Project Update Report

Title: Face Recognition System (Group - 5)

1. Introduction and Understanding the Problem

At the beginning of this project, I focused on understanding the core requirement and objective. After discussing with my supervisor and reviewing the project description, I realized that the main goal is to build a system capable of identifying 10 individuals based on their facial images. The system must demonstrate its effectiveness in recognizing known individuals and rejecting unknowns. This clarified that my project falls under the domain of face recognition**.**

2. Literature Review and Initial Research

Once the objective was clear, I began researching how face recognition systems are generally built. I consulted resources from Google, ChatGPT, and reviewed 2-3 academic papers. From this research, I learned that a typical face recognition pipeline consists of three main stages:

- Collecting or sourcing a suitable dataset
- Preprocessing the images
- Selecting and applying a suitable machine learning model

I learned that for my system to work effectively, the dataset must include multiple images per person with varying facial expressions and angles. I also identified that the dataset must be labeled, scaled, and annotated properly.

3. Dataset Exploration

I considered two main options for the dataset:

1. **Collect a dataset from the internet**
2. **Create a custom dataset manually**

Initially, I searched online for existing datasets that match my requirement of having 10 individuals with 15–20 facial images each. If I am unable to find an exact match, I plan to proceed with building my own dataset.

4. Model Selection: Why KNN?

Once I had a basic plan for the dataset, I shifted focus to model selection. Based on my understanding and paper review, I realized my dataset will have only around **200 images**, which is relatively small. I chose **K-Nearest Neighbors (KNN)** as my initial model for the following reasons:

- KNN is simple and well-suited for small datasets.
- It doesn’t require long training time.

- It provides reliable performance when paired with good feature extraction.

To improve performance further, I found in a paper that combining **PCA (Principal Component Analysis)** with KNN achieved **94% accuracy**. PCA helps reduce the dimensionality of image data while retaining critical facial features. This reduces computation and improves classification.

5. Thresholding and Unknown Detection

One challenge I discovered is that KNN **always predicts a class**, even if the input doesn't match any known individual. To handle this, I plan to implement a **distance-based threshold**. If the nearest neighbor is too far, the system will classify the face as "Unknown." This approach is commonly used in face recognition literature.

6. Evaluation Metrics

We have planned to evaluate model performance, for both recognition and rejection. For example:

- **Accuracy** (for known faces)
- **Precision, Recall, F1-Score**
- **False Acceptance Rate (FAR), False Rejection Rate (FRR)** – especially relevant for unknown face detection

7. Future Work and Deployment Plan

To deploy the model I plan to build a simple **web app** using:

- **HTML/CSS** for the frontend
- **Django** for the backend
- **Model integration** via .pkl files

8. Summary

So far, I have completed the following steps:

- Conducted initial research and literature review
- Explored dataset options
- Finalized KNN as the first model i will try and PCA for dimensionality reduction
- Learned about threshold-based rejection for unknown faces
- Planned the system deployment architecture

In the next steps, I will focus on finalizing the dataset and training the model

9. Paper Links

1. https://www.researchgate.net/publication/387406529_Optimizing_Face_Recognition_with_PCA_and_KNN_A_Machine_Learning_Approach
2. https://easychair.org/publications/preprint/gS7Q