A Smart Receptionist Implementing Facial Recognition and Voice Interaction

Osman Mohammed Ahmed *ahmed028@cougars.csusm.edu*
Department of Computer Science and Information Systems
California State University San Marcos
San Marcos, 92096, USA

Yanyan Li *yali@csusm.edu*
Department of Computer Science and Information Systems
California State University San Marcos
San Marcos, 92096, USA

Ahmad Hadaegh *ahadaegh@csusm.edu*
Department of Computer Science and Information Systems
California State University San Marcos
San Marcos, 92096, USA

Abstract

The purpose of this research is to implement a smart receptionist system with facial recognition and voice interaction using deep learning. The facial recognition component is implemented using real time image processing techniques, and it can be used to learn new faces as well as detect and recognize existing faces. The first time a customer uses this system, it will take the person's facial data to create a unique user facial model, and this model will be triggered if the person comes the second time. The recognition is done in real time and after which voice interaction will be applied. Voice interaction is used to provide a life-like human communication and improve user experience. Our proposed smart receptionist system could be integrated into the self check-in kiosks deployed in hospitals or smart buildings to streamline the user recognition process and provide customized user interactions. This system could also be used in smart home environment where smart cameras have been deployed and voice assistants are in place.

Keywords: Face Recognition, Deep Learning, Django Framework, Image Processing.

1. INTRODUCTION

The tasks of a receptionist depend on the sector he/she works in and automation of such a job can be in high demand in hotels, commercial complexes or even for security purposes in certain organizations. Considering a hotel room reservation system, a receptionist needs to interact with the customers, getting his/her information, identification, booking and payment of the rooms, etc. Further, in businesses, scheduling appointments with individuals can be automated where security is an important factor to be considered and these systems can be strengthened by imposing features such as facial recognition, voice interaction and server-side security such that this automation that would help making the tasks of a receptionist system more secure and reliable to recognize and interact efficiently with the customers. The process of accepting, addressing and guiding the customers' needs can be done easily such that the users have no inconvenience to be recognized by the automated receptionist system.

Authentication is necessary in order to implement security and present systems allow for some methods to incorporate authentication. Traditional systems use either biometric authentication or non-biometric authentication. Non-biometric authentication allows for the use of physical objects such as a swiping card or a key. These systems are further classified into object-based authentication or knowledge-based authentication. They also allow for the use of passkeys, tokens, PINS, etc which are prone to be guessed or accessed by attackers. In order to overcome these limitations posed by non-biometric authentication systems, modern systems now use more advanced authentication systems that implement Biometric authentication systems. These systems allow identification of an individual through physiological traits and real-time behaviour which makes the passkey unique. These kind of authentication systems include fingerprints, retina scans, electronic signature, voice or face recognition, which are difficult to imitate or be stolen or duplicated by attackers as they're unique to that user.

Face recognition has gained a lot of popularity in implementing secure systems and have been constantly in use to improve the accuracy of the systems. This is done to tackle the challenging tasks because most methods don't really provide a robust solution to different situations. Example of these simulations include different expressions, pose invariants, lighting variations, etc. Furthermore, capturing real-time face recognition has a higher overhead computational cost when it is implemented with Deep Convolutions Neural Networks.

With advancements in technology, machine learning is now covering vast areas of application fields wherein implementations of certain algorithms allow for systems to be more discrete, secure and time saving to automate a lot of different sectors. The accuracy and easy deployment of these algorithms are the main reasons why these systems are trusted in many fields and require less manpower. The vast areas of application include automatic driving and self-transportation vehicles, smart gaming and other entertainment systems, healthcare, home business security, etc. These systems take in a lot of raw data that and process it to generate desirable outputs. They learn by repeatedly processing the information and use that experience to make the systems better.

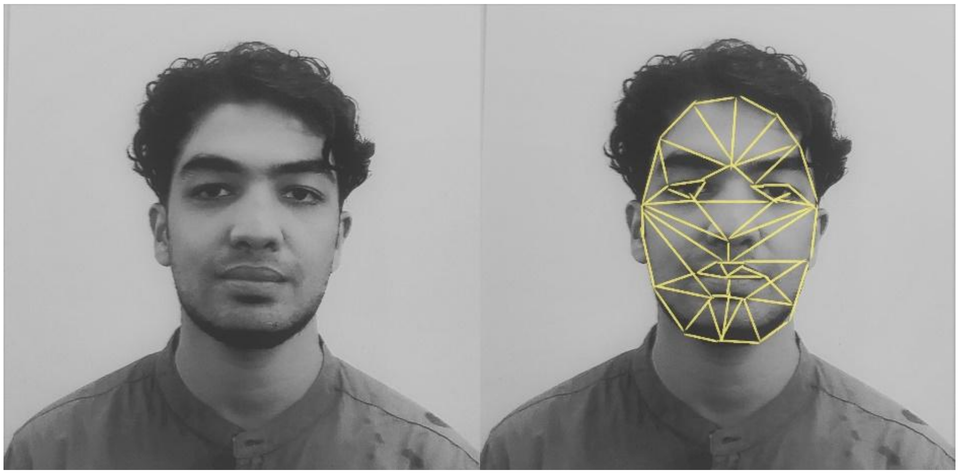


FIGURE 1: Face Recognition Parameters.

One major application is facial recognition that is used in security systems and business companies where certain algorithms can detect the face of an individual, store the data and learn them, and later use the data mining techniques to identify that individual. It is a way to recognize the human face by using certain biometrics to map the different facial features as depicted in Figure 1 and compare them with all the other existing faces in the database to find a match. In business organizations, facial recognition can be used as a security feature, as well as identifying individuals to automate the task of receiving the clients data and proceeding them to their required destination. For a commercial complex or office system, a smart receptionist system can capture the data of the customers and interact with them. This reduce the need of manpower, time, and money. Unlike humans who need to look up for the records, smart receptionists can store large amounts of data in their database and use the machine learning techniques to identify the individuals faster and more efficient.

2. RELATED WORK

Face recognition has been a booming topic in the field of automating certain tasks in establishments and smart receptionist implement several different algorithms and systems to execute efficient reception tasks. In a research done by Hteik et al. [1], face recognition was executed by using a MATLAB program on a PC where the access control is done by a microcontroller. The system that is used in [1] is less efficient as modern-day systems require faster response and interaction, more accuracy, and better security.

In the papers written by Salvador and Foresti [2], facial recognition was done by implementing Regularized Linear Discriminant method that only captures frontal facial data by assuming that user cooperation is present. This might not be the case in every situation and to allow for little or no inconvenience it is more acceptable to have a more robust and dynamic method to have a quick scan of the whole picture in lesser time.

Rohit et al. developed their system [3] using IoT devices by integrating Raspberry Pi to detect a person coming at a door. This allows for efficient face detection but has a higher response time due to the lag that is generated from using IoT devices.

[4] describes their design and implementation of a smart e-receptionist that can greet visitors and talk to them with natural language understanding. However, it can only sense a nearby visitor through motion detection, it doesn't have the capability of recognizing a user, which we have via facial recognition and can thus provide more customized interactions.

An interactive robot receptionist system was proposed and designed in [5] that is able to provide directional guidance using physical gestures and answer simple questions with speech recognition. Similarly, a smart humanoid receptionist was developed in [6] using WeegreeOne robot that is connected to several IoT sensors, camera, databases and AI services to enable the functionalities of user recognition and voice interaction. They tested their humanoid receptionist in a smart office environment and demonstrated its effectiveness.

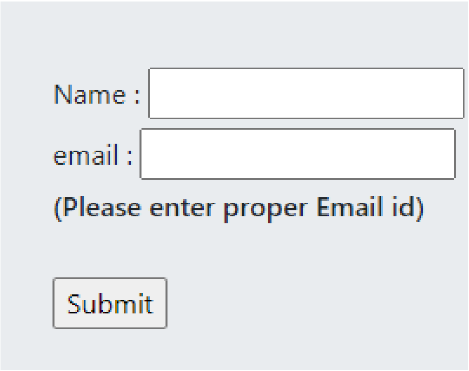
In [7], the authors proposed a cloud-based robot receptionist that works in a home environment to provide both reception and home assistance. The authors in [8] focus on helping the receptionist to gain context-aware capability and to interact with people in a natural way.

Hwang et al. [9] focused on the dialog system in human robot interaction, and proposed a recurrent neural network based dialog system. Their proposed system has been validated in the context of hospital receptionist and their evaluation result shows it is able to efficiently choose responses and gestures to welcome and help check-in users.

In this work, our main focus is on facial recognition based user authentication and voice interaction part of a smart receptionist. The chatting, conversation and question understanding part could be implemented using Google Dialogflow service [10] or word embedding algorithms such as bert or word2vec [11][12].

3. METHODOLOGY

Our goal is to build a system that manages users who login with a face ID using facial recognition concepts of machine learning. We have created a web application that can be set up in offices or business establishments, or even at a certain individual's reception who might have to schedule meetings with other people. The system would recognize people who have already visited the office. For a person who is visiting the first time, the system asks the user to feed in his/her information. Then, the next time the user makes a visit, the system uses the user stored data to recognize the user.



Name :

email :

(Please enter proper Email id)

FIGURE 2: User details for first time users.

Our smart reception system uses facial recognition that allows the user to log in to the system with his/her facial ID which is unique to user. We have implemented different concepts of machine learning to perform deep facial recognition using certain libraries in python which will be discussed later. To be specific, we have used python's libraries that implement OpenCV that uses a form of deep Convolutional Neural Networks to allow for a deeper scan of the picture in real time which can scan all sides including frontal and the sides. Our system takes care of the limitations of existing system and is relatively more accurate.

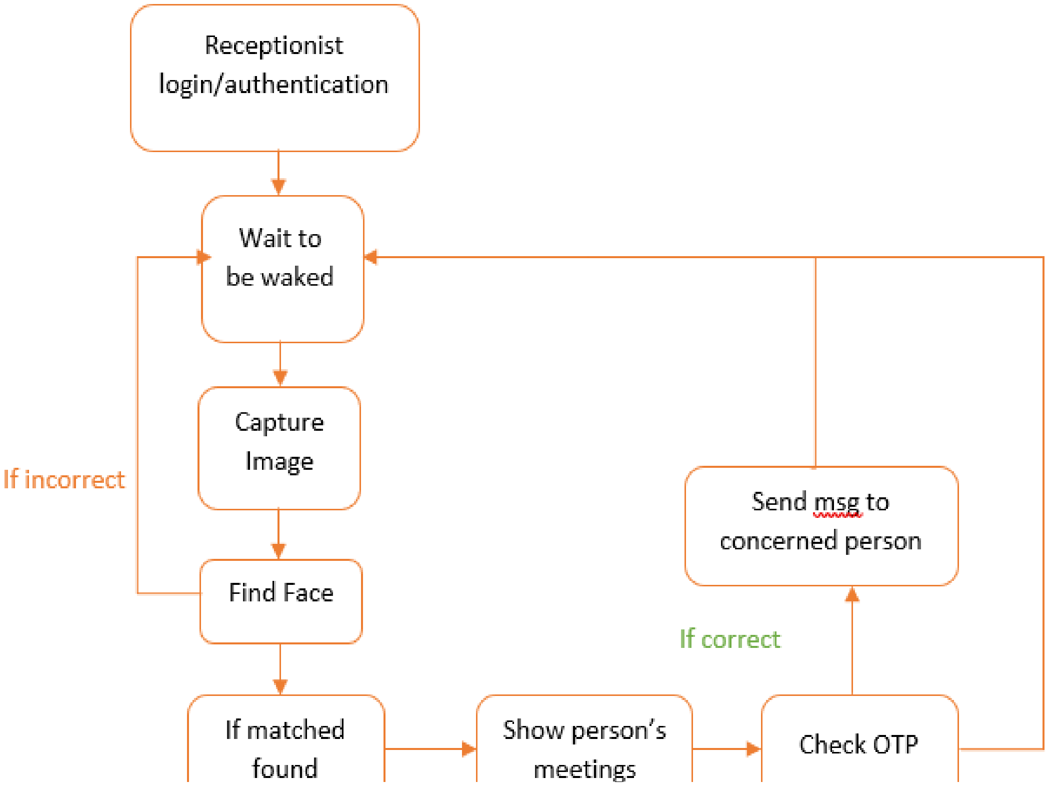


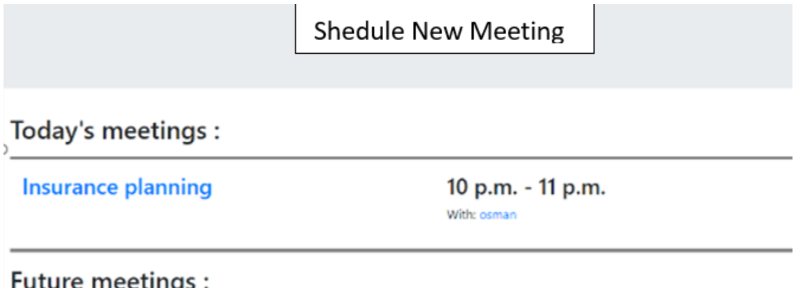
FIGURE 3: System Workflow.

This system can be enhanced further to add other essential tasks of a smart receptionist and thereby creating a perfect application that can be used in large scale commercial complexes. Our system is embedded in a GUI that gives reliability for both client and server side. Figure 2 shows the login page where one can log in as a user or the manager. The manager is the one who manages the Reception.

3.1 Basic Workflow

Figure 3 shows the detailed outline of the system workflow. The system will have the Receptionist's 'Wake me Up' module for a user to come up and use it to record the user face. The steps include:

a. A user clicks on the button that initializes the face recognition. For first time users, the system asks for the name and email address as shown in Figure 2.
b. The manager will then be able to schedule new meetings with that new user or other users as shown in Figure 4.



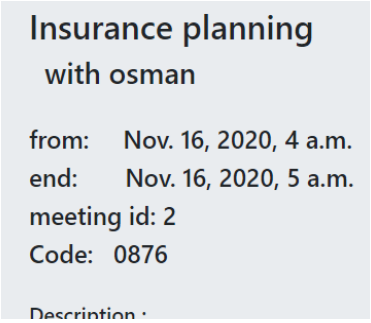
Today's meetings :

Insurance planning 10 p.m. - 11 p.m.
With: osman

Future meetings :

FIGURE 4: List of Meetings.

c. The manager then schedules a new meeting adding necessary details and the user will be able to view the same as in Figure 5.



Insurance planning
with osman

from: Nov. 16, 2020, 4 a.m.
end: Nov. 16, 2020, 5 a.m.
meeting id: 2
Code: 0876

Description :

FIGURE 5: Meeting Details.

d. User receives an email with a code from the manager along with meeting details.
e. User logs in the system again using face recognition that opens the page where he/she enters that code in the system, where the receptionist asks the user to click on a button and enter the meeting ID by giving audio input of the meeting ID.
f. After the user speaks the id, the receptionist again asks the user to enter the meeting code, again giving audio input of that code.
g. If the code is correct, the receptionist sends an alert to the manager saying that the user has arrived, and the manager can authorize that meeting.
h. Finally, the receptionist tells the user to proceed and attend the meeting.

3.2 System Architecture

Figure 6 shows the system architecture of our web application. The back-end system is based on the Django Framework [13] for developing web applications with python. This facilitates for robust and simple managing of the different sections in the entire system. In a system where we need to work with a larger dataset, Django allows for the efficient managing and creating faster access to each of these items whilst managing the whole application with the database. Django uses a system called "Models" that are used to handle the database.

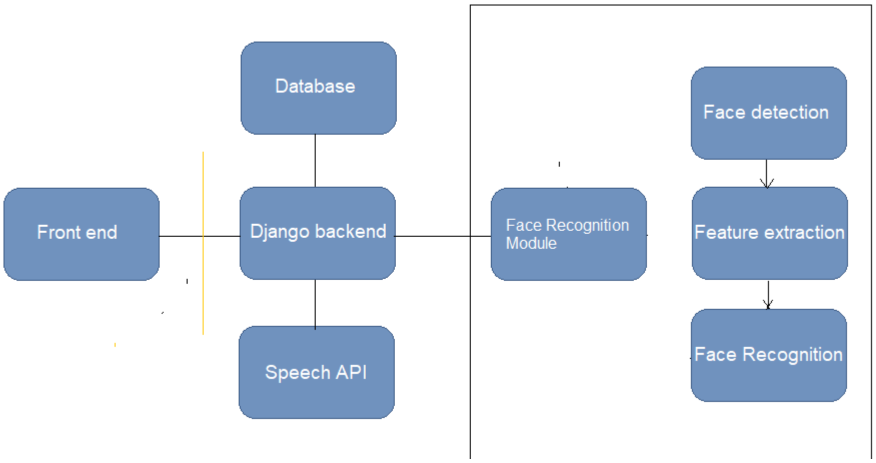


FIGURE 6: System Architecture.

“Models” is a single entity that defines all our information for a field related to a particular dataset. It consists of the important fields and behavioural aspects of our data that we have been storing and maps each of those to a single database table. Every python class is basically a model that has necessary subclasses and different attributes related to it. Combining the models gives a layout of the entire system connecting our database to the web application. With these Models, we can create new tables in the database, and therefore calling models as objects to add rows in the tables of the database.

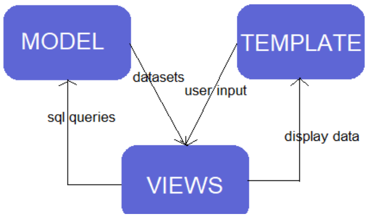
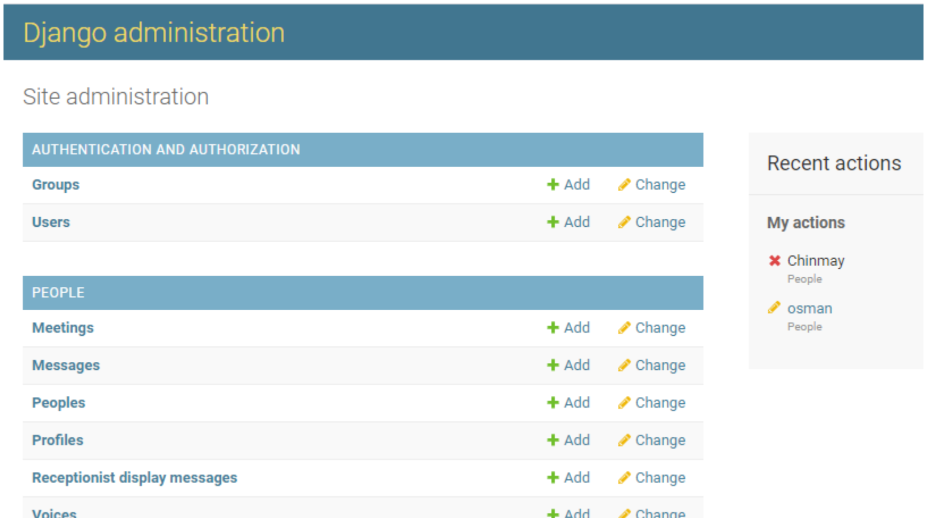


FIGURE 7: Django Framework.



Django administration

Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

PEOPLE		
Meetings	+ Add	Change
Messages	+ Add	Change
Peoples	+ Add	Change
Profiles	+ Add	Change
Receptionist display messages	+ Add	Change
Voices	+ Add	Change

Recent actions

My actions

✖ Chinmay
People

osman
People

FIGURE 8: Admin page backend.

Views in this framework are the logic layers for business models. Hence it is well suited for our application that allows to process the input given by a user and sends back the necessary valid response. The system takes in the input and fetches the required data from the database and sends an output onto the screen. In our application, each page is a different view that has its own GUI to interact with the user such as adding his/her name, email id, and the necessary meeting details. These entities are all connected to allow for a dynamic managing of different sections of the system as shown in Figure 7.

We have written the backend system in python programming language due to its ease of use and compatibility with the Django framework. Python has built-in libraries that we have used for our face recognition feature, as well as the voice automation. The libraries will be discussed in the subsequent sections. Figure 8 shows how admin side is managed with the help of Django to keep track of all the users and sections in the system.

The voice recognition module has been implemented with python's built-in library based on gTTS that is "Google's text to speech" library that allows us to interface with "Google Translating text to speech" and gives us a vocal speech output. For database management, we have used SQLite which is a widely used highly reliable and self-contained database engine that is well suited to work with the Django Framework.

4. IMPLEMENTATION

In this system, we have used different python libraries with implementation of OpenCV that uses deep neural network that allows us to exploit the face recognition module. We have used the Convolutional Neural Networks (CNN) to extract the key and essential components of images that have been taken as the input without any pre-processing of the raw images. CNN also has the potential to recognize patterns that have different geometrical variations such as rotations in the image, scaling, noise, etc.

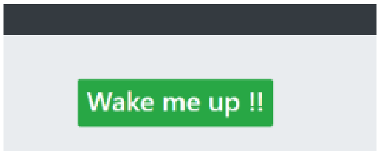
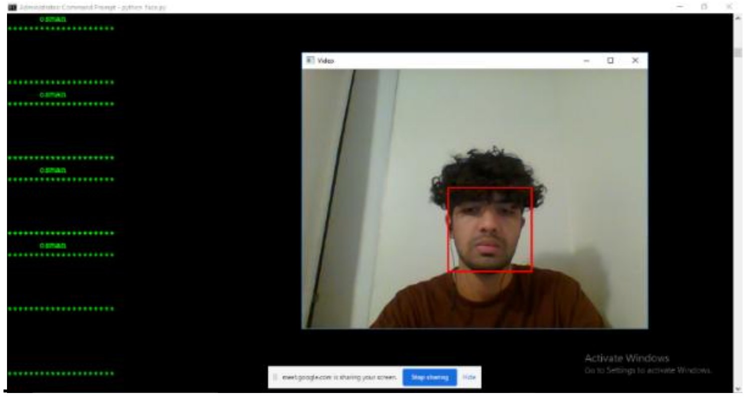


FIGURE 9: Initializing Face Recognition button.

Convolutional Neural Networks reduce the training performance of the traditionally used Back propagation (BP) algorithm [14] by reducing the number of learning parameters in that process to avoid any required pre-processing. The network relationship in CNNs are spatial that allows for the minimizing the pre-processing. Figure 9 is the homepage that initializes the face recognition algorithm on clicking the 'wake me up' button.



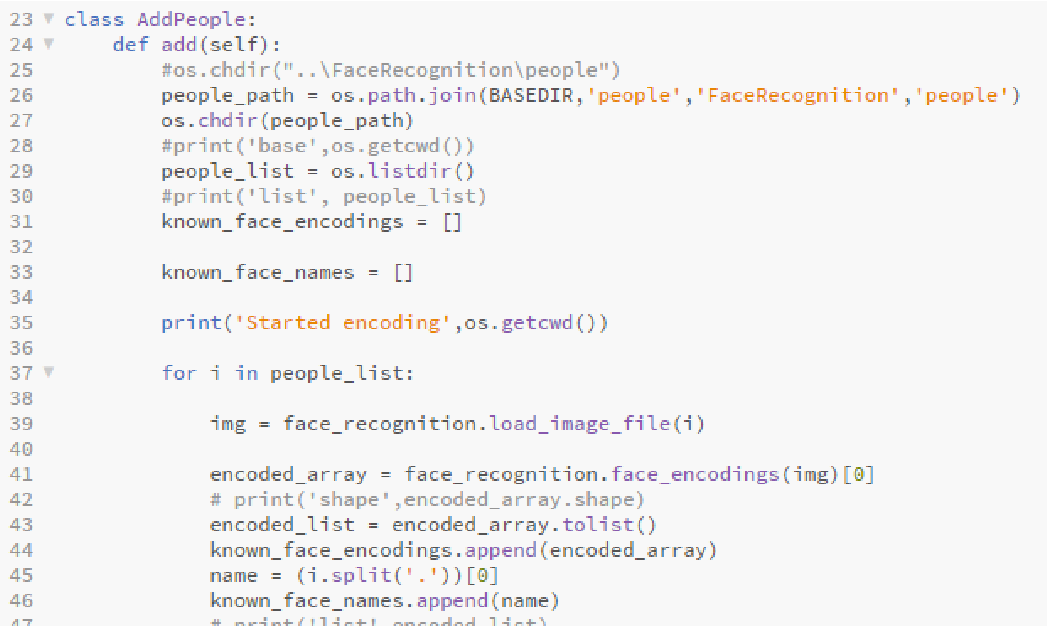
```
      osman
********************


********************
      osman
********************


********************
      osman
********************


********************
      osman
********************


********************



********************
```

meet.google.com is sharing your screen.

Administrator: Command Prompt - python face.py
Video
Stop sharing
Hide
Activate Windows
Go to Settings to activate Windows.

FIGURE 10: Face Recognition with python.

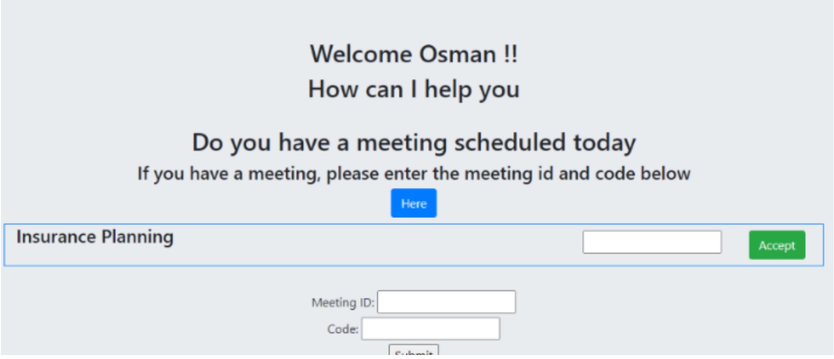
In addition, the main reason for implementation of CNNs is to capture 3D image recognition in all angles and directions, whereas traditional systems using HOG (Histogram of Oriented Gradients) [15] or Regularized Linear Discriminant Analysis (R-LDAs) [16][17] rely mostly on frontal face detection. To make our system more user friendly, we have used the libraries in python that allows for reading and capturing of faces laid in all axis, doing the translations and the rotations as shown in Figure 10.



```
class AddPeople:
    def add(self):
        #os.chdir("..\FaceRecognition\people")
        people_path = os.path.join(BASEDIR,'people','FaceRecognition','people')
        os.chdir(people_path)
        #print('base',os.getcwd())
        people_list = os.listdir()
        #print('list', people_list)
        known_face_encodings = []

        known_face_names = []

        print('Started encoding',os.getcwd())

        for i in people_list:

            img = face_recognition.load_image_file(i)

            encoded_array = face_recognition.face_encodings(img)[0]
            # print('shape',encoded_array.shape)
            encoded_list = encoded_array.tolist()
            known_face_encodings.append(encoded_array)
            name = (i.split('.'))[0]
            known_face_names.append(name)
            # print('list', encoded_list)
```

FIGURE 11: Encoding Faces.

Figure 11 shows a block of code where we have performed the face encoding of some people for whom we were able to deploy the system to. We can use CNNs through OpenCV to train the system to generate 128 measurements for each of the faces. Then for all the people with different measurements, the neural network learns to generate 128 measurements for each person. Next, we run our face images through our pre-trained network to get the 128 measurements for each face and we can generate a string or array for each face that contains the encoded list. Therefore, at the end, each face (or person) has its unique string with its encoded array.



Welcome Osman !!
How can I help you

Do you have a meeting scheduled today

If you have a meeting, please enter the meeting id and code below

Insurance Planning

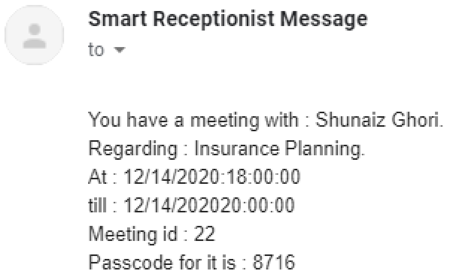
Meeting ID:

Code:

FIGURE 12: View Page after Authentication.

4.1 Receptionist Task

The main task of the smart receptionist in our system is to accept incoming users, record their faces and ask for their names and email address.



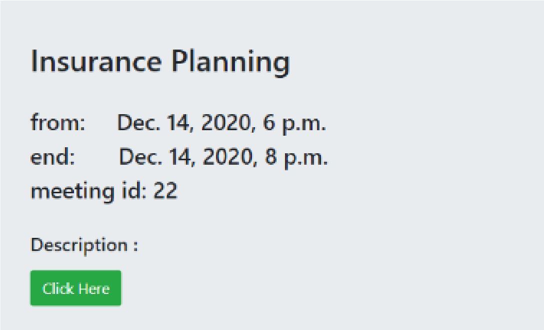
Smart Receptionist Message

to

You have a meeting with : Shunaiz Ghori.
Regarding : Insurance Planning.
At : 12/14/2020:18:00:00
till : 12/14/202020:00:00
Meeting id : 22
Passcode for it is : 8716

FIGURE 13: One-time Code sent via email to User.

When a user logs in for the second time, the system identfies the image of the user by comparing the existing data in the database. The system search the individual and checks if the user has any meeting (Figure 12).



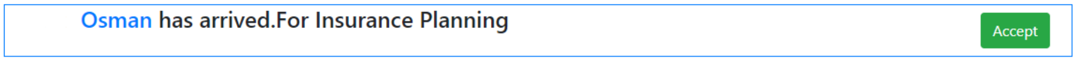
Insurance Planning

from: Dec. 14, 2020, 6 p.m.
end: Dec. 14, 2020, 8 p.m.
meeting id: 22

Description :

Figure 14: Voice Input for Meeting ID and Code

Next, the receptionist generates a one time code after the manager schedules a meeting. The code is sent to the user's email address (See Figure 13) that he/she had entered. The message includes details of the meeting including the time and the description of the meeting. The user will use the code in that email and enter it in the system as shown in Figure 14 in order to aunthenticate himself for that specific meeting.



Osman has arrived.For Insurance Planning

FIGURE 15: Alert to Manager.

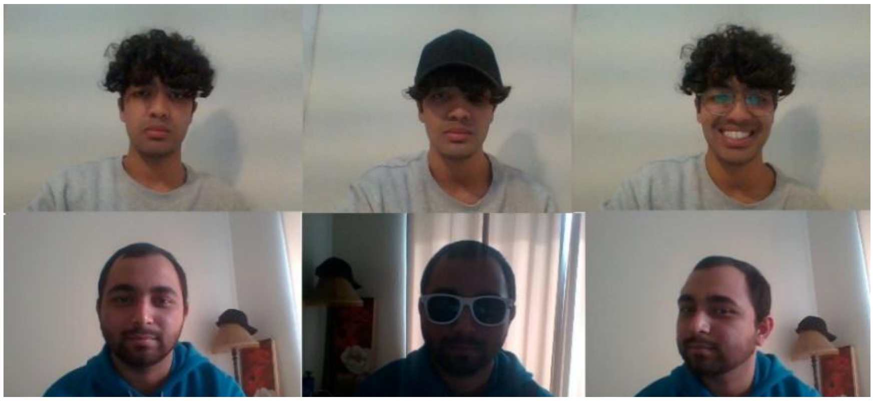


FIGURE 16: Testing.

The receptionist then tells the user to proceed for the meeting, after the manager approves the alert (Figure 15) that was sent to user when the suer had arrived. This is the basic task of the receptionist that we have implemented in this system. Many other functionalities can be added later as the future work.

4.2 Results

We have deployed the application to about 40 different individuals who were able to log into the system after feeding in their facial data, and storing their name and email address into the database. We have allowed them to enter their first image with a regular face. Later, we have tested the accuracy of the system by having them face the camera wearing sunglasses, or hats as we seen in Figure 16. The system was able to detect the faces of those individuals. Each facial data was unique and did not overlap with any of the other data.

For each of the individuals, we were able to record their faces in different lighting and backgrounds and the system was able to recognize and authenticate that individual as well as welcoming the person with his/her name. The model has an accuracy of 99.38% for the face recognition module.

5. CONCLUSION

The smart receptionist system allows for an efficient, robust and dynamic use of the face recognition and voice interaction modules as well as providing an easy GUI that would allow users to come in and authenticate themselves and schedule meetings.

The smart receptionist system can be further enhanced by adding advanced facial recognition to avoid existing security issues. Furthermore, voice automation can be improved by creating a more dynamic interaction between the user and the receptionist to have a proper conversation between the two. It is also possible to assign more tasks to the receptionist. Receptionists in the existing system, are capable of scheduling meetings, and few other minor jobs. The front-end of our application is built on HTML, CSS and Bootstrap, and future work would include refining it to a more dynamic application with buttons and drop-down menus depending on the added tasks of the receptionist. Further, we have used the webcam in laptop HP EliteBook, and future implementations would be extended over to use Raspberry Pi, or other IoT devices.

6. REFERENCES

[1] H. H. Lwin, A. S. Khaing, and H. M. Tun, “Automatic Door Access System Using Face Recognition,” Int. J. Sci. Technol. Res, vol. 4, no. 6, pp. 294–299, 2015.

[2] E. Salvador and G. L. Foresti, “Interactive Reception Desk with Face Recognition-based Access Control,” in 2007 First ACM/IEEE International Conference on Distributed Smart Cameras. IEEE, 2007, pp. 140–146.

[3] R. Khandke, D. Singh, S. Pawar and D. Kadam, “Face recognition door lock using IoT”, International Research Journal of Engineering and Technology, vol. 7, no. 2, 2020

[4] Y. Yang and L. Li, "The Design and Implementation of a Smart e-Receptionist," in IEEE Potentials, vol. 32, no. 4, pp. 22-27, July-Aug. 2013, doi: 10.1109/MPOT.2012.2213851.

[5] R. Karunasena, P. Sandarenu, M. Pinto, A. Athukorala, R. Rodrigo and P. Jayasekara, "DEVI: Open-source Human-Robot Interface for Interactive Receptionist Systems," 2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM), 2019, pp. 378-383, doi: 10.1109/ICARM.2019.8834299.

[6] M. Podpora, A. Gardecki, and A. Kawala-Sterniuk, “Humanoid Receptionist Connected to IoT Subsystems and Smart Infrastructure is Smarter than Expected,” IFAC-PapersOnLine, vol. 52, no. 27, pp. 347–352, 2019

[7] D. Čupková, F. Hendrichovský, P. Papcun and I. Zolotová, "Cloud-enabled Assisted Living: The Role of Robot Receptionist in The Home Environment," 2019 IEEE 17th World Symposium on Applied Machine Intelligence and Informatics (SAMI), 2019, pp. 129-133, doi: 10.1109/SAMI.2019.8782787.

[8] O. Carlsson and D. Strom, “Designing a Virtual Receptionist Using Contextual Information and Natural Interfaces,” Master’s thesis, 2015.

[9] E. J. Hwang, B. Kyu Ahn, B. A. Macdonald and H. Seok Ahn, "Demonstration of Hospital Receptionist Robot with Extended Hybrid Code Network to Select Responses and Gestures," 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 8013-8018, doi: 10.1109/ICRA40945.2020.9197160.

[10] E. Levas, D. Leon and Y. Li, "A Smart Class Chatbot for Improving Student Learning and Engagement", in 2021 1st Computer Science Conference for CSU Undergraduates (CSCSU), Virtual Conference, 2021.

[11] J. Devlin, M.W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2019, pp. 4171–4186.

[12] L. Ni, C. Lu, N. Liu, and J. Liu, “Mandy: Towards A Smart Primary Care Chatbot Application,” in International symposium on knowledge and systems sciences. Springer, 2017, pp. 38–52.

[13] A. Holovaty and J. Kaplan-Moss, The Definitive Guide to Django: Web Development Done Right, Berkeley, CA: Apress, 2009.

[14] M. A. Abuzneid and A. Mahmood, "Enhanced Human Face Recognition Using LBPH Descriptor, Multi-KNN, and Back-Propagation Neural Network," in IEEE Access, vol. 6, pp. 20641-20651, 2018, doi: 10.1109/ACCESS.2018.2825310.

[15] N. Rekha and M. Kurian, “Face Detection in Real Time Based on HOG,” International Journal of Advanced Research in Computer Engineering & Technology, vol. 3, no. 4, 2014.

[16] Y. Pang, L. Zhang, M. Li, Z. Liu, and W. Ma, “A Novel Gabor-LDA based Face Recognition Method,” in Pacific-Rim Conference on Multimedia. Springer, 2004, pp. 352–358.

[17] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, “Regularization Studies of Linear Discriminant Analysis in Small Sample Size Scenarios with Application to Face Recognition,” Pattern recognition letters, vol. 26, no. 2, pp. 181–191, 2005.