SMART DOORBELL-SYSTEM USING COMPUTER VISION AND IOT

PROJECT BY:

WISAM RAMADAN

SAMER HAMMOUD

SUPERVISOR: DR. ABDEL-MEHSEN AHMAD

COMMITTEE MEMBER: DR. SAMIR OMAR

SPRING 2020-21

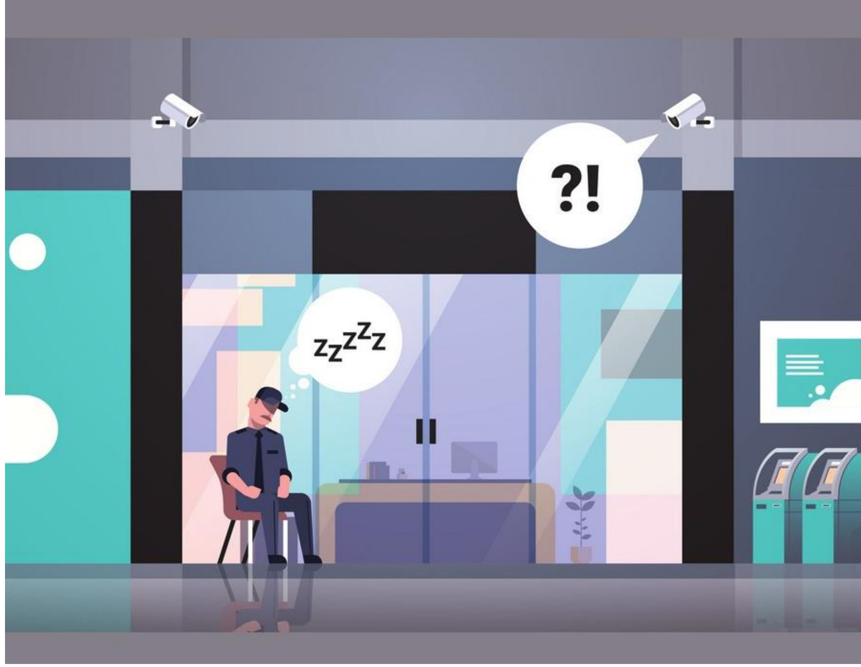
OUTLINE

- INTRODUCTION
- PROBLEM STATEMENT
- RELATED WORK
- PROPOSED SYSTEM
- SYSTEM DESIGN
- SYSTEM IMPLEMENTATION
- RESULTS
- CONCLUSION & FUTURE WORK

INTRODUCTION

INTRODUCTION: SECURITY POSITION

- Traditional security practices.
- Expansion to involve hardware system (e.g., Surveillance Camera, Alarming system).
- What is IoT?
- What is a Doorbell System? How is it different?



INTRODUCTION: PROBLEM STATEMENT

- ➢ Traditional Reporting.
- ➢ Documentation limitation.
- ➢ No prevention contribution.
- ➢ Autonomous Report and fast action?

RELATED WORK

RELATED WORK: GOOGLE NEST HELLO

Advantages:

- Human Detection and Reporting (App notification).
- Facial Recognition (Voice Announcement of Visitor’s Name).
- Two-way Audio.
- Remote Lock Control.

Disadvantages:

- Not autonomous.
- No Deterring module (e.g., Alarming system).

RELATED WORK: ARLO VIDEO DOORBELL

Advantages:

- Object Detection and Reporting (Human, Vehicle, Animal).
- Warning measure available (alarming system).
- Two-way Audio.
- Remote Lock Control.

Disadvantages:

- Not autonomous.
- No Facial Recognition.
- Any object detected is reported to users.

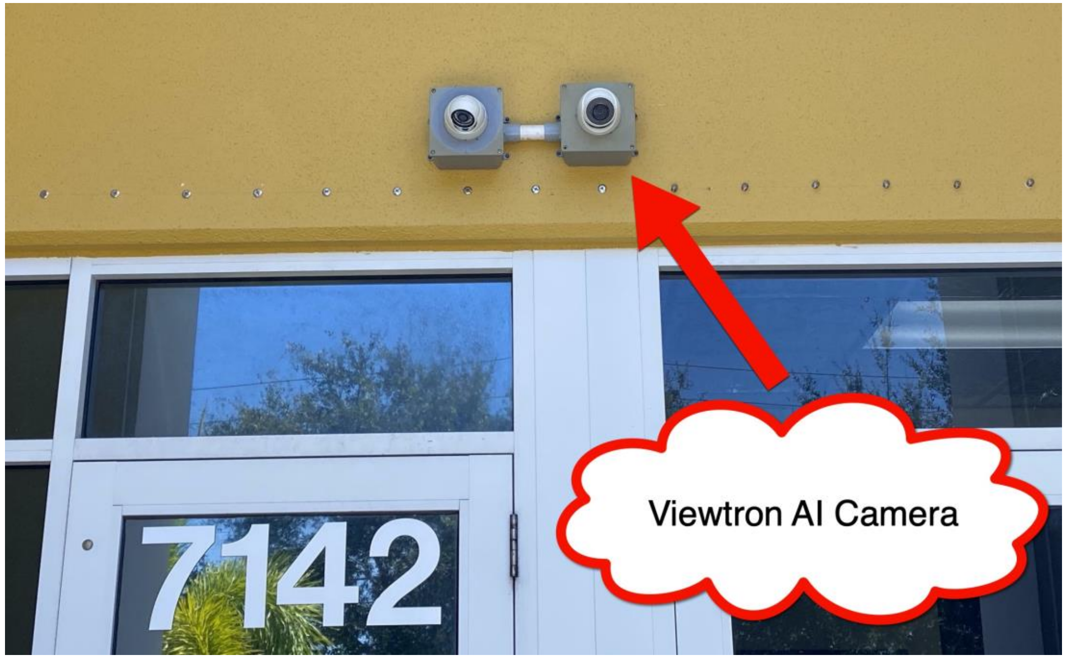
RELATED WORK: AI FACE RECOGNITION SECURITY CAMERA

Advantages:

- Face & Object Detection.
- Facial Recognition.
- Autonomous Operation.
- Control via Mobile App.

Disadvantages:

- No deterring measures available.
- No Liveness Detection (Anti-spoofing measure).



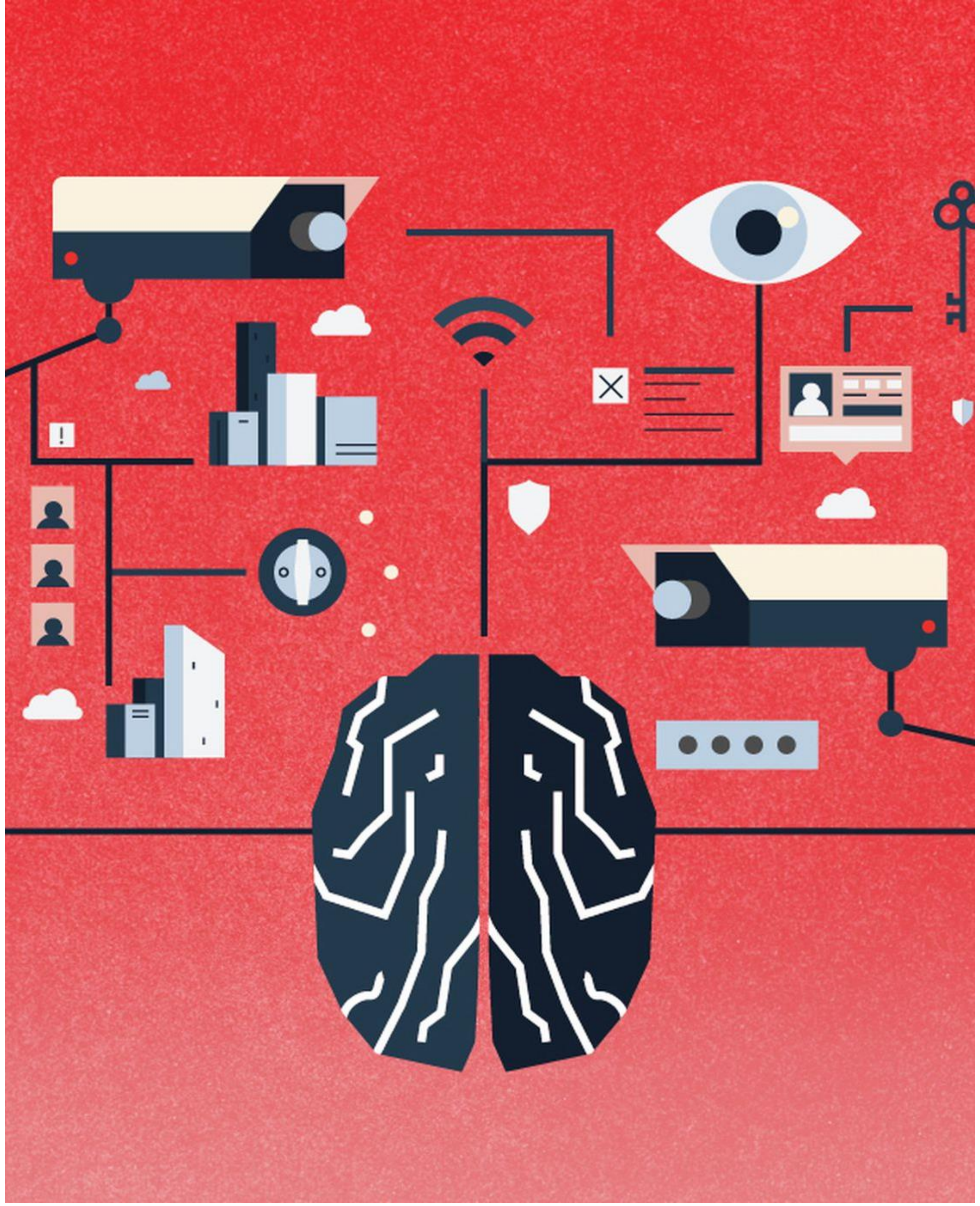
RELATED WORK: COMPARISON

Method	Advantages	Disadvantages
Google Nest Hello	• Interfaces with other google devices (voice announcement, camera…etc.). • Small size. • Managed by smart devices (ok google).	• 298$ cost. • Monthly subscription required (for extra features). • No decent cloud space (for video recordings).
Arlo Doorbell System	• 179$. • Different modes of power (battery & wall outlet). • Easy installation. • Small size. • All features are provided with no additional cost. • Managed by smart devices (Alexa, ok google). • Reliable against weather conditions.	• No decent cloud space.
AI Face Recognition Security Camera	• Facial recognition capability. • Autonomous door unlock. • Captured footage and video recordings stored locally.	• Big size. • Requires separate video recorder. • Installation is hard. • 350-500 cost.

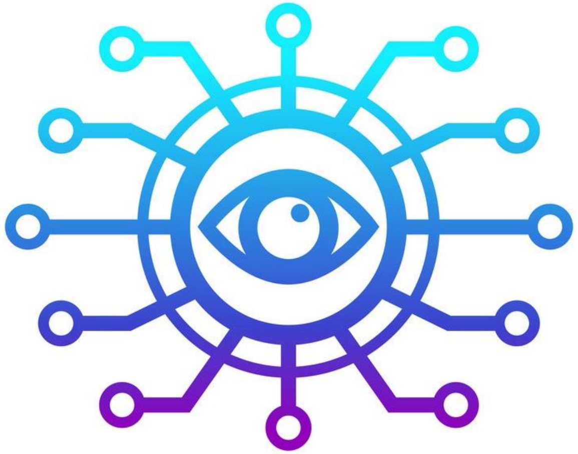
PROPOSED SYSTEM

OUR SOLUTION: MAKE SECURITY SMARTER(1/2)

- ✓ Use advanced technologies to improve performance by:
  - o Face Detection Module.
  - o Liveness Detection (Anti-spoofing).
  - o Facial Recognition.
  - o Real-time Reporting (App notification).
  - o Autonomous Actions (Certain cases).
  - o Remote Control (app control panel).
  - o Deterrence measures (Alarming system).



OUR SOLUTION: MAKE SECURITY SMARTER(2/2)



- ❖ Approach:
  - ✓ Use Computer Vision and Machine Learning to:
    1. Provide Computer with vision capabilities.
    2. Create models for object classification (Face ROIs, Liveness, Features’ vector, ..etc.).
  - ✓ User Internet of Things (IoT) to:
    1. Create a presence for the system on the internet.
    2. Helps create a communication link with the user.

WHERE TO USE THIS SYSTEM?

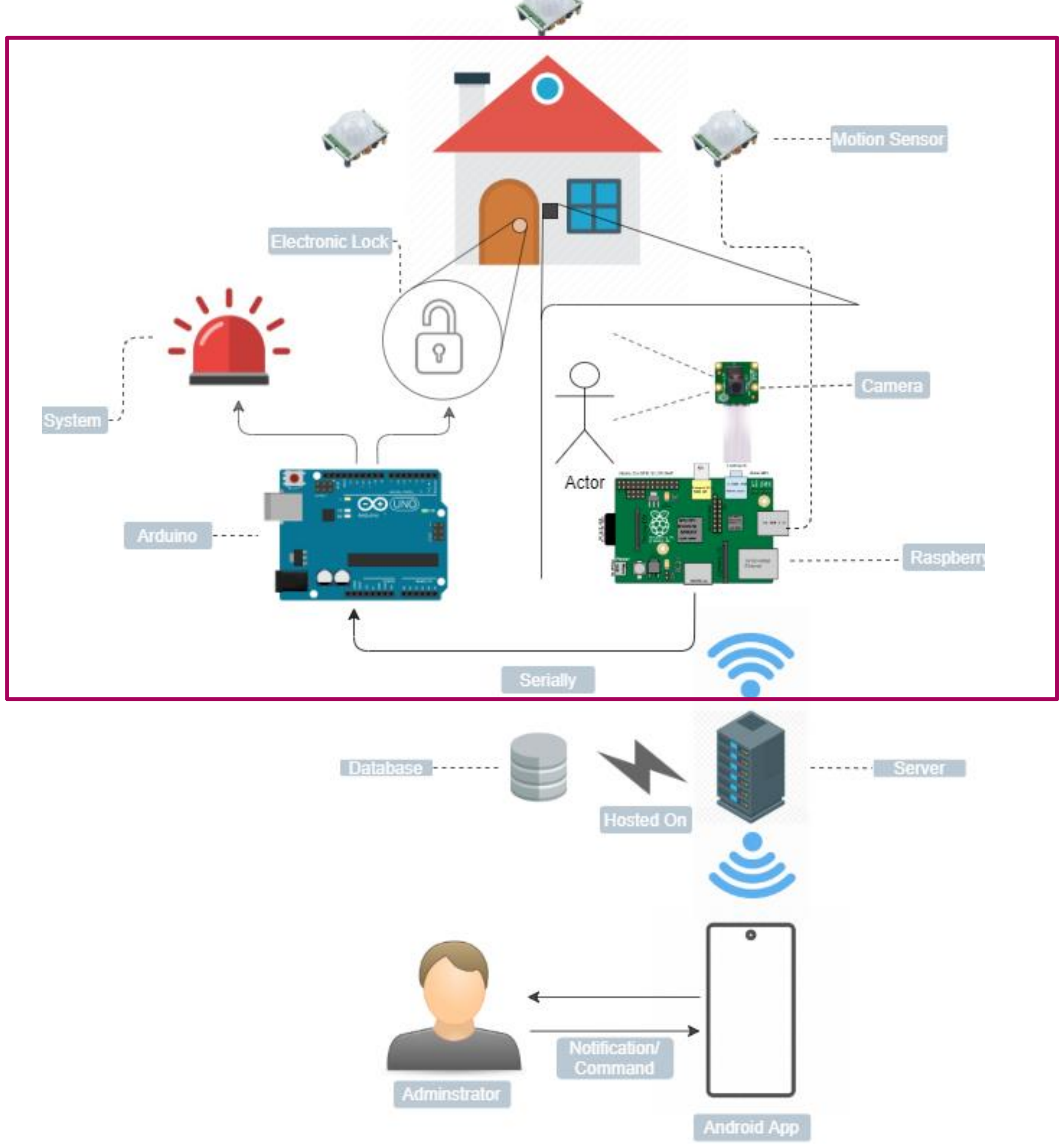
- Private houses:
  - This system is mainly designed for house’s entrance.
  - Can be placed any where around the house (e.g., backyard, windows).
- Offices and Companies:
  - Placed at main entrance to permit employees' access only.
  - Private spots where only certain people allowed to be in.

SYSTEM DESIGN

SYSTEM ARCHITECTURE

❖ Hardware Parts:

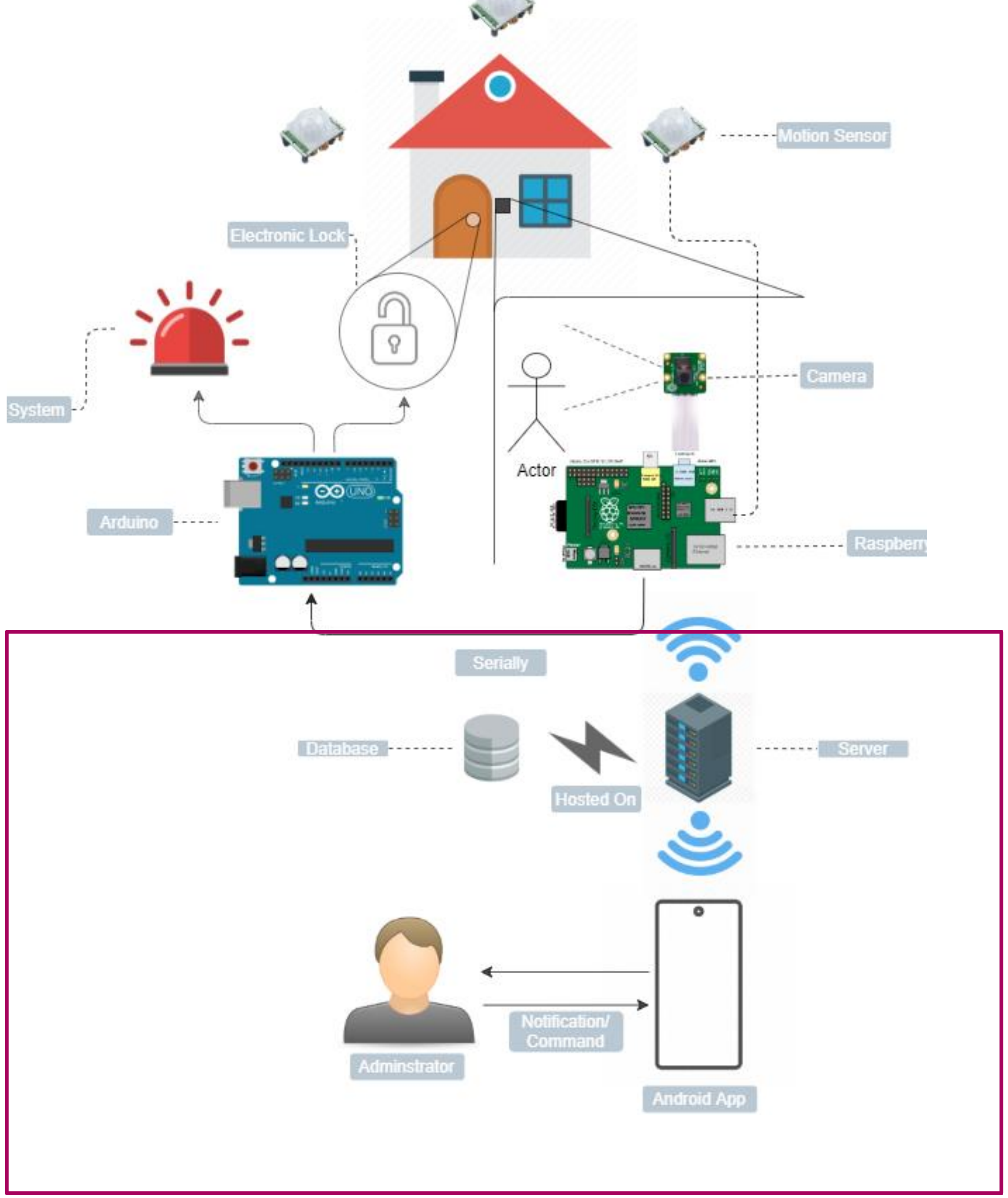
1. Raspberry Pi (Peripherals Connection link).
2. Arduino UNO (Slave Peripheral).
3. Camera Module (Visual Data).
4. Motion Sensors (Motion Detection).
5. Door Lock & Alarming System (Access vs. Block).



SYSTEM ARCHITECTURE

❖ Software Parts:

1. Database (Laptop Local Server).
2. Android Application (Control & Monitoring Panel).



SYSTEM SOFTWARE MODEL

❖Raspberry Pi:

1. Python (OpenCV, TensorFlow).

❖ Arduino UNO:

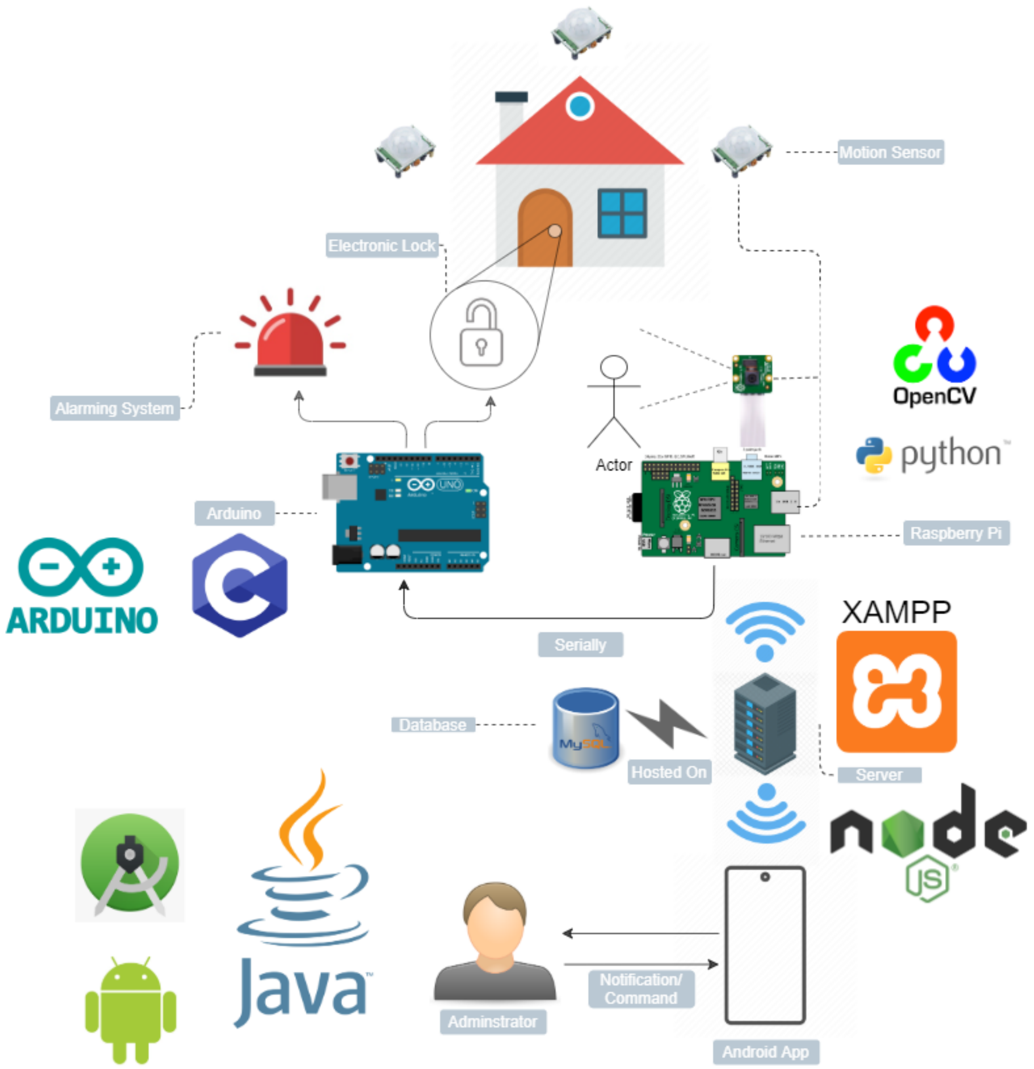
1. C++.

❖ Server:

1. MySQL.
2. PHP.

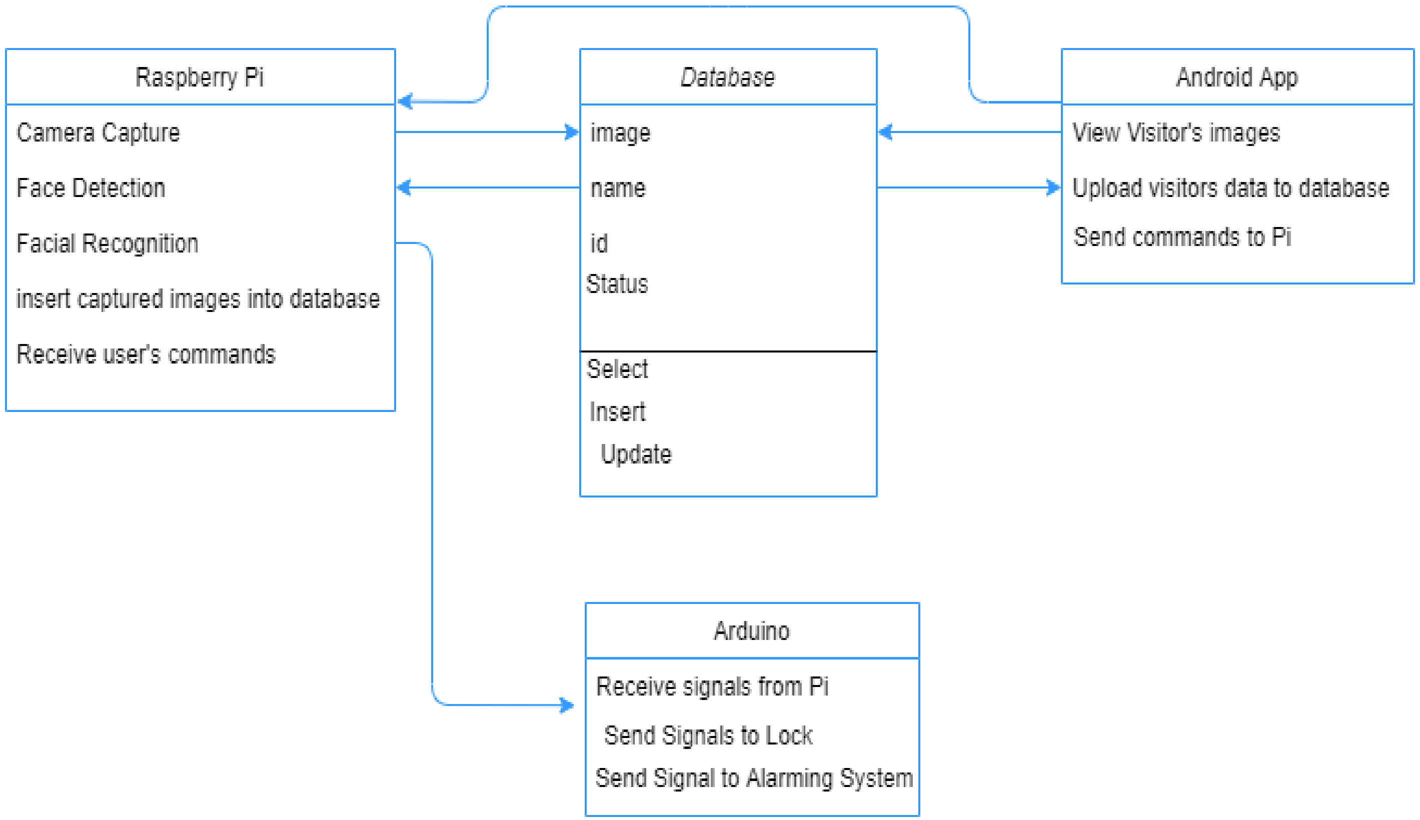
❖ Android Application:

1. Java.
2. XML.



CLASS DIAGRAM

- ❖ Database Class:
  - ✓ For commands and notifications.
  - ✓ Communication link between user and main system.
- ❖ Raspberry Pi Class:
  - ✓ Computer Vision & IoT.
  - ✓ Reporting and Command Execution.
- ❖ Android App Class:
  - ✓ Receive notifications.
  - ✓ Provides Control features.
- ❖ Arduino Class:
  - ✓ Door Lock and Alarming System triggering.



Raspberry Pi

Camera Capture
Face Detection
Facial Recognition
insert captured images into database
Receive user's commands

Database

image
name
id
Status

Select
Insert
Update

Android App

View Visitor's images
Upload visitors data to database
Send commands to Pi

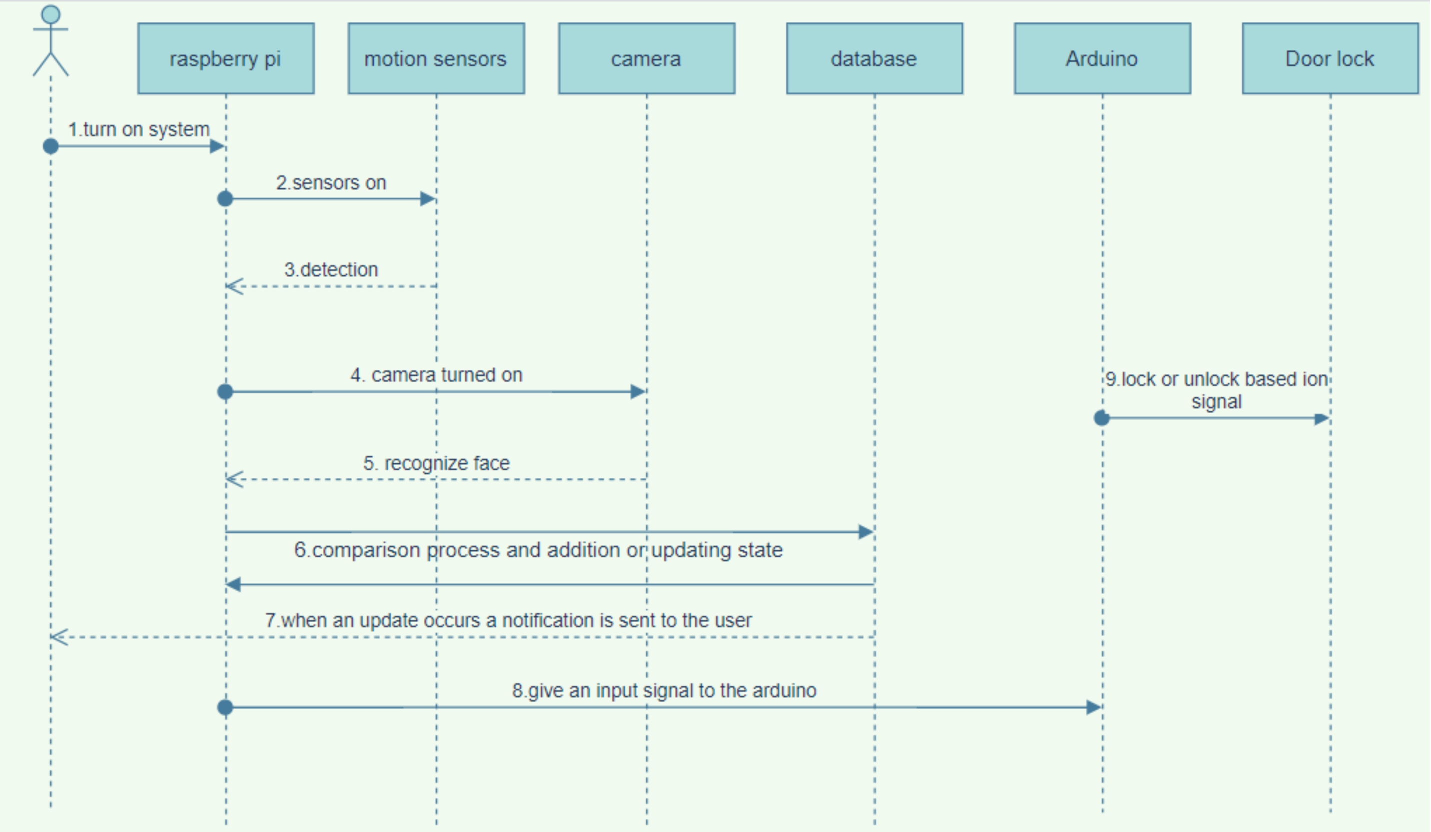
Arduino

Receive signals from Pi
Send Signals to Lock
Send Signal to Alarming System

SEQUENCE DIAGRAM

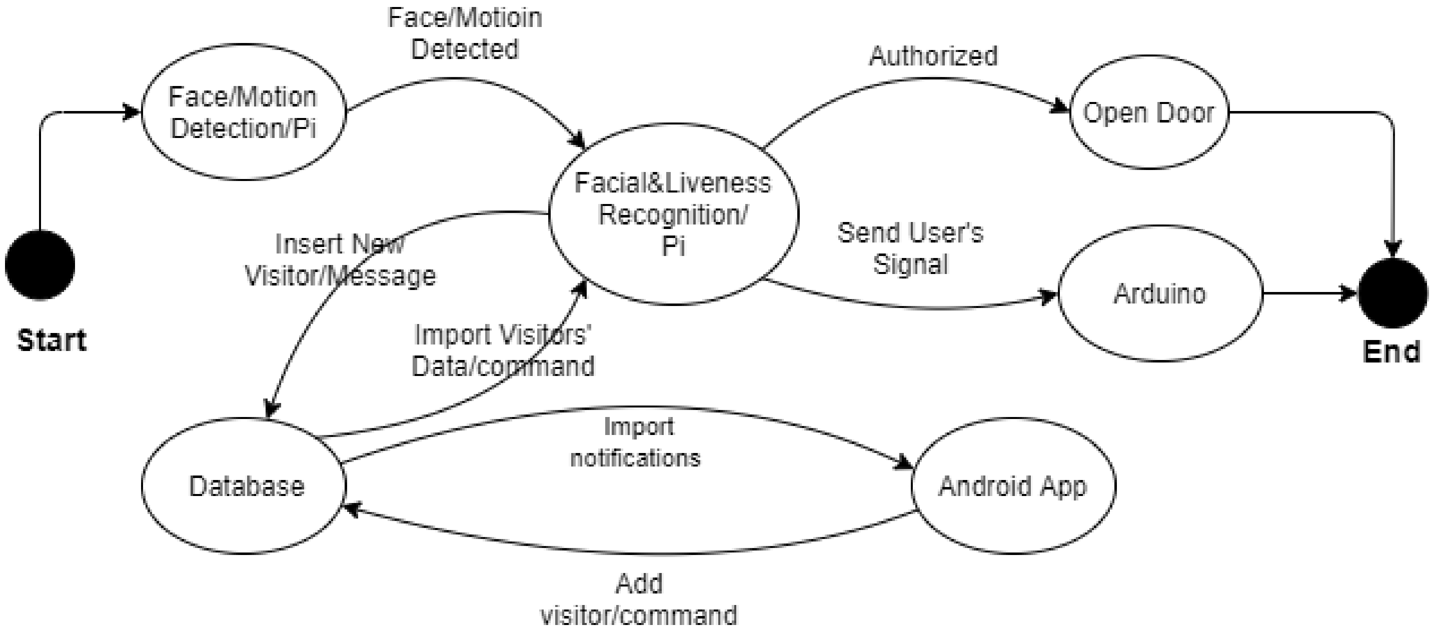
- ❖ Owner will be the one to trigger the system firstly.

- ❖ Automation of the system will get it to situations where the sequence might change.


raspberry pi
motion sensors
camera
database
Arduino
Door lock
1.turn on system
2.sensors on
3.detection
4. camera turned on
9.lock or unlock based ion signal
5. recognize face
6.comparison process and addition or updating state
7.when an update occurs a notification is sent to the user
8.give an input signal to the arduino

STATE DIAGRAM

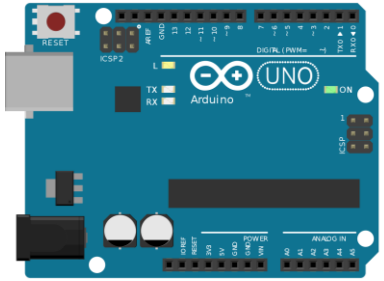
- Human Detection.
- Liveness Detection & Facial Recognition.
- Autonomous Action/User Notification.
- User Control.
- Command Handling.



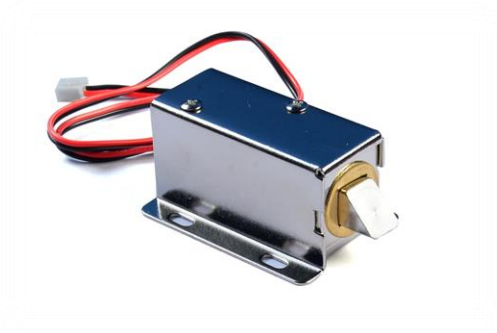
SYSTEM IMPLEMENTATION

HARDWARE TOOLS

PIR Motion Sensor

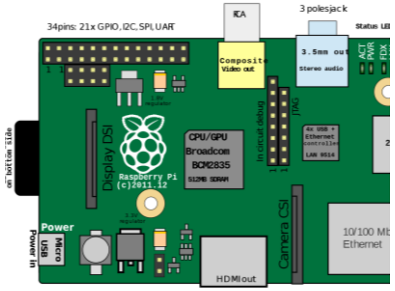


Arduino UNO



Electronic Lock

Raspberry Pi Camera 8MP



Raspberry PI Model B 2GB

Buzzer

SOFTWARE TOOLS

OpenCV Library

TensorFlow Python Library

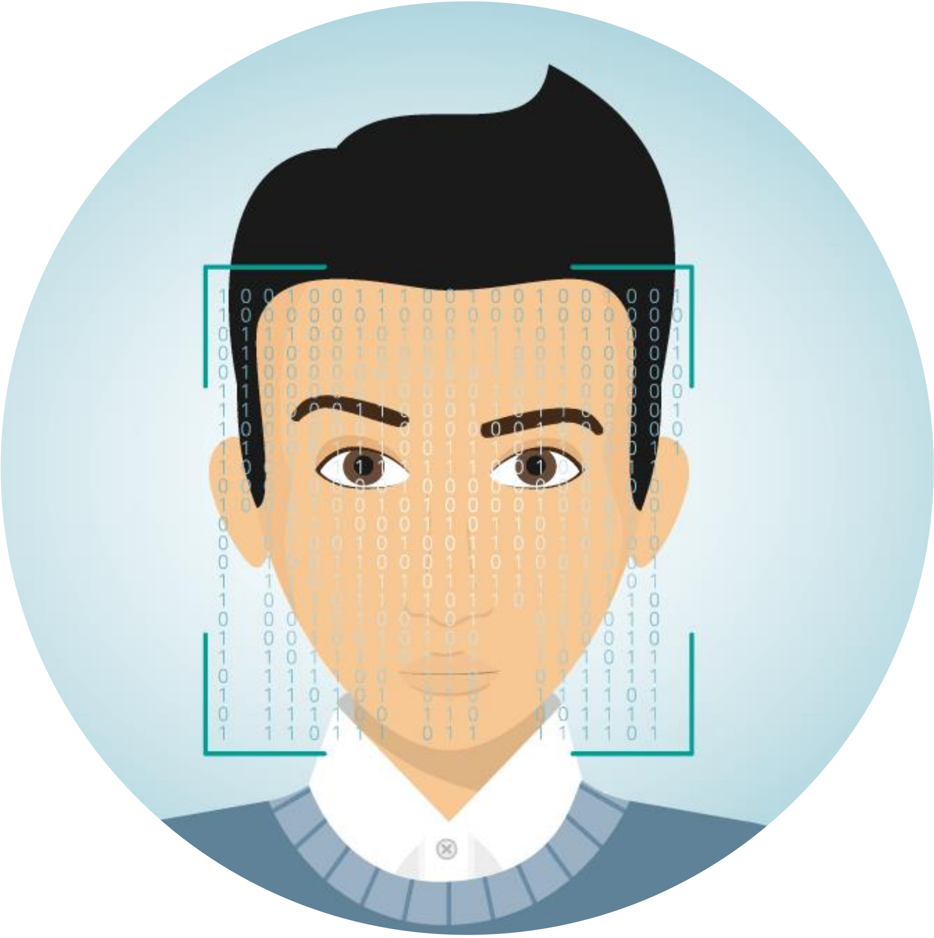
Java

PHP (Server Side)

MySQL (Database)

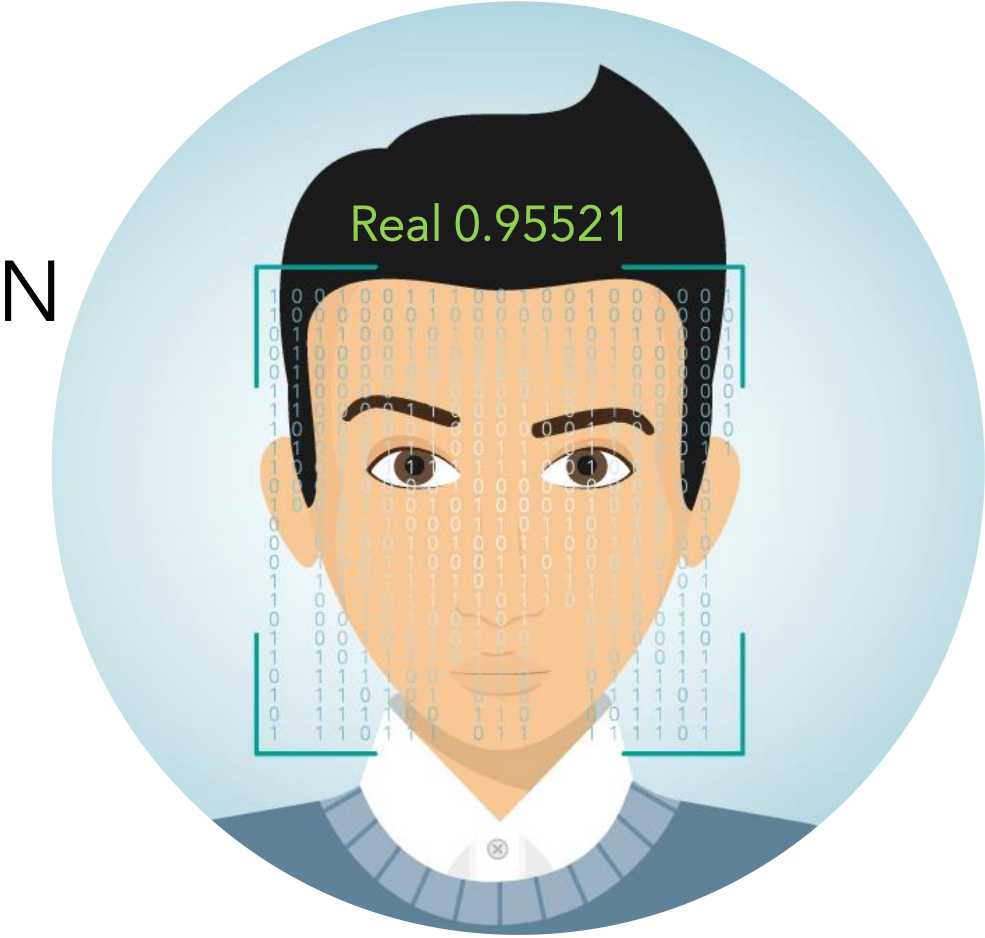
IMPLEMENTATION SUMMARY (1/4): GATHER FACES

- Use OpenCV to collect faces from dataset videos (Fake & Real).
- Save Faces ROIs to fake and real directories.
- Run face detector to detect human presence.



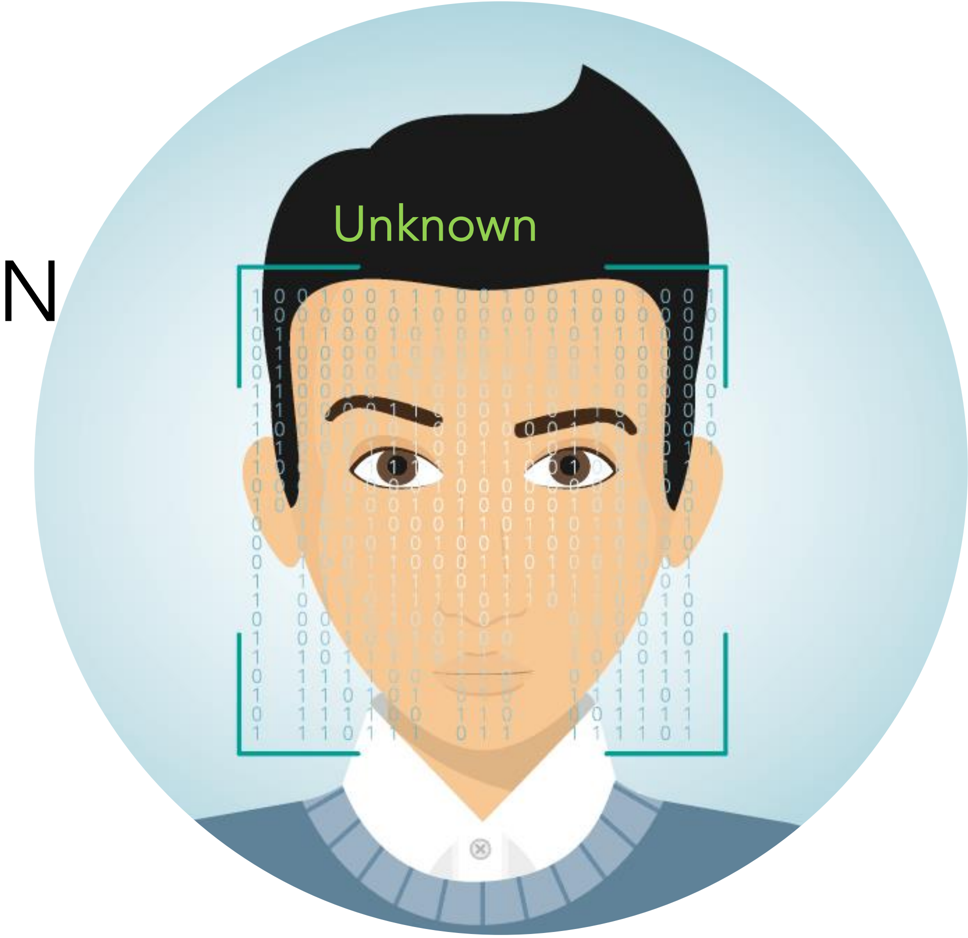
IMPLEMENTATION SUMMARY (2/4): LIVENESS DETECTION

- Based on Machine Learning.
- Classifies faces into real or fake.
- Implemented to detect spoofing attempts.



IMPLEMENTATION SUMMARY (3/4): FACIAL RECOGNITION

- Use Python face-recognition library (face matching).
- Create and store a dataset of authorized people.
- Capture an image and match it with images in prestored dataset.
- If match occurs, extracts file's name; otherwise, do nothing.

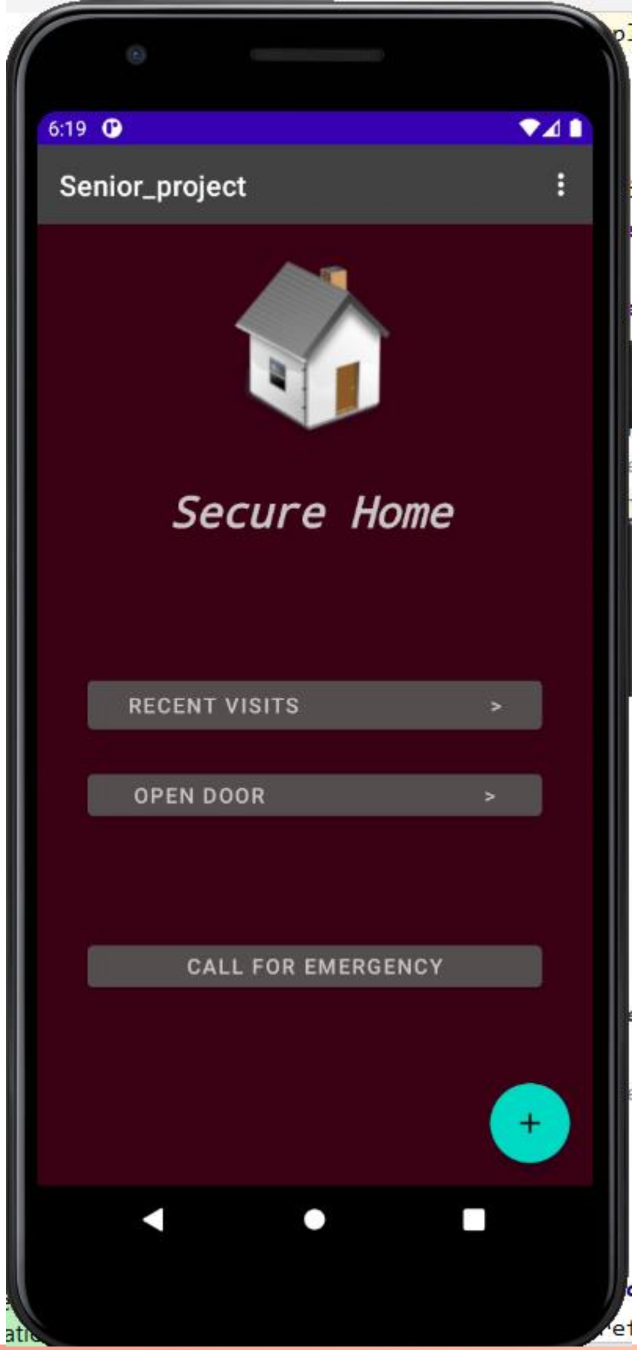


IMPLEMENTATION SUMMARY (4/4): TAKE ACTION

- Face is fake? Push notification => breakdown => wait command.
- Face is real but unknown? Push notification.
- Face is real and recognized? Open door lock.

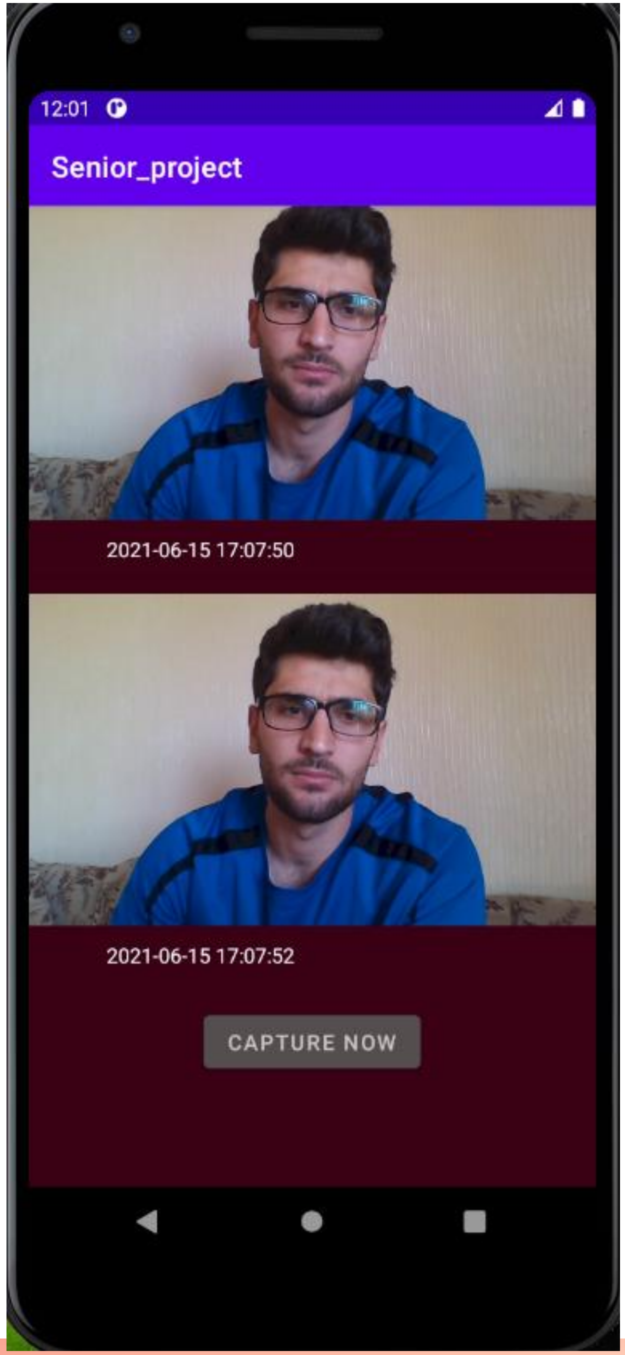
IMPLEMENTATION (5/7): ANDROID APP

- Pop-up activity where the main starts.
- Each button takes the user to another activity.
- While application is running in the background service will keep a track on any new visits



IMPLEMENTATION(6/7): ANDROID APP

- Pictures in this activity are brought from database.
- Images are added to the database with an id.
- The last id one is kept in the users app.

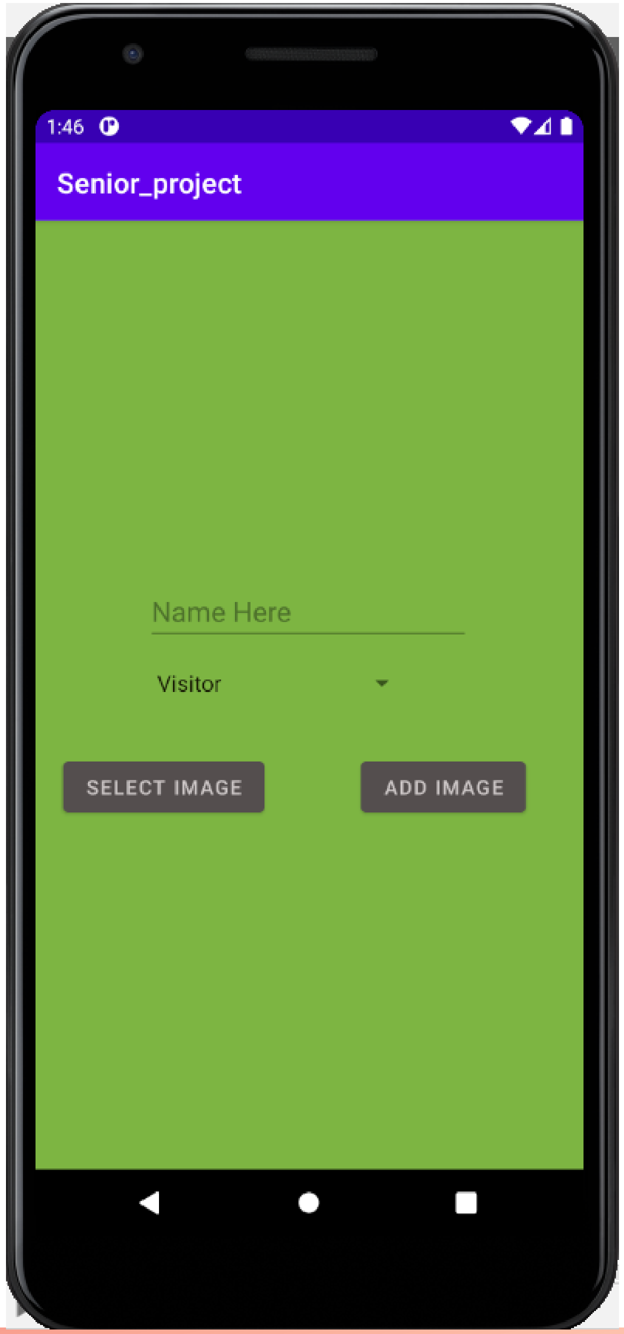


2021-06-15 17:07:50

2021-06-15 17:07:52

IMPLEMENTATION(7/7): ANDROID APP

- ➢ Add guest activity allows the user to add specific people.
- ➢ Face of added ones are token from gallery.
- ➢ The user must provide a name for the person to be added.



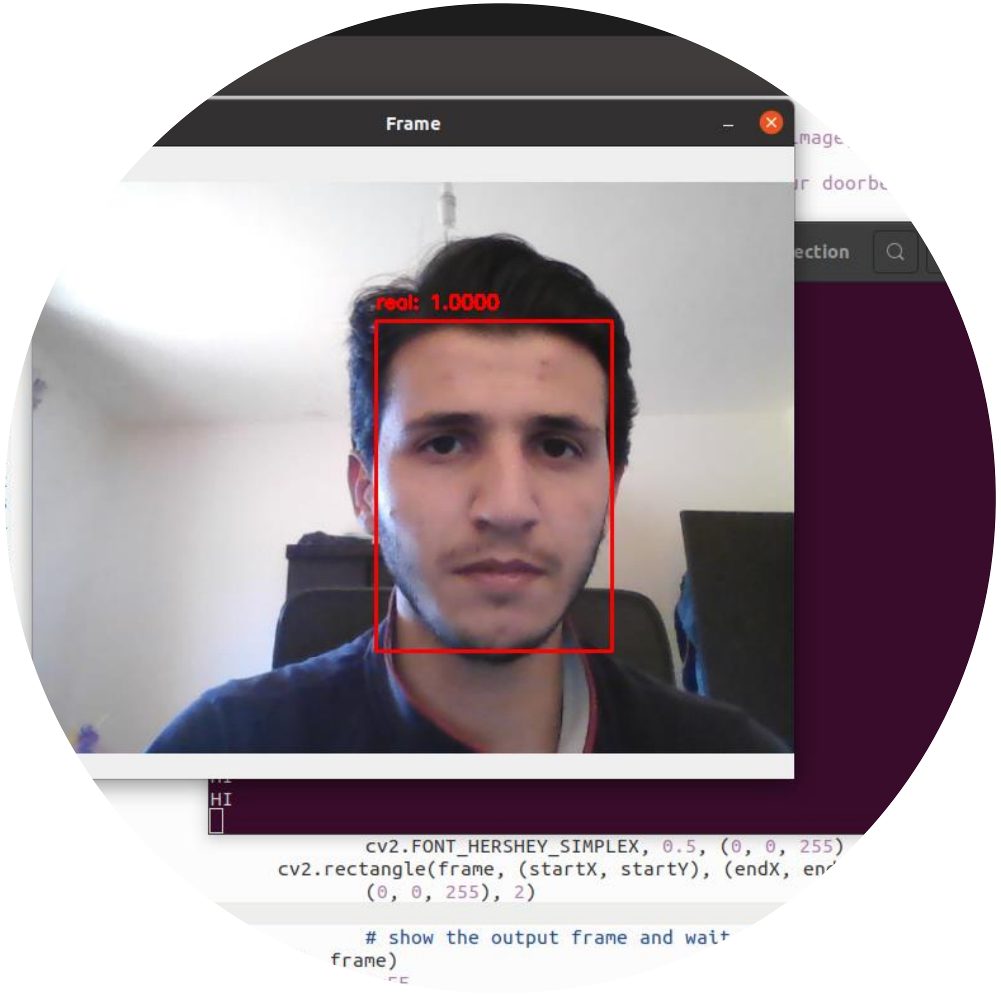
Name Here

Visitor

RESULTS

RESULTS (1/6): LIVENESS DETECTION

- ➢ Liveness Detection showed about 99% accuracy after detecting a real face.



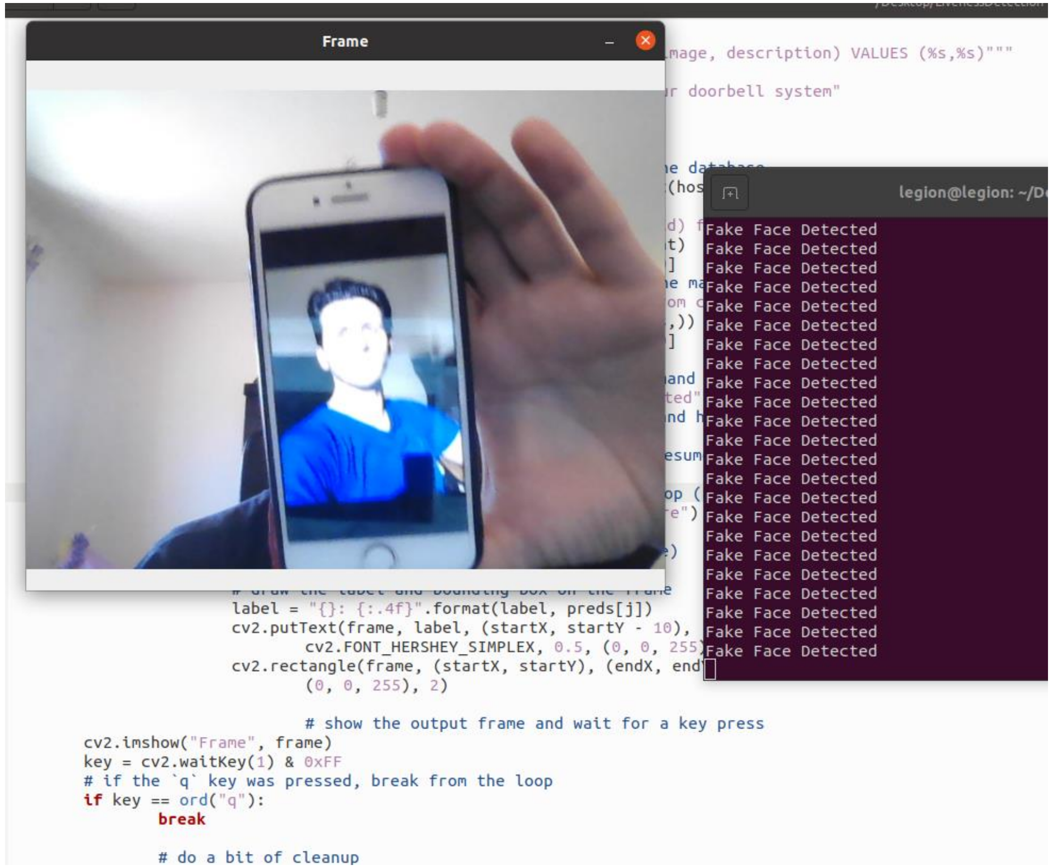
```
HI
```

```
        cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255)
cv2.rectangle(frame, (startX, startY), (endX, en
        (0, 0, 255), 2)

        # show the output frame and wait
    frame)
```

RESULTS (2/6): LIVENESS DETECTION

- Photo pointed at the camera.
- Liveness Detector classified it as fake.



```
mage, description) VALUES (%s,%s)"""

r doorbell system"
```

```
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
```

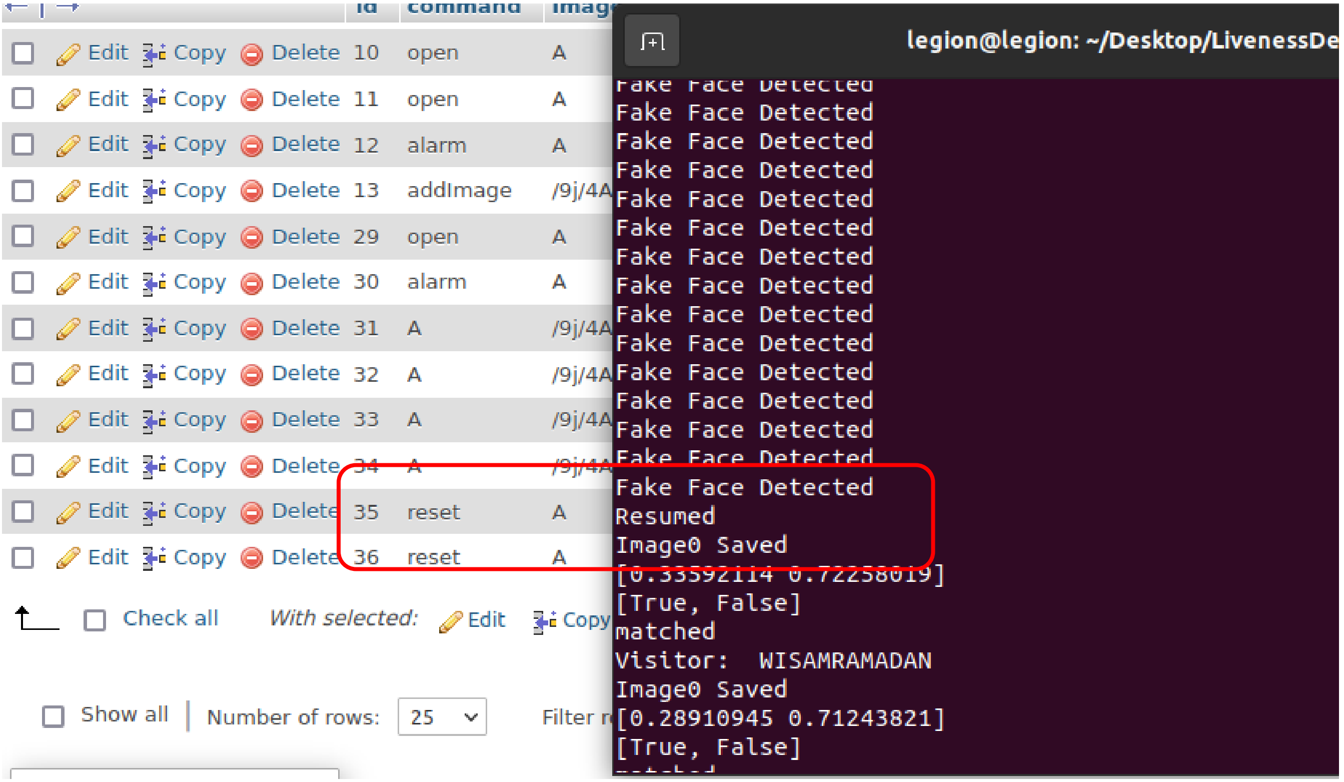
```
                    label = "{}: {:.4f}".format(label, preds[j])
                    cv2.putText(frame, label, (startX, startY - 10),
                            cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255)
                    cv2.rectangle(frame, (startX, startY), (endX, end
                            (0, 0, 255), 2)

                            # show the output frame and wait for a key press
        cv2.imshow("Frame", frame)
        key = cv2.waitKey(1) & 0xFF
        # if the `q` key was pressed, break from the loop
        if key == ord("q"):
                break

                # do a bit of cleanup
```

RESULTS (3/6): SYSTEM RESET

- ➢ Spoofing attempt notification was received.
- ➢ User pressed “reset” to reset the system.



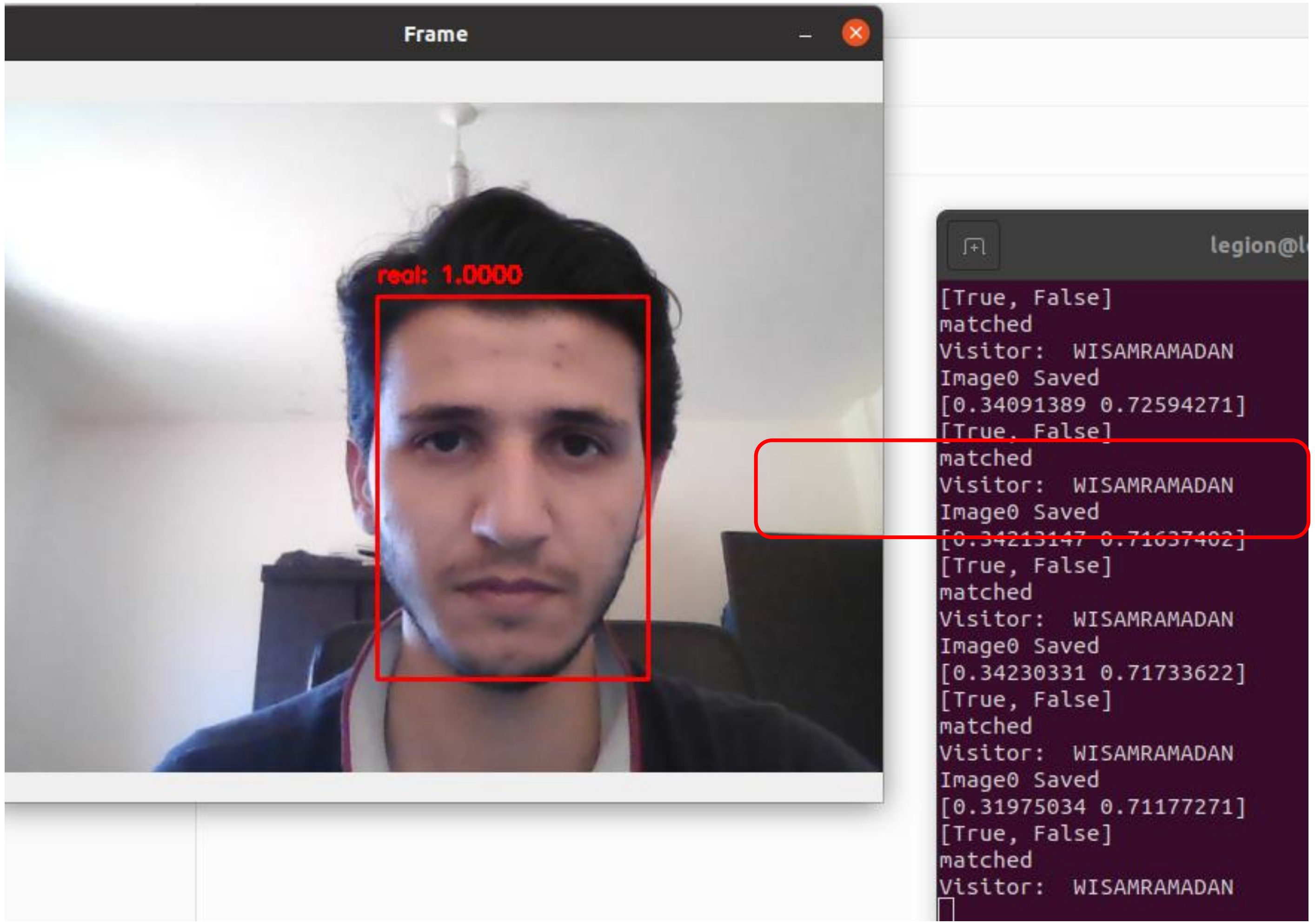
				id	command	image
□	Edit	Copy	Delete	10	open	A
□	Edit	Copy	Delete	11	open	A
□	Edit	Copy	Delete	12	alarm	A
□	Edit	Copy	Delete	13	addImage	/9j/4A
□	Edit	Copy	Delete	29	open	A
□	Edit	Copy	Delete	30	alarm	A
□	Edit	Copy	Delete	31	A	/9j/4A
□	Edit	Copy	Delete	32	A	/9j/4A
□	Edit	Copy	Delete	33	A	/9j/4A
□	Edit	Copy	Delete	34	A	/9j/4A
□	Edit	Copy	Delete	35	reset	A
□	Edit	Copy	Delete	36	reset	A

legion@legion: ~/Desktop/LivenessDe

```
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Fake Face Detected
Resumed
Image0 Saved
[0.33592114 0.72258019]
[True, False]
matched
Visitor:  WISAMRAMADAN
Image0 Saved
[0.28910945 0.71243821]
[True, False]
```

RESULTS (4/6) FACIAL RECOGNITION

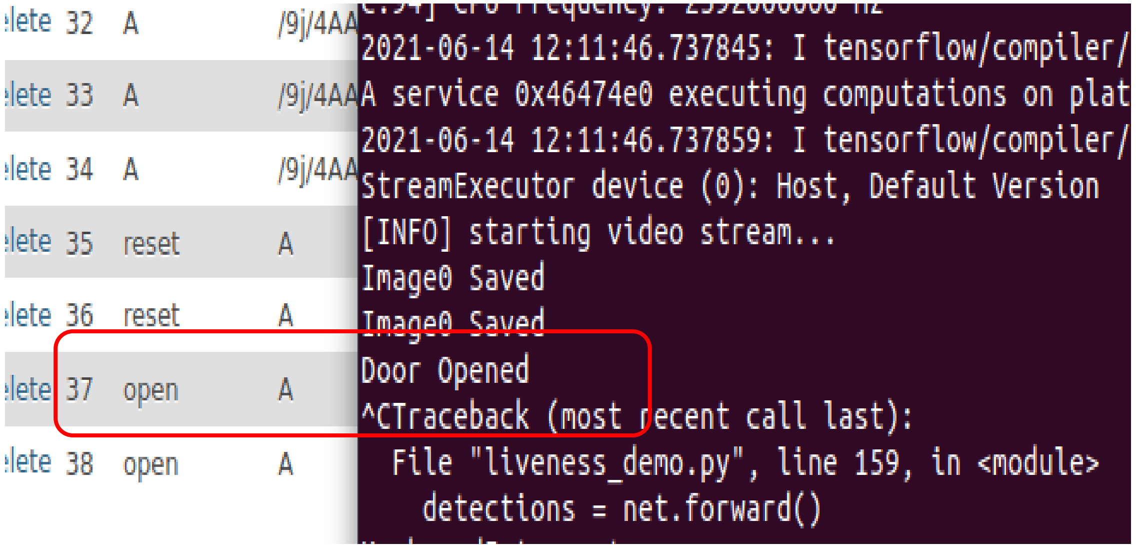
- System classified real person as real.
- It found a match for the face,
- Then, extracted the matched file’s name.



```
[True, False]
matched
Visitor:  WISAMRAMADAN
Image0 Saved
[0.34091389 0.72594271]
[True, False]
matched
Visitor:  WISAMRAMADAN
Image0 Saved
[0.34213147 0.71637402]
[True, False]
matched
Visitor:  WISAMRAMADAN
Image0 Saved
[0.34230331 0.71733622]
[True, False]
matched
Visitor:  WISAMRAMADAN
Image0 Saved
[0.31975034 0.71177271]
[True, False]
matched
Visitor:  WISAMRAMADAN
```

RESULTS (5/6): OPEN DOOR

- ➢ User sent "Open Door" command from android app.
- ➢ The system opened the door.

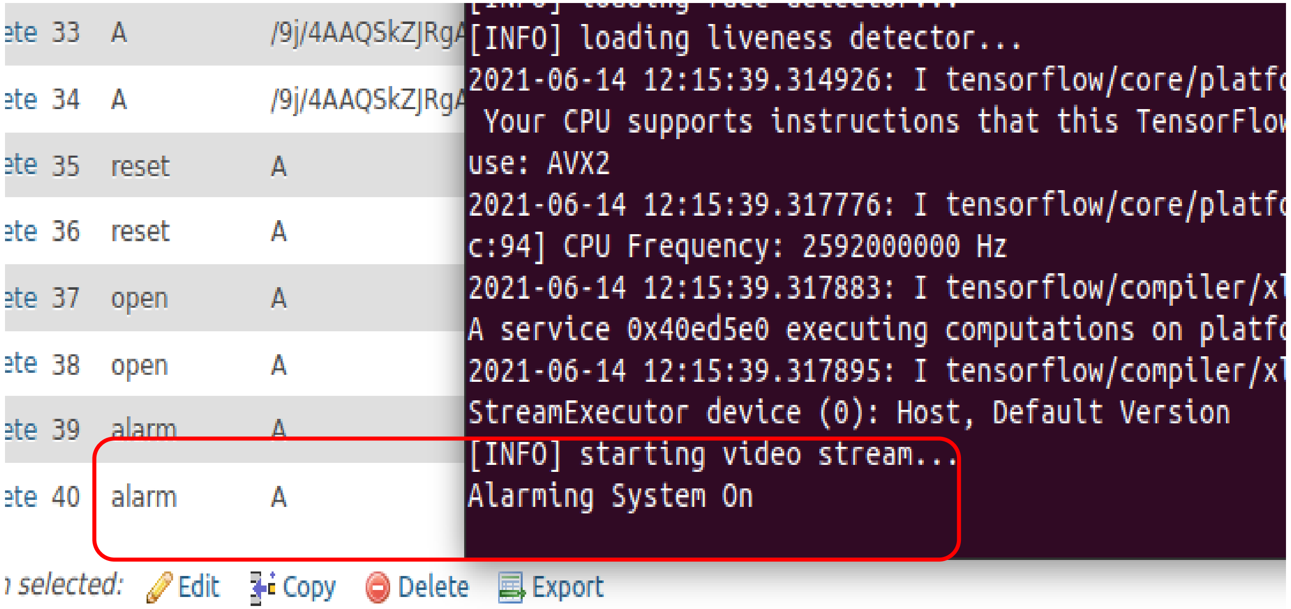


lete	32	A	/9j/4AA
lete	33	A	/9j/4AA
lete	34	A	/9j/4AA
lete	35	reset	A
lete	36	reset	A
lete	37	open	A
lete	38	open	A

```
2021-06-14 12:11:46.737845: I tensorflow/compiler/
A service 0x46474e0 executing computations on plat
2021-06-14 12:11:46.737859: I tensorflow/compiler/
StreamExecutor device (0): Host, Default Version
[INFO] starting video stream...
Image0 Saved
Image0 Saved
Door Opened
^CTraceback (most recent call last):
  File "liveness_demo.py", line 159, in <module>
    detections = net.forward()
```

RESULTS (6/6): ALARMING SYSTEM TURN ON

- ➢ User sent "Turn on alarm" command.
- ➢ The system turned alarming system on.



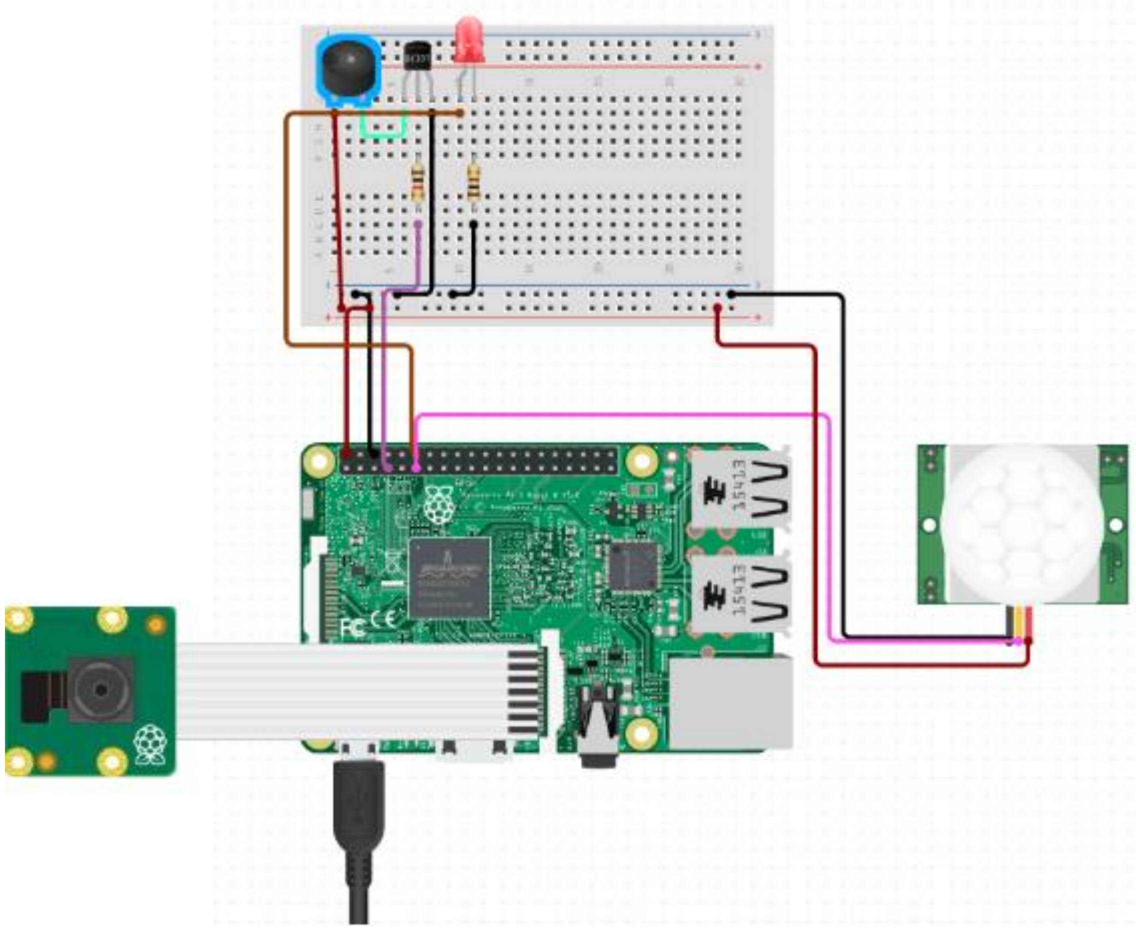
ete	33	A	/9j/4AAQSkZJRgA
ete	34	A	/9j/4AAQSkZJRgA
ete	35	reset	A
ete	36	reset	A
ete	37	open	A
ete	38	open	A
ete	39	alarm	A
ete	40	alarm	A

```
[INFO] loading face detector...
[INFO] loading liveness detector...
2021-06-14 12:15:39.314926: I tensorflow/core/platfo
 Your CPU supports instructions that this TensorFlow
use: AVX2
2021-06-14 12:15:39.317776: I tensorflow/core/platfo
c:94] CPU Frequency: 2592000000 Hz
2021-06-14 12:15:39.317883: I tensorflow/compiler/xl
A service 0x40ed5e0 executing computations on platfo
2021-06-14 12:15:39.317895: I tensorflow/compiler/xl
StreamExecutor device (0): Host, Default Version
[INFO] starting video stream...
Alarming System On
```

CONCLUSION AND FUTURE WORK

CONCLUSION

- Machine learning provides great improvement to existing applications.
- This system's improvement is based on Computer Vision and IoT.
- Compared to related works:
  - It operates autonomously.
  - It detects spoofing attempts.
- However, there are still drawbacks with this proposed system.
  - Requires a large dataset (real & fake).
  - Dataset is hard to create.



CHALLENGES

- Not all failure conditions may be predicted.
- Camera systems are affected highly by various weather conditions.
- It was not feasible to test the quality of the system under different conditions.
- Liveness Detection and Facial Recognition may operate with less accuracy in darkness.
- This system was not tested with night vision camera.

FUTURE WORK

- ➢ The system can be improved by extending it to support:
  - ✓ One main administrator.
  - ✓ Two-way audio communication.
  - ✓ Video Recording.
  - ✓ Live Streaming.
  - ✓ More advanced infrared camera.
  - ✓ Interfaces to connect to other security systems (e.g., alarming system).
  - ✓ Using a more efficient computer.

THANK YOU!