

List of Scripts and their function

Script Name	Function
Prediction_HW_Server.py	Using height and Weight, the identity is predicted
Prediction_Img_Server.py	Prediction of Identity is reinforced by Image Processing
FaceRecognizer.py	Image Processing – Face Recognition Through LBPH Algorithm
TrainHeightAndWeight.py	Trains the height and weight data to make prediction model

List of Tables

Table Name	Use
SmartDoor_Diagnostics	Error Reports on Smart Door
SmartDoor_PeopleEntryExitDetail	Occupancy Details – Person Entering/Exiting
SmartDoor_PeopleCount	Occupancy Count
SmartDoor_Face_Identity	Face Database Identity vs Name
SmartDoor_Face_PredictionRank	Top 3 Ids thro Face Recognition
SmartDoor_HW_PredictionRank	Top 3 Ids thro Height and Weight Classification

Training:

1. Height and Weight Data:

1. Go to the Folder “SmartDoor_New”

2. Run the script as :

python TrainHeightAndWeight.py <From which SessionID> <Of Room ID :>

2. Image:

1. Go to the Folder “SmartDoor_New/TrainingForImages”

2. Collect images by recording video using the script “VideoRecord.py”

and give the LabMemberId as per the detail in the table “SmartDoor_Face_Identity” and the video record number.

The video gets stored in the directory “Recording”

3. To extract the images, run the script “Convenient_DataSetCreator.py”

and give the LabMemberId and the video record number.

A directory <ID> (in the directory “LabFacesByID”) will be updated with extracted images (or if it doesn't exist, it will be newly created)

4. Delete the blurred or unwanted images (if needed , ie, optional) in the directories in the directory “LabFacesByID”.

5. Copy all the images to one single directory “LabMembersFacesDB”

6. Go to the folder “SmartDoor_New/FacialImageTraining”
7. Run the script “Trainer.py”
8. Trained file will be stored in the directory
“SmartDoor_New/FacialImageTraining/RecognizerFiles”
9. Update the table to map the identity with the name

The screenshot shows the MySQL Workbench interface. The left sidebar contains the 'MANAGEMENT' and 'SCHEMAS' panels. The 'SCHEMAS' panel shows a list of databases including 'datapool', 'dummy', 'ninja_seil', 'phpmyadmin', 'SCC', 'SEILAPPLIANCES', 'seil_monitoring', and 'wordpress'. The 'datapool' database is selected. The main window displays a query editor with the following SQL query: `SELECT * FROM datapool.SmartDoor_Face_Identity;`. Below the query editor, the 'Result Set Filter' is empty. The query results are displayed in a table with the following columns: '#', 'Identity', 'Room_No', and 'Name'. The table contains 16 rows of data, with the last row marked with an asterisk (*). The 'Action Output' panel at the bottom shows the execution of the query, with a message indicating that 16 rows were returned.

#	Identity	Room_No	Name
1	1	1	Anshul
2	2	1	Akshay
3	3	1	Avisha
4	4	1	Ayush
5	5	1	Bhushan
6	6	1	Chaitra
7	7	1	Hareesh
8	8	1	Karan
9	9	1	Krithi
10	10	1	Parth
11	11	1	Priyanka
12	12	1	Rohit
13	13	1	Shaunak
14	14	1	Shobana
15	15	1	Siddharth
16	16	1	Spoorthy
*	NULL	NULL	NULL

Running:

1. Change the directory to:
`cd /home/stark/SmartDoor_New`
2. Run the Prediction according to Height and Weight Script as follows: Parameters: RoomID (SEIL - 1, ERTS - 2), INFOPORT (According to RPi)
`python Prediction_HW_Server.py 1 12345`
3. Run the Prediction according to Video Processing Script as follows: Parameters: RoomID (SEIL - 1, ERTS - 2), CAMPORT (According to RPi)
`python Prediction_Img_Server.py 1 12347`
4. `ssh pi@10.129.23.214`
`cd ~/SmartDoor_New/`
`sudo python SmartDoor_Client_Run.py`