is crucial to scale the features such that each one has a mean of zero and a standard deviation of one.
  2. **Calculate the Covariance Matrix**: Given **n** standardized samples of dimension **d**, form a **d x d** covariance matrix **C**. Each element Cij*Cij*​ of **C** measures the covariance between the **i-th** and **j-th** features.
  3. **Calculate Eigenvectors and Eigenvalues**: Obtain eigenvectors and corresponding eigenvalues of the covariance matrix. Eigenvectors represent the directions of maximum variance in the data, and the eigenvalues give the magnitude of this variance.

Again, much of this is abstracted away due to the conveniences of modern libraries.

#### Eigenfaces with SciKit Learn:

SciKit Learn offers an effective toolkit for PCA implementation. The library's **PCA** module streamlines Eigenface extraction, eliminating the manual computation of covariance matrices and eigenvalues.

* **Face Projection**: SciKit Learn allows for the transformation of new faces into the Eigenface subspace, representing them as a linear combination of the selected Eigenfaces.
* **Recognition Mechanism**: The projected representation can be juxtaposed with known faces using distance metrics. A minimized distance suggests successful facial recognition

#### PCA in SciKit Learn:

The PCA class in the sklearn.decomposition module is a dimensionality reduction tool that uses singular value decomposition of the data and can project it to a lower-dimensional space.

The PCA object in SciKit Learn provides a user-friendly interface to perform Principal Component Analysis, making it easier for data scientists and researchers to apply this powerful technique for dimensionality reduction and feature extraction. However, a clear understanding of its workings and inherent assumptions is crucial to appropriately apply it to real-world datasets [15].

### Support Vector Machine (SVM)

#### SVM Overview:

Support Vector Machine (SVM) is a powerful classification method that has been implemented efficiently in the Scikit-learn module. This section provides an in-depth overview of how SVM operates, detailing the mathematical formulations of the hyperplanes and margin optimization. Particularly, it explains how Scikit-learn's SVM implementation (`sklearn.svm.SVC`) fits within the facial recognition paradigm [15].

#### Kernel Selection:

One of SVM's standout features is its flexibility, primarily using kernel functions. The Scikit-learn module offers built-in support for a variety of these functions. This sub-section delves into different kernel functions available in Scikit-learn, such as linear (“linear”), polynomial (“poly”), and Radial Basis Function (“RBF”), and discusses their relevance to face recognition. Additionally, it examines the decision-making process for kernel selection in Scikit-learn's SVM and how to apply these kernels within the system effectively [15]. For this project, it was found that the linear kernel was the best choice over RBF. The polynomial option, admittedly, was never tried due to the strong performance of the linear Kernel.

#### Hyperparameter Tuning:

SVM's performance, especially when implemented via Scikit-learn, is highly sensitive to specific hyperparameters. Key among these are the regularization parameter (C) and the kernel coefficient (gamma). This section sheds light on the methodologies available in Scikit-learn, like `GridSearchCV` and `RandomizedSearchCV`, to fine-tune these hyperparameters. It further expounds on the implications of selecting different values, emphasizing potential issues such as overfitting and the trade-offs concerning model complexity [15].

Incorporating Scikit-learn's utilities and tools can streamline the SVM modeling process, making tasks like kernel selection and hyperparameter tuning more systematic and efficient.

### Training

#### Dataset Generation:

Creating a diverse and representative dataset is critical. This section discusses the principles and practices of dataset construction, including the selection of sources, diversity in facial expressions, lighting, angles, and potential biases. SciKit Learn provides direct access to the fetch\_lfw\_people function, which grants access to the Labeled Faces in the Wild (LFW) Dataset. This data set is used to generate the false cases, and the true cases are presently jpeg images of the author of this paper. Effectively, the system is trying to learn to authenticate on images of author.

#### Data Splitting:

The data is partitioned into training (75%) and testing (25%) sets, ensuring an unbiased evaluation. The partitioning of data into training and testing sets is a delicate process, impacting model evaluation. This is done using SciKit Learn’s train\_test\_split functionality [15].

#### Validation Strategy:

Cross-validation, e.g., 5-fold, is employed to minimize overfitting and provide a robust estimate of model performance. Cross-validation, such as k-fold validation, is a robust methodology to ensure that the model's performance is generalizable.

The data contains True (1) and False (0) cases, the experiment aims to understand the effect of weighting True to False in different ratios. The system aims to produce data sets that contain 5%, 10%, 15%, … , 90% True images and then run them through the algorithm. A cross validation report will be produced for each case.

### Reporting

The Face Recognition, more specifically, the FaceRecognizer, can generate classification reports describing how well the FaceRecognizer performed throughout the conditions it was tested under. The FaceRecognizer chooses an output file at runtime, and then each successive instantiation of FaceRecognizer appends to that same file. An example is shown below in Figure 5, where the 4 runs of 5% True data set are shown. The four runs include SMOTE resampling Enabled/Disabled and Face Detection Enabled/Disabled.

This is the first glimpse at how the system is truly being tested.

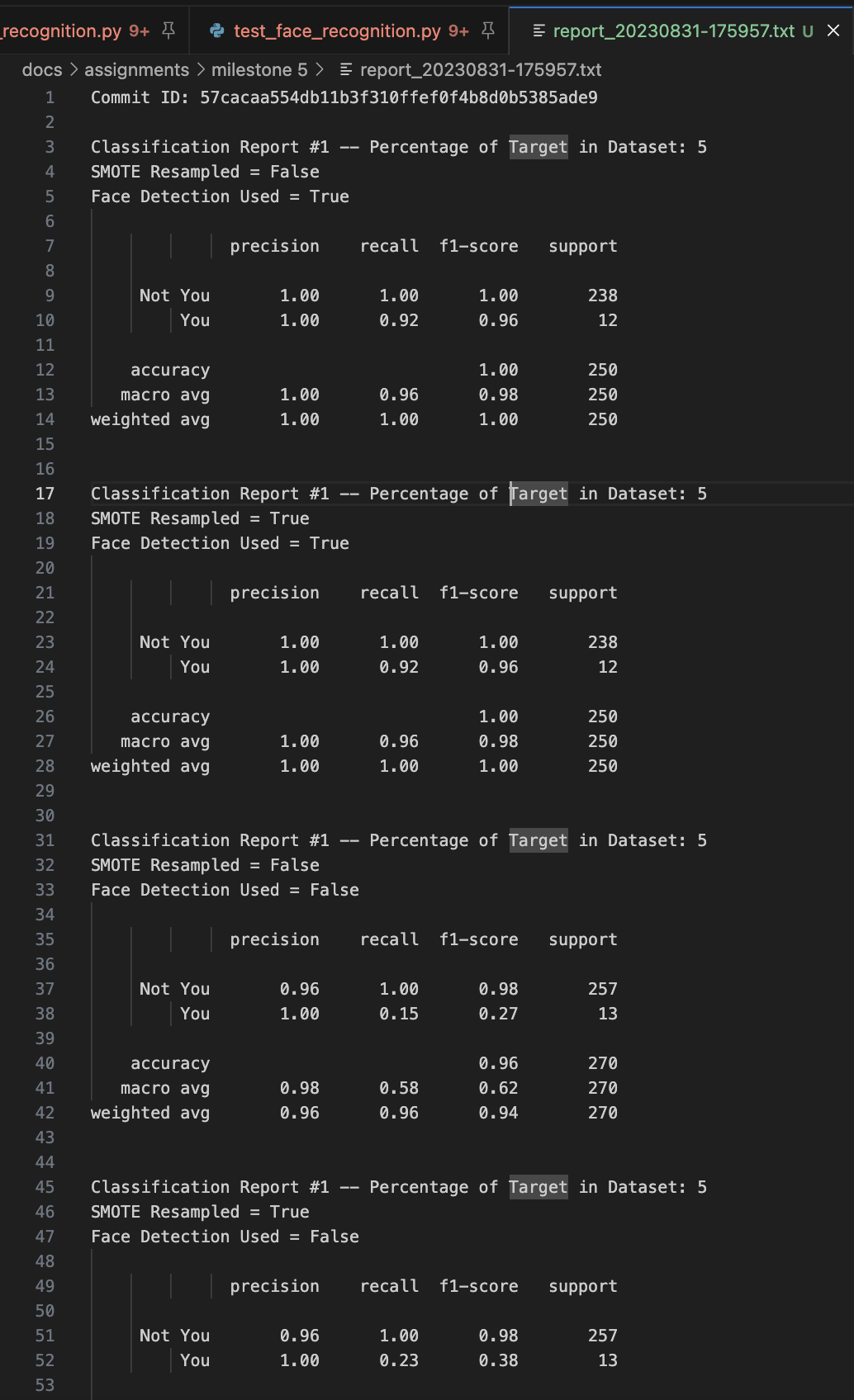


Figure : FaceRecognizer Abridged Report Output

## Experiment Procedure

### Data Collection

Data is collected from the LFW data set and from the user who would like to be verified:

When using the LFW (Labeled Faces in the Wild) dataset, the faces have already undergone preprocessing. This means that the facial features are well-aligned and centered within the frame. Such preprocessing is crucial for consistent training in facial recognition systems. However, the challenge arises when the images being recognized, particularly the target person or the “you” case, do not possess the same alignment or scale characteristics as the LFW dataset. The images could be in color, blurry, or rotated.

To mitigate this discrepancy, the system employs a face detection mechanism. This mechanism ensures that the faces in the target images are identified and aligned consistently with the training data.

**Implementation Details:**

The system uses OpenCV, a widely used computer vision library, for the face detection task. The main class responsible for this in our system is called **FaceDetector**. Here is a deeper dive into its workings:

1. **Initialization**: The **FaceDetector** is initialized using a given Haar or LBP cascade file. This cascade file contains features that the detector uses to identify faces. The class can also be set up to generate an HTML report showcasing detected faces, controlled by the **generate\_report** flag.
2. **Face Detection**: The **detect\_faces** method is the core of this class. It attempts to detect faces by rotating the image through various angles - 0, 90, 180, and 270 degrees. This rotation ensures that faces are detected regardless of their orientation in the image. Once the image is rotated, it is converted to grayscale, which simplifies the detection process. The detected faces are then filtered based on their aspect ratios to ensure that only valid faces are considered.
3. **Labeling:** The faces are then labeled “0” and “1” where “0” represents the “Not You” case and “1” represents the “You” case.
4. **Image Saving**: Detected faces and source images can be saved to predefined directories for further analysis. This feature is particularly useful when generating reports.
5. **Visualization**: The system offers two utilities, **matplotlib\_face** and **matplotlib\_faces**, for visualizing the detected faces using the **matplotlib** library. This is beneficial for quick verification and quality checks.
6. **Report Generation**: If the **generate\_report** flag is set during initialization, the **finalize\_report** method can generate an HTML report that showcases all the source images and their detected faces. This report provides a visual summary of the face detection process and its outcomes.

In summary, the face detection system ensures that the target images are processed and aligned consistently with the training data, thereby enhancing the accuracy and reliability of the facial recognition process. This step is crucial, especially when the input images have varied orientations and scales.

Face detection is an option when building the training and testing data. It can be turned on and off at the will of the programmer, and experiments can be run on whether its usage contributes to better outcomes.

### Preprocessing

Both the data from LFW and the detected faces from the FaceDetector are passed through a common preprocessor. They are preprocessed by converting the image to a grayscale image via Python’s Pillow (PIL) library. These images are resized to 47x62 pixels The images are then normalized to values between 0 and 1 by first checking if normalized to this range, then by dividing each pixel by 255 if it is not. The resulting image is then flattened into a 1D array.

### Training the Recognizer:

Training includes applying PCA for dimensionality reduction and then feeding these features to the SVM for classification. The 1D arrays from the preprocessing have been labeled, and the Recognizer receives data either with or without Face Recognition enabled and with or without SMOTE resampling enabled. 4 cases total.

### Evaluation

Evaluation of the system is done through reporting the classification capabilities of the recognizer. The system is fed its training data, as mentioned in the previous section, and then it is fed testing data to evaluate the system:

Metrics such as precision, recall, and F1-score provide a quantitative assessment. The outcomes from different inputs and parameters are generated and reported as mentioned in earlier sections.

## Conclusion of Design

The experiment's design encapsulates a holistic approach to facial recognition, from detection to classification. The meticulous planning, selection of algorithms, validation strategies, and robust evaluation showcase a well-rounded and rigorous design. The next phases will involve implementation, tuning, and extensive testing to ensure that the design's promise translates into an effective real-world application.

This expanded content adds depth to each section, detailing the steps, choices, and considerations in the design of the facial recognition system. Feel free to modify or ask for further details on specific areas.

# Results:

## Introduction

After extensive work on the system, through creation of a valid dataset, proper preprocessing of the images and faces, and finally in creation of the tests, the experiment’s results have been collected and prepared. The system is truly robust in many circumstances, and, based on what the results will show, it is fair to conclude that this is a valid approach to face verification. With a high degree of accuracy, the system can recognize the face of one, out of many.

## Evaluation

The Results will focus on the classification report outputted by the FaceRecognizer. The results will show strong results in most areas in F1 Score, Recall, and Precision:

In the context of binary classification, precision and recall are two pivotal metrics that provide insights into different facets of model performance. Precision focuses on the correctness of positive predictions, denoting the fraction of predicted positive instances that are truly positive. On the other hand, recall gauges the model's capability to identify all positive instances, reflecting the fraction of actual positive cases that the model correctly detected. While precision is crucial in scenarios where the cost of false positives is high, like drug approvals, recall is paramount in situations where missing a positive instance carries significant consequences, such as in disease diagnostics. Both metrics, when considered together, provide a comprehensive perspective on a model's ability to discern between classes and the trade-offs involved.

The F1 score represents the harmonic mean of precision and recall, offering a balanced measure for binary classification, especially in imbalanced datasets. The Weighted Average F1, on the other hand, calculates the F1 scores for each class in multiclass classification and then takes a weighted average, ensuring the score reflects the importance or prevalence of each class in the dataset.

## Discussion

Further discussion of the collected metrics on the trained models is to follow. These models have been trained as described in previous sections. SMOTE resampling has been toggled on and off, along with Face Detection. In the sections to follow, you will see a the outcomes of these tests and a comparison of each test case against the others, in terms of the aforementioned metrics, precision, recall, and F1 scores.

### Precision

A graph with numbers and a bar chart

Description automatically generated with medium confidence

Figure : Precisions for "You" Classification

A blue and white chart with numbers and a bar chart

Description automatically generated with medium confidence

Figure : Precisions for "Not You" Classification

Precision performance is an important metric in Facial Recognition. A False positive can be quite costly in some applications such as security verification. A bad actor could be granted access via this system on a false positive. A model that has a precision of 1 for all cases, produces no false positives.

In our specific model, the goal is to distinguish between "You" and "Not You". A false positive in the "You" category would mean the system mistakenly identifies a person that is "Not You" as "You". This error has significant security implications, as it means the system could potentially grant access to an unauthorized individual. The outcomes of testing the model for this are evident in Figure 6.

On the other hand, a false positive in the "Not You" category would imply the system erroneously identifies "You" as "Not You". While this error would prevent authorized access, it does not pose a security risk. It is more of an inconvenience or operational challenge. This could be mitigated by something like a 2nd attempt. Results detailing the precision performance in the "Not You" category can be found in Figure 7.

The model performs admirably in both cases. Making use of face detection is shown to be ever so slightly worse in the “You” case, however, it is hard to make a case for not employing face detection in the algorithm. It is not that costly computation wise, and it is also a necessary preprocessing step to get all the training and testing images into the correct shape and orientation.

What is shocking, is the limited effect of SMOTE resampling. It seems to cut back on a few false positives in the “Not You” cases. With SMOTE enabled, we can achieve higher performance by showing the model fewer images of the target “You” compared to images of other people.

### Recall

A blue and white chart

Description automatically generated with medium confidence

Figure : Recalls for "You" Classification

A graph with numbers and a bar chart

Description automatically generated with medium confidence

Figure : Recalls for "Not You" Classification

Recall is a crucial performance metric in Facial Recognition, particularly when the consequences of false negatives can be detrimental. In contexts like security verification, a false negative implies that a legitimate user or entity, labeled as "You", is mistakenly denied access. In essence, if a model achieves a recall score of 1 across categories, it signifies that all genuine positive instances are correctly recognized by the system.

For our specific model, which aims to discern between "You" and "Not You", a false negative in the "You" category indicates that the system mistakenly perceives "You" as "Not You". Such an error translates to an operational challenge, where a genuine user might be locked out or required to undergo additional verification steps. The recall results for the "You" classification are visualized in Figure 8.

Conversely, a false negative in the "Not You" category implies that someone who is not "You" is incorrectly identified as "You". This error poses a significant security threat, as it means the system might overlook an unauthorized individual, mistaking them for the genuine user. The recall outcomes for this classification are showcased in Figure 9.

The model's recall performance is commendable for both classifications. Again, face detection only slightly impacts recall in the "You" category, but, as mentioned, the overall benefits of using face detection far outweigh the negatives of using it.

By employing SMOTE, we observe a marginal reduction in false negatives for the "Not You" classification. This indicates that even with fewer "You" samples compared to other instances during training, the model can potentially recognize genuine users more effectively. Again, the benefit seen is quite small, and in the “Not You” case, the model is performing very well to begin with. So long as we show the model an adequate number of images of “Not You,” then the system does not produce false negatives in this category. Having SMOTE and Face Recognition enabled is ideal in the “You” category.

### F1 Score

A blue and yellow chart

Description automatically generated with medium confidence

Figure : F1 Scores for "You" Classification

A blue and white chart

Description automatically generated

Figure : F1 Scores for "Not You" Classification

In facial recognition systems, the F1-score holds paramount importance as it strikes a balance between precision and recall, ensuring that both false positives and false negatives are minimized. Especially when considering applications with security implications, achieving a high F1-score means that the system is both reliably identifying legitimate users and accurately filtering out unauthorized ones. An F1-score of 1 represents perfect precision and recall, eliminating both false positives and false negatives.

In our specific model, which discerns between "You" and "Not You", an optimal F1-score in the "You" category would indicate the system's impeccable ability to correctly identify genuine users while minimizing the risk of denying them access due to misidentification. Insights into the F1-score performance for this category are provided in Figure 10.

For the "Not You" classification, a high F1-score ensures that unauthorized individuals aren't misclassified as genuine users, safeguarding against potential security breaches while also ensuring that legitimate users aren't misclassified as impostors. This balance is vital for both operational efficiency and security robustness. The F1-score results for this category are illustrated in Figure 11.

The model showcases impressive F1-scores for both categories. Although implementing face detection minutely influences the F1-score for the "You" category, the broader advantages of face detection—mainly, ensuring consistent image preprocessing for training and testing—justify its integration. Interestingly, the SMOTE resampling's impact on F1 is noteworthy. With SMOTE activated, the model demonstrates enhanced balance between precision and recall, particularly in the "Not You" classification. This suggests that by presenting the model with a balanced training dataset, its ability to harmonize false positives and negatives is optimized.

### Weighted Average F1 Score

A blue and white chart

Description automatically generated with medium confidence

Figure : Weighted Average F1 Scores

The weighted average F1-score is a pivotal metric in facial recognition systems, particularly when there are imbalances in class distributions. It provides a holistic view of a model's performance by calculating the F1-scores of each class proportionally, based on their presence in the dataset. This ensures that more frequent classes have a greater influence on the overall score. Achieving a high weighted average F1-score signals a system's capability to maintain a balance between precision and recall across all categories, factoring in their respective prevalence. An optimal score of 1 denotes the system's unparalleled proficiency in mitigating both false positives and false negatives across all classifications. The insights into our model's performance, represented by the weighted average F1-score across the global dataset, are vividly depicted in Figure 12.

This metric is especially crucial for our model, given the distinct class distributions between "You" and "Not You". It affirms the model's robustness in handling disparities in sample sizes while maintaining accuracy in identification. Remarkably, leveraging techniques like face detection and SMOTE resampling has shown to subtly influence this aggregated score, emphasizing their roles in fine-tuning the model's overall efficacy.

# Future Work

The following improvements to the system could be implemented in the future given the promise shown by the results of the experiment.

1. **Algorithmic Refinements**: Our model exhibited promise in discerning "You" from "Not You", but the ongoing evolution of neural architectures and algorithms means there is potential for further optimization. Exploring deeper networks or integrating transfer learning could yield improvements in accuracy and computational efficiency.
2. **Diverse Datasets**: To ensure the robustness and universality of our model, it would be invaluable to test on datasets encompassing a broader range of demographics. This includes variability in age, ethnicity, lighting conditions, and facial obstructions (like glasses or facial hair).
3. **Real-world Application Testing**: While our research provided valuable insights under controlled conditions, testing the model in real-world, dynamic scenarios—like varied lighting, movement, and diverse backgrounds—can offer insights into its practical efficacy. As we showed, false positives in the Face Detection phase may ramp up as complexity increases.
4. **Alternative Preprocessing Techniques**: Beyond face detection, other preprocessing steps such as facial landmark detection, alignment, and normalization might enhance the model's capability to recognize faces amidst diverse conditions.
5. **Security Measures**: With the growing concern about adversarial attacks on neural networks, future endeavors should investigate enhancing the model's resistance against such attempts. This could ensure the system remains reliable even when targeted by sophisticated adversaries.

In conclusion, while our current research has laid a solid foundation in facial recognition metrics and performance, the horizon of possibilities remains vast. There are a myriad opportunities to refine, expand, and ethically advance in this domain.

# Conclusion

Facial recognition has grown into an indispensable tool in a myriad of applications, especially in security and authentication systems. Through this study, we've delved deep into evaluating various models and preprocessing techniques, emphasizing their performance through metrics like precision, recall, F1 score, and the weighted average F1. The implications of false positives and negatives in security applications underscore the paramount importance of achieving optimal values in these metrics. Our findings illuminate the exemplary performance of our model in discerning "You" from "Not You," while highlighting the intricate nuances of each metric's significance.

Incorporating face detection, though slightly influencing individual scores, proved invaluable for consistent image preprocessing—a foundational step for any successful facial recognition system. Surprisingly, the nuanced influence of SMOTE resampling, particularly in adjusting the balance between precision and recall for the "Not You" classification, affirms the technique's potential in scenarios with imbalanced datasets.

In summary, this research underscores the intricacies of building an effective facial recognition system. It serves as a testament to the continuous advancements in the domain, providing insights and benchmarks for future endeavors. As technology advances and the importance of reliable facial recognition continues to escalate, studies like this lay the groundwork for ensuring systems are both accurate and secure.

Code can be found here: <https://github.com/jm0rt1/cis-663-final-project>

**Bibliography**

[1] Y. Taigman, M. Yang, M. Ranzato and L. Wolf, “DeepFace: Closing the gap to human-level performance in face verification,” 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014, pp. 1701-1708, doi: 10.1109/CVPR.2014.220.

[2] F. Schroff, D. Kalenichenko and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 815-823, 2015.

[3] R. Chellappa, C. L. Wilson and S. Sirohey, "Human and machine recognition of faces: a survey," in *\_Proceedings of the IEEE\_*, vol. 83, no. 5, pp. 705-741, May 1995, doi: 10.1109/5.381842.

[4] M. Turk and A. Pentland, “Eigenfaces for recognition,” Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 71-86, Jan. 1991, doi: 10.1162/jocn.1991.3.1.71.

[5] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus and Y. LeCun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” arXiv preprint arXiv:1312.6229, 2013.

[6]Yang, H., Jia, X., Loy, C. C., & Robinson, P. (2015). An empirical study of recent face alignment methods. arXiv preprint arXiv:1511.05049.

[7] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, “Gradient-based learning applied to document recognition,” in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[8]A. Krizhevsky, I. Sutskever and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in Neural Information Processing Systems, pp. 1097-1105, 2012.

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov et al., “Going deeper with convolutions,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-9, 2015.

[10] Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015, September). Deep face recognition. In bmvc (Vol. 1, No. 3, p. 6).

[11] Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Le, Q. V., ... & Ng, A. Y. (2012). Large scale distributed deep networks. In Advances in neural information processing systems (pp. 1223-1231).

[12] Y. Sun, Y. Chen, X. Wang and X. Tang, “Deep learning face representation by joint identification-verification,” 2014 28th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 2014, pp. 1988-1996.

[13] Y. Sun, Y. Chen, X. Wang and X. Tang, “Deep learning face representation by joint identification-verification,” 2014 28th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 2014, pp. 1988-1996.

[14] K. W. Bowyer, K. Chang, and P. Flynn, "A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition," Computer Vision and Image Understanding, vol. 101, no. 1, pp. 1–15, 2006.

[15] Scikit-learn developers. Scikit-learn: Machine Learning in Python. Documentation, version 1.3.0. Available: https://scikit-learn.org/stable/documentation.html. [Accessed: August 15, 2023].