Face Recognition

S. E. Computer Engineering

By

Misaal D'souza	**25**
Shruti Kamath	**41**
Yogita Likhi	**46**

Supervisor:

Ms. Priya Chaudhari

Designation

Introduction :

Face recognition is one of the many wonders that AI research has brought forward to the world. 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 multiple 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 also described as a Biometric Artificial Intelligence based application that can uniquely identify a person by analysing patterns based on the person's facial textures and shape.

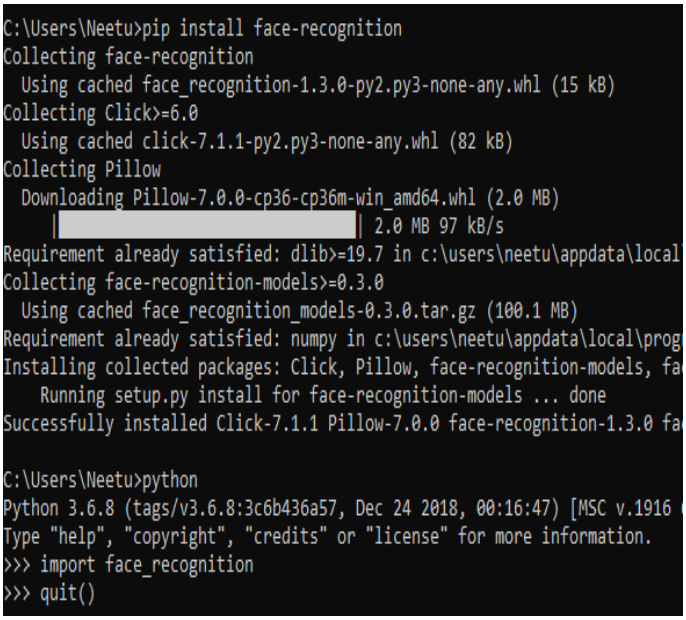
To understand how a machine can recognize faces, we can start with asking ourselves — how do we recognize a face? Most images of human faces have two eyes, a nose, lips, forehead, chin, ears, hair… That rarely changes. Yet, faces are different from each other. What makes them different? At the same time, face of the same person changes with emotion, expression, age… In fact just change in orientation creates a different image. How do we identify a person in spite of all that?

On a gross level, one can say that there are some components of a face are related to age, emotion and orientation. While there are some other components that are stick to the person irrespective of the age, emotion, etc. Further, we can say that these components are not orthogonal or independent. We have all seen people who look alike sideways, but very different otherwise. Or, a kid in the family reminds people of his parents at that age, etc.

So, it is not so easy to logically identify these individual components. But, one can say that there are several overlapping components of the face — which are individually responsible for the perception of emotion, age and the person himself. Essentially, we know that there is “some relation that is too complex for logic” — that is where **machine learning** shows up!

Libraries :

1. face_recognition (including Click, Pillow , SciPy and Numpy)



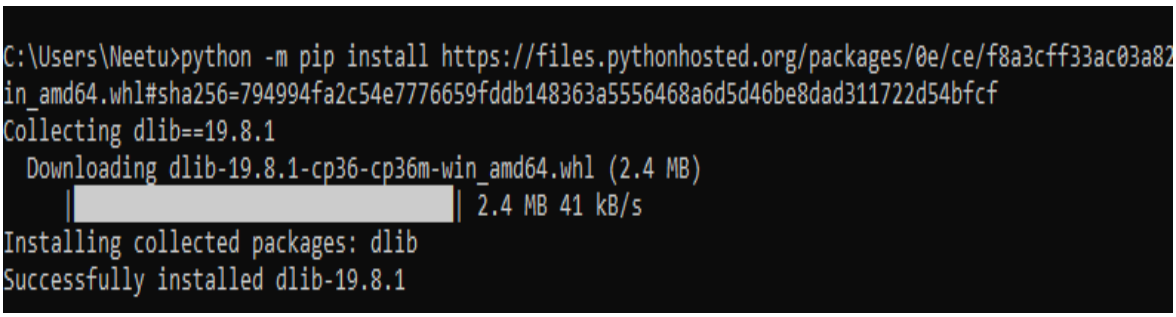
The library works in three simple steps :

- Identify a face in a given image
- Identify specific features in the face
- Generate a face encoding vector of 128 values

Based on this encoding, we can measure the similarity between two face images — that can tell us if they belong to the same person.

2. dib

Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems. It is used in both industry and academia in a wide range of domains including robotics, embedded devices, mobile phones, and large high performance computing environments.



Steps :

The library works in three simple steps :

- Identify a face in a given image
- Identify specific features in the face
- Compare it with the unknown images
- If found , it generates a face encoding vector of 128 values

Based on this encoding, we can measure the similarity between two face images — that can tell us if they belong to the same person.

Implementation :

Code:

#findfaces.py

```
import face_recognition
image = face_recognition.load_image_file('./img/groups/team1.jpg')
face_locations = face_recognition.face_locations(image)
#Array Of Co-ordinates Of Each Face
print(face_locations)
print(f'There are {len(face_locations)} people in this image')
```

Output :

```
[(39, 411, 101, 349), (101, 294, 163, 232), (38, 95, 90, 43), (72, 164, 124, 112)]
There are 4 people in this image
```

To print face_landmarks :

```
face_landmarks = face_recognition.face_landmarks(image)
print(face_landmarks)
```

Output:

```
[{'chin': [(62, 98), (63, 111), (65, 123), (67, 135), (72, 146), (79, 157), (89, 166), (100, 172), (112, 174), (124, 172), (133, 165
), (141, 156), (147, 146), (152, 136), (155, 125), (157, 115), (159, 103)], 'left_eyebrow': [(74, 92), (82, 86), (91, 84), (101, 85)
, (109, 89)], 'right_eyebrow': [(120, 88), (129, 86), (138, 86), (146, 88), (152, 93)], 'nose_bridge': [(115, 95), (115, 104), (115,
 113), (115, 123)], 'nose_tip': [(103, 125), (108, 128), (114, 131), (120, 128), (125, 126)], 'left_eye': [(84, 97), (89, 94), (95,
94), (100, 98), (95, 99), (89, 99)], 'right_eye': [(127, 98), (132, 95), (138, 95), (142, 99), (138, 100), (132, 99)], 'top_lip': [(
91, 136), (102, 136), (109, 137), (114, 138), (119, 137), (126, 137), (134, 137), (131, 138), (119, 139), (114, 139), (109, 139), (9
4, 138)], 'bottom_lip': [(134, 137), (126, 145), (119, 148), (114, 149), (108, 148), (101, 144), (91, 136), (94, 138), (108, 143), (
114, 144), (119, 143), (131, 138)]}]
```

#facematch.py

```
import face_recognition
image_of_bill = face_recognition.load_image_file('./img/known/Bill Gates.jpg')
bill_face_encoding = face_recognition.face_encodings(image_of_bill)[0]
unknown_img = face_recognition.load_image_file('./img/unknown/gates_lookalike.jpg
')
unknown_face_encoding = face_recognition.face_encodings(unknown_img)[0]
#Compare Faces
results = face_recognition.compare_faces([bill_face_encoding],unknown_face_encodi
ng)
if results[0]:
    print('This is Bill Gates')
else:
    print('This is NOT Bill Gates')
```

Output:

1. Comparing Bill Gates image from a known folder with another Bill Gates image in an unknown folder

```
This is Bill Gates
```

1. Comparing Bill Gates with an unknown person

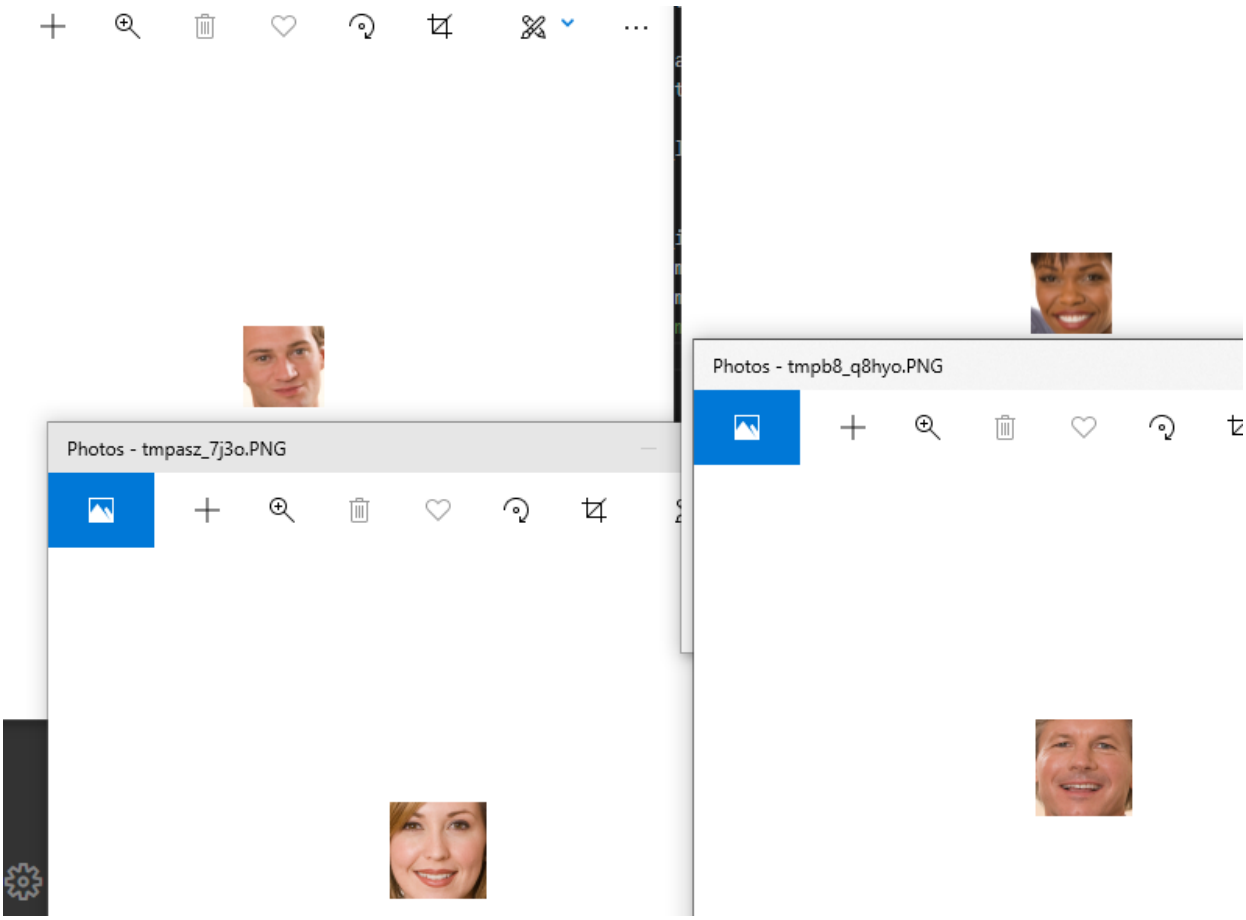
```
This is NOT Bill Gates
```

#pullfaces.py

```
from PIL import Image
import face_recognition

image = face_recognition.load_image_file('./img/groups/team1.jpg')
face_locations = face_recognition.face_locations(image)

for face_location in face_locations:

    top , right , bottom , left = face_location

    face_image = image[top:bottom,left:right]
    pil_image= Image.fromarray(face_image)
    pil_image.show()
 # Save each faces from an image
 # pil_image.save(f'{top}.jpg')
```

Image Used :

Output:



#findfaces.py

```
import tkinter as tk
import face_recognition
from PIL import Image, ImageDraw
image_of_albert = face_recognition.load_image_file('./img/known/Albert Einstein.j
pg')
albert_face_encoding = face_recognition.face_encodings(image_of_albert)[0]
image_of_alexander = face_recognition.load_image_file('./img/known/Srinivasa Rama
nujan.jpg')
alexander_face_encoding = face_recognition.face_encodings(image_of_alexander)[0]
image_of_kalam = face_recognition.load_image_file('./img/known/APJ Abdul Kalam.jp
g')
kalam_face_encoding = face_recognition.face_encodings(image_of_kalam)[0]
image_of_raman = face_recognition.load_image_file('./img/known/CV Raman.jpg')
raman_face_encoding = face_recognition.face_encodings(image_of_raman)[0]
image_of_galileo = face_recognition.load_image_file('./img/known/Galileo.jpg')
galileo_face_encoding = face_recognition.face_encodings(image_of_galileo)[0]
image_of_newton = face_recognition.load_image_file('./img/known/Isaac Newton.jpg'
)
newton_face_encoding = face_recognition.face_encodings(image_of_newton)[0]
image_of_marie = face_recognition.load_image_file('./img/known/Marie Curie.jpg')
marie_face_encoding = face_recognition.face_encodings(image_of_marie)[0]
image_of_thomas = face_recognition.load_image_file('./img/known/Thomas Edison.jpg
')
thomas_face_encoding = face_recognition.face_encodings(image_of_thomas)[0]
#  Create arrays of encodings and names
known_face_encodings = [albert_face_encoding,alexander_face_encoding,kalam_face_e
ncoding,raman_face_encoding,galileo_face_encoding,newton_face_encoding,marie_face
_encoding,thomas_face_encoding]
known_face_names = ["Albert Einstein","Alexander Graham Bell","APJ Abdul Kalam","
CV Raman","Galileo","Isaac Newton","Marie Curie","Thomas Edison"]
# Load test image to find faces in
address=[]
n=int(input("\nHow many images do you want to scan: "))
for i in range(n):
  c=input("Image Address: ")
  address.append(c)
for i in address:
  test_image = face_recognition.load_image_file(i)
# Find faces in test image
  face_locations = face_recognition.face_locations(test_image)
  face_encodings = face_recognition.face_encodings(test_image, face_locations)
# Convert to PIL format
  pil_image = Image.fromarray(test_image)
# Create a ImageDraw instance
  draw = ImageDraw.Draw(pil_image)
# Loop through faces in test image
  for(top, right, bottom, left), face_encoding in zip(face_locations, face_encodi
ngs):
    matches = face_recognition.compare_faces(known_face_encodings, face_encoding)
```

```
    name = "Unknown Person"
  # If match
    if True in matches:
      first_match_index = matches.index(True)
      name = known_face_names[first_match_index]
  # Draw box
    draw.rectangle(((left, top), (right, bottom)), outline=(255,255,0))
  # Draw label
    text_width, text_height = draw.textsize(name)
    draw.rectangle(((left,bottom - text_height - 10), (right, bottom)), fill=(255
,255,0), outline=(255,255,0))
    draw.text((left + 6, bottom - text_height - 5), name, fill=(0,0,0))
  del draw
# Display image
  pil_image.show()
# Save image
  pil_image.save('identify.jpg')
```

Output:

Applications :

1. Companies across the world are using facial recognition to secure their premises.

The fact that machines can today accurately recognize individuals, presents a slew of opportunities for the security sector, chief among them the ability to identify unauthorized access to locations where non-authorized people shouldn't be. It's a well-known fact that IP cameras today can be equipped with facial recognition software, to enable complex access control of premises, with enabling of individual whitelists and blacklists for specific locations, enabling perimeter and asset monitoring, on top of threat and intrusion detection.

Security companies also regularly equip employees with body cameras, to enable video capture and recording during sensitive interactions and potential altercations when deploying security personnel to handle security intrusions. This is especially useful if security forces are engaging or handling intrusions in an area that may not be covered by fixed CCTV cameras. In fact, many of the different applications of facial recognition technologies we will go through in the later points, are centred around security enhancements in application to a specific industry segment's needs.

2. How Facial Recognition is used in Healthcare

Facial recognition is slowly gaining momentum in healthcare, partially due to improvements in artificial intelligence, which has made it possible to use the technology in different ways. AI-powered face recognition systems in the market have allowed healthcare professionals to consider using the technology in several ways that go beyond security purposes.

A. Emotion Detection & Sentiment Analysis

Real-time emotion detection is yet another valuable application of face recognition in healthcare. It can be used to detect several emotions patients exhibit during their stay in the facility and analyze the data so as to determine how they are feeling. The results of the analysis may help identify where the patients need more attention in case they're in pain or sad.

B. Securing Hospital Facilities

One of the most popular applications of face recognition is for security. Law enforcement personnel can use this technology to detect and identify individuals by scanning anyone entering the hospital. They can then compare each person to a list of flagged individuals. The technology can also be used in hospitals to identify individuals who might be drug seekers or individuals previously ejected whom the hospital no longer allows to access the facility.

helps diagnose rare genetic diseases

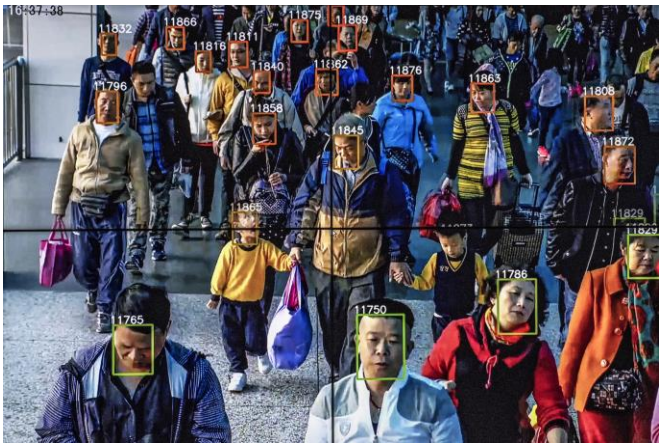
predicts unsafe behaviour of patients in the intensive care unit (ICU), such as accidentally removing their breathing tube.

“Using images of a patient’s face and eyes we were able to train computer systems to recognise high-risk arm movement,” said study lead author Akane Sato from Yokohama City University in Japan.

3. Making Cities Even Smarter

A. To find missing children and victims of human trafficking

Up to 800,000 children go missing every year. But face recognition can make it easier to bring them home. In addition to adding dangerous individuals to a face recognition database, smart cities can also add missing children and victims of human trafficking to their face recognition database. Let’s say a missing child is detected by a public camera feed, the victim could be instantly identified and authorities could be alerted.



B. Protect Public Events

Unfortunately, public events remain attractive targets for terrorists and mass-shooters. By augmenting cameras at events with facial recognition software, it’s possible to instantly identify individuals who present a threat.
Security professionals at events have a tremendously difficult job, and they do their best. But it’s impossible to remember the names and faces of every individual who may pose a threat to an event. This is compounded by the fact that it is difficult for humans to recognize these individuals in a crowd. Face recognition can solve this problem. It works well in crowds and can alert human security personnel the moment that a threat is detected.

HOW FACIAL RECOGNITION IS HELPING THE WORLD TO BEAT COVID-19

Indian police use face recognition app to track Covid-19 patients

30 March 2020 15:44 GMT

Jump to comments

Police in India's Pune state have developed a facial recognition software system for surveillance of 'Home Quarantined' persons in Pune and Pimpri-Chinchwad cities.

Police Commissioner Dr K Venkatesham said in a press release that the selfie-app based face recognition and location tracking system enables a person to register himself/herself with their details like Name, Phone Number, Selfie and other relevant information.

The person uploads his/her registration details using the app, following which a verification is made against the existing master list of persons.

Once the registration is approved, the selfie app enables a passenger to upload attendance in the form of a selfie with location tagging in realtime.

This information is then sent to a private cloud based server which runs a real time Artificial Intelligence based face recognition and GPS algorithm and compares against the registered information.

Russia uses facial recognition to tackle virus

As Russian cities go into lockdown to try to contain coronaviru Moscow is using the latest technology to keep track of reside

City officials are using a giant network of tens of thousands of cameras - installed with facial recognition software - which the plan to couple with digital passes on people's mobile phones. prompted concern about whether such widespread surveillanc will ever be rolled back.

The government in China are now scanning every citizens face (including nose images)and the images are then stored in the central database after which ,it is analysed using machine learning algorithms.The CCTV cameras locate the location of that person easily and if they are found to have the symptoms of the virus , then they are sent a message saying " You have to quarantine yourself for the next 14 days" and the same message is sent to the people who have been in contact to that particular person.

Outside, drones hovered above streets, yelling at people to get inside and scolding them for not wearing face masks, while elsewhere in China facial-recognition software, linked to a mandatory phone app that color-coded people based on their contagion risk, decided who could enter shopping malls, subways, cafes and other public spaces.

Conclusion :

Face recognition technology has come a long way in the last twenty years. Today, machines are able to automatically verify identity information for secure transactions, for surveillance and security tasks, and for access control to buildings etc. These applications usually work in controlled environments and recognition algorithms can take advantage of the environmental constraints to obtain high recognition accuracy. However, next generation face recognition systems are going to have widespread application in smart environments -- where computers and machines are more like helpful assistants.

Successfully Implemented Facial Recognition.