The University of Texas at Arlington

CSE-5333-CLOUD COMPUTING
Project Proposal

“Facial Recognition Attendance System”

Vishnu Gopal Rajan	Chinmaya B Venkatesh	Ipsa Mishra
1001755911	1001778014	1001759616

1. Abstract

Automatic attendance management will replace the manual process, which takes a lot of time. And it's hard to manage. There are several biometric processes in operation, identification of the face is the best form. . In this process, the camera is fixed in the classroom to capture the image or the instructor uploads the picture, the faces are identified, and then the image is recognized and the attendance is updated in the database.

2. Introduction

Automation is when the context and variables are set, and a certain task must be performed on the basis of these factors. This can be called automation because this job is performed automatically. Automation is all around us now! It may not be advanced automation, which involves complicated calculations and structures with sensors and actuator controls, but it is still automation. Examples of everyday automation that most people encounter on a daily basis; washing machine, dishwasher, refrigerator, bus door, car air conditioning systems, converting to a full home theater system with only one press, etc.

AI-based computer vision software is highly applied in a wide variety of manufacturing and customer-related applications. But wherever it is implemented, it usually brings productivity and creativity across the line to what is possible in the automation market.
AI-powered machine vision systems or software optimize the production process within the automotive sector vertically where any mistake in a highly competitive industry can be a waste of time.

Usage of facial recognition for the purpose of attendance

- Marking is the smart way to handle attendance System.
- The identification of the face is more precise and quicker using this technique, among other methods, and decreases the probability of proxy's attendance.
- Face recognition include passive recognition, Identification of the person to be named. Does not need to take any decision on its identification.

3. Background

Many Research papers have explored the idea of automating the attendance management in a classroom using face recognition with a wide variety of mathematical models to recognise the models. Eigen faces were used in one of the papers where a set of Eigenvectors are used in computer vision part of facial recognition.
Most of the papers first find the faces by the method of sliding windows to detect faces. Then they get some feature vectors such as HoG features to map the face and use a CNN model to recognise the label or the person.

3.1 Our Goals:

- Create a Face Recognition program to process classroom picture frames to provide a full attendance report without the challenge of calling out student names.
- Interaction between GCP resources such as Compute Engine, Cloud Storage and Cloud SQL in order to provide a flawlessly functioning framework.
- Provide a contactless and non-invasive means of attending classrooms.
- It also favors time efficiency and prevents human error in attendance.
- Provide fool proof method of taking attendance.

Future Goals

- Sentiment Analysis in classroom using Facial Features
- Eye tracking to maintain focus levels in classroom

Image 1

3.2 Components of our project:

Computer Vision:
Computer vision is an interdisciplinary research area that focuses on how computers can achieve high-level understanding from visual images or videos. From the point of view of engineering, it aims to understand and automate the tasks that the human visual system can perform.

Computer equipment can capture images with clarity and information that far surpasses the human vision system. Computers can also detect and quantify color variations with very high precision. With advances in ML and DL, computer vision has advanced to the degree that it can exceed humans in certain tasks related to the identification and marking of items.

Cloud Components:

Google Cloud Platform (Cloud Service Provider) : Cloud computing is the delivery of on-demand computing services -- from applications to storage and processing power -- typically over the internet and on a pay-as-you-go basis. Computer vision with ML requires high computing and storage requirements with low latency reads and writes.
No more buying servers, updating applications or operating systems, or decommissioning and disposing of hardware or software when it is out of date, as it is all taken care of by the supplier.
Using cloud services means companies can move faster on projects and test out concepts without lengthy procurement and big upfront costs, because firms only pay for the resources they consume.

4. Project Design:

The language that we are using is python 3.0 as we can import a lot of machine learning frameworks which are pre-built into the language. We are using the dlib libraries which run on cv2 or OpenCV framework to use it in face recognition as a multistep pipeline.
The workflow is designed as follows:
The frames from the real time video capture are passed through a set of functionalities listed below.

4.1 Face Detection

The first step in facial recognition is Face detection. The HOG algorithm gives the gradient for a black and white picture based on the oriented gradients. It basically has pointers from lighter area to darker area in an image as shown in Image. Using a sliding window to scan through the pixels from top left to bottom right corner of the image. We use the sliding window technique to detect whether the face exists in a given frame or not.

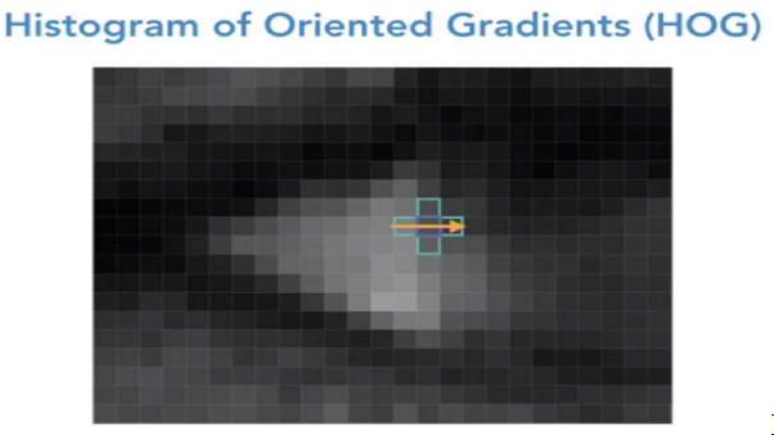


Image 2 : HOG

4.2 Face Landmark estimation

The second step of our face recognition pipeline is called face landmark estimation. Face landmark estimation is where we identify key points on a face, such as the tip of the nose and the center of the eye. On the left is a face that we extracted from a photograph using face detection model. On the right is what that image looks like after we use face landmark estimation to detect points on the face. Each of the points we've detected on the face is called a face landmark because it represents a well-known place on the face like the tip of the nose or the edge of the eyes. Face landmark estimation works by starting with a known set of points that should appear on any face. Then it moves those points around until they match the face image. In this case, we've located the eyebrows, eyes, nose, lips, and chin line. And because the points are predefined, we should already know which of the points make up the eyes, which make up the nose, and so on, so we can connect them with a line.

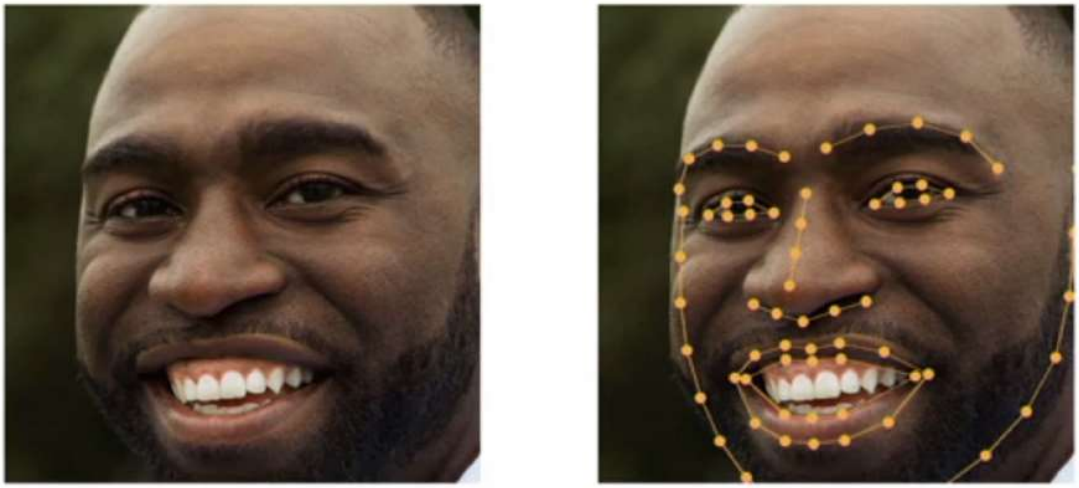


Image 3: Landmark Estimation

4.3 Face Encoding

Representation of the face using a set of measurements. As a human, you can tell two people apart even if they look very similar because you recognize differences in the structure of their faces. We want our algorithm to be able to do the same thing. The solution is to come up with a way to measure each face. For example, we might want to measure the width of each eye, the size of the cheekbones, and the width of the mouth, and so on. Then, if we want to compare two faces to see if they match, all we need to do is see if they have similar measurements. Then, if we want to compare two faces to see if they match, all we need to do is see if they have similar measurements. If the measurements of two faces are close enough, we can assume they are the same person. This is a better approach than comparing images directly because it takes the structure of the face into account. Also, it is much quicker to compare a few measurements than it is to compare an entire image with thousands of pixels.

```
[-0.23, -0.54, ..., 0.27]
```

Image 4: Face Encodings

4.4 Face Recognition

Euclidean distance is just the distance between two points in space along a straight line. It can be hard to visualize a model with 128 measurements. Using Euclidean distance to check for face matches has a lot of nice properties. First, calculating Euclidean distance is fast, so it's easy to compare an unknown face against a large set of known faces quickly.

- This system is built using OpenCV and Deep Learning with the use of *scikit-learn* library.
- Scikit-learn library helps Detect faces, Compute Face embeddings, Train SVM model using embeddings and Recognize faces.
- To detect faces and recognize, the library extracts 128-d feature vectors to quantify each face.

These features are used to match with unknown faces in input class images in order to recognize the students in the classroom

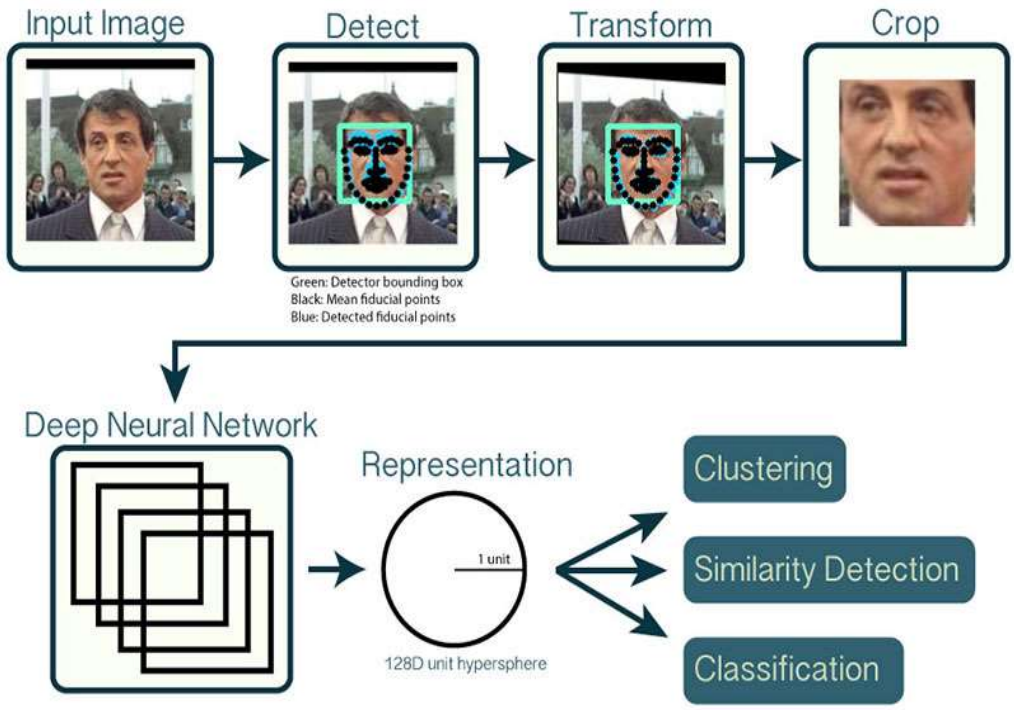
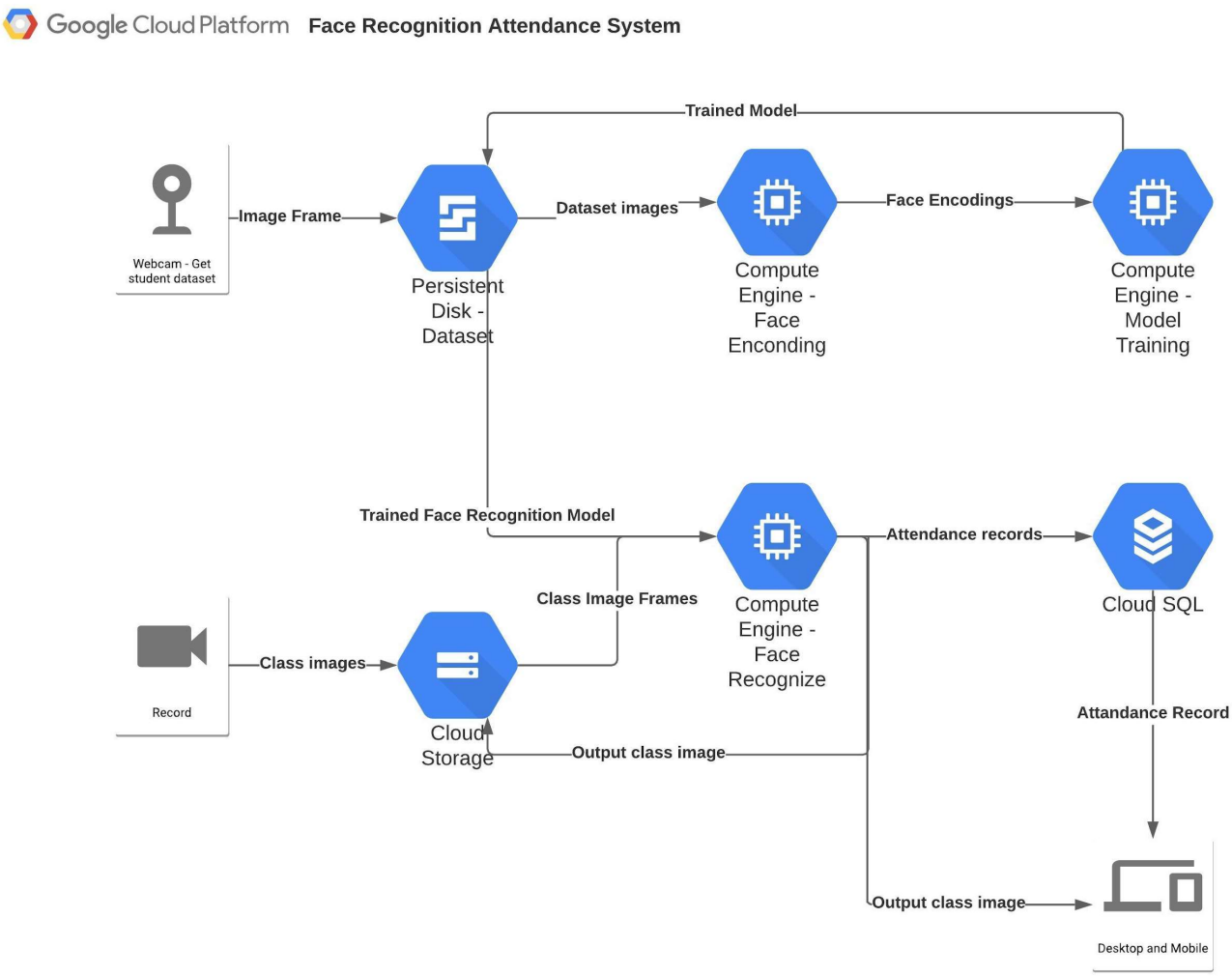


Image 5 Overview of OpenCV face recognition pipeline. https://cmusatyalab.github.io/openface/

5. Architecture:



The Architecture above represents the conceptual diagram of the system we have built in place. We have used the Google Compute Engine to carry out the computations necessary to achieve the task at hand. It serves as the VM instance which is used to host our application.

The students upload the images when they register which is stored in the persistent storage and used as the dataset for training the model. Next the encodings are extracted in each of the images/video uploaded which is stored in a pickle file. Then these encodings are used to train the model which is then used in the deployment cycle.
The model which is trained based on the uploaded pictures is also stored locally on the Compute Engine.

The Teacher uploads or has a camera in class which sends the data to the application or the compute engine whenever the class is held(ideally). Now a copy of the image is stored in Google Cloud Storage and it goes through the facial recognition pipeline to get classified as one of the students who attended the class. This classified image is stored into Google Cloud Storage and can be accessed by the professor in case a student raises an issue with the Attendance report. After the list of students are returned from the face recognition module the Flask API code along with mysql connectors updates the mysql instance on CloudSQL which authenticates using certificates. These updated attendance reports are viewed by the students once they login to their accounts.

6. Implementation:

Cloud Service: Google Cloud Platform:

We have made use of a custom dataset and a custom ML model in this project. This Custom Face Dataset, ML model and python OpenCV code is stored on Compute Engine Instance Persistent Disk.

This project is deployed on top of **Conda Environment (Python)** in the instance. The custom dataset is trained using the custom ML model in the Instance when a new user is added to the dataset.

The Compute Engine Instance communicates with the front end using **Flask API**. This enables us to connect to the Database and Compute engine from the front end.

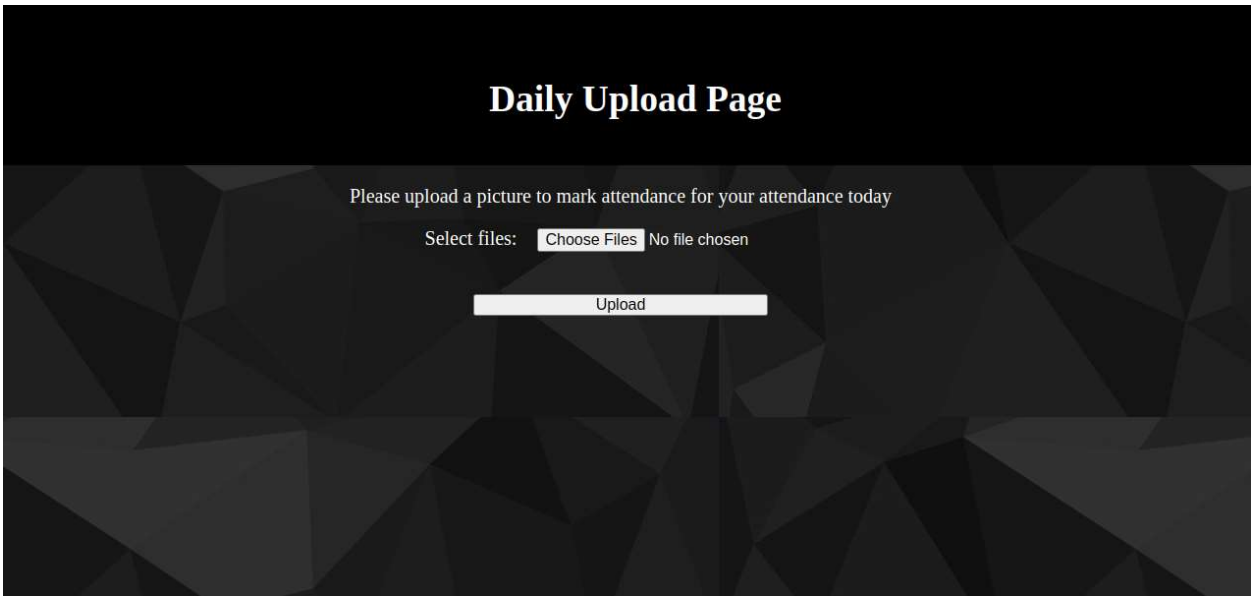
We have connected our Application to the CloudSQL as an external application using SSL certificates where the host IP is public and the External Application Authenticates with the certificates generated and downloaded from the CloudSQL instance.

The many required libraries that were used in building this project are listed in the requirements.txt file which can be viewed using a text editor. We have also used blob storage to interact with Cloud Storage using the Gcloud storage API.

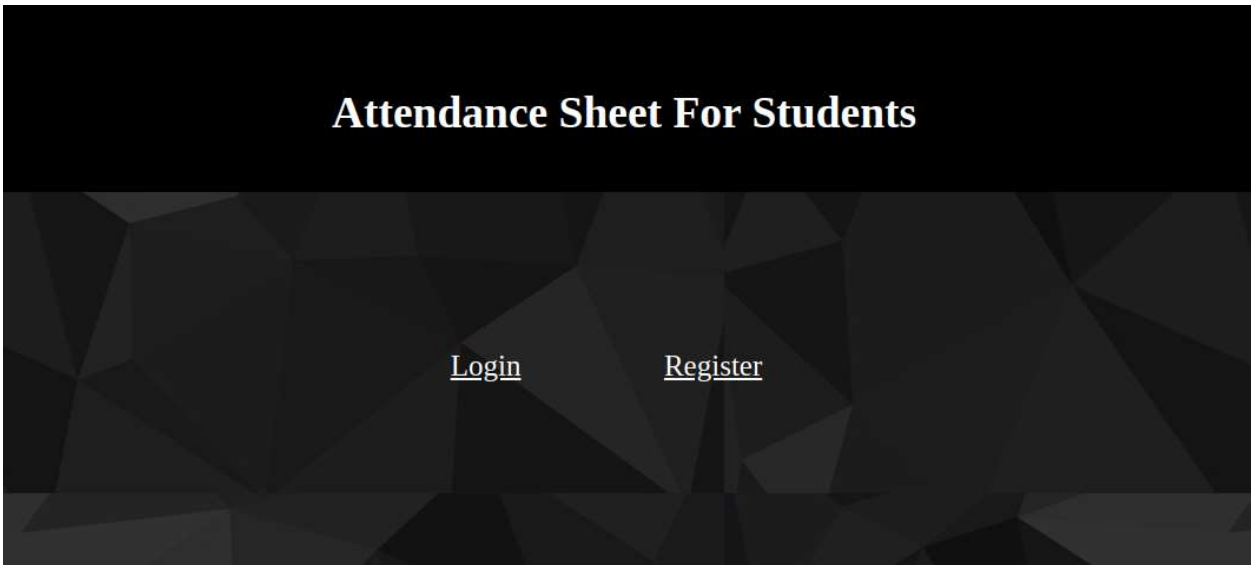
The front end consists of login page, register page and attendance page for students. It contains daily image upload page for the teachers. We have used html and css for designing the web framework UI. We have used various html tags such as meta tags, header tags, footer tags and style tags for making the page responsive and html buttons

for login, register and uploading multiple images. The register page contains name field, email field, password and confirm password field and an image upload button.

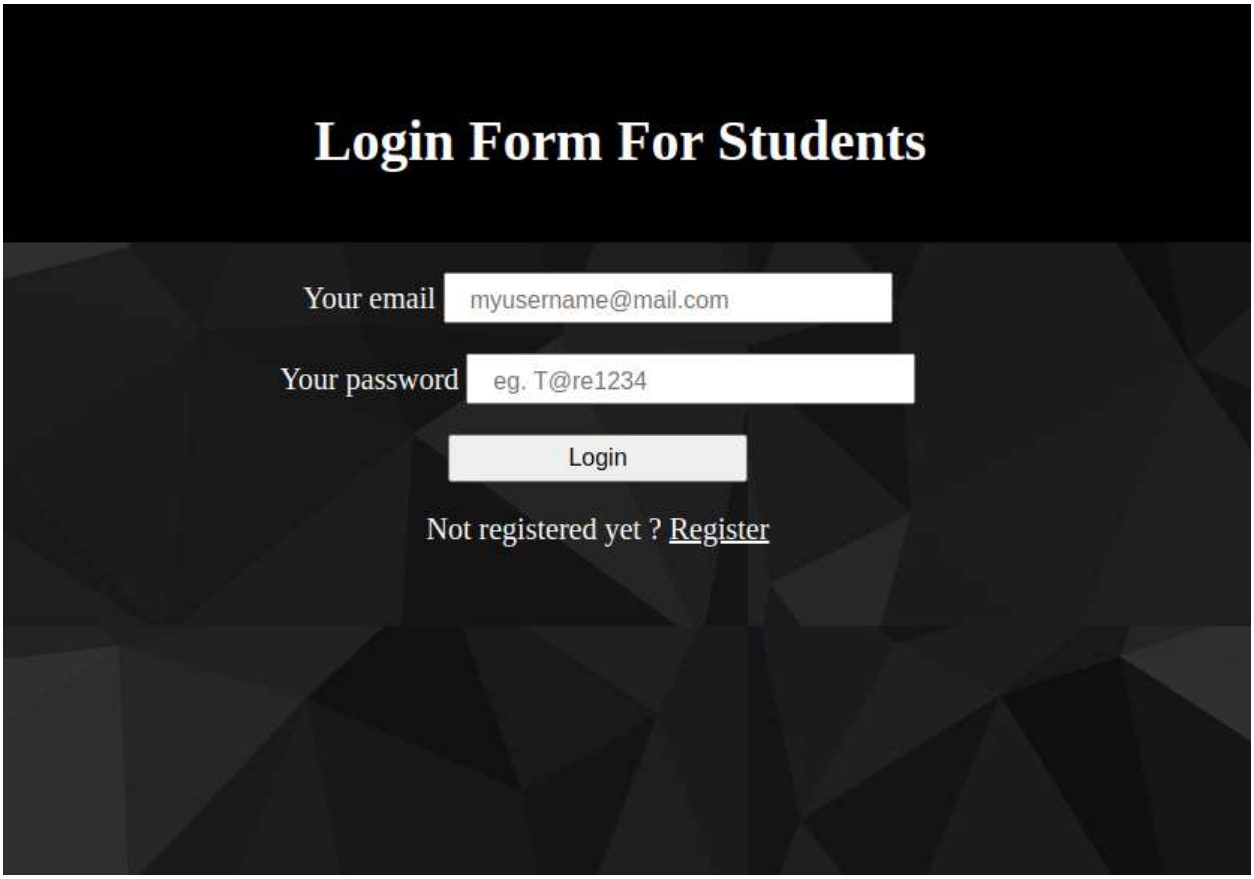
This page allows a student to enter their details and upload 5 photos of them and register as a new user. Then it goes to the attendance page and displays the percentage as 0% as the student has just registered as a new user and there is no attendance to show. The daily image upload page allows the teacher to upload class images for that day. As soon as he uploads the class image, it passes through the model trained for recognizing faces and updates the attendance of each student for that day by recognizing their faces. Then the login page contains email field and password field. It allows a student to login and view their attendance report.



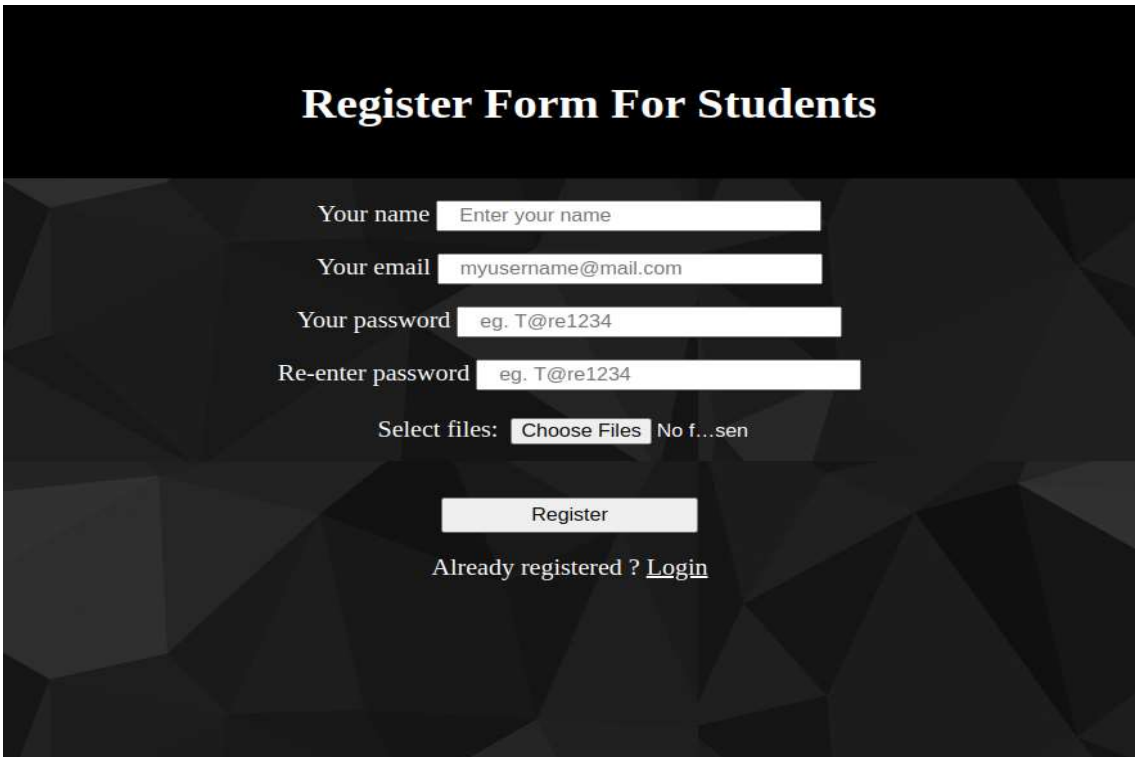
Upload Screen for when class was held



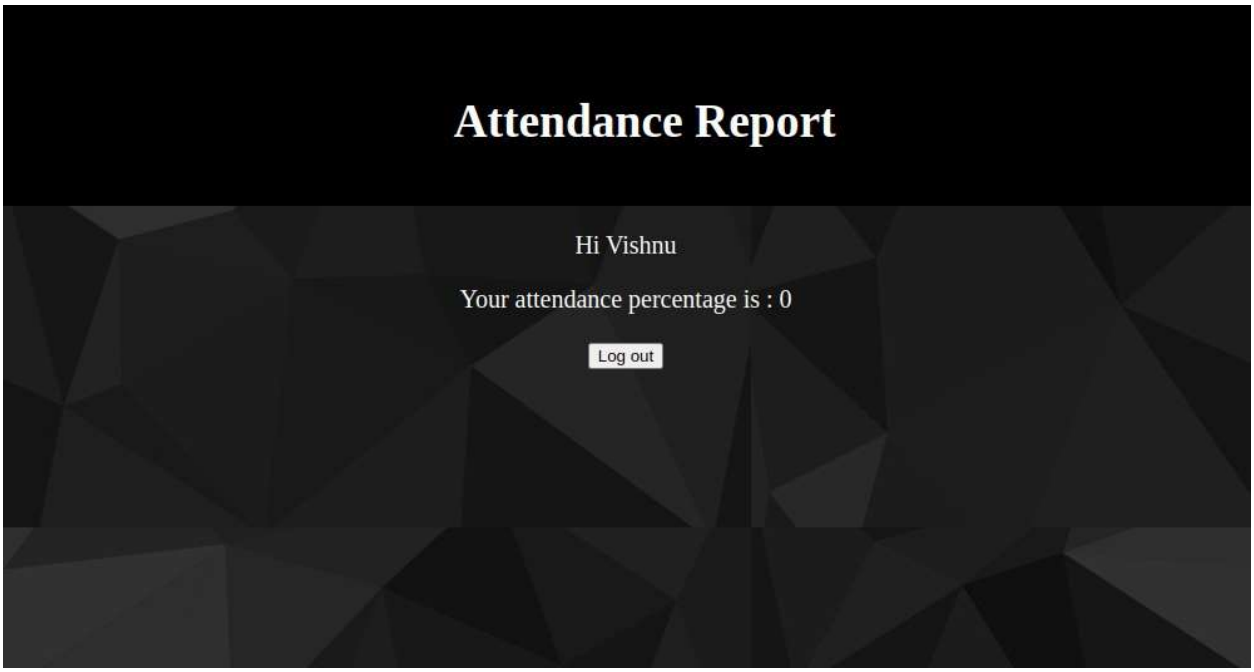
Student Home Page



Student Login Page

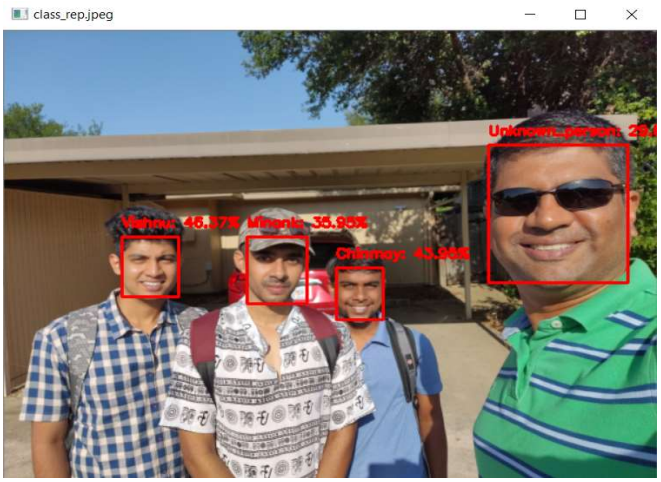


Register for Student



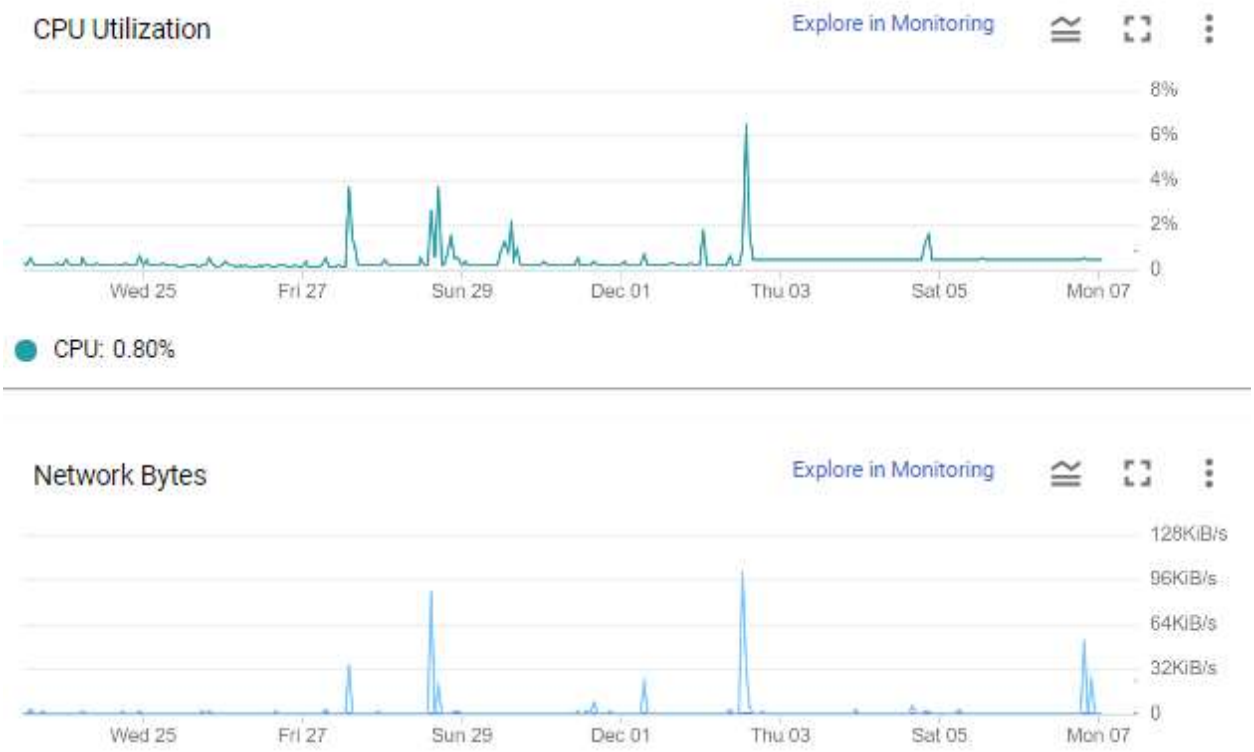
Attendance Report

7. Output


class_rep.jpeg
Unknown_person: 29.
Vinay: 46.37%
Minanki: 35.95%
Chinmaya: 43.98%


Minanki: 50.83%
Chinmaya: 57.17%

8. Testing



Custom ML Model , Custom Dataset running on Compute Engine Instance connected to Cloud storage and Cloud SQL hosted on Flask API.

SVM model : <2% CPU , <16KiB/s
Face_recognition dlib : ~5% CPU, ~100 KiB/s

Unittest module was used to conduct unit tests on functions in python by using assertion based testing. Web testing was performed to verify the links and other interactions that the user can have. Integration tests was performed to verify that the modules integrate to create the desired result and no malfunctions while working together. As we are using a light weight model we can see that the CPU utilization is less.

Advantages of Cloud:

- ***Scaling***: Our dataset is ever growing in size and requires scaling which is provided with Cloud Storage and Cloud SQL.
- ***Computing Power:*** The Cloud instance can be customized to provide the best performance for our service, We can upgrade computing power with GPU for the use of larger dataset and better ML model.
- ***Reduced Costs*****:** No costs on hardware or software. Easy deployment with reduced costs.
- ***Flexibility*****:** Cloud resources can be flexible during the different times of the School year to improve efficiency.
- Easy, Contact-less and non-invasive method of collecting and viewing attendance records on a cloud based service.

Problems encountered:

- Unavailability or hard to use input peripherals like cameras to get image or video real time feed
    - Could not enable real time dataset collection during registration using webcam.
- Difficult to maintain and keep track of the state of data between each Cloud instance.
- Unavailability of GPU in all regions. Test performances:
    - SVM Model (25 images) : 6 seconds
    - face_recognition dlib (25 images) : 60+ seconds

9. Conclusion

- Thus, we have used various components like computer vision, google cloud platform services like compute engine, cloud storage, cloud sql and web technologies such as html and css for our work.
- This work designs a face recognition system which can take the attendance of students without the hassle of calling out their names for attendance. This process is time effective and less prone to errors as everything is done in an automated manner over here.

- This also avoids proxy entries which makes the attendance process inconsistent. It is feasible for students and teachers to view the attendance report from anywhere. This process is contactless and non-invasive which makes it way more useful.
- For future work, we have thought of implementing sentiment analysis in classroom which would allow us to know the moods of the students. We have also thought of enabling eye tracking which would be able to detect the attention and focus of the student.

10. References

[1] RoshanTharanga, J. G., et al. "Smart attendance using real time face recognition (smart-fr)." Department of Electronic and Computer Engineering, Sri Lanka Institute of Information Technology (SLIIT), Malabe, Sri Lanka (2013)

[2]Anil K. Jain, Arun Ross and Salil Prabhakar, “An introduction to biometric recognition”, Circuits and Systems for Video Technology, IEEE Transcations on Volume 14, Issu 1, Jan 2004 Page(s):4-20

[3] E. Varadharajan, R. Dharani, S. Jeevitha, B. Kavinmathi and S. Hemalatha, "Automatic attendance management system using face detection," 2016 Online International Conference on Green Engineering and Technologies (IC-GET), Coimbatore, 2016, pp. 1-3, doi: 10.1109/GET.2016.7916753.

[4] J. Lee, J. Bang and S. Yang, "Object detection with sliding window in images including multiple similar objects," 2017 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, 2017, pp. 803-806, doi: 10.1109/ICTC.2017.8190786.

- https://www.pyimagesearch.com/2018/02/26/face-detection-with-opencv-and-deep-learning/
- https://www.pyimageconf.com/
- https://github.com/ageitgey/face_recognition
- https://towardsdatascience.com/how-to-do-everything-in-computer-vision-2b442c469928