POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

EMERGING TECHNOLOGIES IN COMPUTER ENGINEERING

"Face Recognition"

In Partial Fulfillment of the Requirements for the

Bachelor of Science in Computer Engineering

By:

HICANA, JHANNA MAE G.

To:

Prof. Simon Salvador Tidon

2022

Table of Contents

Introduction

Facial recognition is a way of identifying or confirming an individual's identity using their face. Facial recognition systems can be used to identify people in photos, videos, or in real-time. Facial recognition is a category of biometric security. Other forms of biometric software include voice recognition, fingerprint recognition, and eye retina or iris recognition. The technology is mostly used for security and law enforcement, though there is increasing interest in other areas of use.

How does Face Recognition Works?

Many people are familiar with face recognition technology through the FaceID used to unlock iPhones (however, this is only one application of face recognition). Typically, facial recognition does not rely on a massive database of photos to determine an individual's identity — it simply identifies and recognizes one person as the sole owner of the device, while limiting access to others. Beyond unlocking phones, facial recognition works by matching the faces of people walking past special cameras, to images of people on a watch list. The watch lists can contain pictures of anyone, including people who are not suspected of any wrongdoing, and the images can come from anywhere — even from our social media accounts. Facial technology systems can vary, but in general, they tend to operate as follows:

Step 1. Face Detection

The camera detects and locates the image of a face, either alone or in a crowd. The image may show the person looking straight ahead or in profile.

Step 2. Face Analysis

Next, an image of the face is captured and analyzed. Most facial recognition technology relies on 2D rather than 3D images because it can more conveniently match a 2D image with public photos or those in a database. The software reads the geometry of your face. Key factors include the distance between your eyes, the depth of your eye sockets, the distance from forehead to chin, the shape of your cheekbones, and the contour of the lips, ears, and chin. The aim is to identify the facial landmarks that are key to distinguishing your face.

Step 3. Converting Image to Data

The face capture process transforms analog information (a face) into a set of digital information (data) based on the person's facial features. Your face's analysis is essentially turned into a mathematical formula. The numerical code is called a faceprint. In the same way that thumbprints are unique, each person has their own faceprint.

Step 4. Finding a match

Your faceprint is then compared against a database of other known faces. For example, the FBI has access to up to 650 million photos, drawn from various state databases. On Facebook, any photo tagged with a person's name becomes a part of Facebook's database, which may also be used for facial recognition. If your faceprint matches an image in a facial recognition database, then a determination is made. Of all the biometric measurements, facial recognition is considered the most natural. Intuitively, this makes sense, since we typically recognize ourselves and others by looking at faces, rather than thumbprints and irises. It is estimated that over half of the world's population is touched by facial recognition technology regularly.

Definition of Deep Learning for Face Recognition

In reality, face recognition has been in development since the 1960s.But only because of modern digital technology developments and methods like artificial intelligence, Internet of Things, Artificial Intelligence facial recognition is now a highly developing technology. Furthermore, several factors such as the rise of public security concerns and the need for more reliable identity verification methods have contributed to the increasing interest in face recognition. Pattern recognition, face analysis, machine learning, and deep learning are the main contributors to the development of facial recognition systems. A subset of ML (machine learning), deep learning uses ML algorithms and lots of data to educate deep neural networks for enhanced accuracy. Deep learningmlearns through an ANN (Artificial Neural Network) and thus considered as more human-like.

Real-Time Face Recognition with Python and OpenCV

Face Recognition is a technology in computer vision. In Face recognition / detection we locate and visualize the human faces in any digital image. It is a subdomain of Object Detection, where we try to observe the instance of semantic objects. These objects are of particular class such as animals, cars, humans, etc. Face Detection technology has importance in many fields like marketing and security.

Cascade Classifiers and Haar Features

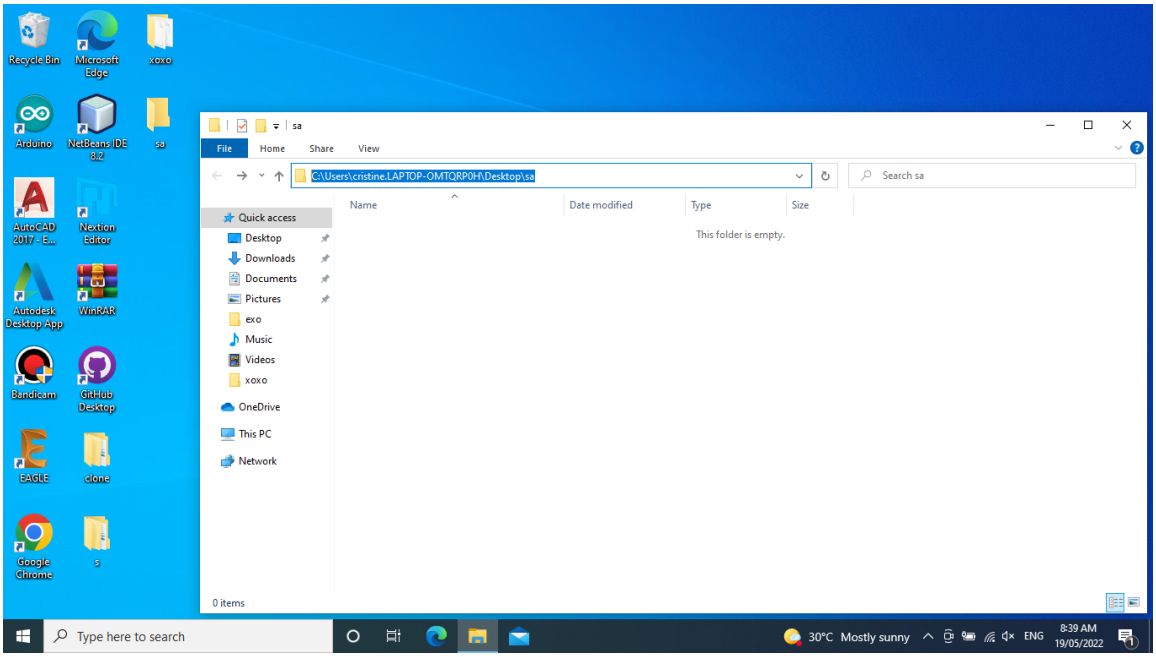
Cascade Classifiers and Haar Features are the methods used for Object Detection. It is a machine learning algorithm where we train a cascade function with tons of images. These images are in two categories: positive images containing the target object and negative images not containing the target object. There are different types of cascade classifiers according to different target objects. In our project, we will use a classifier that considers the human face to recognize it as the target object. Haar Feature selection technique has a target to extract human face features. Haar features are like convolution kernels. These features are different permutations of black and white rectangles. In each feature calculation, we find the sum of pixels under white and black rectangles.

Anaconda Navigator

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository. It is available for Windows, macOS, and Linux. In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages and use multiple environments to separate these different versions. The command-line program conda is both a package manager and an environment manager. This helps data scientists ensure that each version of each package has all the dependencies it requires and works correctly. Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages, and update them – all inside Navigator.

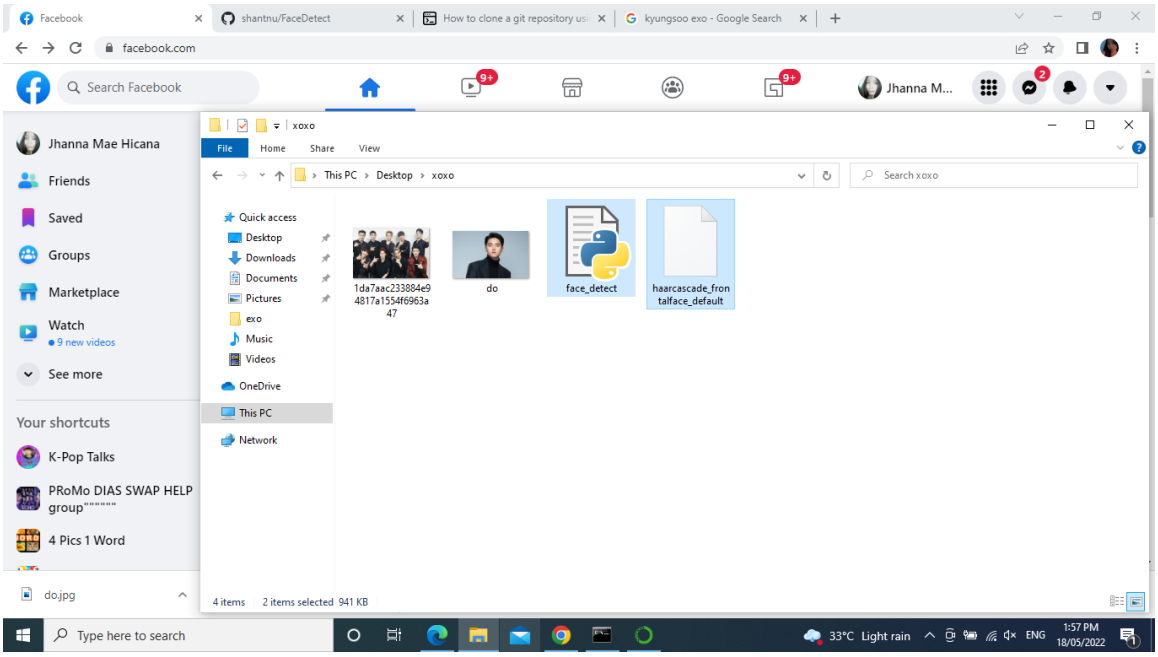
Steps to implement human face recognition with Python & OpenCV

1. **Create a new Folder**

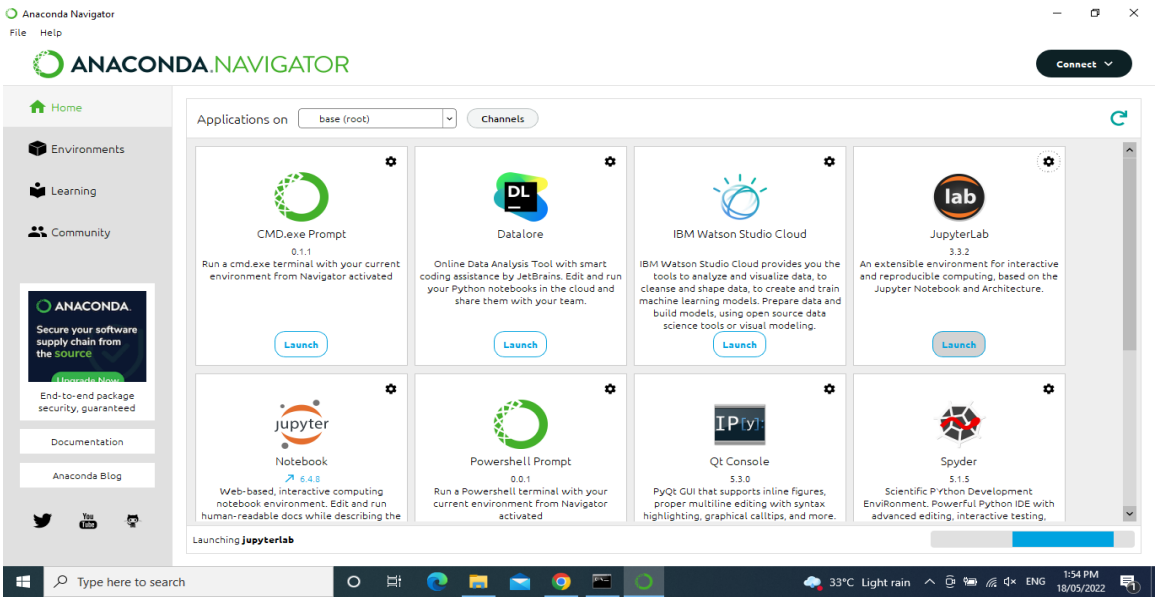


2. Create a new Folder

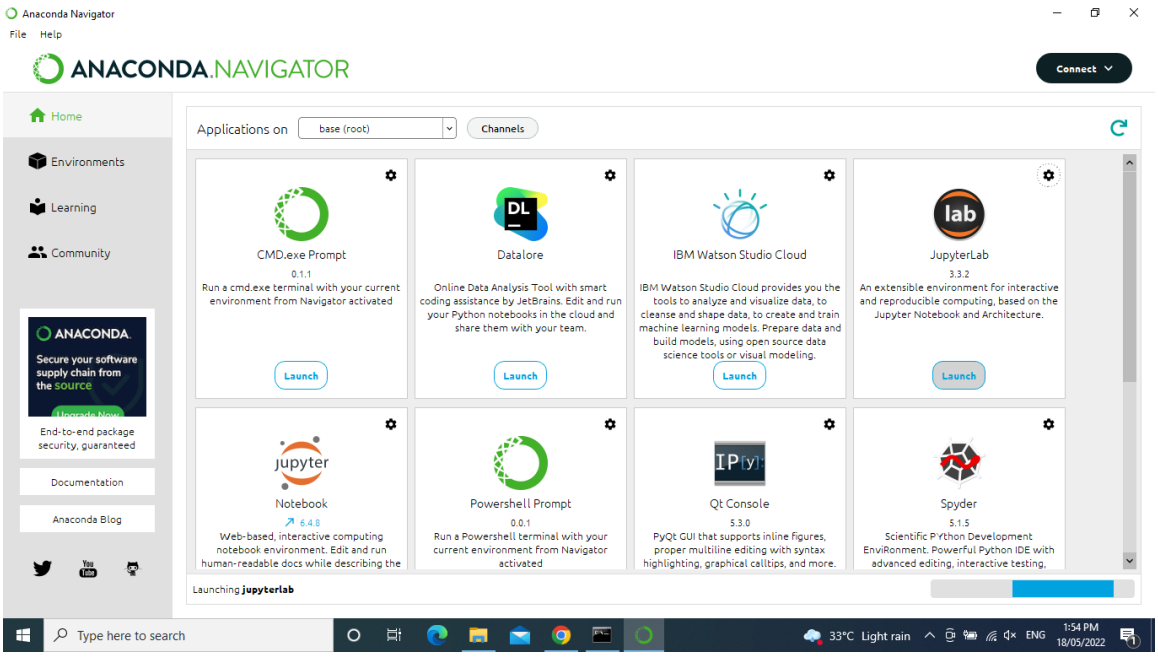
After creating a new folder, paste the copied files together with a single photo and a group photo.



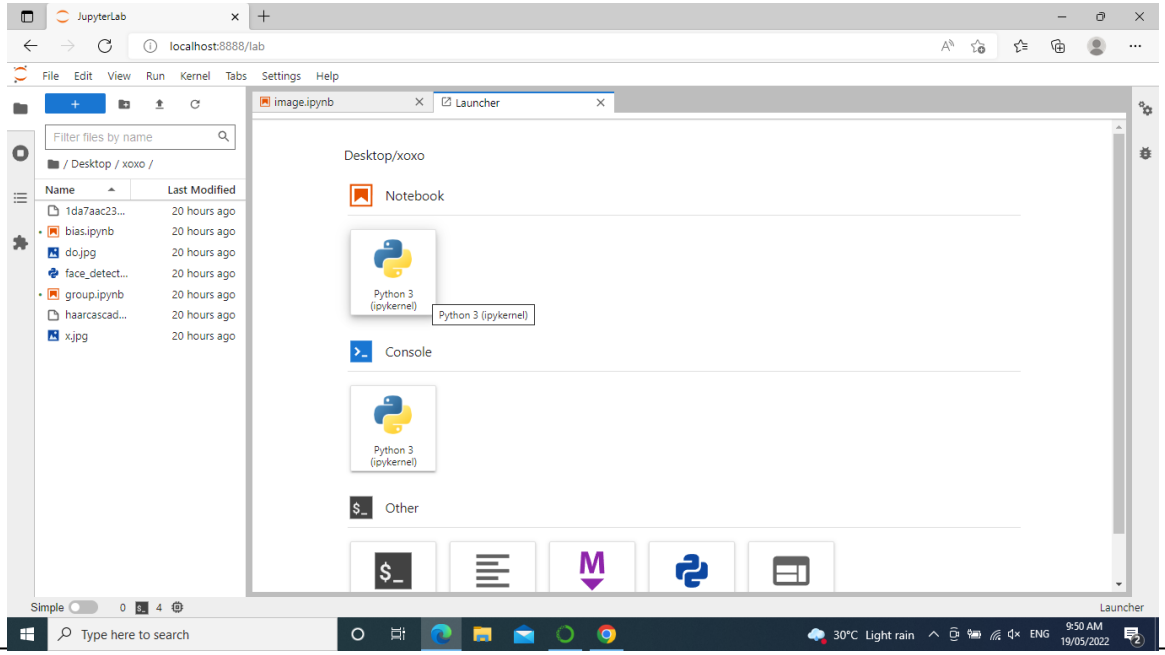
3. Open Anaconda Navigator



4. Launch Jupyter Lab

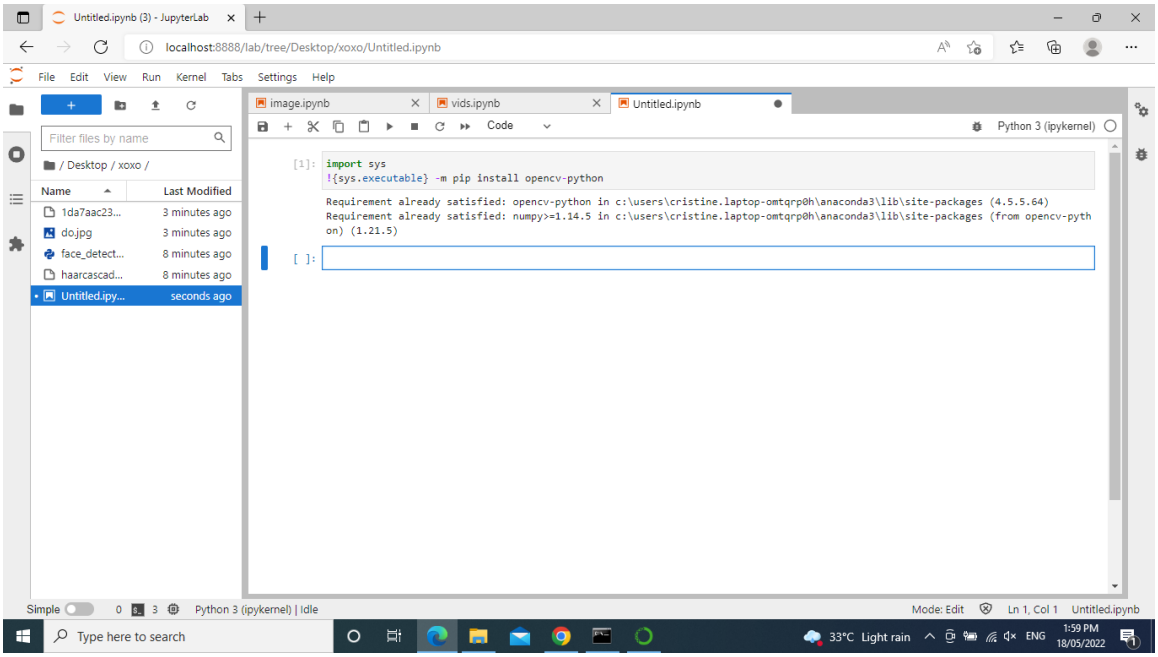


5. Create new Notebook



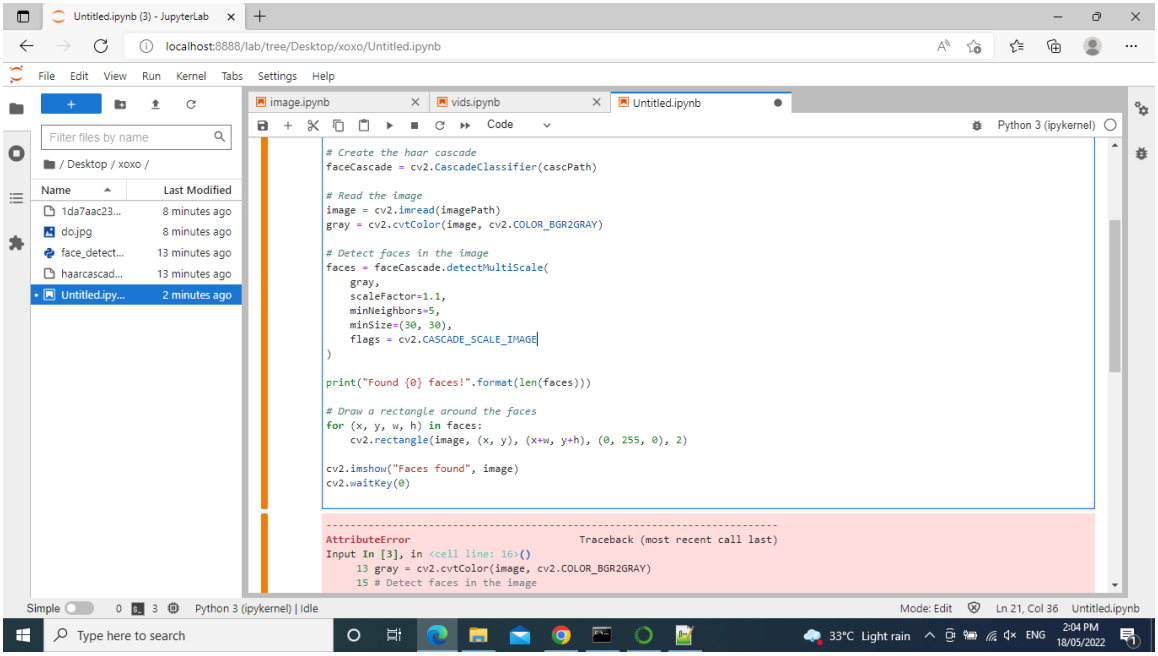
Single Photo Source Code

```
import sys!.executable} -m pip install opencv-python
```

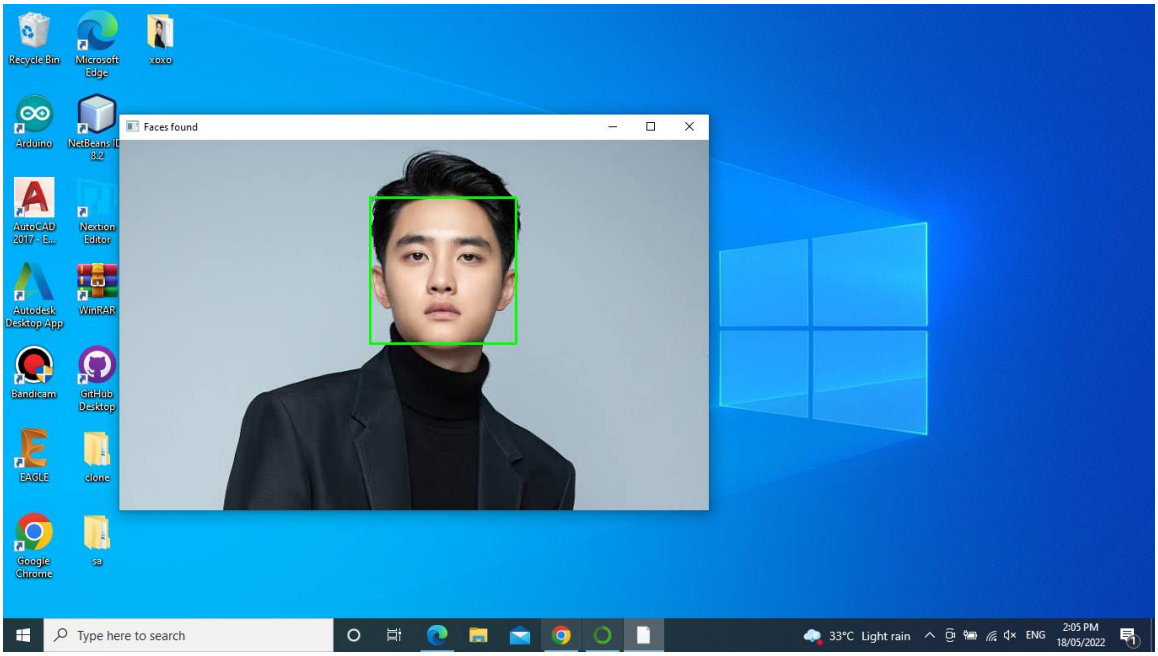



```
import cv2

import sys


# Get user supplied values

imagePath = “do.jpg”

cascPath = "haarcascade_frontalface_default.xml"


# Create the haar cascade

faceCascade = cv2.CascadeClassifier(cascPath)
```

```
# Read the image
image = cv2.imread(imagePath)
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)


# Detect faces in the image
faces = faceCascade.detectMultiScale(
    gray,
    scaleFactor=1.1,
    minNeighbors=5,
    minSize=(30, 30),
    flags = cv2.CASCADE_SCALE_IMAGE
)


print("Found {0} faces!".format(len(faces)))


# Draw a rectangle around the faces
for (x, y, w, h) in faces:
cv2.rectangle(image, (x, y), (x+w, y+h), (0, 255, 0), 2)
cv2.imshow("Faces found", image) cv2.waitKey(0)
```



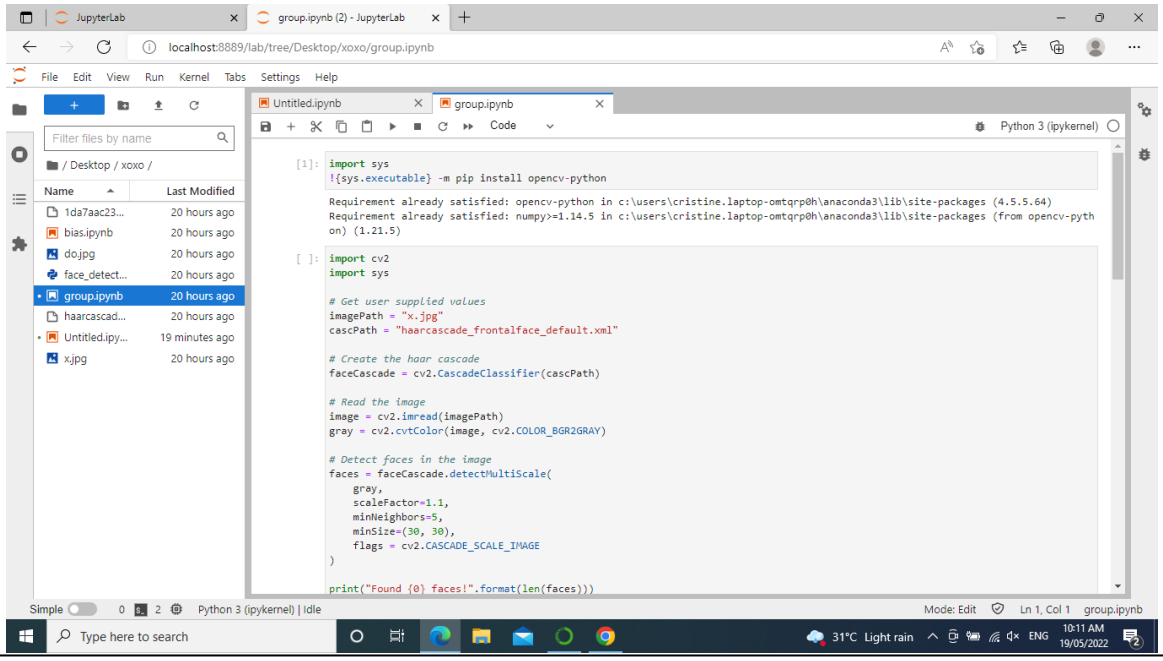
Debugging of Single Photo Code



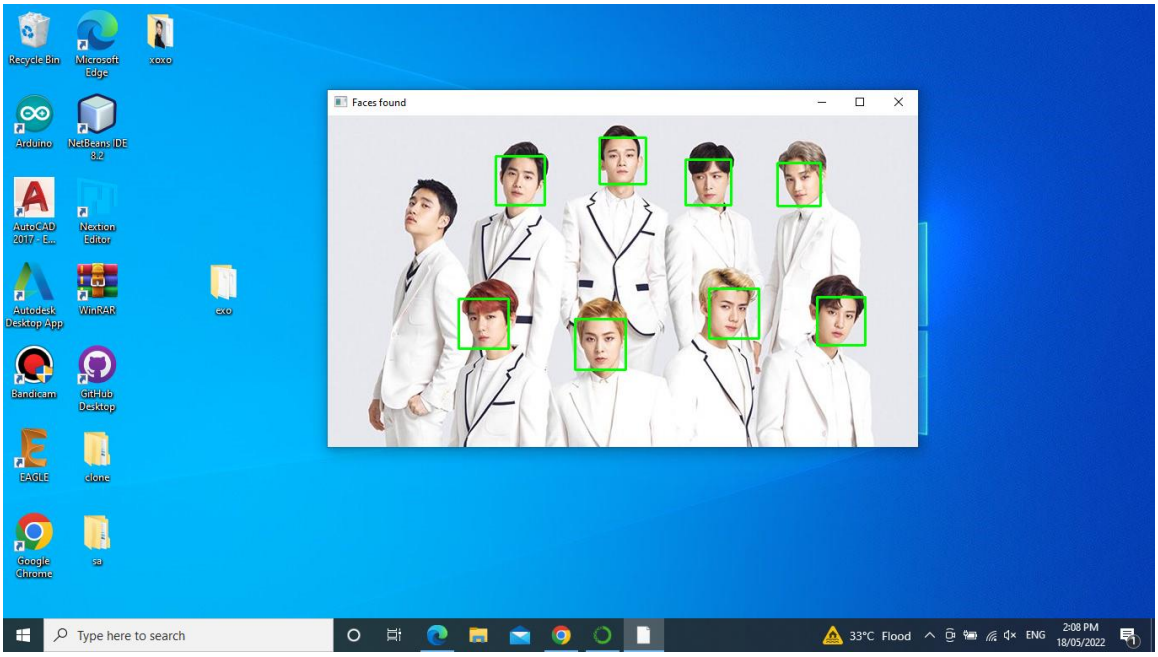
Source Code for Group Photo

```
import sys
!{sys.executable} -m pip install opencv-python
import cv2
import sys

# Get user supplied values
imagePath = "x.jpg"
cascPath = "haarcascade_frontalface_default.xml"

# Create the haar cascade

faceCascade = cv2.CascadeClassifier(cascPath)

# Read the image
image = cv2.imread(imagePath)
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Detect faces in the image
faces = faceCascade.detectMultiScale(
    gray,
    scaleFactor=1.1,
```

```
    minNeighbors=5,
    minSize=(30, 30),
    flags = cv2.CASCADE_SCALE_IMAGE
)

print("Found {0} faces!".format(len(faces)))

# Draw a rectangle around the faces
for (x, y, w, h) in faces:
    cv2.rectangle(image, (x, y), (x+w, y+h), (0, 255, 0), 2)

cv2.imshow("Faces found", image)
cv2.waitKey(0)
```



Debugging of Group Photo Source Code



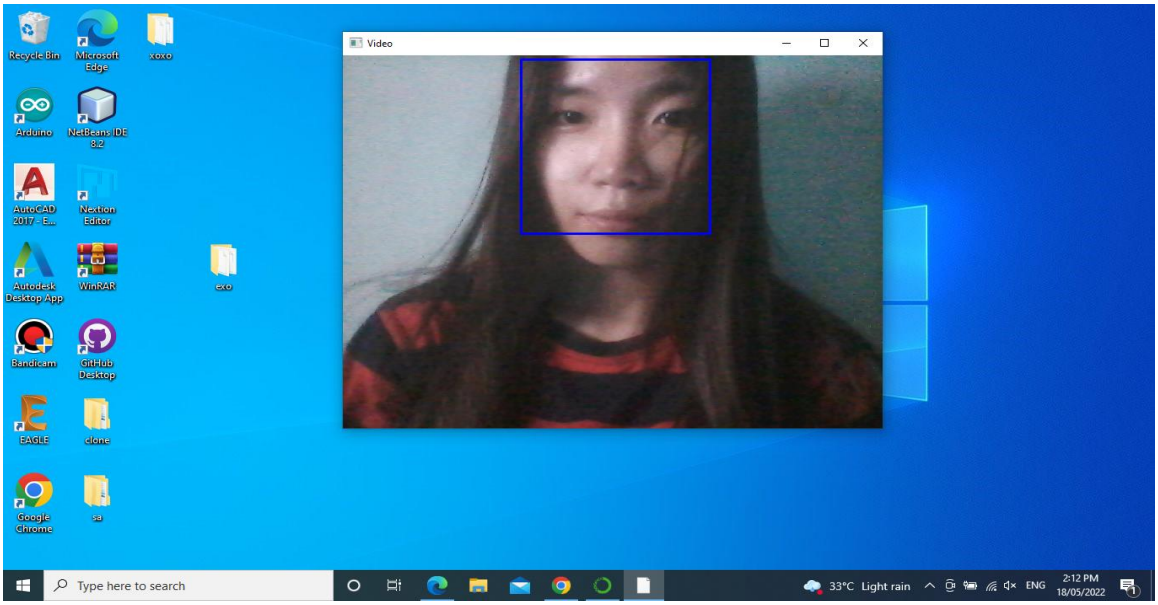
Real time Face Detection using a Laptop Camera

Source Code

```
import cv2


# Load the cascade
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')


# To capture video from webcam.
cap = cv2.VideoCapture(0)


while True:
    # Read the frame
```

```
    _, img = cap.read()

    # Convert to grayscale
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # Detect the faces
    faces = face_cascade.detectMultiScale(gray, 1.1, 4)

    # Draw the rectangle around each face
    for (x, y, w, h) in faces:

    cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)

    # Display
    cv2.imshow('Video', img)

    # Stop if escape key is pressed
    k = cv2.waitKey(30) & 0xff
    if k==27:
        break

# Release the VideoCapture object
cap.release()
```

Debugging


Video
Recycle Bin
Microsoft Edge
xoxo
Arduino
NetBeans IDE 8.2
AutoCAD 2017 - E...
Nextion Editor
Autodesk Desktop App
WinRAR
exo
Bandicam
GitHub Desktop
EAGLE
clone
Google Chrome
sa
Type here to search
33°C Light rain
ENG
2:12 PM
18/05/2022