Abstract

A facial recognition system is a technology capable of identifying or verifying a person from a digital image or a video frame from a video source. There are multiples methods in which facial recognition systems work, but in general, they work by comparing selected facial features from given image with faces within a database. It is typically used in security systems and has also become popular as a commercial identification and marketing tool. Face recognition has been getting pretty good at full frontal faces and 20 degrees off, but as soon as you go towards profile, there've been problems. Besides the pose variations, low-resolution face images are also very hard to recognize. This is one of the main obstacles of face recognition in surveillance systems. Face recognition is less effective if facial expressions vary. The key challenge of face recognition is to develop effective feature representations for reducing intra-personal variations while enlarging inter-personal differences. The project proposes to bring out the comparison between Face Recognition Service offered by two of the most leading and well known cloud computing service providers of all time - Amazon Web Services and Microsoft Azure. This project deals with Face Rekognition of AWS and Face API of Azure. Python code on Linux platform is chosen as the method of implementation. Data Set consists of black and white face images of people taken with varying pose (straight, left, right, up), expression (neutral, happy, sad, angry) and eyes (wearing sunglasses or not). A quantitative and qualitative analysis is done for the same and a set of comparison is made between the two the services to help users make the right choice.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

AWS	Amazon Web Services
API	Application Program Interface
IT	Information Technology
JPG	Joint Photographic Experts Group
PNG	Portable Network Graphics
IaaS	Infrastructure as a Service
SaaS	Software as a Service
PaaS	Platform as a Service
IDE	Integrated Development Environment
LDA	Linear Discriminant Analysis
FERET	Face Recognition Technology
PCA	Principle Component Analysis
LBP	Local Binary Pattern
SIFT	Scale Invariant Feature Transform
FRT	Facial Recognition Technology

CHAPTER 1

INTRODUCTION

1.1 Background

Cloud Computing

Cloud Computing is the concept of delivery of computing services like servers, storage, databases, networking, software, analytics and more over the Internet. Companies offering these computing services are called cloud providers and typically charge for cloud computing services based on usage. There are three types of cloud services: IaaS, PaaS and SaaS. SaaS (Software as a Service) is a method for delivering software applications over the Internet, on demand and typically on a subscription basis. Cloud computing is the most prominent technology that drives the industry now-a-days. Cloud computing is the back bone of most of the technically advanced organizations because it provides ample services to its users. Cloud environment can provide any type of services, depending upon the requirement. It is an on-demand self-service that allows a variety of users to access a wide range of service. It is a pay-as-you-go type of a technology that charges users according to the amount of service consumed. Cloud computing has taken a firm place in this short span of time because of its scalability, that is, it can easily add up or shrink down the services depending upon the customer necessities.

The cloud service consumers or the clients may be heterogeneous, in the sense; they can be thick or thin clients. All that they want is service, from the cloud. The service may be usage of resources for a particular time, for which they will be metered, monitored and charged for. These resources are a pool of physical and virtual resources which are available anywhere, anytime on cloud.

System can access any virtual machine, which is randomly allotted by the cloud, and access service.

These are the characteristics of cloud can be briefly described as follows:

On demand self-service: The cloud provides features to users according to their demands

Network access: The cloud resources can be accessed through internet instantly. All the resources that an organization require can be obtained from the cloud via internet connection in a pay per use basis

Resource pooling: The cloud provides an efficient method of distributing resources to its multiple clients. Hence each client can use a part of the huge resources available in the cloud.

Flexibility: The cloud provides flexibility according to the users preferences. Users can switch their data between private, public and hybrid cloud (which will be dealt shortly).

Measured service: Services are provided based upon the payment.

TYPES OF CLOUD COMPUTING

The major problem faced by the organization is the amount of data increases day by day at a higher proportion. Hence with increasing number of users, the time and costs to handle the number of user is also increasing. Hence cloud provides the solution to this problem. There are few organizations which provide all the cloud computing services to the users and also perform the resource sharing among the users. There are three types of cloud environment.

Private Cloud

Private cloud is dedicated to one single organization and it's highly secure. It can be handled by the same organization or other 3rd party organization. As it is solely dedicated to one single organization, the organization would have complete control over it. Hence accessibility of resources is well achieved. Privacy is ensured for the organization which is the most important aspect for any organization. The organization can switch the less sensitive information to public cloud.

Public Cloud

A public cloud consists of pool of resources which is maintained by a third party organization. It provides efficient resource sharing among the clients. The public cloud provides high level of scalability where the companies can pay for the resource they use. It is like a centralized system that efficiently provides resource to the clients and hence organization need not have a separate resources and work forces. They can use the resource from the cloud which reduces costs. There is also less wastage of resources because organization can use resources by subscribing them and release them if they are no more in need and thereby the wastage of resources is avoided.

Hybrid Cloud

Hybrid cloud is an integration of both private and public cloud. Individual organizations which provide cloud computing services can join together and can provide an integrated service. Even individual organizations provide hybrid cloud services. The hybrid cloud keeps specific resources as public and certain resources as private. Hence organization can switch sensitive information to private and less sensitive information to public cloud. It provides scalability to some extent because public resources must be shared. It is more cost effective as compared to private cloud because in private cloud, all the data irrespective of it sensitiveness, as kept private and it is not shared, whereas in hybrid we can switch less sensitive information to public cloud. It is more flexible and also secure

ARCHITECTURE ELEMENTS OF CLOUD COMPUTING

Cloud providers deliver services in 3 different ways. All these methods are used to run applications over the internet and to store data online. These methods provide different level of flexibility and control over the data and resources available in the cloud. The user can choose the best among these methods which would satisfy their business requirements. There are 3 architectural elements of cloud:

i. Software as a service
ii. Infrastructure as a service
iii. Platform as a service

Software as a Service (SaaS)

Software as a service provides application as a service to clients. It serves multiple clients. It makes one software accessible by multiple users. This can be made available in pay per use basis or free of costs by getting paid from advertisements etc. This is very efficient method of accessing the software where the software can be accessed through the internet. The SaaS serves as a centralized system from which the software can be accessed by many clients. Organizations do have a complicated software stacks. Hence if the software is upgraded, the whole software stack is likely to fall. Upgrading software is an essential part in an IT company which is inevitable. Now, with the help of SaaS, there is no need to upgrade the software or to employ technicians to maintain them. Applications which require internet usage can make a best use of SaaS. It uses web to deliver its software resources. It doesn’t require any new software to be installed. It can be directly used from the browser by the client.

Infrastructure as Service (IaaS)

IaaS offers computing platform for the client who access the cloud. It allows the client to store data on the cloud. To store the large amount of data, organization must require more servers and hardware resources. With the high rate of increase in data, the organizations have to concentrate lot on the scalability issues in hardware also, which is time-consuming and very expensive. Hence, IaaS provide a solution for this by allowing the companies to store their data in cloud and access through the internet. And, as the data increases, the companies can subscribe more and store their data. Therefore organizations need to concentrate on their business logic and not on the increasing hardware resources to handle the bulk of data.

Platform as a Service (PaaS)

Cloud computing system deals with a colossal amount of data; so we need to set a platform for those data. Platform as a service provides a scaffold for the developers to refine their application and helps to use the tools provided by the provider. PaaS sits between Infrastructure as a Service (IaaS) and Software as a Service (SaaS). Integrated Development Environment (IDE) along with data security, application Hosting, Scalable Infrastructure, backup and recovery to the Business Applications are being provided by PaaS.

In a project that is being developed from different parts of the world, the team members will be able work together from anywhere to build the same application. Since we can get all the components over the internet and more than that the developer and the vendor will no longer need to endow for the physical infrastructure because they can deal their business over the internet. In Platform as a Service system, it provides the accession called data-centric approach to bestow the secure rostrum for the data which can extend and adapt as much as possible throughout the Cloud Environment. This accession provides two rostrums called Software as a Service (SaaS) and Infrastructure as a Service (IaaS) for managing the sensitive data in the Cloud Environment. The developers can model the PaaS architecture according to their environment. Some companies like HP, IBM, Engine Yard, Amazon, and Salesforce are created the scattered architecture for the PaaS based on their environment.

Face detection

Face detection is a computer technology being used in a variety of applications that identifies human faces in digital images. Face detection also refers to the psychologic process by which humans locate and attend to faces in a visual scene. Face detection can be regarded as a specific case of object - class detection.

Face Recognition

A face recognition system is a computer application for automatically identifying or verifying a person from a digital image or a video frame. One of the ways to do this is by comparing selected facial feature from the image and a facial database. Face recognition is an effective means of authenticating a person the advantage of this approach is that, it enables us to detect changes in the face patterns of an individual to an appreciable extend the recognition system can tolerate local variations in the face expression of an individual. The characteristic features called "Eigen faces‟ are extracted from the storage images using which the system is trained for subsequent recognition of new images. Face recognition can typically be used for verification or identification. In verification an individual is already enrolled in the reference database.

Face Recognition and Cloud Computing

In a Face Recognition system incorporating cloud services, the FR engine is located in the cloud, not in the local processing unit. This attribute renders the system broadly accessible, in addition to providing the capability for quick and reliable integration with other applications. The cloud incorporation facilitates high scalability in order to ensure that the system can be adapted to a wide user base. Several prominent commercial applications follow the client server model, where the query face is captured by the user and transmitted to the cloud server for conducting authentication with the gallery faces of the FR database located on the cloud.

1.2 Problem Statement

We intend to perform a comparative study on Microsoft Azure and AWS in the area of Face Recognition.

1.3 Purpose of Study

The purpose of this project is to work on the two facial recognition algorithms and make comparisons in respect of their accuracy and time taken. The compared facial recognitions algorithms have been widely utilized by cross-domain applications that range from mainstream commercial software to critical law enforcement applications. The study will give the readers an insight into the working and differences in performance with respect to changes in different features like orientation, expression and presence of sunglasses. The study, when extended to a huge extent, can prove to provide a major guidance in helping readers in making up their mind to choosing the right service for their usage and requirement.

1.4 Specific Objective

In our project, the services that we have considered for comparison are Face Rekognition and Face API, which belong to AWS and Azure respectively. We write the python code in Linux platform and use them as an interface to interact with the services. The output of the code gives us the index to the human subject that the test image is identified as, and the time taken for its execution.

1.5 Scope and Importance of the study

Face recognition is widely used in security, natural user interface, image content analysis and management. The cloud computing system can implement a better security of computational time to detect and recognize the human faces with big database than those of others. Azure and AWS have been competing for the past five years in this area. Hence it is highly necessary that we find the best option possible to have highest possible accuracy at our work.

1.6 Limitations

Our project uses 120 images as the data set that might not be sufficient to accurately bring out the results of the comparison. The project works with single face images and not for multi face images. We confine the image format to jpg alone. The project deals only with orientation, expression and presence of sunglasses. Yet there are many other features like size, illumination and image format that determine our results.

CHAPTER 2

LITERATURE SURVEY

The following section provides a review of the literature related to the comparison of face recognition algorithms. The project involves selection of appropriate dataset with variations in deterministic properties of a face that can affect the performance of a face recognition algorithm. It is crucial to decide which facial features contribute to a good recognition and which ones are no better than added noise. The literature survey is aimed towards that area.

Discriminant analysis for recognition of human face images: The paper makes a study on the linear discriminant analysis (LDA) of different aspects of human faces which will result in objective evaluation of the significance level of visual information in different structural parts of the face. It helps us identify a human subject accurately. A small set of features are obtained from the study which carries the most relevant information based on which classification is done. Probabilistic or evidential approaches to multisource data analysis are used. It can provide excellent classification accuracy for databases of average size with low-dimensional feature vectors. The main challenge in feature extraction is to represent the input data in a low-dimensional feature space, in which points corresponding to different poses of the same subject are close to each other and far from points corresponding to instances of other subjects' faces. However, there is a lot of within-class variation that is due to differing facial expressions, head orientations, lighting conditions, etc., which makes the task more complex. Closely tied to the task of feature extraction is the intelligent and sensible definition of similarity between test and known patterns. The task of finding a relevant distance measure in the selected feature space, and thereby effectively utilizing the embedded information to identify human subjects accurately, is one of the main challenges in face identification. They focus on feature-extraction and face-identification processes. The application of LDA to study the discriminatory power of various facial features in spatial and wavelet domain is presented. Also, an LDA-based feature extraction for face recognition is proposed and tested.

The FERET evaluation methodology for face-recognition algorithms: It says that two of the most critical requirements in support of producing reliable face-recognition systems are a large database of facial images and a testing procedure to evaluate systems. One of the primary objectives of the test was to measure algorithm performance on large databases. To date, 14,126 images from 1199 individuals are included in the FERET database, which is divided into development and sequestered portions. In September 1996, the FERET program administered the third in a series of FERET face-recognition tests. The primary objectives of the third test were to (1) assess the state of the art, (2) identify future areas of research, and (3) measure algorithm performance on large databases. It concluded that algorithm performance is dependent on the gallery and probe set.

Comparison of face Recognition Algorithms on Dummy Faces: In the age of rising crime face recognition is enormously important in the contexts of computer vision, psychology, surveillance, fraud detection, pattern recognition, neural network, content based video processing, etc. Face is a non-intrusive strong biometrics for identification and hence criminals always try to hide their facial organs by different artificial means such as plastic surgery, disguise and dummy. The availability of a comprehensive face database is crucial to test the performance of these face recognition algorithms. However, while existing publicly-available face databases contain face images with a wide variety of poses, illumination, gestures and face occlusions but there is no dummy face database is available in public domain. The contributions of this research paper are: i) Preparation of dummy face database of 110 subjects ii) Comparison of some texture based, feature based and holistic face recognition algorithms on that dummy face database, iii) Critical analysis of these types of algorithms on dummy face database.

This paper presents a methodology for creating dummy faces database preparation and demonstrates the percentage identification accuracy. It does comparison of some texture based, feature based and holistic face recognition algorithms and critical analysis of these types of algorithms on dummy face database of 110 subjects. It tested the accuracy of PCA, LDA, iSVM, LBP and SIFT face recognition algorithms.

It concludes that when we increase the gallery size the identification accuracy of each algorithms increases

This paper highlights the potential and limitations of the technology, noting those tasks for which it seems ready for deployment, those areas where performance obstacles may be overcome by future technological developments and its concern with efficacy extends to ethical considerations. For the development of FRT face image database is needed. Several researchers have developed so many real face databases with a lot of covariates. They have designed and tested many algorithms for recognition and identification of human faces and demonstrated the performance of the algorithms but the performance of face recognition algorithms on dummy and fake faces are not reported in the literature. Since face is non-intrusive physiological biometrics for the verification of identity claim therefore in the age of increasing crime, criminals always pay more attention to hide or tamper their facial organs by using so many artificial techniques such as plastic surgery, disguise, mask and dummy faces. Preliminary researches have also been attempted on plastic surgery and disguised face recognition or identification. For the dummy face datasets, they have evaluated the three types of face recognition algorithms: Holistic Performance Based, Local Feature Based and Texture Based. A series of findings were made. As we compress the images there is loss of some of its important features and therefore in higher level of compression accuracy decreases. When we increase the number of gallery images the algorithms gives the better results.

Face Recognition Using PCA: The purpose of the proposed research work is to develop a computer system that can recognize a person by comparing the characteristics of face to those of known individual. PCA technique is used for feature extraction and data representation. The proposed method have been evaluated using the Microsoft cognitive service API with Azure API of databases that can implement a better security of computational time to detect and recognize the human faces with big database.

The idea of the proposed system is to enable us to detect changes in the face pattern of an individual to an appreciable extent. The recognition system can tolerate local variations in the face expressions of an individual. The Proposed system can tolerate some variation in the new face image. When the new image of a person varies from the image of that person stored in the database, the system will be able to recognize the new face and identify person.

In this the facial feature detection is the process to detect the presence and location of feature like eyebrow, nose, eyes, lips, mouth, emotion, tag, computer vision, face size, pose, name, etc. The face recognition system has to identify the facial features such as happiness, neutral of each of individual’s expression. The most of calculating the ages of the human face is detected with the skin colour. The image processing in face API of face recognition system for uploading and storing the Images with big size database takes place in high computational time.

It concluded that the eyes of person must be open and without glass. If such obstacles are on the face of person, the feature detection is complicated. The main limitation of the available face recognition system is that they only detect upright face looking at the camera.

CHAPTER 3

SYSTEM SPECIFICATIONS

The proposed system requires a PC or laptop with the following specifications.

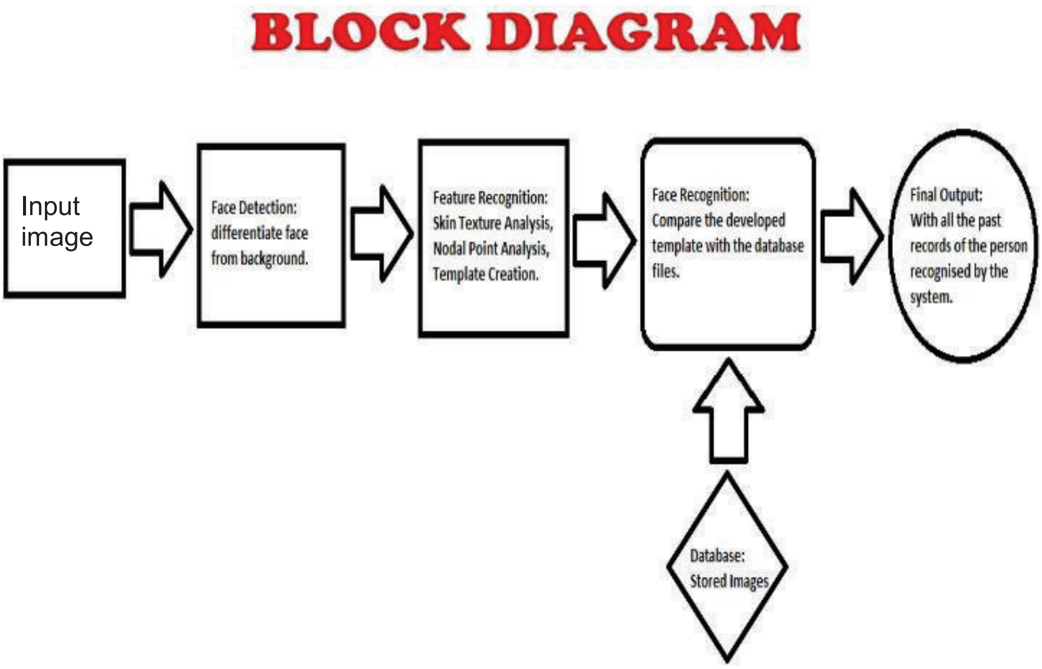
- Ubuntu Operating System
- Packet data facility (minimum of 30 kbps)
- Wi-Fi facility
- Python 2.7
- Cognitive Services API account with Face API.
- AWS account

CHAPTER 4

ARCHITECTURE

4.1. General Architecture:

The general Architecture diagram of “Face Recognition Algorithm” adopted by the work is shown below


BLOCK DIAGRAM
Input image
Face Detection: differentiate face from background.
Feature Recognition: Skin Texture Analysis, Nodal Point Analysis, Template Creation.
Face Recognition: Compare the developed template with the database files.
Final Output: With all the past records of the person recognised by the system.
Database: Stored Images

COMPONENTS OF THE ARCHITECTURE:

1. Input Image: We have to upload the image dataset to the cloud and then train the data into groups.

2. Face Detection: It takes a photo and returns the facial features. To detect faces, all you need to do is submit a JPG or PNG photo. You can submit the photo either as a publicly accessible URL or Base64 encoded. Face detection is the process of automatically locating human faces in visual media (digital images or video). A face that is detected is reported at a position with an associated size and orientation. Once a face is detected, it can be searched for landmarks such as the

eyes and nose. The uploaded image will be cropped into a rectangle box like structure.

- Face rectangle (left, top, width, and height) indicating the face location in the image is returned along with each detected face.
- Optionally, face detection extracts a series of face related attributes such as pose, gender, age, head pose, facial hair, and glasses.
- Landmarks are the facial features like eyes, nose and ears.

3. Face recognition automatically determines if two faces are likely to correspond to the same person. A facial recognition system is a technology capable of identifying or verifying a person from a digital image or a video frame from a video source.

Another emerging trend uses the visual details of the skin, as captured in standard digital or scanned images. This technique, called skin texture analysis, turns the unique lines, patterns, and spots apparent in a person's skin into a mathematical space. Tests have shown that with the addition of skin texture analysis, performance in recognizing faces can increase 20 to 25 percent.

Facial skin exhibits various small scale structures in the surface (wrinkles, scars) and the texture that stands out from normal skin appearance and represents potentially valuable references for individual distinction. Their predictable appearance, also under changing illumination, facilitates detection.

Nodal points are end points used to measure variables of a person's face, such as the length or width of the nose, the depth of the eye sockets and the shape of the cheekbones. These systems work by capturing data for nodal points on a digital image of an individual's face and storing the resulting data.

Face templates have a significant influence on the correct classification rate, while the number of temples is directly proportional to the classification time.

4. A solution may be found in designing an adequate face template database and in applying appropriate face template creation algorithms. The number of face

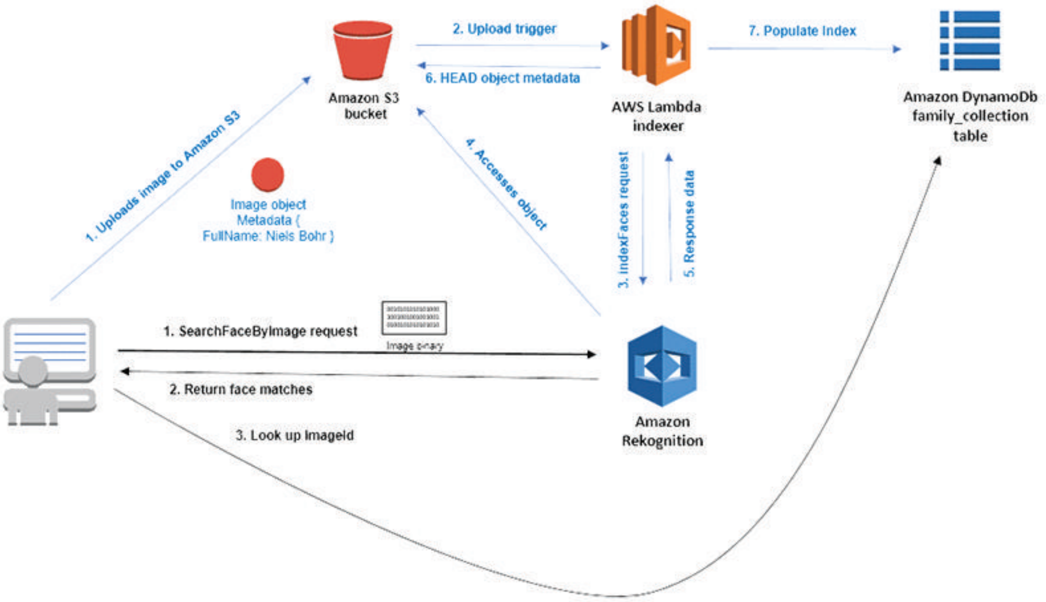
templates influences the classification time; a higher number of face templates requires a longer time for the classification which causes a delay in the identification process. In some applications, e.g. face verification applications or surveillance camera systems, long identification time is not acceptable, which necessitates the reduction of face templates. This urges us to develop methods that produce just one face template per individual.

5. Database stores all the images uploaded by the user.

4.2. Application workflow of Amazon Rekognition

It’s separated into two main parts:

1. Indexing (blue flow) is the process of importing images of faces into the collection for later analysis.
2. Analysis (black flow) is the process of querying the collection of faces for matches within the index.



Steps :

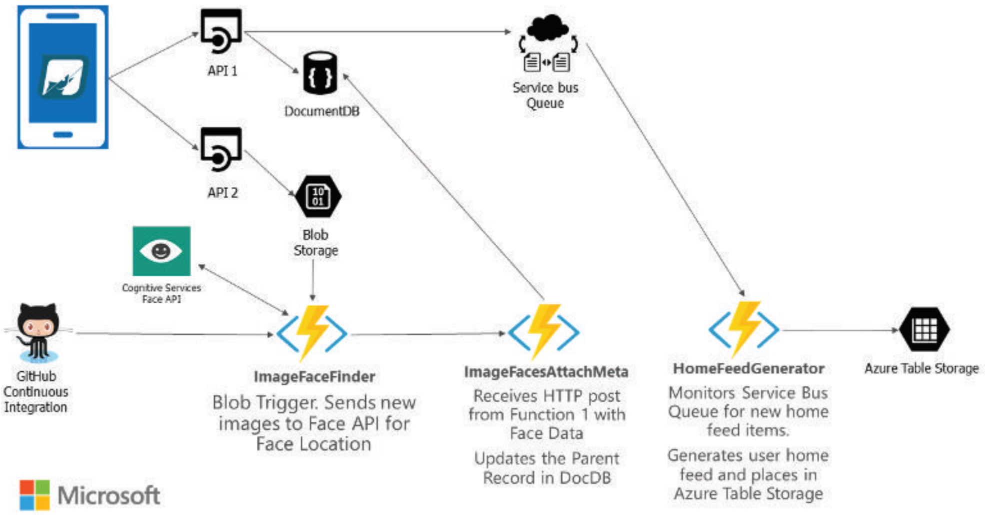
1. AWS uses the Amazon Rekognition IndexFaces API to detect the face in the input image and adds it to the specified collection.
2. If successful, it retrieves the full name of the person from the metadata of the object in Amazon S3.

3. It then stores this as a key-value tuple with the FaceId in the DynamoDB table for later reference.
4. We now need to upload our images to Amazon S3 to seed the face collection. For this, we use a small piece of Python code that iterates through a list of items that contain the file location and the name of the person within the image.

5. We add additional metadata to the objects in Amazon S3. The Lambda function uses this metadata to extract the full name of the person within the image.
6. Depending on our need, we can alter this process to include additional metadata or to load the metadata definition from another source, like a database or a metadata file.
7. Once the collection is populated, we can query it by passing in other images that contain faces. With the use of SearchFacesByImage API we can provide the parameters. Two parameters are mandatory: the name of the collection to query, and the reference to the image to analyze. We can provide a reference to the Amazon S3 bucket name and object key of the image, or provide the image itself as a byte stream.
8. Note: Amazon Rekognition tries to find a match for only the most prominent face within an image. If the image contains multiple people, we first need to use the DetectFaces API to retrieve the individual bounding boxes of the faces within the image. We can then use the retrieved x-y coordinates to cut out the faces from the image and submit them individually to the SearchFacesByImage API.

   We can extend the boundaries of face boxes in all directions to simplify the definition of the box to crop. For tilting the head, we can adjust the orientation and location of the box or can extend the size of the crop-out. This works because Amazon Rekognition tries to detect a person for only the largest face within an image. Slightly extending the size of the crop-out by up to 50 percent won’t unveil another face that is larger than the original detected face.

4.3. Azure Internal Architecture:


API 1
DocumentDB
Service bus Queue
API 2
Blob Storage
Cognitive Services Face API
GitHub Continuous Integration
ImageFaceFinder
Blob Trigger. Sends new images to Face API for Face Location
ImageFacesAttachMeta
Receives HTTP post from Function 1 with Face Data
Updates the Parent Record in DocDB
HomeFeedGenerator
Monitors Service Bus Queue for new home feed items.
Generates user home feed and places in Azure Table Storage
Azure Table Storage
Microsoft

The following process was put into place using Azure Functions **and Cognitive Services Face API:**

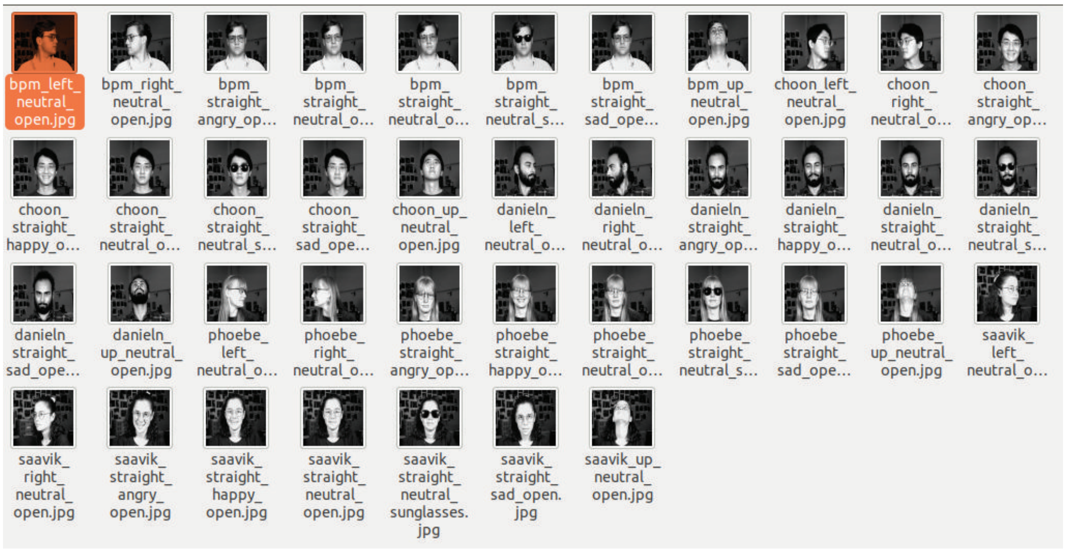
1. User creates a new "catch" including photos and catch information.
2. This information is then sent to two separate existing APIs:
    - API1 – Creates record in DocumentDB with all the catch information and metadata. In addition, it creates a record in a Service Bus queuing up the catch for processing.
    - API2 – Takes the user's photos and uploads them into blob storage.
3. HomeFeedGenerator function monitors the Service Bus queue. As new records are created, the function triggers and generates items for the user's home feed.
4. ImageFaceFinder function monitors the container in blob storage where the photos are uploaded. As new photos are added, the image is sent to Cognitive Services Face API and an object is returned with the location of all faces in the picture.
5. ImageFacesAttachMeta function receives an HTTP POST from the previous function with the source CatchId and a JSON object containing all of the face data. It then takes this data and attaches it to the original document inside of DocumentDB.

CHAPTER 5

DATASET

This data consists of 80 black and white face images of 15 people taken with varying pose (straight, left, right, up), expression (neutral, happy, sad, angry), eyes (wearing sunglasses or not) in jpg format.

SAMPLE DATA SET:


bpm_left_neutral_open.jpg
bpm_right_neutral_open.jpg
bpm_straight_angry_op...
bpm_straight_neutral_o...
bpm_straight_neutral_o...
bpm_straight_neutral_s...
bpm_straight_sad_ope...
bpm_up_neutral_open.jpg
choon_left_neutral_open.jpg
choon_right_neutral_o...
choon_straight_angry_op...
choon_straight_happy_o...
choon_straight_neutral_o...
choon_straight_neutral_s...
choon_straight_sad_ope...
choon_up_neutral_open.jpg
danieln_left_neutral_o...
danieln_right_neutral_o...
danieln_straight_angry_op...
danieln_straight_happy_o...
danieln_straight_neutral_o...
danieln_straight_neutral_s...
danieln_straight_sad_ope...
danieln_up_neutral_open.jpg
phoebe_left_neutral_o...
phoebe_right_neutral_o...
phoebe_straight_angry_op...
phoebe_straight_happy_o...
phoebe_straight_neutral_o...
phoebe_straight_neutral_s...
phoebe_straight_sad_ope...
phoebe_up_neutral_open.jpg
saavik_left_neutral_o...
saavik_right_neutral_open.jpg
saavik_straight_angry_open.jpg
saavik_straight_happy_open.jpg
saavik_straight_neutral_open.jpg
saavik_straight_neutral_sunglasses.jpg
saavik_straight_sad_open.jpg
saavik_up_neutral_open.jpg

Data Characteristics

Each image can be characterized by the pose, expression and eyes. Images of 10 human subjects with straight orientation, neutral expression and without sunglasses is chosen as the training set. Test cases:

1.straight-neutral-sunglasses(s-n-g)

2.straight-angry-open(s-a-o)

3.straight-happy-open(s-h-o)

4.straight-sad-open(s-s-o)

5.up-neutral-open(u-n-o)

6.left-neutral-open(l-n-o)

7.right-neutral-open(r-n-o)

CHAPTER 6

IMPLEMENTATION

FACE API IMPLEMENTATION

Prerequisites:

- Install either Python 2.7+ or Python 3.5+.
- Install pip.
- Install the Python SDK for the Face API using the command: pip install cognitive_face
- Obtain a subscription key for Microsoft Cognitive Services.

Steps:

1. Authorize the API call
2. Create the PersonGroup

   2.1. Define people for the PersonGroup

   2.2. Detect faces and register them to correct person

3. Train the PersonGroup
4. Identify a face against a defined PersonGroup

Code to identify a person in a PersonGroup:

```
import httplib, urllib, base64

headers = {
    # Request headers
    'Content-Type': 'application/json',
    'Ocp-Apim-Subscription-Key': '{subscription key}',
}
params = urllib.urlencode({
})
try:
    conn = httplib.HTTPSConnection('westus.api.cognitive.microsoft.com')
    conn.request("POST", "/face/v1.0/identify?%s" % params, "{body}", headers)
    response = conn.getresponse()
    data = response.read()
    print(data)
    conn.close()
except Exception as e:
    print("[Errno {0}] {1}".format(e.errno, e.strerror))
```

Sample Ouput:

```
{
  [
    {
      "faceId": "c5c24a82-6845-4031-9d5d-978df9175426",
      "candidates": [
        {
          "personId": "25985303-c537-4467-b41d-bdb45cd95ca1",
          "confidence": 0.70
        }
      ]
    }
  ]
}
```

FACE REKOGNITION IMPLEMENTATION:

Steps:

1. Install the latest Boto 3 release via **pip**:

```
pip install boto3
```

2. Set up authentication credentials: Configure your credentials file.

```
aws configure
```

3. Upload images to AWS S3 Bucket:

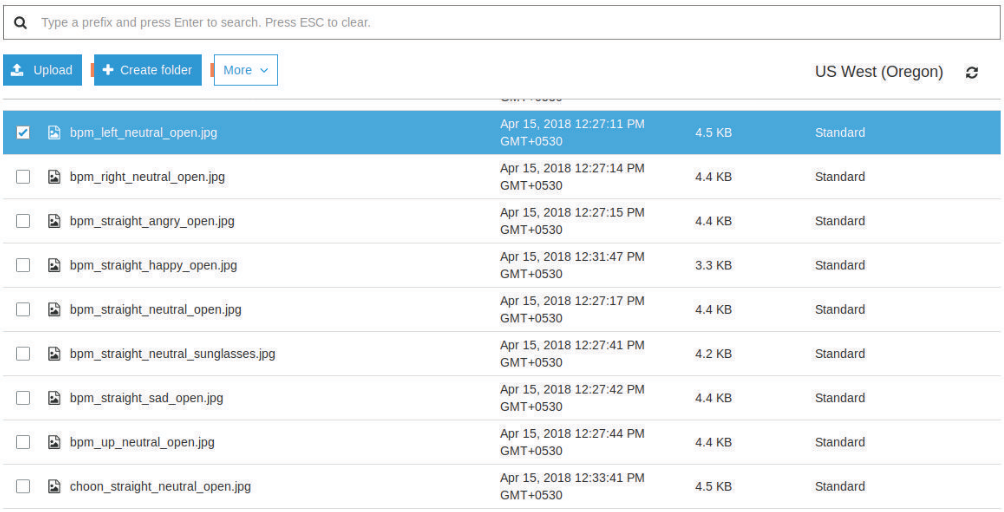
```
import boto3

s3 = boto3.resource('s3')

data = open('test.jpg', 'rb')

s3.Bucket('my-bucket').put_object(Key='test.jpg', Body=data)
```

S3 Bucket:


Type a prefix and press Enter to search. Press ESC to clear.
Upload
Create folder
More
US West (Oregon)

☑	bpm_left_neutral_open.jpg	Apr 15, 2018 12:27:11 PM GMT+0530	4.5 KB	Standard
□	bpm_right_neutral_open.jpg	Apr 15, 2018 12:27:14 PM GMT+0530	4.4 KB	Standard
□	bpm_straight_angry_open.jpg	Apr 15, 2018 12:27:15 PM GMT+0530	4.4 KB	Standard
□	bpm_straight_happy_open.jpg	Apr 15, 2018 12:31:47 PM GMT+0530	3.3 KB	Standard
□	bpm_straight_neutral_open.jpg	Apr 15, 2018 12:27:17 PM GMT+0530	4.4 KB	Standard
□	bpm_straight_neutral_sunglasses.jpg	Apr 15, 2018 12:27:41 PM GMT+0530	4.2 KB	Standard
□	bpm_straight_sad_open.jpg	Apr 15, 2018 12:27:42 PM GMT+0530	4.4 KB	Standard
□	bpm_up_neutral_open.jpg	Apr 15, 2018 12:27:44 PM GMT+0530	4.4 KB	Standard
□	choon_straight_neutral_open.jpg	Apr 15, 2018 12:33:41 PM GMT+0530	4.5 KB	Standard

4. Python code

Create Collection:

```
import boto3
client = boto3.client('rekognition')
response = client.create_collection(
    CollectionId='friends')
print(response)
```

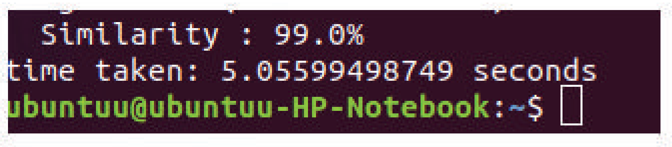
Index Faces:

```
import boto3
response = client.index_faces(
    CollectionId='friends',
    Image={
        'S3Object': {
            'Bucket': 'faceset',
            'Name': 'bpm_left_neutral_open.jpg',
        }
    },
    ExternalImageId='choon',
    DetectionAttributes=[
        'ALL',
    ]
)
print(response)
```

Search Face By Image:

```
import boto3
import time()
strat_time=time.time()
response = client.search_faces_by_image(
    CollectionId='string',
    Image={
        'S3Object': {
            'Bucket': 'faceset',
            'Name': ' bpm_left_neutral_sunglasses.jpg",
        }
    },
    MaxFaces=1,
    FaceMatchThreshold=0.0
)
print(response)
print (“time taken: %s seconds “ % (time.time() – start_time))
```

Sample Output



```
  Similarity : 99.0%
time taken: 5.05599498749 seconds
ubuntuu@ubuntuu-HP-Notebook:~$ 
```

CHAPTER 7

RESULT

Quantitative Analysis:

Formula Used:

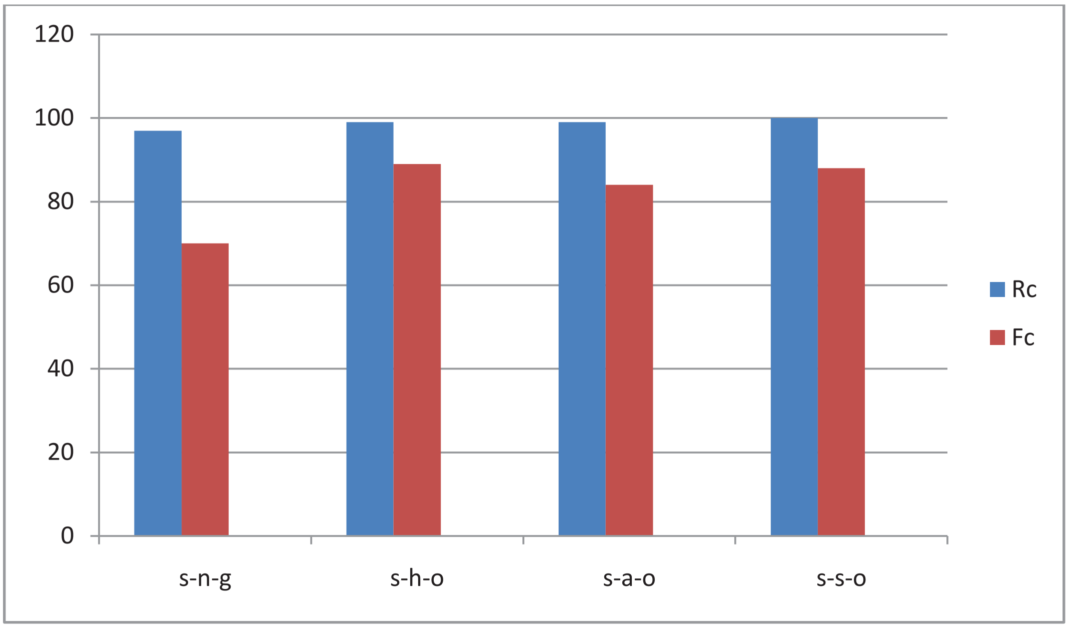
- If Fc>Rc: (Fc-Rc)/Rc
- If Rc>Fc: (Rc-Fc)/Fc

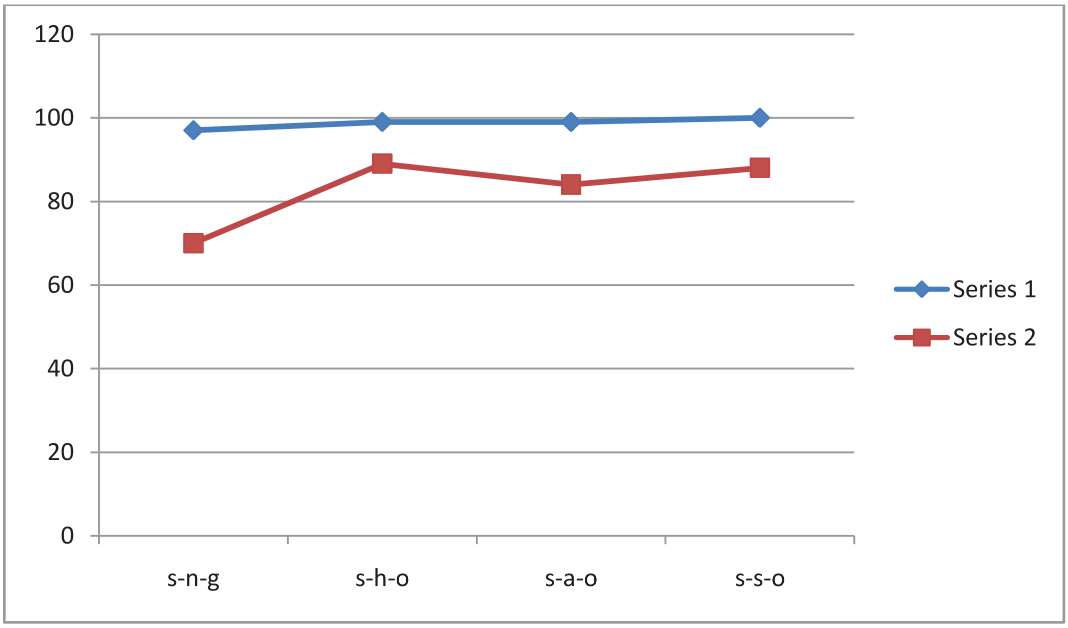
The confidence and delay are obtained from the implementation of the code on the dataset. We use these values to have a quantitative comparison of the two services. The formula given above is then applied to give the amount by which one is more accurate than the other in different fields of comparison.

Table obtained from implementing the two services:

Test Set	**Rekognition Confidence (Rc)**	**Rekognition Time-Taken (Rt)**	**Face API Confidence (Fc)**	**Face API Time-Taken (Ft)**	**Value obtained from the formula**
s-n-g	97	4.7	70	0.09	38.5
s-h-o s-a-o s-s-0	99 99 100	3.971 6.464 2.771	89 84 88	0.0340 0.222 0.717	11.2 17.85 13.63
l-n-o r-n-o u-n-o	90 70 63	4.35 4.391 4.25	Does not detect	-	Rekognition is better in identifying when there is a change in orientation

GRAPHS:





Graphs obtained after comparing the two services

RESULT:

From the values obtained, we infer that AWS Face Rekognition is more accurate at recognizing a person by a factor of 38% when the presence of an obstacle like sunglasses is present on the image and by a factor of 15% on average when the expression is changed. Since Face API does not detect faces with any orientation other than straight pose (upright face looking at the camera), it makes it obvious for us to understand that Face Rekognition is indeed preferable over Face API in such a case. Although it seems more accurate in some aspects, Azure is much faster. Hence, we need to choose from the two services depending on our requirement and usage.

CHAPTER 8 CONCLUSION

8.1 Future Work

For our work, we have considered only three main features for testing. They are: Orientation, expression and w and w/o glasses. We have also considered only accuracy and time elapsed during it execution as the basis for comparison. There can be various other basis on which they can be compared. The project can further be extended to more number of human subjects with a combination of feature variations from the training set. This can help us get a deeper insight into the behavioural aspects of the algorithm that feature variation can bring out when they are combined in different ways. The project can also be extended in the direction of using additional feature variation like age and illumination that plays a vital role as a determining factor for the performance as well. The project deals only with images in jpg format. It can be studied and performed on other formats that are classified as lossy and lossless compressed formats. The relative study of the formats with the performance of the algorithms can also be taken as a new area of discussion in future.

8.2 Limitations

The project is confined only to a few major discriminant factors of comparison. The project deals only with orientation, expression and presence of sunglasses. Yet there are many other features like size, illumination and image format that determine our results. Hence, the accuracy of our result might not be sufficient enough to understand the two services completely. The image format used is jpg. The compression level of the image plays a major role as well in determining the confidence. Moreover, the two underlying algorithms in the services might behave differently with different image formats. Our project uses 120 images as the data set that might not be sufficient to accurately bring out the results of the comparison. The project works with single face images and not for multi face images. We confine the image format to jpg alone.

REFERENCES

1. Etemad, Kamran, and Rama Chellappa. "Discriminant analysis for recognition of human face images." *JOSA A* 14, no. 8 (1997): 1724-1733.
2. Phillips, P. Jonathon, Hyeonjoon Moon, Syed A. Rizvi, and Patrick J. Rauss. "The FERET evaluation methodology for face-recognition algorithms." *IEEE Transactions on pattern analysis and machine intelligence* 22, no. 10 (2000): 1090-1104.
3. Singh, Aruni, Sanjay Kumar Singh, and Shrikant Tiwari. "Comparison of face recognition algorithms on dummy faces." *The International Journal of Multimedia & Its Applications* 4, no. 4 (2012): 121.
4. Bhuvaneshwari, K. V., A. Abirami, and N. Sripriya. "Face Recognition Using PCA." *International Journal Of Engineering And Computer Science* 6, no. 4 (2017).
5. R.Baron "Mechanisms of human face recognition" Int. j. Man machine studies 15,137-178 (1981).
6. G.Devis ,H Ellis , and E. J. Shepherd, Perceiving and Remembering Faces (Academic, New York, 1981).
7. Yacoob and L. S. Davis, ''Computing spatio-temporal representations of human faces,'' in Proceedings of the IEEE Computer Society Conference on Computer Vision and PatternRecognition(IEEE)ComputerSociety,LosAlamitos,Calif., 1994), pp. 70–75.
8. K. Etemad, R. Chellappa, "Discriminant analysis for recognition of human face images", *ICASSP '96*, pp. 2148-2151, 1996.
9. B. Moghaddam, C. Nastar, A. Pentland, "Bayesian face recognition using deformable intemsity surfaces", ***Proc. Computer Vision and Pattern Recognition 96***, pp. 638-645, 1996.
10. A.Pentland, B. Moghaddam, T. Starner, "View-based and modular eignspaces for face recognition", ***Proc. Computer Vision and Pattern Recognition 94***, pp. 84-91, 1994.

11. Lucas D.Introna, H.Nissenbaum.: “Facial Recognition Technology, A survey of policy and implementation Issues”, CCPR.

12. W. Zhao, R.Chellpa, A.Rosenfield, P.J.Phillips: “Face Recognition A Literature Survey”.

13. Juthi Raut, Snehal patil and prof.Mamta Meena , “Face detection and recognition in video”.