**Facial recognition and analysis**

Facial recognition is a way of recognizing a human face through technology. A facial recognition system uses biometrics to map facial features from a photograph or video.  Face detection technology can be applied to various fields -- including security, biometrics, law enforcement, entertainment and personal safety -- to provide surveillance and tracking of people in real time.

**Importance’s**

Facial detection has progressed from fundamental computer vision applications to advances in machine learning to increasingly sophisticated artificial neural networks and related technologies with continuous performance improvements. It now plays an important role as the first step in many key applications. Facial detection and analysis have a significant effect on how sequential operations will perform in the application.  The primary purpose of the detection algorithm is to determine if the image has a face. Such identification helps to identify age, gender and emotions using expressions and to identify which parts of the image or video to focus on. It is a component of object detection and can be used in many fields such as security, biometrics, law enforcement, entertainment, personal security, etc.

**Benefits**

**Enhanced security**

Face detection improves surveillance efforts and helps track down criminals and terrorists. In a facial recognition system, which maps an individual's facial features mathematically and stores the data as a faceprint. Requirements to identify what parts of an image or video are needed to create a faceprint. Once identified, the new fingerprint can be compared with the archived fingerprint to determine if there is a match. Personal security is also enhanced as hackers have nothing to steal or change, such as passwords.

**Efficient shopping**

While identifying and finding missing persons and criminals are arguably the most important benefits of facial recognition, they extend beyond security to convenience. Instead of making cash or credit purchases at stores, facial recognition technology can recognize your face and charge the goods to your account. Use of this increased during the pandemic to serve both convenience and security purposes, as well as help manage the smaller ratio of staff to customers, but retailers also see the tech being used in the future to recognise and advertise to loyalty club members and clock employees in and out.

**Challenges**

Face detection can consider a substantial part of face recognition operations.  According to its strength to focus computational resources on the section of an image holding a face. The method of face detection in pictures is complicated because of variability present across human faces such as pose, expression, position and orientation, skin colour, the presence of glasses or facial hair, differences in camera gain, lighting conditions, and image resolution. Recent years have brought advances in face detection using deep learning, which presents the advantage of significantly outperforming traditional computer vision methods.

**Example**

Colab notebook using camera to capture real time images for facial feature detection at runtime.

CELL1:

!pip3 install face\_recognition

%matplotlib inline

CELL2:

from IPython.display import display, Javascript

from google.colab.output import eval\_js

from base64 import b64decode

def take\_photo(filename='base\_pic.jpg', quality=0.8):

  js = Javascript('''

    async function takePhoto(quality) {

      const div = document.createElement('div');

      const capture = document.createElement('button');

      capture.textContent = 'Capture';

      div.appendChild(capture);

      const video = document.createElement('video');

      video.style.display = 'block';

      const stream = await navigator.mediaDevices.getUserMedia({video: true});

      document.body.appendChild(div);

      div.appendChild(video);

      video.srcObject = stream;

      await video.play();

      // Resize the output to fit the video element.

      google.colab.output.setIframeHeight(document.documentElement.scrollHeight, true);

      // Wait for Capture to be clicked.

      await new Promise((resolve) => capture.onclick = resolve);

      const canvas = document.createElement('canvas');

      canvas.width = video.videoWidth;

      canvas.height = video.videoHeight;

      canvas.getContext('2d').drawImage(video, 0, 0);

      stream.getVideoTracks()[0].stop();

      div.remove();

      return canvas.toDataURL('image/jpeg', quality);

    }

    ''')

  display(js)

  data = eval\_js('takePhoto({})'.format(quality))

  binary = b64decode(data.split(',')[1])

  with open(filename, 'wb') as f:

    f.write(binary)

  return filename

CELL3:

from IPython.display import Image

try:

  filename = take\_photo()

  print('Saved to {}'.format(filename))

  display(Image(filename))

except Exception as err:

   print(str(err))

CELL4:

from PIL import Image, ImageDraw

import face\_recognition

#from google.colab.patches import cv2\_imshow

image = face\_recognition.load\_image\_file("base\_pic.jpg")

#below returns all facial features

face\_landmarks\_list = face\_recognition.face\_landmarks(image)

#works with multi face too

print("found {} face(s) in this photograph.".format(len(face\_landmarks\_list)))

landmark\_image = Image.fromarray(image)

drawer = ImageDraw.Draw(landmark\_image)

for landmarks in face\_landmarks\_list:

    #print features

    for facial\_feature in landmarks.keys():

        print("The {} in this face has the following points: {}".format(facial\_feature, landmarks[facial\_feature]))

    for facial\_feature in landmarks.keys():

        drawer.line(landmarks[facial\_feature], width=3)

CELL5:

landmark\_image