Class attendance using facial recognition.

**Group members:**

Tanuja Polineni

[tanujapolineni@my.unt.edu](mailto:tanujapolineni@my.unt.edu)

Yaswanth Bandaru

[yaswanthbandaru@my.unt.edu](mailto:yaswanthbandaru@my.unt.edu)

Mahan Ghosh

[mahanghosh@my.unt.edu](mailto:mahanghosh@my.unt.edu)

Sampreeth Nelavelli

[sampreethnelavelli@my.unt.edu](mailto:sampreethnelavelli@my.unt.edu)

Tejaswini Mallela

[tejaswinimallela@my.unt.edu](mailto:tejaswinimallela@my.unt.edu)

|  |  |
| --- | --- |
| Tanuja Polineni | Project lead will be involved in the overall workflow of the project, data integration with the GUI and document preparation. |
| Yaswanth Bandaru | Lead programmer will be involved in coding the backend and data integration. |
| Mahan Ghosh | Will be involved in coding the front end, data integration and documentation. |
| Sampreeth Nelavelli | Will be involved in data set training and documentation. |
| Tejaswini Mallela | Will be involved in data set training and documentation. |

**Abstract:**

The objective of the proposal is to recognize a face of a person and note it to the attendance registry. The solution recognizes the face and displays the name of the person and marks present in the registry. We use a webcam to detect the faces and record the attendance.

This would help reduce the time spent in classes spent on roll calls and provide a feasible and accurate alternative to it.

This can further be extended to verifying the attendee during exams given online which is especially useful given the online classes becoming mainstream.

**Data Specification:**

The training data is derived from the mp4 video of each student being enrolled into the class during registration. Then the backend extracts 30 jpeg pictures from the video to use as the training data.

**Project Design:**

1. **MTCNN** or Multi-Task Cascaded Convolutional Neural Networks has been used for the face detection as it provides higher accuracy than any other method available.
2. Libraries used are NumPy, OpenCV and FaceNet had to be installed to achieve the face recognition.
3. FaceNet Keras is used to extract the features from the training data. From each picture 128 vector embeddings are extracted. This allowed for high accuracy with very low data training time.
4. SVM(Support vector machine) has been used for feature classification which creates an optimum hyperplane to classify the classes of training dataset based on different features of the face.
5. Django has been used to create the GUI that allowed us to create an elegant and user friendly UI for easy implementation of the attendance system.

**Milestones:**

1. Creating the backend code that can store and train the data into the model in real time and is able to detect the user, based on the trained model. **Python** will be used for the coding and **FaceNet Keras** for face detection.
2. Create a simple to use and register GUI for first time user complete with registration and login page. **Python, Django** and possibly CSS and Java will be used for the GUI.
3. Integrate the Backend model with the GUI and test its performance in real time.

**Hardware required at user end:**

* Web cam enabled laptop or desktop would suffice from the user end and a web browser.

**References:**

1. <https://medium.com/clique-org/how-to-create-a-face-recognition-model-using-facenet-keras-fd65c0b092f1> (MTCNN is the method suggested for the most accuracy for face detection at this blog and that is what we intend to use.)
2. <http://blog.altabel.com/2018/08/14/facial-recognition-technology-where-is-it-being-used-and-how-does-the-technology-work/>
3. <https://www.bayometric.com/identification-verification-segmented-identification/>
4. <https://learnopencv.com/face-detection-opencv-dlib-and-deep-learning-c-python/>