**Door Sensors That Help You (Smart Doors)**

*Submitted by-*

*Group-23, Shivam Sharma (15ucs130)*

(c) jack\_1729

Licensed under the [MIT License](https://github.com/jack17529/Smart-Doors/blob/master/LICENSE)

**DEPARTMENT OF ELECTRICAL ENGINEERING**

**LNM INSTITUTE OF INFORMATION TECHNOLOGY, JAIPUR**

**May 2018**

© 2018, Shivam Sharma. All rights reserved

**DECLARATION**

I hereby certify that,

a) the work contained in this report is original and has been done by me under the guidance of my supervisor(s).

b) the work has not been submitted to any other Institute for any degree or

diploma.

c) I have followed the guidelines provided by the Institute in preparing the report.

d) I have conformed to the norms and guidelines given in the Ethical Code of

Conduct of the Institute.

e) Whenever I have used materials (data, theoretical analysis, figures, and text)

from other sources, I have given due credit to them by citing them in the text

of the report and giving their details in the references. Further, I have taken

permission from the copyright owners of the sources, whenever necessary.

(Signature of the Student)

Shivam Sharma (15ucs130)

Date:

09-05-2018

**CERTIFICATE**

This is to certify that the Dissertation Report entitled, “Smart Doors” submitted

by Mr “Shivam Sharma” to LNMIIT, Jaipur.

India, is a record of bonafide Project work carried out by him under my/our

supervision and guidance and is worthy of consideration for the award of the degree of Bachelor of Technology in Electrical Engineering of the Institute.

Name of the Supervisor, Name of the Supervisor

(Signature) (Signature)

Department of

LNM Institute of Information Technology

Jaipur

Date:

09-05-2018

**ACKNOWLEDGMENT**

This document is prepared by the inspiration received from Professor “.............”,

Head, and Department of CSE. Many colleagues at LNMIIT have carefully read and improved the document; their contributions are gratefully acknowledged.

Student Name and Roll Number

**Abstract**

It happens to everybody , when one is not in condition of opening the room door , but has to open it for someone. Opening door in the middle of urgent work or sleep is always painful. My project solves this problem using door sensors.

If you are not in the right situation of opening the door and someone knocks, if he/she is your friend then the program send email to him/her automatically. It uses face recognition to check weather he/she is your friend or not and sends the email if he/she is. First it checks your schedule for the current time and if you are busy or sleeping, then it sends the mail to your friends so that they can know when you are awake and free. Helping them to even know your next free slot is done , so that everyone is happy and stress free in the end.

.

The storage used in the project is Amazon S3 it has your friends pictures . Using google calendar API so as to access your calendar . Using Amazon Face Rekognition API for face recognition. Any Ip Webcam can be used , even a smart phone will do when an app named Ip Webcam is installed , which is free of cost. Using smtp protocol to send mails . Using concepts of networking to connect to the ip webcam using Wireless Network written in OpenCV .The whole code is written in Python and is durable and works well in less time complexity

The project can be seen here -

<https://github.com/jack17529/Smart-Doors>

**TABLE OF CONTENTS**

DECLARATION ................................................................................................... ii

CERTIFICATE ..................................................................................................... iii

ACKNOWLEDGEMENTS .................................................................................. iv

ABSTRACT........................................................................................................... v

CHAPTER 1 (INTRODUCTION).............................................................. 1

1.1. ................................................................................................................. 5

1.2. ................................................................................................................. 8

CHAPTER 2 (OTHER MAIN HEADING) ......................................................... 13

3.1..................................................................................................................... 15

3.2..................................................................................................................... 17

3.2.1. ............................................................................................................ 19

3.2.2. ............................................................................................................. 20

3.2.2.1. ................................................................................................... 21

3.2.2.2. .............................................................................................. 22

3.3..................................................................................................................... 23

CHAPTER 4 (OTHER MAIN HEADING) ......................................................... 30

4.1..................................................................................................................... 36

4.2..................................................................................................................... 39

CHAPTER 5 (RESULTS AND DISCUSSIONS)............................................... 40

CHAPTER 6 (CONCLUSIONS AND FUTURE SCOPE) .................................. 49

REFERENCES... .................................................................................................... 49

APPENDIX A ......................................................................................................... 50

APPENDIX B ......................................................................................................... 52

Introduction

What is the last time you opened your door for someone(friend) , when you were actually sleeping , or were busy ?

Yes I know it's painful .

That’s why I made this project.

Problem Statement

How to keep doing your important work or sleeping when someone rings the bell.

State of Art

Automatically sending Emails to friends when you are not in the condition of opening the door.

I am focusing on sending info to friends when I can’t tell them thus automating the task of telling when will I be free.

Proposed Solution

My project solves the problem by sending emails , to the person or a friend standing on the door . It also tell the next time I will be active to talk to him/her.

So that I can keep doin my important work and my friend too leaves stress free cause he/she knows that I am busy doing something or in deep sleep.

Steps in the process

1. Checks day schedule –

Program checks my google calendar using google calendar API.

Checks the event on the current time. And if I am busy or sleeping.

2. Takes a pic –

Takes the picture using OpenCV library using to the webcam connected to the door. Using secure SSL certificate.

3. Does Face Recognition –

This done using Amazon face Rekognition API , which requires use of several libraries like boto3 and auth.

4. Putting in the S3 Database –

The picture gets stored by the name Unknown.jpg in the Target bucket of my S3 account.

5. Running Face Recognition Algorithm –

Does face recognition from the source bucket of my S3 account . In the source bucket the pictures of all the known persons are stored.

Finds the largest image in the Unknown.jpg and does face recognition.

6. Checking Email –

If the face of the person is recognized , then the program checks the name of the person and email written in the database.

7. Sending Email –

If the name of the person along with email is found in the database , then it sends email using the smtp library.

Usage of Hardware and Software

Hardware –

1. Ip webcam or a phone with Ip-webcam app installed.
2. A pc to run the program

Software –

1. Python 2.7 must be installed in the pc to run the program.
2. Libraries that need to be installed –
   1. Numpy
   2. Boto3
   3. Opencv
   4. Google-api-python-client

Results

I have made a software that sends emails to a known person when you are busy or sleeping or have any urgent work. Giving them the next time you can meet.

The software accomplishes its goal to send emails.

Also reducing the tension in stressful situations.

Future Work

1. I am thinking of putting the software on Heroku so that it can be active 24/7 and can be used anytime.
2. I am thinking of solving the problem of the bell ringing in the end.

What if the person does not understand the seriousness of the situation then I need to disable the bell after certain period of time.

Implementing an hardware bell that stops when the email is delivered.

This is the current proposal that was made by me to the problem to disable the disturbance caused.

I need to start the python code when the bell starts and then disable the bell with the same python code when the email is delivered .

Difficulties –

After researching for a week , it seems like this require some serious hardware skills . Even process priority skills which is beyond my report submitted before. Average time 2-3 months.

References

1. PythonRekognitionDemo by [drpventura](https://github.com/drpventura/PythonRekognitionDemo/tree/fda89bcc14c1607f0967c313e573f9f56fdace45)

<https://github.com/drpventura/PythonRekognitionDemo/tree/fda89bcc14c1607f0967c313e573f9f56fdace45>

1. python-aws-s3 by [keithweaver](https://github.com/keithweaver/python-aws-s3)

<https://github.com/keithweaver/python-aws-s3>

1. AWS docs by Amazon

<https://docs.aws.amazon.com/rekognition/latest/dg>

1. Amazon Cli User guide by Amazon

<https://docs.aws.amazon.com/cli/latest/userguide>

Response Time ?

For an uptime check (http check) the response time is calculated as the time it takes to perform a HTTP GET to the specified URL, so the response time is calculated in three parts:

* Time to first byte
* Time to receive headers
* Time to load HTML of the site

Load Time?

what we use to describe how long a specific page took to load in its entirety, this includes all images, scripts, CSS and third party resources (as well as the HTML of course) that might be found on a website.

These reports will combine the load time of each element on the page to give you the total page load time, and this is why the load time of a website often is a lot higher than the response time.

Why measure Response Time ?

Everyone who operates an online business understands the importance of having fast website [response times](https://www.websitepulse.com/kb/response_time). When webpages are simple to perceive, the user will spend more time on your pages, and are much more likely to spend money while they are there.

Fast website response times can also be critical to [influencing buying behavior](http://www.iresearchservices.com/5-common-factors-influencing-consumer-behavior/), as your website performance is often judged as a reflection of the quality and competence of your business skills.

“For example, Amazon calculated that a page load slowdown of just one second could cost it $1.6 billion in sales each year.”

Factors that Affect Response Time ?

**1. COMPLEXITY**

Too often, organizations get wrapped up in adding so much functionality that performance actually suffers. Complexity can be on the client side as well as the application side

### 2. INTERDEPENDENCIES

The top factor impacting website response time is application/infrastructure/endpoint interdependencies. Shifting dynamics across these interdependencies can cause latencies, outages, security breaches and wreak havoc on end user experience.

### 3. CONFIGURATION AND COMMUNICATION OF COMPONENTS

Today's website infrastructure consists of a lot of components. Some of these components aren't even located in the same country. The installation and configuration of these components is the biggest factor of slow website response times.

### 4. LATENCY

A platform approach that unifies monitoring of servers and back-end infrastructure and front-end API and application performance is the key to ensuring speed and responsiveness that meet user expectations.

### 5. DEMAND PEAKS

 When problems rear their ugly head it's typically during peak times. Think Black Friday or Cyber Monday. These may be extreme examples but they illustrate a very good point. Infrastructure must be to be scaled to handle peak rates rather than average rates. Peaks in demand may only last for a short time, sometimes only milliseconds but they have a much longer lasting effect, impacting not only the web server and supporting systems but more importantly user experience.

Industry standard of Response Time ?

Ideally, optimal server response time is about 200ms.

Sub-second (<1000ms) response time is the industry standard, from my experience.     
  
1ms is less than most network latency, outside the local network.  Ping [Google](http://google.com/" \t "_blank)and you will proabably see 4-5ms, but the entire page load of the sparse [Google](http://google.com/) loads in less than 200ms.  However, Google is a bad example because their home page is almost devoid of anything.  
  
A user should see, at the very least, the page structure and layout load with almost all text and some images.  It is acceptable for some images to load after the initial load.  
  
The user should have the perception that something is happening when they take the action of entering a URL or clicking on a link.

Recent Research in this Field –

* **0.1 second** is about the limit for having the user feel that the system is **reacting instantaneously**, meaning that no special feedback is necessary except to display the result.
* **1.0 second** is about the limit for the **user's flow of thought** to stay uninterrupted, even though the user will notice the delay. Normally, no special feedback is necessary during delays of more than 0.1 but less than 1.0 second, but the user does lose the feeling of operating directly on the data.
* **10 seconds** is about the limit for **keeping the user's attention** focused on the dialogue. For longer delays, users will want to perform other tasks while waiting for the computer to finish, so they should be given feedback indicating when the computer expects to be done. Feedback during the delay is especially important if the response time is likely to be highly variable, since users will then not know what to expect.

References - https://stackoverflow.com/questions/164175/what-is-considered-a-good-response-time-for-a-dynamic-personalized-web-applicat