# *Live Face Detection.py*

## *How it works?*

Facial recognition is made easy on python with the ability to download the package face_recognition. To get live detection working you first need a picture to compare for us this is a profile picture. This is taken with the following code.

```
capture =
cv2.VideoCapture(0)

                    while True:

                        ret, frame = capture.read()
                        cv2.imshow("Take profile picture", frame)

                        #Save on pressing "Spacebar" then exit
                        if(cv2.waitKey(1) & 0xFF == ord('q')): #picture is taken by
                    pressing q
                            cv2.imwrite("RegisterPhoto.jpg", frame)
                            cv2.destroyAllWindows()
                            break

                    capture.release()
```

This code takes a picture when the user presses the spacebar.

Once the picture is saved it will now open a live capture and using face_recognition package read in the picture just taken.

```
video_capture =
cv2.VideoCapture(0)

                    # Load a sample picture and learn how to recognize it.

                    Reg_Photo = face_recognition.load_image_file("RegisterPhoto.jpg")
                    Reg_Photo_encoding = face_recognition.face_encodings(Reg_Photo)[0]
```

Next we create and array of face encodings and the names of the users.

```
# Create
arrays of
known
face
encodings
and their
names
            known_face_encodings = [
             Reg_Photo_encoding

            ]
            known_face_names = [
                "Name of user goes here"

            ]

            # Initialize some variables
            face_locations = []
            face_encodings = []
            face_names = []
            process_this_frame = True
```

Next we resize the frame so it will work faster and change the image from BGR to RGB.

```
 small_frame =
cv2.resize(frame,
(0, 0), fx=0.25,
fy=0.25)

                    # Convert the image from BGR color (which OpenCV uses) to RGB
                color (which face_recognition uses)
                    rgb_small_frame = small_frame[:, :, ::-1]
```

Next, we can process the frames to find all the faces and face encodings in the live feed.

```
  face_locations =
face_recognition.face_locations(rgb_small_
frame)
                                                face_encodings =
                                                face_recognition.face_encodings(rgb_small_
                                                frame, face_locations)
```

Once the frames are processed, we can simply check to see if there is and matches.

```
  matches = face_recognition.compare_faces(known_face_encodings, face_encoding)

face_distances = face_recognition.face_distance(known_face_encodings, face_encoding)
 best_match_index = np.argmin(face_distances)

if
matches[best_match_index]:
                                name = known_face_names[best_match_index]

                                                face_names.append(name)
```

If we get a match, we will add the name from the array of known users.

Now that we have a match, we can display a box around the face with their name to make it clearer that the face has been found.

```
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)

                                    # 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()
```

This is all the code we need to recognise faces. This code however will be altered to add in images and names from a database.