# CSE 473/573 - Introduction to Computer Vision and Image Processing – Spring 2020

# Visual Welcome Center

**Srikar Challa**
50312357
University at Buffalo
Buffalo, NY 14260
*srikarch@buffalo.edu*

**Suhelbeer Singh Sandhu**
50337041
University at Buffalo
Buffalo, NY 14260
*suhelbee@buffalo.edu*

**Uneeth Akula**
50317480
University at Buffalo
Buffalo, NY 14260
*uneethak@buffalo.edu*

**Abstract**

The applications of Artificial Intelligence are on the rise in order to make many tasks in our daily lives more convenient and easier to perform. The tasks of face recognition and verification have particularly gained more attention because of its vast biometric applications as it is more convenient than the traditional biometric authentication methods of fingerprint identification, iris scanners, and voice analysis. One such simple application is the visual welcome center and in this project, we present an interface, called **UB Visual Welcome Center**, which interacts with a user, recognizes his face, registers him to the system if the user is new, greets and clocks in the user and updates his image every time he interacts with the system. Additionally, details such as the current events that are being held in the department, directions to those places are also provided for the user.

## 1 Introduction

This visual welcome center is a one-stop center that provides general information to the users such as the ongoing events in the department, additional details, and directions related to those events. It is an interface that runs continuously and gets triggered only when a person comes in front of the camera and his face is recognized. Once the face is detected, the system waits for a few seconds and then recognizes the person again. If the same person who initiated the interaction is standing in front of the camera then it compares it with the records in the database. If the person is found not registered, then the system will ask permission whether he wants to enroll himself in the system. If he says “yes”, then the system will click his photo and then, it will request him to fill out a form in order to register him to the database. If the person is already registered, then the system will start interacting with him. The mode of interaction of the user with the system is visual and speech only. The messages from the system are displayed to the user on the screen and also are spoken out by the system. In response, the user interacts with the system using voice commands. When he is done interacting with the system, the system sleeps and waits to get triggered by another face. The system runs continuously in this loop.

## 2 System Architecture

The face recognition and speech recognition are the main components of this architecture. This architecture of this system can be broadly classified into four parts:

## 3.1 Face Recognition

Face detection, along with recognition and verification is the most important component of this system. For these tasks, we have used the python library of face recognition [1] which was built using dlib's [2] state-of-the-art face recognition built with deep learning. The model has an accuracy of 99.38% on the Labeled Faces in the Wild [3] benchmark dataset.

In this project, we have used the webcam to take the live feed and the system will wake up when a face is detected. It stores the face encodings and locations of the person in front of the camera. To ensure that the person is trying to interact with our system and is not a by passer, we capture his face for a few seconds and verify that it is the same person that is standing in front of us. When the user is ready, a picture of his face is clicked and checked with the image set we have in our storage. If it already exists, the system updates the image, greets him, and starts interacting with him. These interactions are based on the voice commands of the user. If the image does not match with any of the images in the set of images, we have in our file storage, then the system considers the user as new and displays a QR code for registering him into the system.

## 3.2 Database Management

Another major aspect of this architecture is the database storage. A QR code is displayed to an unregistered user when a face is detected. The QR code leads to a google form that the user needs to fill out the details such as his name, e-mail address, phone number, designation, department, and website if any. All these details along with the image location are stored in the database. We have MySQL Database for this purpose.

There are three different tables that are being maintained in the database. One is the user table, which stores the basic information of the user that is obtained from the form the user has submitted. The second table is the image table, which contains the location of the image that is stored in the local storage. Each user has a unique image and is stored separately. The third table is the timesheet table, which stores the date and time whenever the user clocks-in and interacts with the system. Whenever a registered user interacts with the system, the old image of the user gets replaced by the new image in the local storage. Hence, we have only one single entry for each user in the image table. But we store all the dates and times when the user clocks-in. Hence, a single user can have multiple entries in the timesheet table.

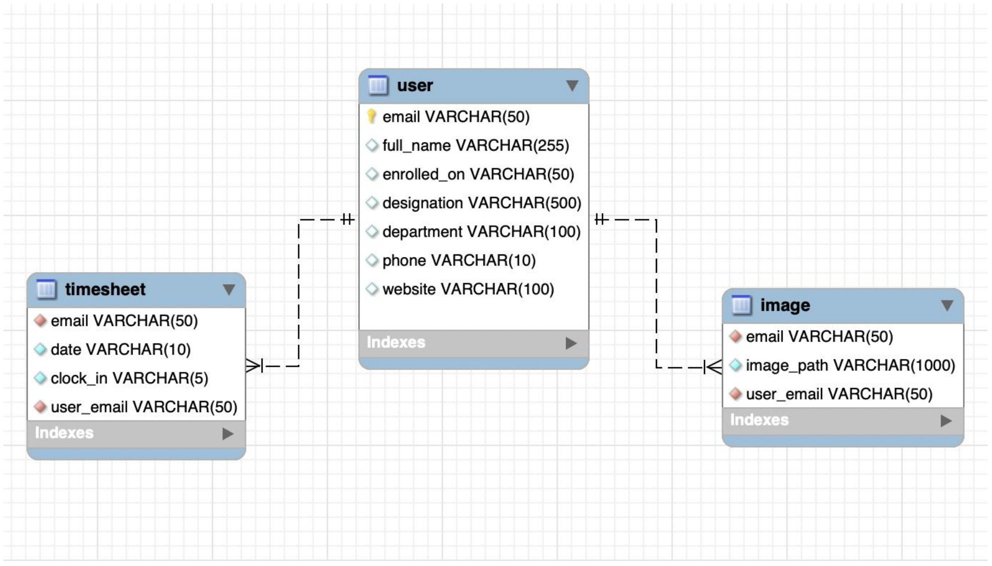


**Figure - 1** The three tables that are stored in the database

A user is uniquely identified by his email address. Hence, this field is used as the unique ID for the user in the database. This is also used for naming the image file that is obtained after capturing a new user. This naming convention ensures that a particular person can register only once using a single email address. This also ensures that the old image file is overwritten by the new image file, whenever the user clocks-in, as they both will have the same file name.

## 3.3 Speech Recognition

The system interacts with the user using simple voice commands from the user. The user speaks simple words, such as, “yes” or “no” to interact with the system. For this purpose, we have used the speech recognition python library [4]. The user can use the microphone to interact with the system. To provide convenience for the user, we have included a small restricted vocabulary of words that correspond to the different styles of “yes” and “no” the person can speak. This also allows the user to say multiple times “yes” or multiple times “no” but not both at a time. It will then extract the keywords necessary to complete an action. A registered user can ask for his previous entries (timesheet), and all the clock-in details of the user is displayed. Additionally, he can ask for any other information and the system displays that information on the screen.

## 3.4 Graphical User Interface

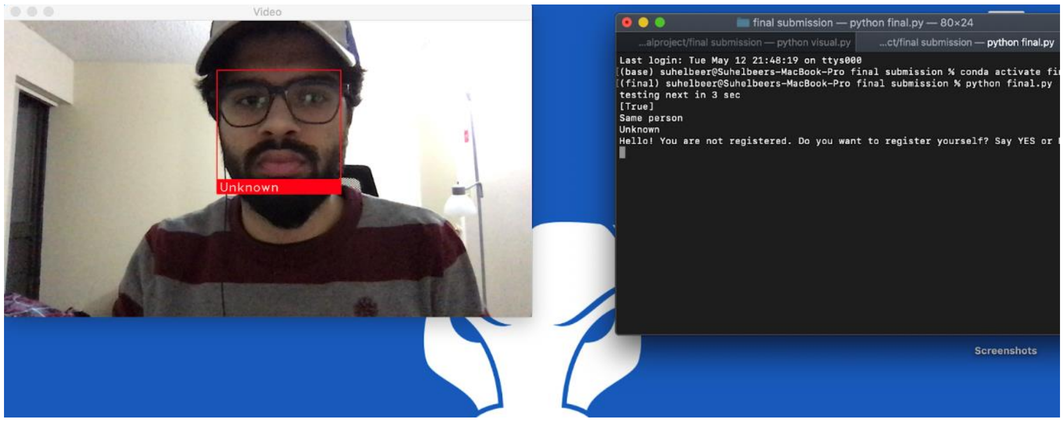
The Graphical User Interface part has been kept simple. Tkinter [5], the standard python library for the graphical user interface, has been used for this purpose. Most of the output is shown in the python console. The webcam feed and the QR code are displayed in separate windows. The webcam uses an OpenCV frame and another frame is used to display the QR code. While the webcam feed is displayed all the time to the user, the QR code pops up when a new user tries to register himself for a max of 100secs.

**Figure 2** – The QR code which is provided to the user for registration

# 4 Results

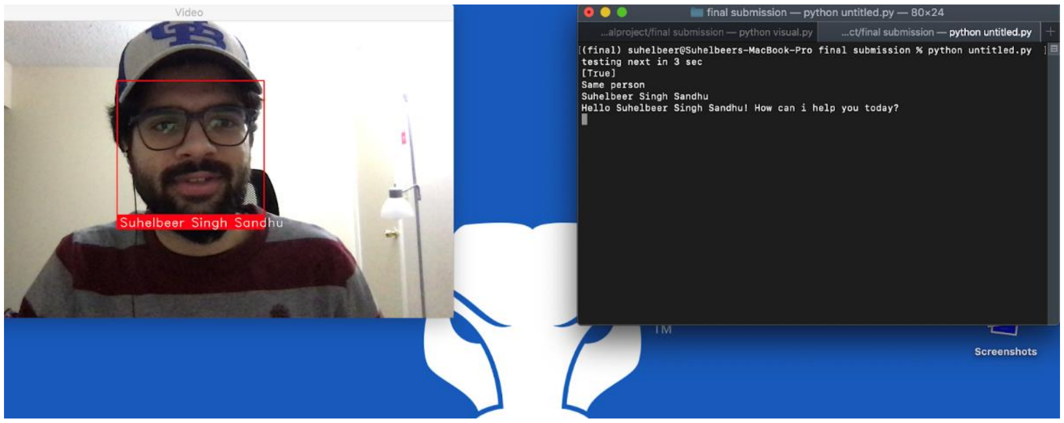
## 4.1 Screenshots

The screenshots of the working model of our interface are as shown below


Video
Unknown
final submission — python final.py — 80×24
...alproject/final submission — python visual.py
...ct/final submission — python final.py
Screenshots

```
Last login: Tue May 12 21:48:19 on ttys000
(base) suhelbeer@Suhelbeers-MacBook-Pro final submission % conda activate fi
(final) suhelbeer@Suhelbeers-MacBook-Pro final submission % python final.py
testing next in 3 sec
[True]
Same person
Unknown
Hello! You are not registered. Do you want to register yourself? Say YES or
```

**Figure 3** – The system recognizing the face as an unknown user and asking to register


Video
Suhelbeer Singh Sandhu
final submission — python untitled.py — 80×24
...alproject/final submission — python visual.py
...ct/final submission — python untitled.py
Screenshots

```
(final) suhelbeer@Suhelbeers-MacBook-Pro final submission % python untitled.py
testing next in 3 sec
[True]
Same person
Suhelbeer Singh Sandhu
Hello Suhelbeer Singh Sandhu! How can i help you today?
```

**Figure 4** – The system recognizing the face after registration and interacting with the user

## 4.2 Video

To make the interaction between the user and the system clear, a video demonstration of this interaction with the user is provided. Initially, the person randomly moves in front of the system. The system recognizes the face, but it does not wake up from sleep. When the person is really standing in front of the system, without moving away, for a few seconds, then it starts interacting with him. Here, the user is a new user when he first interacts with the systems. The system prompts him to get registered. the system clicks a photograph and displays the QR code for a few seconds. After he fills out the form and submits it, the system adds his details to the database and greets him. The user then starts to interact with the system. After the interaction is done, the system waits for another user.

Link : https://buffalo.app.box.com/s/wy862s7o1dahtxu9lcxkmodrrz1uszn8

## 5 Challenges

One of the first challenge that we faced was to make the system run in real-time. It should not recognize everyone that pass by; therefore, a sleep time is introduced so as to check if a person is standing in front of the system for some time. Doing so will stop the system from activating redundantly.

Another challenge was to verify at every step that the same person is using the system that started the interaction in the first place. A specific function of verify_face was defined to accomplish this task. This stops the user from making repeated entries in the database.

In order to interact with users, speech recognition is used to convert audio to text. But the limited vocabulary of yes and no was difficult to understand due to ambient noise. So, a time limit of a few seconds was introduced to recognize shorter sentences.

To make the interaction smoother with the system, no keyboard or mouse inputs are taken. So, in order to register a user into the system, He is requested to fill out a google form by scanning a QR code displayed on the screen. A count of entries in the spreadsheet is monitored to see if there is any change.

Many of the challenges that we have faced had to do with storing the images in the database. The number of images that are to be stored is very large and when each user passes by, we need to compare his image with the entire images set and verify him. This process of retrieval of the entire image set to the local storage, performing operations, and again storing the new image into the database is a tedious process. Hence, we have used the local storage to store the images and stored the location of these images as a reference in the database. So, whenever the system detects a user and clicks his picture, it compares that image with all the images in the database and if there is a match, takes the location of that image as the reference, searches the database for this reference and retrieves only that corresponding information and displays the user. If there is no match between the images, the system clicks a picture, stores the location of it in the database, and registers the new user. Every time a person interacts with the system, his or her latest image is stored in the database so as to keep up with any change in look.

The other challenges that we have faced are related to the User Interface. Due to the time constraints, we have limited ourselves to the OpenCV for displaying images, but there are many other options that can be used. The continuous displaying of images and processing them requires parallel processing which is out of scope for this project taking into consideration our time constraints.

## 5 Conclusion and Future Work

We have successfully created an interface that interacts with the user, clicks a picture of the user, recognizes him, and walks him through. If the user is new to the system, a prompt is made which displays a QR code and the user gets registered using the registration form. The system interacts with the user using voice commands and provides the necessary info that can be provided to him.

Furthermore, it would be interesting to perform smile detection for clicking the picture instead of clicking a picture randomly.  Instead of asking the user to speak “yes” or “no”, we can just ask the user to smile which implies a “yes”. This would reduce the necessity of the speech recognition part of the system but introduce other complexities to the system such as detection of the smile accurately. Another interesting update could be to recognize the natural language spoken by the user, instead of the simple “yes” or “no” commands. Though it increases the complexity, interacting with the user in natural language is an interesting option to consider. Another interesting feature that could be added is the hand gesture recognition, where the user can interact with the system using hand gestures of symbols.

In this project, we have considered the local storage for storing our images set. Though this can be used for our system, it is not a feasible solution when we deploy it. A better storage system such as cloud storage or a storage server could be used but the retrieval time is the main constraint that is to be considered. We have tried using the Google Firebase Storage and Database for this purpose but limited ourselves to local storage to keep it simple. This could be a better improvement as cloud storage is efficient, fast, and secure.

Few more improvements can be done to this interface. We are replacing the old image of the user with the new image each time he clocks-in and interacts with our system. Instead of that, we can store these images additionally, keeping the old images as it is. This can be done by creating a new directory for the additional images with a tag for each number that indicates the number of the image for that particular person. This also improves the accuracy of our model as we have multiple images to verify the person. As said above, the Graphical User Interface can be made more attractive by using different front-end tools/frameworks that are available. The use of a web page designed using HTML/CSS or BootStrap and connecting it to our backend python program would be more attractive.

In this project, we have used the QR code to provide the link to enroll in our system. The user fills the form and submits it. Instead of this, we can directly interact with the user to get his details using voice commands or natural language. Another option that can be considered here is the Optical Character Recognition (OCR). The user can show his business card or some other visiting card that contains his details. The system then captures the image and tries to recognize the text in the image. The main challenge while working with this is the differentiation of name, email, website, phone number from other texts that are present on the card. Another interesting option is to consider a chatbot for interacting with the user which takes the natural language spoken by the user as input.

Finally, we have developed an interface that can interact with the user. The list of improvements that can be made to this keeps going on and on but we have limited our implementation considering the time constraints.

## Individual Contributions

| NAME | Contribution |
|---|---|
| Srikar Challa | Face Recognition, Database, Forms handling, Report |
| Suhelbeer Singh Sandhu | Face Recognition, Speech Recognition, Code Documentation, Video |
| Uneeth Akula | Face Recognition, Database, Speech Recognition, Report |

## Acknowledgements

We would like to thank our CSE 473/573 Introduction to Computer Vision and Image Processing course instructor Dr. David Doermann for his insightful lectures and his immense contribution in igniting our interest in the subject. Additionally, we would also like to thank our teaching assistants William Liu, Xinguo Zhu and Xuan Gong for being helpful throughout the semester.

### References

[1]. https://pypi.org/project/face-recognition/

[2]. http://dlib.net/

[3]. http://vis-www.cs.umass.edu/lfw/

[4]. https://pypi.org/project/SpeechRecognition/

[5]. https://docs.python.org/3/library/tkinter.html

[6]. https://github.com/ageitgey/face_recognition