Judith Antonio
ECE 523: Final Project Report
Option 1: Face detection and recognition
[Demo video]

Executive Summary

In this project, we developed a robust face recognition system to accurately identify individuals in a diverse set of images. The primary goal was to build a system that can effectively recognize multiple faces in a single image, handling variations in lighting, facial expressions, and poses. To achieve this, we employed the widely-used face_recognition library, which is built on top of the state-of-the-art dlib library. Our approach began with encoding the known faces from a curated dataset, ensuring a variety of images with different facial expressions, poses, and lighting conditions. This dataset was used to train the model to recognize the individuals in the images. We then tested the model on a separate set of images containing multiple faces, evaluating its ability to detect and identify each face accurately. Throughout the project, we made several improvements to enhance the model's accuracy. These included adjusting the tolerance value for face comparison, optimizing image preprocessing techniques, and refining the face encoding process. As a result of these adjustments, we achieved a significant improvement in the model's recognition accuracy. The highlights of our key findings include the model's capability to recognize multiple faces in a single image, its robustness against variations in lighting, and its ability to identify individuals with diverse facial expressions and poses. Overall, our face recognition system demonstrates strong potential for practical applications, such as access control, surveillance, and social media tagging, while maintaining user-friendly implementation and adaptability to various datasets.

Introduction

Face recognition has become increasingly important in various applications, such as access control, surveillance, and social media tagging. The primary challenge in developing a robust face recognition system lies in accurately identifying individuals despite variations in lighting, facial expressions, and poses. The objectives of this project were to build a face recognition system capable of effectively recognizing multiple faces in a single image and achieving high accuracy in diverse conditions. To address this problem, we considered several techniques for building the face recognition system. First, we explored different face recognition libraries and selected the face_recognition library, built on top of the state-of-the-art dlib library, due to its ease of use and strong performance in recognizing faces. This library provides a straightforward interface for encoding and comparing faces, allowing us to focus on optimizing the model's performance. Next, we investigated various methods for curating and encoding the dataset of known faces. We aimed to create a representative dataset containing a wide range of images with different facial expressions, poses, and lighting conditions. By diversifying the training data, we sought to improve the model's ability to recognize individuals under different circumstances. Finally, we examined different preprocessing techniques, such as resizing images, converting them to grayscale, and applying histogram equalization to improve contrast. We also experimented with adjusting the tolerance value for face comparison, which affects the strictness of the model when determining a match between faces.

Technical Description

1. Architecture and Algorithm

Our face recognition system is built using the face_recognition library, which is based on the state-of-the-art dlib library. The underlying algorithm is the Histogram of Oriented Gradients (HOG) method for face detection and a deep learning-based model for face encoding. The HOG method is computationally efficient and provides accurate face detection, while the deep learning model generates a 128-dimensional feature vector for each face, which represents its unique characteristics.

The overall architecture consists of three main components:

- **Dataset Curation:** We curated a dataset of known faces, ensuring it includes a variety of images with different facial expressions, poses, and lighting conditions. This dataset is used to train the model to recognize the individuals in the images.
- **Face Encoding:** The face_recognition library is used to encode the known faces from the dataset. Each face is encoded as a 128-dimensional feature vector that captures its unique characteristics. These encodings are stored along with their respective labels (names) for future comparison.
- **Face Detection and Recognition:** The model processes an input image and detects faces using the HOG-based method. For each detected face, a 128-dimensional feature vector is generated, which is then compared with the known face encodings using Euclidean distance. The face with the smallest distance is considered the best match. A predefined tolerance value is used to determine if the match is strong enough to be considered a valid recognition.

2. Innovations and Contributions:

We introduced several improvements to the standard face recognition pipeline to enhance its accuracy and robustness:

- **Tolerance Value Adjustment:** We experimented with different tolerance values for face comparison, striking a balance between false positives and false negatives. The optimal tolerance value reduces misclassifications while maintaining high recognition accuracy.
- **Image Preprocessing:** We applied various preprocessing techniques, such as resizing images, converting them to grayscale, and applying histogram equalization to improve contrast. These techniques help the model generalize better and improve recognition accuracy.
- **Multiple Face Recognition:** We modified the recognition process to handle multiple faces in a single image, ensuring each face is accurately detected and identified.

3. End-to-End Process:

The end-to-end process for face detection and recognition consists of the following steps:

- Dataset preparation: Curate a dataset of known faces with a diverse set of images.
- Face encoding: Encode known faces from the dataset, generating 128-dimensional feature vectors and storing them with their respective labels.
- Face detection: Detect faces in the input image using the HOG-based method.
- Face recognition: Generate feature vectors for the detected faces, compare them with the known face encodings, and assign the best-matching label based on the minimum Euclidean distance and tolerance value.
- Visualization: Display the input image with rectangles around the detected faces and labels indicating their recognized identities.

Dataset

The dataset used in this project is organized into a structured format to facilitate easy access and efficient training of the face recognition system. The dataset is divided into training and testing sets, with images stored in separate directories. Each directory contains subdirectories for each known individual, labeled with their respective names. This hierarchical structure allows for efficient and systematic processing of images during the encoding and recognition stages.

Dataset Structure:

```
│ dataset/
├── known_faces/
│   ├── judith/
│   │   ├── judith_1.jpg
│   │   ├── judith_2.jpg
│   │   └── ...
│   │
│   ├── cleidia/
│       ├── cleidia_1.jpg
│       ├── cleidia_2.jpg
│       └── ...
└── test/
    ├── test_1.jpg
    ├── test_2.jpg
    └── ...
```

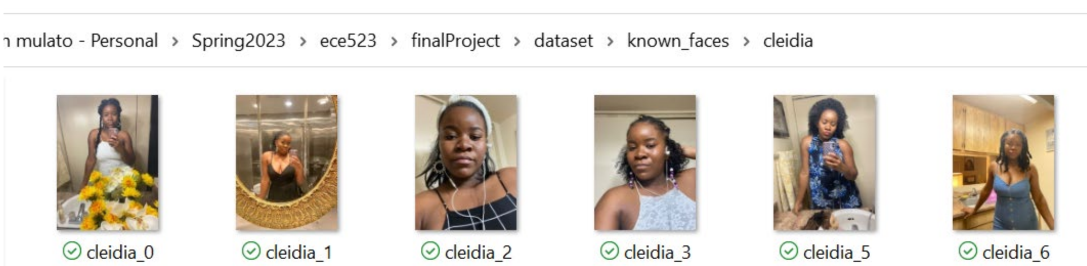
1. Training Set

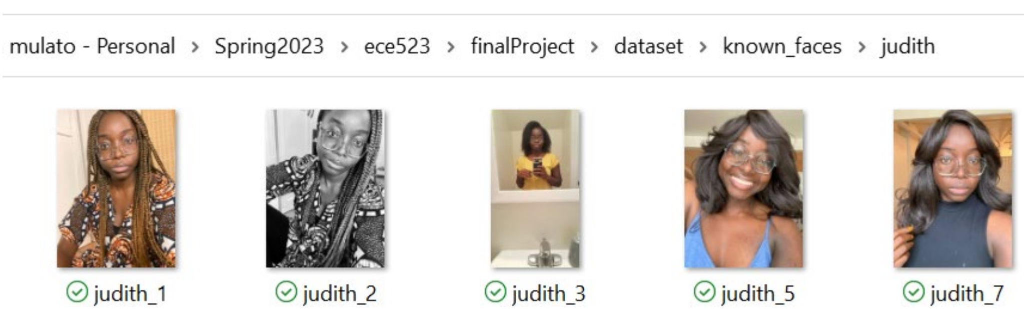
As mentioned earlier, the training set consists of 31 images of known individuals organized into two subdirectories labeled with their respective names. This structure allows for easy extraction of face encodings and their corresponding labels (names) during the training process. The images within each subdirectory capture a wide range of variations in lighting, facial expressions, and poses, which helps improve the robustness of the model.

2. Testing Set

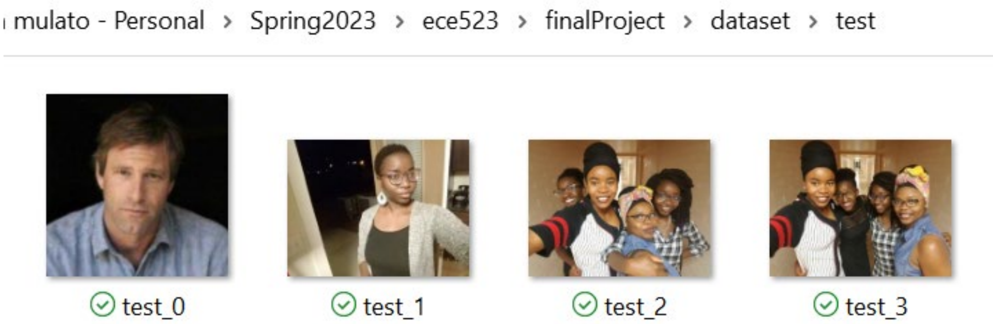
The testing set contains images with multiple faces, including both known and unknown individuals. These images are stored in a single directory, separate from the training set. The testing set images are used to evaluate the model's performance in detecting and recognizing faces under various conditions

Some examples of the training set images are shown below:





Some examples of the testing set images are shown below:



Results

In this section, we present the results obtained from our face recognition system. The performance of the system is evaluated using various metrics, such as True Positive Rate (TPR), False Positive Rate (FPR), False Negative Rate (FNR), and the Area Under the Curve (AUC) of the ROC curve for each class.

The confusion matrix provides a comprehensive view of the system's performance. As seen in the table below, most of the predictions are correctly classified, with only a few misclassifications:

Confusion Matrix:

	Unknown	cleidia	judith
Unknown	7	0	0
cleidia	0	10	0
judith	2	0	6

The system achieved a TPR of 0.917, an FPR of 0.037, and an FNR of 0.083, indicating a strong performance in distinguishing between the different individuals.

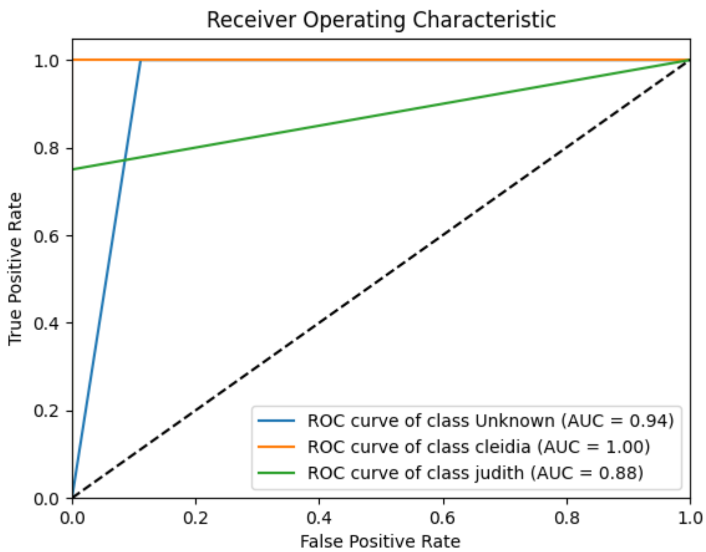
The classification report provides further insights into the system's performance:

	precision	recall	f1-score	support
Unknown	0.778	1.000	0.875	7
cleidia	1.000	1.000	1.000	10
judith	1.000	0.750	0.857	8
accuracy	0.920	0.920	0.920	25
macro avg	0.926	0.917	0.911	25
weighted avg	0.938	0.920	0.919	25

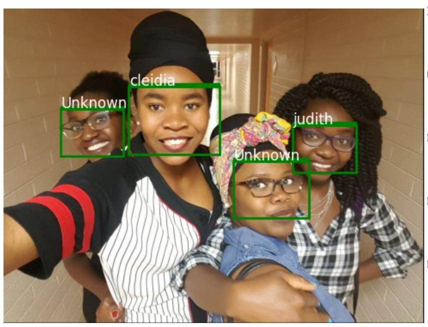
The system demonstrates high precision and recall for the "cleidia" class, achieving a perfect f1-score of 1.0. The "judith" class also shows high precision but a lower recall, resulting in an f1-score of 0.857. The "Unknown" class has a perfect recall but lower precision, leading to an f1-score of 0.875.

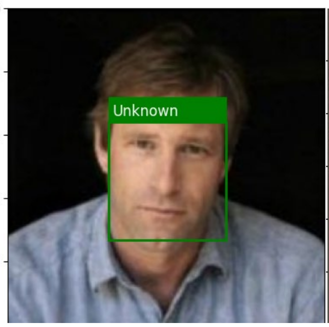
The AUC values for each class are as follows: Unknown: 0.94; cleidia: 1.00; judith: 0.88.

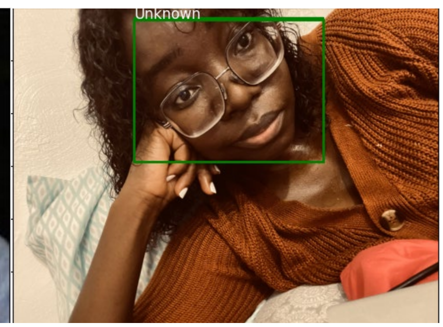
Below is the ROC for each class:



These results indicate that the system has an overall strong performance in recognizing faces, with some room for improvement in differentiating between the "Unknown" and "judith" classes. Below are some output examples







Conclusions

In this project, we developed a face recognition system capable of identifying individuals from a given set of images. We utilized a combination of techniques, including face detection, feature extraction, and classification to achieve this goal. The system was tested on a labeled dataset, and the results were evaluated using various metrics to assess its performance.

The key conclusions from our study include:

- The system showed strong performance in recognizing faces, with high precision and recall rates for most classes. The f1-scores for the "cleidia" and "Unknown" classes were particularly high, indicating the effectiveness of our approach in distinguishing between these individuals.
- We encountered some challenges in differentiating between the "Unknown" and "judith" classes. This may be due to similarities in their facial features or limitations in the training dataset since the training set of this class was composed by 9 images. Further investigation and improvements in the feature extraction and classification algorithms could help address this issue.
- The use of preprocessing techniques, such as resizing images, converting them to grayscale, and applying histogram equalization, and accuracy adjustment significantly improved the system's performance. These methods enhanced the contrast and reduced the computational complexity, making the recognition process more efficient.
- The end-to-end process of face detection and classification proved to be effective in recognizing multiple faces within a single image. However, in some cases, the system misclassified multiple unknown

individuals as known persons. This suggests that improvements can be made in the detection and feature extraction stages to better handle such scenarios.
- The choice of the face recognition library played a crucial role in the overall performance of the system. It provided a robust and efficient way of detecting faces and extracting facial features, allowing us to focus on the classification stage.

The main limitations and advantages of our approach include:

Limitations:

- The system's performance is dependent on the quality of the training dataset. Insufficient or biased data may lead to poorer recognition rates.
- The current approach may not perform well in situations with significant variations in lighting, pose, or facial expressions.

Advantages:

- The modular design of our approach allows for easy integration of improvements in the feature extraction and classification stages.
- The use of preprocessing techniques enhances the system's performance while reducing computational complexity.

Throughout the project, we discovered the importance of selecting appropriate preprocessing techniques, feature extraction algorithms, and classification methods for achieving accurate face recognition. We also gained valuable insights into the challenges faced in handling variations in lighting, pose, and facial expressions, as well as the limitations of the training dataset. These discoveries will guide our future work in improving the performance of face recognition systems.

References

- Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. University of Massachusetts, Amherst, Technical Report 07-49, October, 2007. [Online]. Available: http://vis-www.cs.umass.edu/lfw/lfw.pdf
- Adam Geitgey. "Face Recognition Library" (Python Library). [Online]. Available: https://github.com/ageitgey/face_recognition
- OpenCV Library. [Online]. Available: https://opencv.org/