Cosaint

A MINI PROJECT REPORT

Submitted by

THOMAS CHERAIN (0015113148)

Under the guidance of

MS. LINDA JOSEPH

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

HINDUSTAN INSTITUTE OF TECHNOLOGY AND SCIENCE

CHENNAI - 603 103

OCTOBER 2017

Cosaint

A MINI PROJECT REPORT

Submitted by

THOMAS CHERAIN (0015113148)

Under the guidance of

MS. LINDA JOSEPH

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

HINDUSTAN
INSTITUTE OF TECHNOLOGY & SCIENCE

HINDUSTAN INSTITUTE OF TECHNOLOGY AND SCIENCE

CHENNAI - 603 103

APRIL 2017

HINDUSTAN
INSTITUTE OF TECHNOLOGY & SCIENCE

BONAFIDE CERTIFICATE

Certified that this project report **"Cosaint"** is the bonafide work of **"THOMAS CHERIAN (0015113148)"** Who carried out the project work under my supervision during the academic year 2016-2017.

SIGNATURE

SUPERVISOR

MS. LINDA JOSEPH

INTERNAL EXAMINER

Name: _____

Designation: _____

Institution Name: _____

EXTERNAL EXAMINER

Name: _____

Designation: _____

Project Viva - voice conducted on _____

ACKNOWLEDGEMENT

We wish to express our heartfelt gratitude to **Mrs. Rajeswari Mukesh, Head of the Department, School of Computer Science Engineering** for much of her valuable support, encouragement in carrying out with this project.

I am very grateful to my project supervisor MS. LINDA JOSEPH**, Assistant Professor, Department of Computer Science and Engineering** for the guidance, inspiration and constructive suggestions that helpful me in the preparation of this project. I won't forget to also mention class mates for their wonderful and skillful guidance in assisting me with the necessary support to ensure that my project is a success. I also thank my parents and family at large for their moral and financial support in funding the project to ensure successful completion of the project.

ABSTRACT

Cosaint is a tiny helper for people whose life and work revolve around their machines. Having said that, it would be a disaster for them to allow another person or a little kid to accidently work on their machines.

Cosaint (Irish for Protect) tries to help out a little. It has facial recognition built in so that only you can use it, and no one else. It has an age detection system -> connected to Microsoft azure face API using which if the unknown person is a minor then the device will automatically shutdown at that instant without giving the 30 second interval it gives for the adults.

In addition to the above two modules it helps to remind you of basic things like drinking water and going out for a stretch. Based on your environment lightning it also sets the screen's brightness so as to save your eyes from excessive glare from the screen.

TABLE OF CONTENTS

CHAPTER NO. **TITLE** **PAGE NO.**

Contents

LIST OF FIGURES

CHAPTER 1

1. INTRODUCTION

1.1 INTRODUCTION:

Cosaint (Irish for Protect) has facial recognition built in so that only you can use it, and no one else. It has an age detection system -> connected to Microsoft azure face API using which if the unknown person is a minor then the device will automatically shutdown at that instant without giving the 30 second interval it gives for the adults.

In addition to the above two modules it helps to remind you of basic things like drinking water and going out for a stretch. Based on your environment lightning it also sets the screen's brightness so as to save your eyes from excessive glare from the screen.

1.2 ABOUT:

Cosaint as the name suggests is built over my use case scenario and helps out in the following ways.

It has the following functions

1. Face Recognition
2. Smart Brightness setting
3. Automatic lockdown
4. Age detection

5. Alert to go and drink water
6. Alert to go out and do some stretching
7. Periodic display of random comic to lighten the mood

1.3 SUMMARY:

This document contains the design of a module for the Cosaint. This module offers the above functions. It uses high level azure API's and offline DLIB modules for face recognition.

CHAPTER 2

2 PROJECT DESCRIPTION

2.1 INTRODUCTION:

This chapter reviews how the existing system and proposed system works as well as to produce a better alternative for its improvement.

2.2 EXISTING SYSTEM:.

1. Asus Smart logon

2. Lenovo smart logon

3. Other third party login methods

2.2.1 SHORT COMES OF EXISTING SYSTEM:

Are not very accurate.

Can be fooled using a photo

Cannot detect while the computer is open

Cannot detect Age and auto shutdown

2.2.2 PROBLEMS OF THE EXISTING SYSTEM:

- Proprietary software
- Can work on a few machines only
- Other third party options are not so good
- None of them have a age recognition feature.

2. 3 PROPOSED SYSTEM:

- Automates all the above problems and does a great job at it.

2.3.1 BENEFITS OF PROPOSED SYSTEM

- One shot learning

- Better detection
- Instant shut off

2.4 SUMMARY:

In this chapter we have discussed about short coming of the existing system, problems of the existing system .so that we can overcome with the new proposed system. We have also discussed about advantages of proposed system.

CHAPTER 3

3 SYSTEM DESIGN

3.1 INTRODUCTION:

System design is the specification or construction of a technical, computer-based solution for the business requirements identified in system analysis. It gives the overall plan or model of a system consisting of all specification that give the system its form and structure (i.e. the structural implementation of the system analysis).

3.2 OBJECTIVES OF THE DESIGN:

The objective of designing the new system is to provide an automated response to all the problems mentioned in the previous chapter.

3.2.1 FACTORS CONSIDERED IN THE DESIGN:

The following are the factors put into consideration during the design of the new system;

To design

- Should not slow the computer
- Should work at all times
- A time effective system.

3.3 OUTPUT DESIGN:

The output design contains all the necessary details of the processed input by the system which is drawn from different data inputs to the machine learning module and the other input it gets during use of the laptop.

3.5 PROGRAM DESIGN:

This is where the programs that will run the modules identified in the control center are specified. This will enable the researcher to capture the complete working picture of the application, how each component is related to another, the calling program modules and the ones called. The Cosaint consists of various program modules. Each module of the system is a complete module and part of the entire system. Below are the various modules contained in the application.

3.5.1 Main module

The main module is responsible for making the processes and running the other modules efficiently without slowing down my system.

3.5.2 Face Recognition Module:

The first step in our pipeline is face detection. Obviously we need to locate the faces in a photograph before we can try to tell them apart! If you've used any camera in the last 10 years, you've probably seen face detection in action:

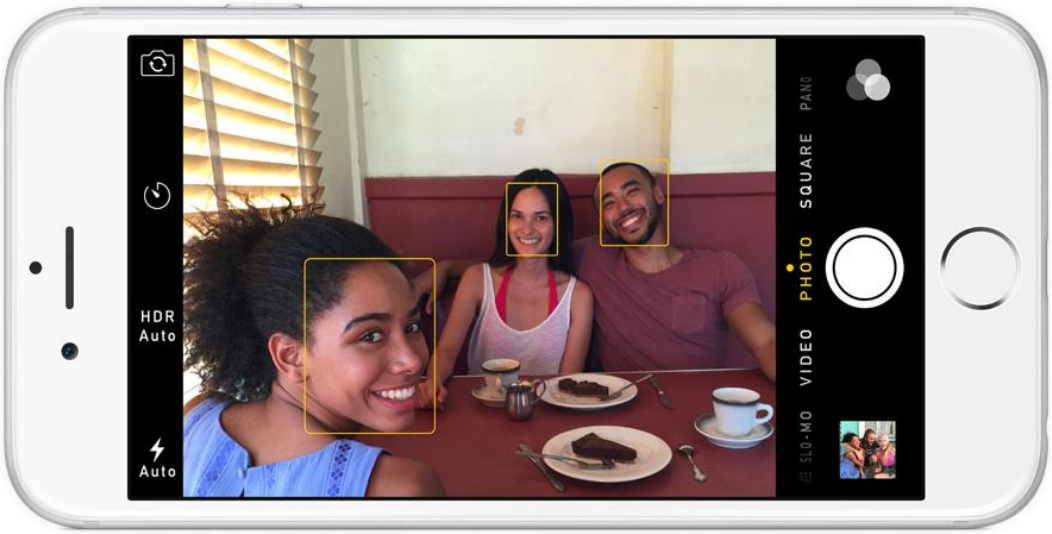


Fig 3.5.2: Face Detection

To find faces in an image, we'll start by making our image black and white because we don't need color data to find faces:

Fig 3.5.3: Will Farrell

Then we'll look at every single pixel in our image one at a time. For every single pixel, we want to look at the pixels that directly surrounding it:

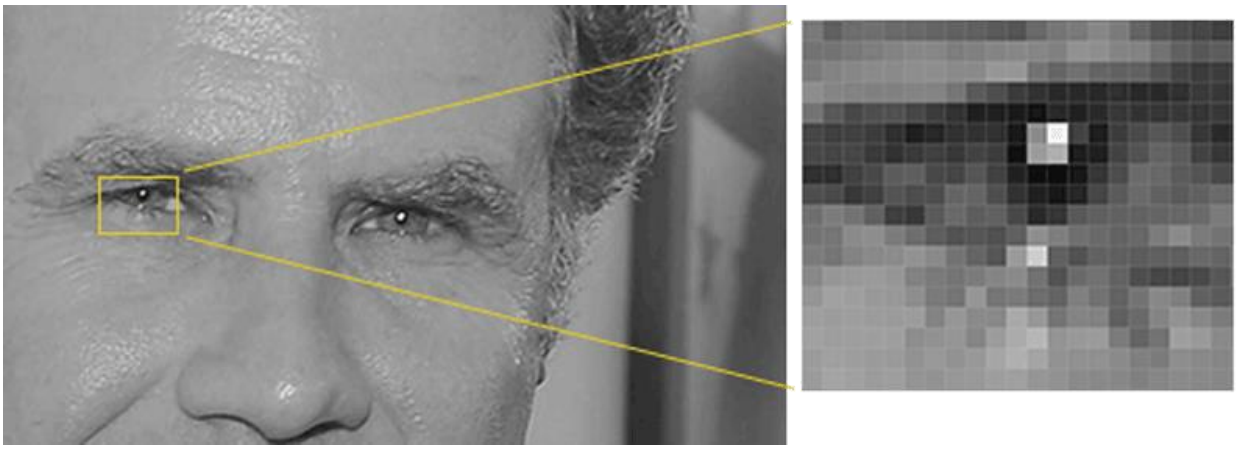


Fig 3.5.4: Eye Detection

Our goal is to figure out how dark the current pixel is compared to the pixels directly surrounding it. Then we want to draw an arrow showing in which direction the image is getting darker: Looking at just this one pixel and the pixels touching it, the image is getting darker towards the upper right. If you repeat that process for every single pixel in the image, you end up with every pixel being replaced by an arrow. These arrows are called gradients and they show the flow from light to dark across the entire image.

To find faces in this HOG image, all we have to do is find the part of our image that looks the most similar to a known HOG pattern that was extracted from a bunch of other training faces:

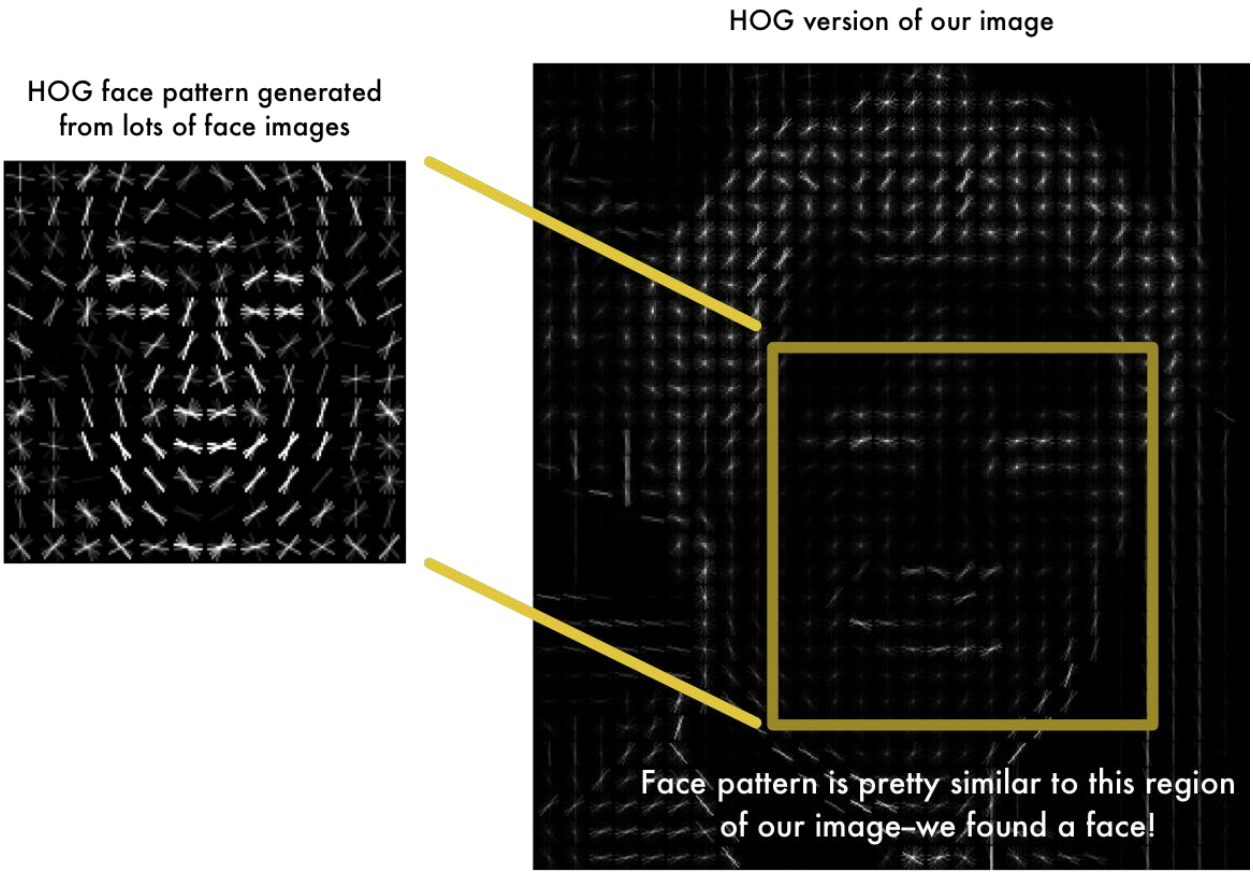


Fig 3.5.5: Face Pattern Detection

Using this technique, we can now easily find faces in any image:

Fig 3.5.6: Face Pattern Detected

Encoding our face image

This process of training a convolutional neural network to output face embeddings requires a lot of data and computer power. Even with an expensive NVidia Telsa video card, it takes about 24 hours of continuous training to get good accuracy. But once the network has been trained, it can generate measurements for any face, even ones it has never seen before! So this step only needs to be done once. Lucky for us, the f ine folks at OpenFace already did this and they published several trained networks which we can directly use. Thanks Brandon Amos and team! So all we need to do ourselves is run our face images through their pre-trained network to get the 128 measurements for each face. Here's the measurements for our test image:

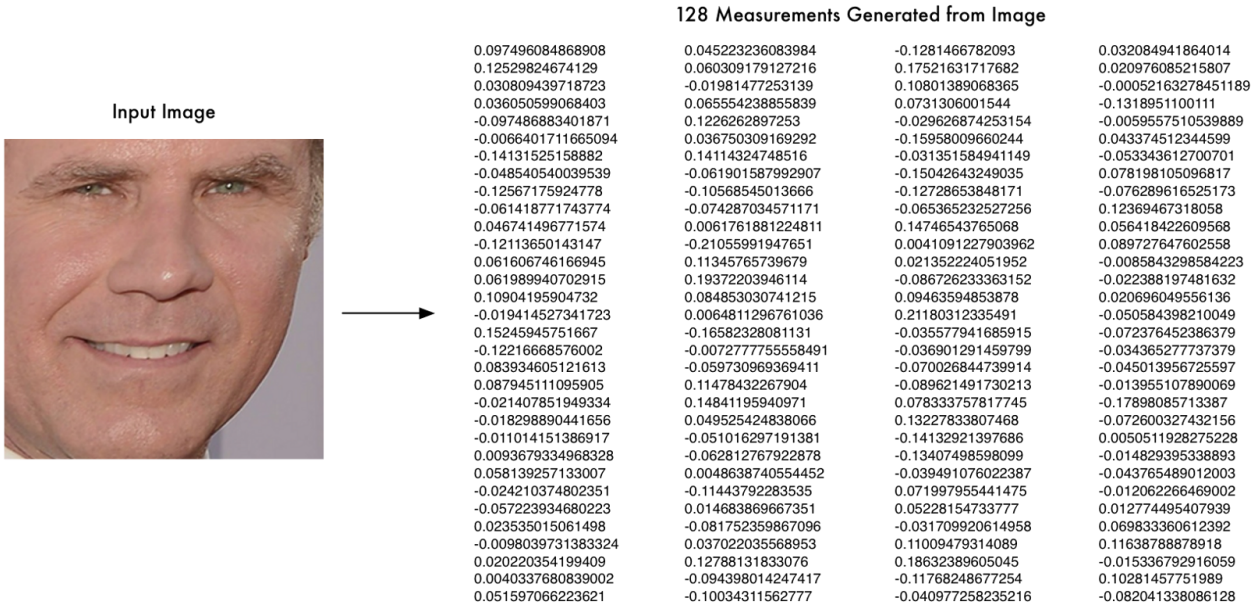


Fig 3.5.7: Face Encoding

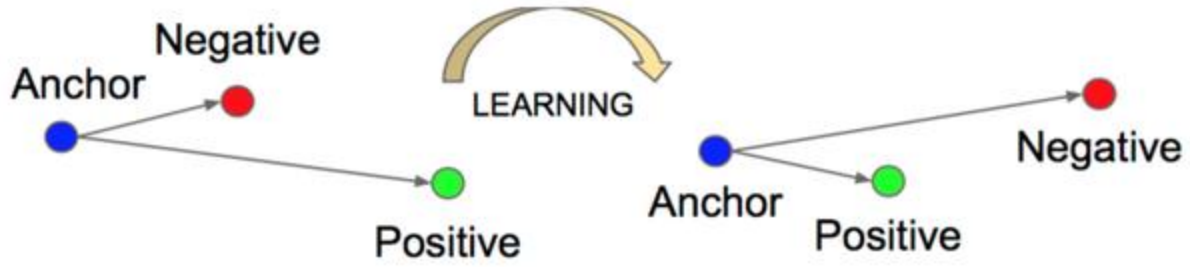
Face Recognition Module:

In 2015, researchers from Google released a paper, FaceNet, which uses a convolutional neural network relying on the image pixels as the features, rather than extracting them manually. It achieved a new record accuracy of 99.63% on the LFW_dataset.

FaceNet: In the FaceNet paper, a convolutional neural network architecture is proposed. For a loss function, FaceNet uses "triplet loss". Triplet loss relies on minimizing the distance from positive examples, while maximizing the distance from negative examples.

$$\sum_{i}^{N} \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+$$

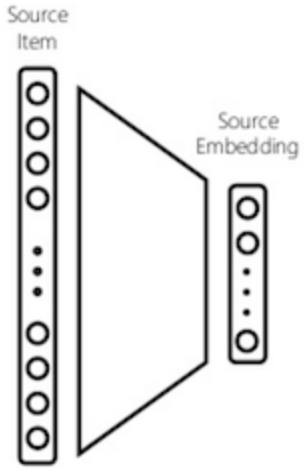
Triplet loss equation



Triplet loss Learning

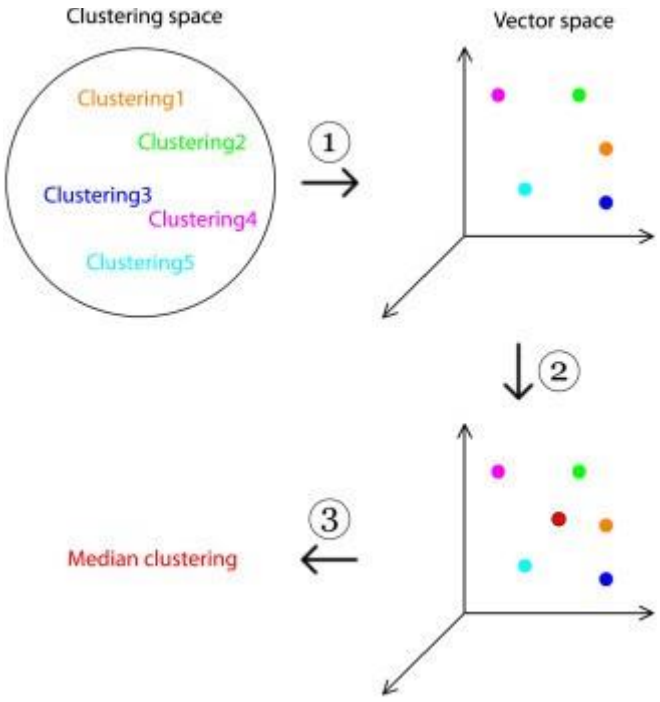
Conceptually, this makes sense. Faces of the same identity should appear closer to each other than faces of another identity.

Vector Embeddings: For this tutorial, the important take away from the paper is the idea of representing a face as a 128-dimensional embedding. An embedding is the collective name for mapping input features to vectors. In a facial recognition system, these inputs are images containing a subject's face, mapped to a numerical vector representation.



Mapping input to embedding

Since these vector embeddings are represented in shared vector space, vector distance can be used to calculate the similarity between two vectors. In a facial recognition context, this can vector distance be applied to calculate how similar two faces are. Additionally, these embeddings can be used as feature inputs into a classification, clustering, or regression task.



3.5.2 Brightness Module:

To set the brightness of the screen, you must first find what the brightness of the room is. To do that, you must first get an image of the room. Then follow the following steps

1. Convert image into array. Done by converting the image from an object and converting it into numpy array.
2. To understand this further let us first understand what each pixel is. A pixel is nothing but a tuple of three values- Red, Green and Blue.

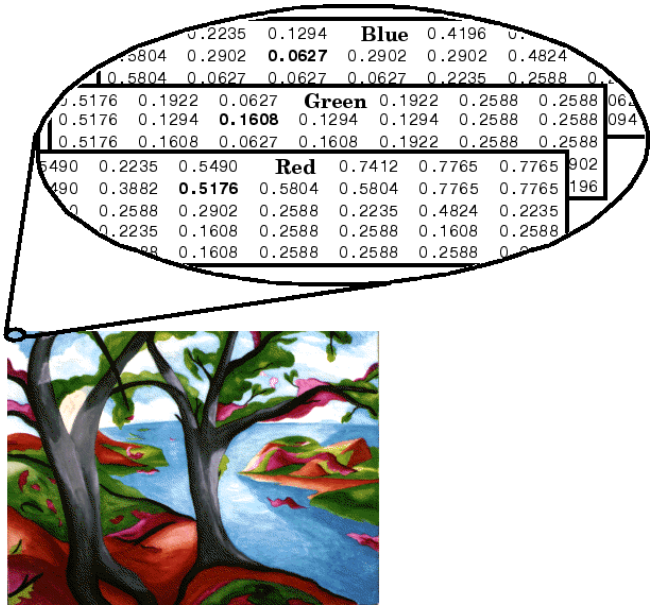


Fig 3.5.8: Pixel Encoding

3. Now we calculate a mean of these values to get the mean brightness in the entire picture of the room
4. Now using the formula

   value = int(10 / 13) * value

   we calculate the next brightness value for the screen.

3.5.3 Age Module:

The age module goes into Microsoft azure and uses the face API to calculate and get the age.

This happens using the following steps

1. Gets an unidentified face
2. Sends face to azure
3. Extracts age from the json response
4. Sends back the age

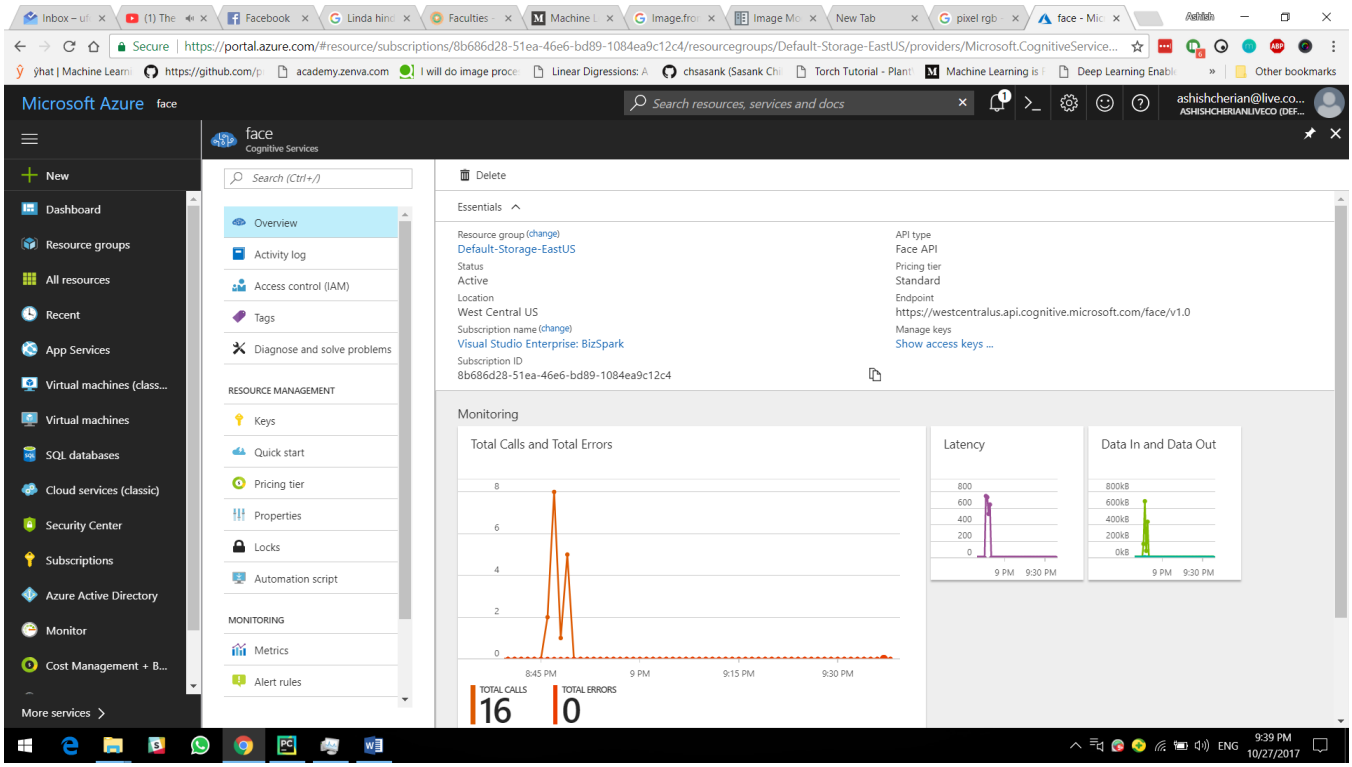


Fig 3.5.9: Azure Face API

3.5.4 Comics Module:

The comics module does the following tasks

1. Queries the webpage for the latest comic number
2. Generates a random comic number from 1 – latest

3. Gets the comic from the webpage
4. Saves the comic as jpg
5. Open the comic to show

Fig 3.5.10: Random comic

3.5.5 Get out Module:

Just alerts me every 2 hours to go out of my room and stretch and also drink water.

3.7 SUMMARY:

In this chapter we have discussed about the how the image processing, face detection, face recognition and other modules using in Cosaint.

CHAPTER 4

4. TESTING

4.1 INTRODUCTION:

Firstly we will start testing with individual module and will perform unit testing on that. So we successfully verify and validate all modules by successively integrating each module. Moreover checking the work done was very important to reduce risk factor. Checking was being ultimately handled by testing but interim checking was required.

4.2TESTING METHODS:

4.2.1 Testing your face recognition:

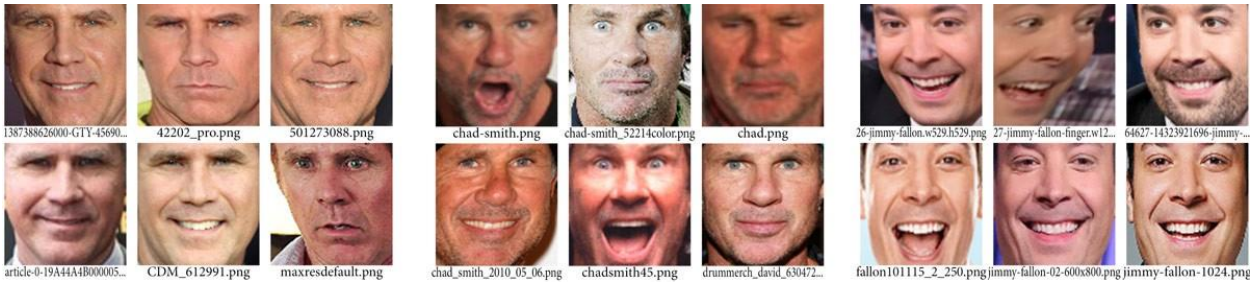


Fig 3.5.11: Labeled faces

Have a dataset filled with labelled faces to carry out the testing.

We will use the Yale Face Database that contains 165 grayscale images of 15 individuals in gif format, There are 11 images for each individual. In each image, the individual has a different facial expression like happy, sad, normal, surprised, sleepy etc. Indeed, there are 166 images with 12 images for the first individual. We will use this database by using 10 images of the total 11 images of each individual in training our face recognizer and the remaining single image of each individual to test our face recognition algorithm.

The images corresponding to each individual are named like `subject<number>.<facial_expression>` where `number` ranges from 01, 02,

03…, 14, 15 and `facial_expression` is the expression that the individual has in the image.

```
bikz05@jaswinder:~/Desktop/FaceRecognition$ facerecognizer.py
init done
opengl support available
15 is Correctly Recognized with confidence 25.2105434172
5 is Correctly Recognized with confidence 34.3565508048
9 is Correctly Recognized with confidence 46.2072675938
13 is Correctly Recognized with confidence 32.6527994538
8 is Correctly Recognized with confidence 99.8276423452
7 is Correctly Recognized with confidence 44.3297146637
11 is Correctly Recognized with confidence 39.0980390296
6 is Correctly Recognized with confidence 25.8154211953
14 is Correctly Recognized with confidence 25.5569113665
10 is Correctly Recognized with confidence 19.4849737279
4 is Correctly Recognized with confidence 0.0
2 is Correctly Recognized with confidence 29.15139655
1 is Correctly Recognized with confidence 37.6819645242
3 is Correctly Recognized with confidence 32.3938790423
12 is Correctly Recognized with confidence 30.7063923459
bikz05@jaswinder:~/Desktop/FaceRecognition$
```

Fig 3.5.12: Labeled faces Prediction

Now we just test each of the faces and see if you get a correct label.

4.3 SUMMARY:

In this chapter we have discussed about testing and also importance of testing and also we have discussed about testing method. Mainly these testing methods are useful to reduce the risk factor.by using these testing methods we can easily know what are all the risks are involved in the project or software.

5 SOFTWARE REQUIREMENT ANALYSIS

5.1 INTRODUCTION:

Currently since the work load is less, the hardware and the software requirements for the script is less. As it increases in both numbers and also in data, the hardware as well as the software requirements might increase.

5.2 PROBLEM DEFINITION:

The system's rule-based approach is simple enough to enable users to write their own rules, yet it is powerful enough to catch many typical errors. Most rules are expressed in a simple XML format which not only describes the errors but also contains a helpful error message and example sentences. Errors which are too complicated to be expressed by rules in the XML file can be detected by rules written in Python. These rules can also easily be added and do not require any modification of the existing source code.

An error corpus will be assembled which will be used to test the software with real errors. The errors will be collected mostly from mailing lists and websites. The errors will be categorized and formatted as XML. Compared to the previous version, many new rules will be added which detect typical errors found in the error corpus.

5.3 FEASIBILITY ANALYSIS:

A feasibility study was an evaluation of a proposal designed to determine the difficulty in carrying out a designated task. Generally, a feasibility study precedes technical development and project implementation.

5.3.1 ECONOMIC FEASIBILITY:

To develop the proposed system, it needs no extra facilities and devices. All dependencies are satisfied from the open source modules that includes language tools and python. All tools used are free, open source and the programming language is python and hence its development is economical.

5.3.2 TECHNICAL FEASIBILITY:

Proposed system is technically feasible because the proposed system requires only those H/Wand S/W tools that are available in the system. More over expandability will be maintained in the new system. New modules can be added later on the application, if required in the future. Additionally the application will have user-friendly forms and &screens.

5.3.3 BEHAVIORAL FEASIBILITY:

Behavioral feasibility determines how much effort will go in the proposed information system, and in educating and training the users on the new system. Since the user interface is very simple and easily understandable, no training is required for using this software.

5.4 METHODOLOGY:

The establishment and use of sound engineering principles in order to obtain economically developed software that is reliable and works efficiently on real machines is called software engineering.

Software engineering is the discipline whose aim is:

➢Production of quality software

➢Software that is delivered on time

➢Cost within the budget

➢Satisfies all requirement's

Software process is the way in which we produce the software. Apart from hiring smart, knowledgeable engineers and buying the latest development tools, effective software development process is also needed, so that engineers can systematically use the best technical and managerial practices to successfully complete their projects.

A software life cycle is the series of identifiable stages that a software product undergoes during its lifetime .A software lifecycle model is a descriptive and diagrammatic representation of the software life cycle .A life cycle model represents all the activities required to make a software product transit through its lifecycle phases .It also captures the order in which these activities are to be taken.

5.5 LIFECYCLE MODELS

There are various life cycle models to improve the software processes.

➔WATERFALL MODEL

➔PROTOTYPE MODEL

➔ITERATIVE ENHANCEMENT MODEL

➔EVOLUTIONARY MODEL

➔SPIRAL MODEL

5.6 SOFTWARE AND HARDWARE REQUIREMENTS:

Platform forms the foundation on which the architecture, design, and implementation of a product is built. &system specification defines the full functionality of the system. In many systems we work on, some functionality performed in hardware and some in software. System specification documents can thus be defined as the requirement's documentation that formally specifies the system level requirements of an application. This application developed in windows platform.

Hardware Requirement:

Processor	:	Intel Core Duo 2.0 GHz or more
RAM	:	2 GB or More
Hard disk	:	1GB or more
Monitor	:	not required, can run on a server
Keyboard	:	Normal or Multimedia
Mouse	:	Compatible mouse
Webcam	:	Any USB compatible camera
Network	:	Proper net connection to work with API's

Software Requirement:

Front End	:	None
Back End	:	python server
Operation System	:	Windows/ MAC/ Linux

5.7 SYSTEM ANALYSIS:

System analysis, sometimes called requirement analysis, is the process of gathering information about the current system, identifying its strengths and problems, and analyzing them to produce a

concept for the new system. The goal of this analysis stage is to truly understand the requirements for the new system and develop a system concept that addresses them. In the current case, there is a current system that used globally. The analysis that will be done would be on how a current system works and the new requirement to improve the timetable system.

5.8 SUMMARY:

In this chapter we have discussed about software and hardware requirements that are used in the process of a project and also we have discussed about feasibility analysis. These feasibility analysis mainly describes the difficulty in carrying out a designated task and also we have discussed about the methodology which mainly describes the Production of quality software, Software that is delivered on time ,Cost within the budget and Satisfies all requirement's ,and also we have named various life cycle models.

CHAPTER 6

6. CONCLUSION

Thus the bot made using python and several other API's is deployed successfully and functions without a hitch. All modules running properly. The grammar module has a lot of scope for improvement.

6.1 FUTURE ENHANCEMENT

- Better use of multithreading
- Better use of dataset
- Better learning model
- Faster recognition
- Faster training
- Offline age detection

Hence my aim would be to improve upon the face recognition module and implement better deep learning models to make sure the prediction is faster and better.

CHAPTER 7

7. RFERENCES

[1] Python Image Module,

http://pillow.readthedocs.io/en/3.1.x/reference/Image.html

[2] Face recognition using open CV

http://hanzratech.in/2015/02/03/face-recognition-using-opencv.html

[4] Face Recognition with deep learning

https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78

[5] yale face dataset:

https://azure.microsoft.com/en-in/services/cognitive-services/face/

[6] Dlib

http://dlib.net/

[7] Dlib Face detection

http://dlib.net/face_detector.py.html

APPENDIX

Source Code

Main Activity Code

```
import apscheduler
from apscheduler.schedulers.background import BackgroundScheduler
from multiprocessing import Process
from run import capture_video
sched = BackgroundScheduler()
sched.start()
from comics import GetComics
from brightness import setBrightness
from bacha import getage
from getOUT import showAlert
from run import brightness

sched.add_job(brightness, 'interval', seconds=30)
sched.add_job(showAlert, 'interval', seconds=7200)
sched.add_job(GetComics, 'interval', seconds=10)

if __name__ == '__main__':
    p = Process(capture_video())
    p.start()
```

Comics.py

```
import urllib.request, random
from PIL import Image
import io
import re
from bs4 import BeautifulSoup


def f(n):
    try:
        page = 'http://xkcd.com/' + n + '/'
        response = urllib.request.urlopen(page)
        text = str(response.read())
        # Now finding the link of the comic on the page
        ls = text.find('embedding')
        le = text.find('<div id="transcript"')
        link = text[ls + 12:le - 2]  # + ".jpg"
        # Now finding the title of the comic
        ts = text.find('ctitle')
        te = text.find('<ul class="comicNav"')
        title = text[ts + 8:te - 8]
        img = title + '.jpg'
        # Now downloading the image
        # print('Now downloading - ' + img)
        print(link, img)
        # urllib.request.urlretrieve(link, img)
        fd = urllib.request.urlopen(link)
        image_file = io.BytesIO(fd.read())
        im = Image.open(image_file)
        im.show()
    except:
```

```
        pass


def getNum(link):
    num = ""
    for c in link:
        if c.isdigit():
            num = num + str(c)
    return num


def latest():
    try:
        comic = urllib.request.urlopen("http://xkcd.com")
        # content = comic.text
        content = str(BeautifulSoup(comic.read().decode('utf-8', 'ignore'), "lxml"))
        # Now finding the latest comic number
        ns = content.find('this comic:')
        newstring = content[ns: ns + 40]
        link = re.findall(r'(https?://[^\s]+)', newstring)[0]
        latest = int(getNum(link))
        return int(latest)
    except Exception as e:
        print(e)
        print('Try again later')
        exit()
        return 0


def GetComics():
    val = str(random.randint(1, latest()))
    f(val)
```

Brightness.py

```
import wmi


def setBrightness(value):
    value = int(10 / 13) * value
    wmi.WMI(namespace='wmi').WmiMonitorBrightnessMethods()[0].WmiSetBrightness(value,
0)
```

Run.py

```
from comics import GetComics
from brightness import setBrightness
from bacha import getage
from getOUT import showAlert
import face_recognition
import cv2
from apscheduler.schedulers.background import BackgroundScheduler
import ctypes
from datetime import datetime
from PIL import Image, ImageStat
import wmi
value = 0

def brightness():
    setBrightness(value)
```

```
def whatage():
    try:
        if getage() < 15:
            print("BACHA SPOTTED: ")
            startTime = 0
            ctypes.windll.user32.LockWorkStation()
    except:
        print("Bag image caught")

# def dostuff():
def capture_video():
    video_capture = cv2.VideoCapture(0)
    user_image = face_recognition.load_image_file("training/ashish1.jpg")
    bachaImage = face_recognition.load_image_file("training/pp.jpg")
    user_face_encoding = face_recognition.face_encodings(user_image)[0]
    bacha_face_encoding = face_recognition.face_encodings(bachaImage)[0]
    known_faces = [
        user_face_encoding,
        bacha_face_encoding
    ]
    face_locations = []
    face_encodings = []
    face_names = []
    process_this_frame = True
    flag = False
    startTime = 0
    while True:
        # Grab a single frame of video
        ret, frame = video_capture.read()
        cv2_im = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        pil_im = Image.fromarray(cv2_im)
        stat = ImageStat.Stat(pil_im)
        value = int(stat.mean[0])
        # Resize frame of video to 1/4 size for faster face recognition processing
        small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)
        # Only process every other frame of video to save time
        if process_this_frame:
            # Find all the faces and face encodings in the current frame of video
            face_locations = face_recognition.face_locations(small_frame)
            # print(face_locations)
            face_encodings = face_recognition.face_encodings(small_frame,
face_locations)

            face_names = []
            for face_encoding in face_encodings:
                # See if the face is a match for the known face(s)
                match = face_recognition.compare_faces(known_faces, face_encoding)
                name = "Unknown"

                if match[0]:
                    name = "Ashish"
                    flag = False
                    startTime = datetime.now()
                elif match[1]:
                    print("BACHA DETECTED BYE BYE")
                    # LOCK Computer
                else:
                    flag = True
                    cv2.imwrite("bacha.jpg", frame)
                    whatage()
                    # p.daemon = True
                face_names.append(name)
```

```
        process_this_frame = not process_this_frame
        # Display the results
        for (top, right, bottom, left), name in zip(face_locations, face_names):
            # Scale back up face locations since the frame we detected in was scaled
to 1/4 size
            top *= 4
            right *= 4
            bottom *= 4
            left *= 4
            # Draw a box around the face
            cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)
            # Draw a label with a name below the face
            cv2.rectangle(frame, (left, bottom - 35), (right, bottom), (0, 0, 255),
cv2.FILLED)
            font = cv2.FONT_HERSHEY_DUPLEX
            cv2.putText(frame, name, (left + 6, bottom - 6), font, 1.0, (255, 255,
255), 1)

        # Display the resulting image
        cv2.imshow('Video', frame)
        if flag and type(startTime) != 0:
            timenow = datetime.now()
            seconds = (timenow - startTime).total_seconds()
            if seconds > 30:
                flag = False
                ctypes.windll.user32.LockWorkStation()

        # Hit 'q' on the keyboard to quit!
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    # Release handle to the webcam
    video_capture.release()
    cv2.destroyAllWindows()
```

bacha.py

```
import http.client, urllib.request, urllib.parse, urllib.error, base64, requests, json

def getage():
    subscription_key = '2c2c93c599a841e5ac01a2e0635fe3be'
    uri_base = 'https://westcentralus.API.cognitive.microsoft.com'
    # Request headers.
    headers = {
        'Content-Type': 'application/octet-stream',
        'Ocp-APIm-Subscription-Key': subscription_key,
    }
    # Request parameters.
    params = {
        'returnFaceId': 'true',
        'returnFaceLandmarks': 'false',
        'returnFaceAttributes':
'age,gender,headPose,smile,facialHair,glasses,emotion,hair,makeup,occlusion,accessorie
s,blur,exposure,noise',
    }
    filename = 'bacha.jpg'
    f = open(filename, "rb")
    body = f.read()
    f.close()

        # Body. The URL of a JPEG image to analyze.
```

```
    body = body
    try:
        # Execute the REST API call and get the response.
        response = requests.request('POST', uri_base + '/face/v1.0/detect', data=body,
headers=headers,
                                    params=params)
        parsed = json.loads(response.text)
        print(parsed)
        age = parsed[0]['faceAttributes']["age"]
        return(age)

    except Exception as e:
        print('Error:')
        print(e)
```

getOUT.py

```
import ctypes
def showAlert():
    ctypes.windll.user32.MessageBoxW(0, "Time to skeedadle", "Go human GO!!", 1)
```