# CSN - 254
# SOFTWARE ENGINEERING
# YouWho

Final Report
April 17, 2022
Version 1.0

Department of Computer Science and Engineering
Indian Institute of Technology
Roorkee

GROUP - 15

20114076 – Pratyush Kumar
20114050 - Lokendra Singh Parmar
20114033 – Dheravath Vikas
20114007 - Alok Raj
20114075- Prashant Bhookya
20114102 – Uday Jangir

## Github repository

## https://github.com/echidna11/YouWho.git

## Problem Proposal

YouWho is a modern face recognition web-application which uses Deep Learning to automatically detect all the faces in a given image and automatically identify the people therein. One major use-case of YouWho will be as an attendance taking system. YouWho would first require the input images of all the employees/students of the organization to train the model and fill up the dataset. Once this is done, whenever a person steps up to the camera installed, YouWho would automatically detect the name of the person and log their attendance in the backend, thus speeding up the whole attendance process.

### Technology Used

- Bootstrap & CSS
- Python
  - face recognition
  - Dlib
  - OpenCV
  - Numpy
  - pandas
- Django
- SQLite database

## Application

A major problem that plagues almost all the organizations in today's world is that of keeping track of the employees/students involved. Precious time and energy which could be used in a productive manner is wasted away in taking attendance and roll calls. This greatly reduces the efficiency of the organization. For example, think about the lectures that we attend. A significant portion of the lecture duration is spent in taking the attendance of all the students. This time could be much more fruitfully utilized for learning. Let us not forget about the extra energy that the professor has to spend in taking the roll-call as well. Similarly, a good chunk of time is wasted in any workplace to keep track of the entry and exit timings of all the employees.

The overall efficiency of any organization would be greatly improved if there was a faster method to take care of this attendance process. This is where YouWho steps in. YouWhois a modern face recognition web-application which uses Deep Learning (DL) to identify the person in the image we feed to the application in real-time. Using YouWho would greatly minimize the time taken in the attendance process because now, rather than individually calling out the name of each student, the professor could just use YouWho and the whole attendance process would be done in under a minute! Apart from the time reduction, it would also save a lot of the professor's energy, making it easier for him/her to take the lecture comfortably.

One method that most organizations today use to tackle the attendance problem is the use of biometric scanners. These are better than manual attendance and are more uniform, but they require physical contact and ever since covid-19 shook the world, it has become very risky to rely on this method of attendance. YouWho would be the perfect solution considering this problem as well, since it does not require any physical intervention to identify the person and the whole process is carried out in a no-contact way.

## Innovation

- The biggest innovation in YouWho is definitely the speed and ease with which it identifies the people in the camera feed. Because of the algorithms used in the implementation of our software, it has an impressive accuracy of **93.38**%
- Because of the face landmark estimation algorithm that we have used in the implementation of our python program, it doesn't matter if the face of the student is tilted sideways or if he/she is looking in a different direction. We identify 68 specific points in any face and then project it so that the main points are centered. This way, our software can accurately identify people, regardless of the orientation of their faces.
- Because of the HOG algorithm we use to detect the faces in a particular frame, we can accurately identify the subject regardless of if it is a bright picture or a dull one, since this algorithm takes into account the gradients of the pixels rather than the actual pixel colors.

- Since there is no physical contact required in this attendance process, it is both extremely safe as well as very convenient for the professors taking the attendance.
- One of the features of our application is that it allows the admin to view attendance data in the form of graphs and plots. This can help the professors to check the attendance stats of the students.

## Description Of Implementation

### Development Process

Agile SDLC model is a combination of iterative and incremental process models with a focus on process adaptability and customer satisfaction by rapid delivery of working software products.

Our project needs a lot of iterative processes to be followed, Agile software methodology would be best suited.

Since our project needs a lot of functionalities to be implemented and regular interaction with customers is important. So, at once we can't decide the entire scope of our project and the complete customer requirements can't be known beforehand. So, we decided to once release the project with only some of the functionality implemented and then taking customer feedback into input and allowing further changes to our features and then a better and upgraded version of our project would be released.

### Outline

Phase 1 | 5th March - 22nd March:

- Gather the resources and explore the tech stack required for the project.
- Feasibility Study report of the application will be prepared and shared.
- Prepare the Requirement Analysis document that will describe all the requirements of the application.

Phase 2 | 23rd March - 10thApril:

- Designing the machine learning models.
- Setting up the backend architecture of the project.
- Set up communication links between the frontend and backend part of the application.7th

Phase 3 | 10th April - 17th April:

- Testing and cleanup of the code
- Deployment
- One week buffer to compensate for any delay
- Brainstorm about future prospects of the application

## Implementation Detail

### Webcam working

OpenCV provides VideoCapture and VideoWriter classes to capture videos and save them on disk. In our project, VideoCapture objects have been used to work with the webcam feed. The VideoCapture() constructor takes one argument, which can either be a number, specifying the index of the camera device, or the path to a video file. Once a VideoCapture object is created, it can be processed frame by frame, allowing us to apply the image processing techniques required for face detection. The video capture can be released with the release() method.

### Face detection using Histogram of Gradients(HOG) algorithm

We need to locate the faces in a photograph before we can try to tell them apart. We'll start by making our image black and white because we don't need color data to find faces. We'll look at every single pixel in our image one at a time. For every single pixel, we want to look at the pixels that directly surrounding it.
Functionality of the method:

- HOG focuses on the structure of the object. It extracts the information of the edge's magnitude as well as the orientation of the edges.
- It uses a detection window of 64 x 128 pixels, so the image is first converted into (64, 128) shape.
- The image is then further divided into small parts, and then the the gradient and orientation of each part is calculated. It is divided into 8x16 cells into blocks with 50% overlap, so there are going to be 7x15 = 105 blocks in total, and each block consists of 2x2 cells with 8x8 Pixels.

We take the 64 gradient vectors of each block (8x8 pixel cell) and put them into a 9-bin histogram .

## Face Landmark Estimation

Basically, when we isolate the face from the image, the face of the person can be projected in any way, or the person may look in any direction. So, when we create an embedding of an isolated face, the embedding of the the face of the same person projected in different ways may vary a lot. We will try to warp each picture so that the eyes and lips are always in the sample place in the image. This will make it a lot easier for us to compare faces. To do this, we are going to use an algorithm called face landmark estimation. The basic idea is we will come up with 68 specific points (called landmarks) that exist on every face.

Functionality of the algorithm:
No matter how the face is projected the output image having an isolated face will always have the outer eyes and nose positioned at the same Position.

## Facial Recognition

We used the Face Recognition Library (OpenFace). Face Recognition library works on deep metric learning tools to identify, train and classify test data. It is the world's simplest library for recognition and manipulation of faces. This model showed an accuracy of 93.38% on the Labeled Faces in the Wild benchmark indicated by python software foundation. The deep neural network used by it, represents the face on a 128-dimensional unit hypersphere. The embedding is a generic representation for anybody's face.
Unlike other face representations, this embedding has the nice property that a larger distance between two face embeddings means that the faces are likely not of the same person. This property makes clustering, similarity detection, and classification tasks easier than other face recognition techniques where the Euclidean distance between features is not meaningful. Our project has incorporated this library to extract the face encodings and face location from the images.

## Web-application development

Our project involves the creation of a web application that works as an attendance system that uses the facial recognition model already described. For the development of this web-application, we used Django. Django is a high-level Python-based, open-source web framework that allows the creation of websites with ease. It follows the Model-Template-View (MTV) Architecture.

## MTV architecture

Django's MTV architecture is a slight variation of the MVC pattern. This model maintains data, the template acts as the presentation layer, and the view formats the data received from the model and sends it to the template to be displayed. It also processes the input by the user and sends create, update, delete requests to the model. The template layer in Django is akin to the view layer in the typical MVC and consists of HTML, CSS and Javascript code.

## Backend working

The models.py file in our project is where we define our database models. The templates contain HTML files which create the UI layer. The views.py file is used to communicate between the models and the templates. It consists of different functions, or views, which take in web requests and return web responses. As a result, the HTML contents of the associated template, along with linked CSS and JS files, are displayed as a web page, at the url mapped to that view. This url mapping is specified in the urls.py file. Django views are linked with their corresponding 25 contents using the render function, and data to be displayed is passed using the Django templating language.

## Database Management

Django allows developers to connect their applications with databases such as SQLite, MySQL, PostgreSQL. It takes care of the hassle of generating SQL specific to that database by providing us with the ability to write Python data structures instead of the database language. Django includes an Object Relational Mapping (ORM) layer that can be used for this interaction with data from various relational databases. Django also provides Models, which are Python classes which act as a bridge between the database and the server.

Our project uses the built-in User model provided by Django, which stores the information of all employees. This model can be imported using the following command:

```
from django.contrib.auth.models import User
```

Our project also uses two other model classes ‘Present’ and ‘Time’ that store Attendance (present/absent) and time-in, time-out, respectively of all employees. The following code is used to create these two models:

```
class Present(models.Model):
-user=models.ForeignKey(User,on_delete=models.CASCADE)
-date=models.DateField(default=datetime.date.today)
-present=models.BooleanField(default=False)

class Time(models.Model):
-user=models.ForeignKey(User,on_delete=models.CASCADE)
-date=models.DateField(default=datetime.date.today)
-time=models.DateTimeField(null=True,blank=True)
-out=models.BooleanField(default=False)
```

The user field acts as a foreign key to the User model. The date field stores the current date. The present field is a boolean field that stores True for present, and False for absent. The time field stores the time of entry or exit, depending upon the value of the out field, which stores True for time-out, and False for time-in

## Frontend

In order to develop our website’s UI and UX, Bootstrap and CSS styling were used. These CSS files are stored in a separate static folder within the Django project.
CSS- Cascading Style sheets are used for describing the presentation of a document written in a markup language such as HTML. It allows the addition of styles to a webpage with the use of classes and IDs.
Bootstrap is a front end framework that simplifies the process of adding styles, by providing built-in CSS classes that 28 can be added to HTML elements, allowing the developer to forgo the process of writing complicated CSS.

For our project, Bootstrap was added via the Bootstrap CDN. The url of Bootstrap CSS files is simply added within the tag of the HTML file, and Bootstrap classes can then be used throughout the file. The Bootstrap typography, navbar, table, button, card, container, alert classes are used extensively throughout the project.

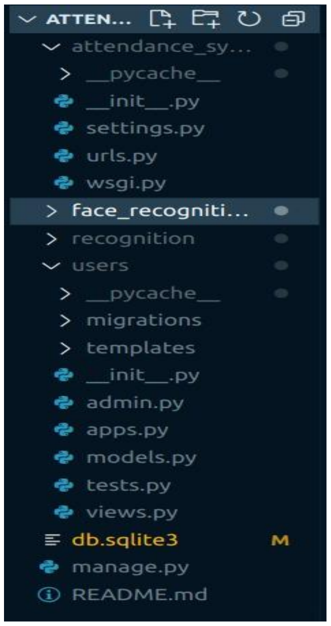
## Django Template Language

Django Template Language is used to send data from the view to the templates. It provides the developer with tags that function similarly to some programming constructs - such as the 'if' tag for boolean tests and the 'for' tag for looping. The templating language has been used throughout the project for interaction between the view and the templates.

## Data Visualization

One of the features of our application is that it allows the admin to view attendance data in the form of graphs and plots. For this purpose, Matplotlib and Seaborn libraries have been used. Data is first fetched from the database in the form of query sets, using Django. Query sets can be converted to Pandas dataframe objects using the django_pandas library. These data frames are then used as direct inputs to Seaborn graphing functions.

## File Hierarchy



```
ATTEN...
  attendance_sy...
    __pycache__
    __init__.py
    settings.py
    urls.py
    wsgi.py
  face_recogniti...
  recognition
  users
    __pycache__
    migrations
    templates
    __init__.py
    admin.py
    apps.py
    models.py
    tests.py
    views.py
  db.sqlite3        M
  manage.py
  README.md
```

## Future Work

YouWho can be of great use not only to various companies, schools and colleges but also to the police and the intelligence agencies of the government. In various police cases, there is access to the CCTV footage which at times captures images of the perpetrators. YouWho would be able to aid in finding out the person from the image, given that there was an image of the person in our dataset before. Alternatively, it could also be used to check if separate images are of the same person or not. We could add this additional functionality to YouWho with time to increase its usability.

YouWho also detects the number of faces in a given picture at a time. This could be of great help to camera manufacturers and photographers, wherein it could inform the photographer in real-time about the number of faces in the frame. This could help him in finding out if he missed out anyone in the photo or if someone isn't looking properly into the camera, thus paving way for better pictures. For example, if someone is clicking a group selfie of 10 people, and if YouWho indicates only 9 faces, then it means that someone isn't in the frame or is looking elsewhere. Thus, making sure that everyone is present in the final picture. Thus, we could work on integrating YouWho with physical cameras or with mobile cameras like GCam through apk extensions.

## References

https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78

http://reports-archive.adm.cs.cmu.edu/anon/2016/CMU-CS-16-118.pdf?ref=https://githubhelp.com

https://docs.djangoproject.com/en/4.0/

https://face-recognition.readthedocs.io/en/latest/readme.html