Project FACE Documentation

**F**ace **A**cknowledgement using **C**omputational **E**lements

# Concept

The concept used in this project is face recognition in which we capture a frame using OpenCV module and convert the frame from BGR to RGB channel and further change it to GRAY channel. Then the module reads the complete image pixel wise and marks arrows on it pointing from light to dark region. The obtained image is called HOG (Histogram of Gradient) which eliminates the issue of matching the image with the template image in varying brightness.

As the image we get have different orientation of face, it appears to the computer as completely different image. To resolve this problem, we use an algorithm called 'Face estimation landmark', which basically find 68 specific points that is present on every face. This algorithm uses machine learning to find these points and rotates the scale of these specific points and tries to centre the eyes and the mouth in the same position as the image in database. It then searches through the database and tries to match with every image. The image which has the closest resemblance and measurements is then given as our match.

As soon as the match is found attendance is marked in an excel sheet having details of student and dates using openpyxl and date & time modules. Openpyxl is also used to iterate through columns and calculate the attendance percent which is latter stored in the excel sheet.

A Condition is applied to check is the attendance is less than 75% and if yes an alert mail is send to the student using send email function which is coded in email.py. The send email function has arguments such as receiver’s email, sender’s mail, subject and message.

# Libraries used

Libraries used are:

* cv2
* numpy
* face\_recognition
* os
* workbook, load\_workbook (from openpyxl)
* date (from datetime)

# Features

* It is user friendly as it marks the attendance just by recognizing the faces and store the data in an excel sheet.
* It eliminates any human intervention as it is automated.
* It provides faster output compared to humans.
* The chances of making errors are very low.
* It does not require any heavy pre-installation set up.
* It sends warning to the specific users when their attendance is low.

# References

* <https://youtu.be/mYVS7QjNjvg>
* <https://youtu.be/7YS6YDQKFh0>
* <https://youtu.be/4AMsBrkkH00>

# Team details

* Pendyala Gnana Sai
* Anirudh Agrawal
* Rishabh Singh