PROGRESS UPDATE REPORT



AttendanceEye: A Computer Vision Attendance System

Mohsin Ali Mirza (20K-0353), Muhammad Waleed Gul (20K-0259), Mehwish Sameer (20K-1895)

**Supervisor:** Dr Muhammad Waqas **Co-Supervisor**:Dr Muhammad Rafi National University of Computing and Emerging Sciences (FAST-NUCES), Karachi, Pakistan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date  Completed | Task/Module | Status | Objective | Progress Completed |
| 28-Dec-23 | Face Detection | COMPLETE | Face Detection Using Yolov8 - 0.96 Accuracy | 100% |
| 18-Feb-24 | Face Recognition | COMPLETE | Face Recognition Using Face-Recognition Library | 100% |
| 21-Feb-24 | User Interface - Client Side | COMPLETE | User Interface which will be used by  students/employees to mark their attendance | 100% |
| 24-Feb-24 | Multiple Camera Integration | COMPLETE | Multiple cameras can be used to mark attendance for multiple students considering | 100% |
| 10-Feb-24 | Database  Connection With Frontend | IN PROGRESS | Database is prepared using MongoDB. Cloud setup for database is pending. | 95% |
| N/A | Batch Inference | IN PROGRESS | Use a single camera to capture multiple faces at once for face recognition | 15% |

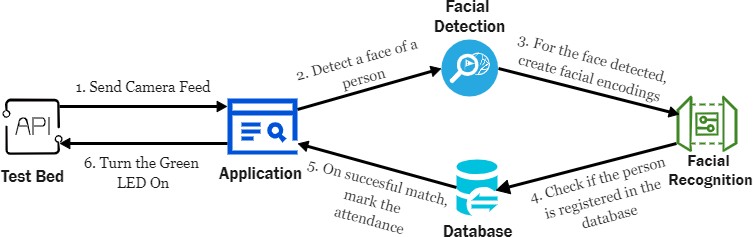
# Project Progress Summary

Following the successful completion of face detection using YOLOv8, achieving an accuracy of 0.96, our subsequent task was to implement face recognition and develop a client-side UI for attendance marking. In the pursuit of efficient face recognition, we explored two prominent libraries: Deepface and FaceRecognition, with and without YOLO integration. After thorough experimentation, we determined that face recognition with YOLOv8 yielded the best performance, boasting an accuracy of 0.777 and an inference time of approximately 0.4 seconds.

In addition to enhancing the face recognition algorithm, we seamlessly integrated the frontend with the database to facilitate the addition of more registered students/employees for recognition. Furthermore, we successfully achieved the final milestone for the first phase deliverable, which involved multiple camera integration. This advancement enables simultaneous face recognition from multiple cameras, ensuring that each camera feed identifies only one student at a time.

Currently, our focus is on fulfilling the last deliverable, which entails implementing batch inference for capturing and recognizing multiple students/employees with a single camera. This enhancement will significantly enhance the efficiency and scalability of our face recognition system.

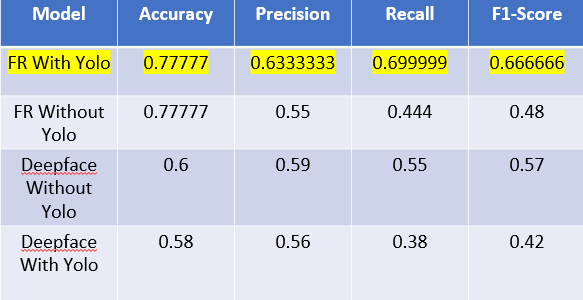
Overall, our project has progressed steadily, with each milestone bringing us closer to a robust and reliable face recognition solution tailored for attendance management in educational and corporate settings.



# Milestone 1: Face Recognition Accuracy and Results:

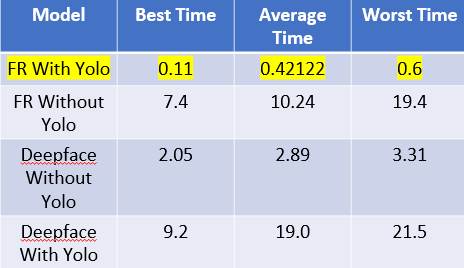
We experimented with two different face recognition libraries: **Deepface** and **FaceRecognition**, both **with and without YOLOv8 for face detection**. The **best performance** was achieved by **Face Recognition** with an **accuracy of 0.7777**, precision of 0.63333, recall of 0.6999, and an **F1 score** of **0.666666**. Following closely was FaceRecognition without YOLO, also with an

accuracy of 0.7777 but with an overall F1-score of 0.4. Deepface without YOLO yielded an accuracy of 0.6 and an F1 score of 0.57. However, Deepface with YOLOv8 only achieved an accuracy of 0.58 and an F1-score of 0.45, rendering it incompatible with YOLOv8.



# Inference Time:

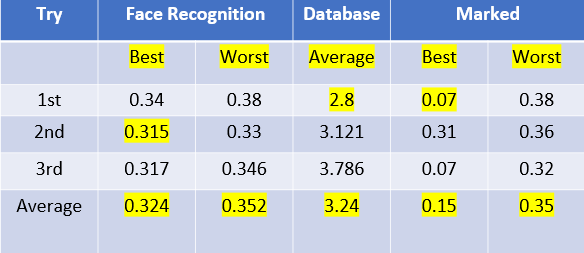
The **best inference time**, which is the time taken to complete the face recognition process, was on **average 0.4 seconds** for **Face Recognition with YOLO**, 2.89 seconds for Deepface without YOLO, 10.24 seconds for Face Recognition without YOLO, and 19.0 seconds for Deepface with YOLO.



# Milestone 2: Multiple Camera Integration Results:

With the constraint of having only one student in the feed for each camera, we successfully synchronized the database with attendance records. We were able to synchronize up to three cameras, with an inference time for face recognition averaging 0.324 seconds. The roundtrip

time for accessing data from the database was 3.24 seconds, and the visual feedback from the UI averaged 0.3 seconds, resulting in a total average processing time of 3.624 seconds.



# Milestone 3: Frontend Integration with Database

The frontend has been successfully integrated with the database, allowing users to add registered individuals to the database for future face recognition purposes. However, we still need to migrate the database to the cloud to establish seamless connectivity between the frontend and

server-side for face recognition.



# Milestone 4: User Interface for Client-Side Face Recognition

The user interface for client-side face recognition has been developed with four different states. When active, users can present their faces to the camera for recognition. If a match is found in the database, attendance, name, ID, and department details are displayed. If the person's

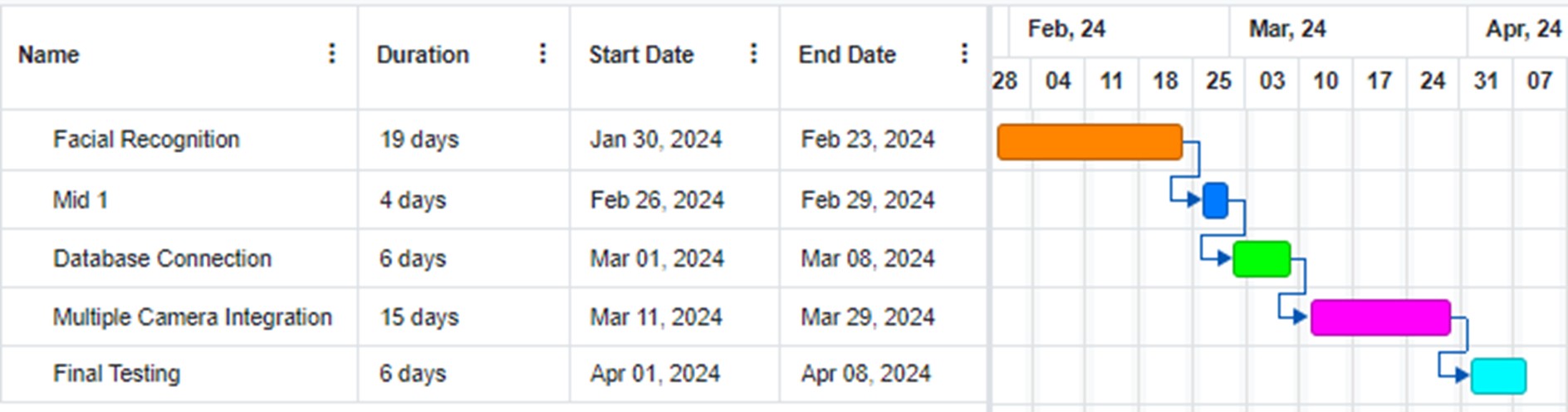
attendance has already been marked; it will indicate "Already Marked" with a 30-second timer to prevent repeated attendance marking. This timer can be adjusted to suit the company's requirements.

# (To Be Completed)

**FYP-2 Final Deliverable: Batch Inference Using a Single Camera**

The final deliverable entails implementing batch inference (multiple face recognitions) within the feed of a single camera. We need to explore whether our current models can support batch inference or if better alternatives are available. Additionally, multiple stress and performance tests need to be conducted to ensure the creation of a viable and effective product.

# FYP - Timeline OLD



**FYP - Timeline NEW**

