

Report Document: Classroom Assistant - Phase 3 Final Report

Team Number: 29

Edwin Lopez (elope083)

Jordan Kaatz (jkaat001)

Overview

Facial Recognition Classroom Attendance Management System is a solution designed to automate the process of attendance taking in educational settings. By leveraging the power of facial recognition and Google Cloud technology, this system aims to address the inefficiencies and inaccuracies associated with traditional manual attendance methods. Boasting extra features such as stranger detection and scalability.

Use Case

- Automating Attendance: Streamlines the attendance-taking process, making it faster and more accurate.
- Enhanced Accuracy and Efficiency: Utilizes facial recognition to ensure accurate identification and attendance recording.
- Unnoticed Presence: Capable of recognizing individuals who may be present in the class but not accounted for in traditional attendance methods.
- Validation and Verification: Attendance is validated through logs and captured images, ensuring authenticity.
- Data Management: Simplifies the export of attendance data for further analysis or record-keeping.

System Design

- Facial Recognition: Utilizes camera and facial recognition technology to capture and log attendance.
- Data Management: Uses Google Cloud Firestore for record-keeping and Firebase for image storage.
- Admin Interface: An admin page facilitates attendance management, data export, and monitoring of unrecognized individuals.
- Jetson Nano: Employs a Jetson Nano to run facial recognition software, updating attendance and images in real time with cost-efficiency considerations.

Services and Technologies:

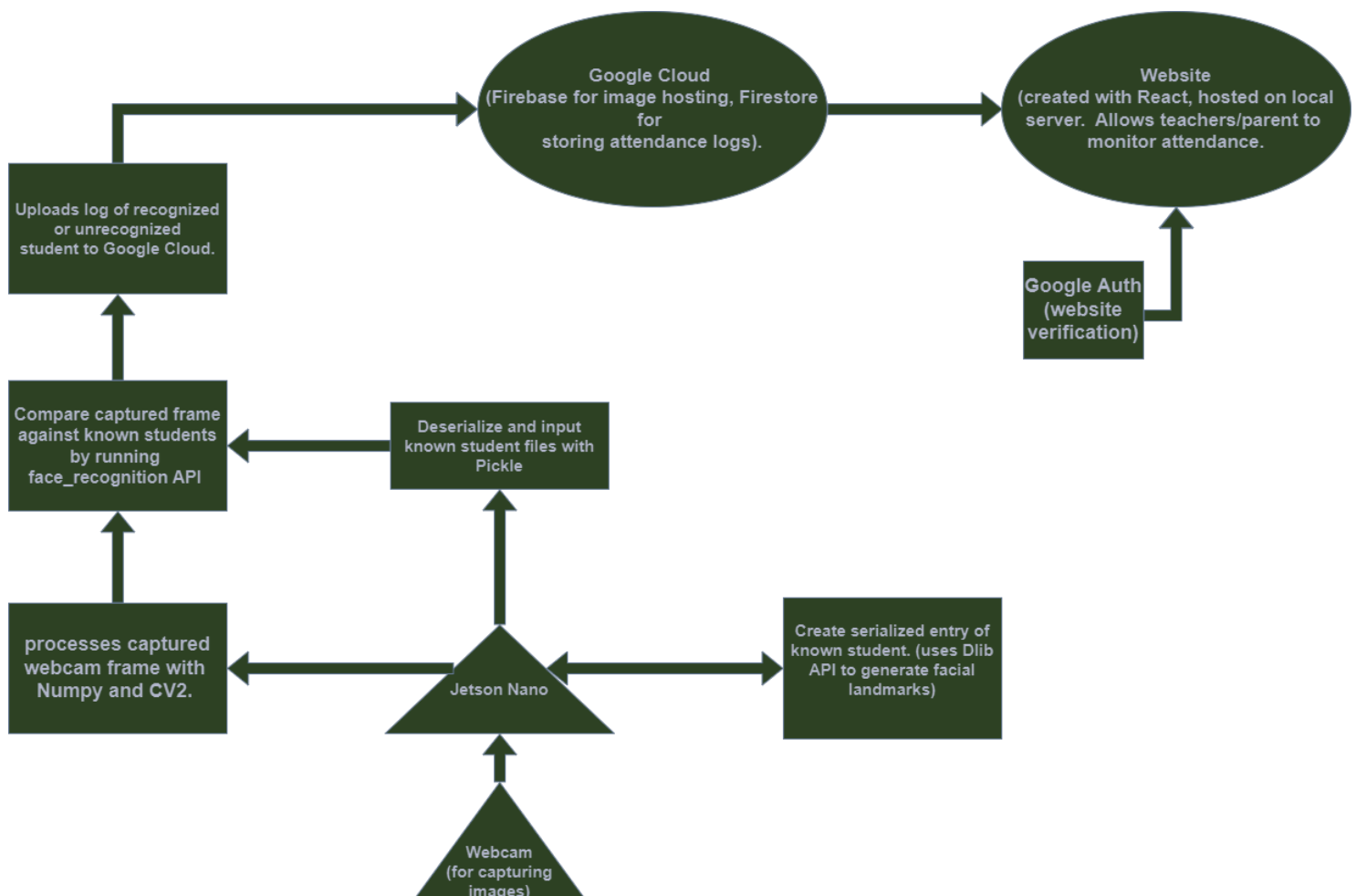
- Image Processing: Utilizing CV2's library's functions Videocapture() and resize() functions to capture and process images for use with face_recognition and dlib.
- Importing Known Faces: Utilizing the dlib library's get_frontal_face_detector() function for initial face detection. Also employing a shape predictor with 68 face landmarks from dlib to adjust and identify facial features, improving the accuracy of face recognition.

- Deserializing known faces: Utilizing Pickle's library's load() function to load and deserialize the known faces and known names files.
- Facial Recognition: Utilizing the face_recognition library's face_locations() and face_encodings() function for initial face detection in captured images. Using its compare_faces() function to find matches to known faces.
- Data Hosting:
 - Firestore: stores logs of classroom attendance.
 - Firebase: stores images of captured faces.
- Website Creation: Utilizing React website technology for website creation. For our project, the website was hosted on our laptop, and that data from Firebase and Firestore is pulled into the website.

Admin Page Features

- Attendance Filtering: Allows filtering of attendance records by class and date.
- Data Export: Enables downloading of attendance lists for specific classes and dates.
- Monitoring: Provides tools for monitoring unrecognized individuals.
- Security: Secured with Google Auth for admin login.

Network Diagram



Device Catalog:

- Jetson Nano 4GB Developer: Positioned within the classroom (with attached VSILE 1080P 110-degree webcam)
 - Is responsible for automated attendance through facial recognition.
 - Enhances security by detecting unauthorized people in the classroom.

Challenges:

- It took some time to learn and implement integrating the React, Google Cloud (Firestore/Firebase) and the facial recognition program together. Getting the Firestore/Firebase log to store the correct student ID as well as getting the Jetson Nano to only upload one log of each person per hour proved to be difficult. It also took some time to get our react website to properly display the correct ID and picture.
- A large portion of the time working on this project was spent trying to get our desired libraries installed and running on the Jetson Nano. We spent hours trying to install and reinstall dlib along with installing and reinstalling various versions of python to try to get dlib working.
- Working on the project between the 9:00am to 5:00pm window in the makerspace (up until week 9) was not always easy. We often did not have much overlap in free time available to work on the project together during this window. There were a couple occasions where we were installing either Python or Dlib in the makerspace, but had to cancel and postpone the installation because the makerspace was closing or we had to go to class. We were, however, granted permission from the professor at the beginning of week 9 to borrow and take home a Jetson Nano so that we could work on the project beyond the 9:00-5:00 window.

Quality Attributes:

- Website utilizes user ID and password to ensure that only authorized users have access to student attendance logs and images.
- Our system only uploads one log/image of each student per class period. This helps with readability and reduces the cost of storing data on Google Cloud.
- Logs and images are securely stored using Google Firestore and Firebase
- Jetson Nano runs facial recognition software and website and cloud take care of everything else to ensure reliability

Messaging Patterns and Communication Protocols:

- Utilizes Google Cloud Storage messaging/communication protocols
 - for uploading images to firebase:
 - Connects to cloud storage service
 - Uses bucket created on the cloud server to send files to
 - Creates a Blob (binary large object) to send the jpeg files in
 - for uploading log entries:
 - Connects to cloud storage service
 - Updates 'log_entries' document with the students name, unique ID, and the timestamp.

Future Improvements

- Enhance the React interface for better image timestamping.
- Integrate automatic features for online lectures, including zooming, recording, and a question system.
- Implement a notification system for strangers.
- Frame by frame tracking to avoid one person being detected as multiple people do to different angles.
- With the prior improvement, on the fly facial recognition training