Task 1: Attendance System with Emotion Detection

Submitted by: Tarrush Saxena

Date: December 2025

1. Introduction

This report documents my work on Task 1 for my internship at Null Class. The project, titled 'Attendance System with Emotion Detection', is a computer vision application designed to modernize classroom attendance. The goal was to build a system that not only marks attendance automatically but also provides insights into student engagement through emotion analysis.

2. Background

Traditional attendance methods are time-consuming and prone to errors. Taking roll calls manually wastes valuable class time. Furthermore, instructors often lack real-time feedback on whether students are engaged or bored. This project addresses these issues by automating the process using facial recognition and adding a layer of analytics with emotion detection (using the FER2013 dataset).

3. Learning Objectives

My primary objectives for this task were:
- To understand and implement real-time Face Detection and Recognition.
- To learn the fundamentals of Deep Learning by training a CNN model for Emotion Detection.
- To integrate multiple AI models into a single, cohesive user interface.
- To handle real-world data challenges like lighting variations and false positives.

4. Activities and Tasks

The project execution involved several key activities:
- Data Collection: creating a script to capture training images for the face recognition system.
- Model Training: Training an LBPH recognizer for faces and a CNN model (using Keras) for emotions.
- System Integration: Writing the main 'dashboard.py' script to run the webcam feed and process frames.
- Logic Implementation: Coding the 'Time Fence' feature to restrict attendance marking to specific hours.

- Testing: Evaluating the models using confusion matrices to ensure accuracy.

5. Skills and Competencies

Through this project, I have developed technical proficiency in:
- Computer Vision: OpenCV (Haar Cascades, LBPH).
- Deep Learning: TensorFlow/Keras (CNN architecture, Model Checkpointing).
- Python Programming: pandas for data logging, os for file management, and datetime for time logic.
- GUI Development: Building interactive desktop applications with Tkinter.
- Problem Solving: Debugging real-time video processing loops.

6. Feedback and Evidence

The system has been tested and verified locally. Evidence of its functionality includes:
- Successfully generated 'assets/cm_emotion.png' showing the confusion matrix for the emotion model.
- Real-time attendance logs generated in the 'reports/' directory.
- Visual confirmation of face bounding boxes and emotion labels on the live camera feed.

7. Challenges and Solutions

I encountered and resolved several meaningful challenges:
- Issue: The system was marking attendance continuously, flooding the CSV file.
  Solution: Implemented a check to only mark attendance once per session per student.
- Issue: Poor accuracy in low light.
  Solution: Added histogram equalization to normalize image brightness before processing.
- Issue: Dependency conflicts (e.g., numpy version issues).
  Solution: Created a strictly versioned 'requirements.txt' to ensure stable environment setup.

8. Outcomes and Impact

The final outcome is a robust, standalone application. It successfully reduces the time taken for attendance from minutes to seconds. The added emotion metrics provide a novel way for instructors to gauge lecture effectiveness, potentially improving teaching quality.

9. Conclusion

Completing Task 1 has been a rewarding experience. I have successfully translated theoretical knowledge of AI into a practical, working tool. This project has laid a strong foundation for my future work in Computer Vision and I look forward to applying these skills in upcoming tasks.