FACE DETECTION

Certainly! I'll explain each line of the code:

pythonCopy code

# Importing necessary libraries

* import cv2, sys, numpy, os
* **cv2**: OpenCV library for computer vision tasks.
* **sys**: System-specific parameters and functions.
* **numpy**: Numerical computing library.
* **os**: Operating system dependent functionality.

# Haar cascade file for face detection

* haar\_file = 'haarcascade\_frontalface\_default.xml'
* **haar\_file**: Path to the Haar cascade classifier XML file for face detection.

# Folder where face images will be stored

* datasets = 'datasets'
* **datasets**: Name of the folder where face images will be stored.

# Subfolder name for the person's data

* sub\_data = 'vivek'
* **sub\_data**: Subfolder name under the 'datasets' folder, representing the person's data.

# Constructing the path to the subfolder

* path = os.path.join(datasets, sub\_data)
* **path**: Complete path to the subfolder where face images will be stored.

# Create the subfolder if it doesn't exist

* if not os.path.isdir(path):

os.mkdir(path)

* Check if the subfolder exists; if not, create it.

# Size of the images

* (width, height) = (130, 100)
* **width** and **height**: Dimensions for resizing the captured face images.

# Initializing the webcam (camera)

* face\_cascade = cv2.CascadeClassifier(haar\_file)

webcam = cv2.VideoCapture(0)

* **face\_cascade**: Creating a Haar cascade classifier object for face detection.
* **webcam**: Initializing the webcam for capturing video frames.

# Loop to capture 30 images of the face

* count = 1

while count < 30:

(\_, im) = webcam.read()

gray = cv2.cvtColor(im, cv2.COLOR\_BGR2GRAY)

faces = face\_cascade.detectMultiScale(gray, 1.3, 4)

* **while** loop: Continuously captures images until 30 images of the face are captured.
* **(\_, im)**: Reads a frame from the webcam, and **im** contains the image.
* **gray**: Converts the image to grayscale for face detection.
* **faces**: Detects faces in the grayscale image.
* for (x, y, w, h) in faces:

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

face = gray[y:y + h, x:x + w]

face\_resize = cv2.resize(face, (width, height))

cv2.imwrite('%s/%s.png' % (path, count), face\_resize)

* Loop over detected faces: Draws rectangles around the faces in the original image (**im**), extracts the face region (**face**), resizes it, and saves it as an image in the specified folder.
* count += 1

cv2.imshow('OpenCV', im)

key = cv2.waitKey(10)

if key == 27:

break

* Updates the count, displays the original image with rectangles around faces, waits for a key press, and breaks the loop if the 'Esc' key (27) is pressed.

This code captures 30 images of the face from the webcam feed, detects faces using the Haar cascade classifier, and saves the resized face images in a specified folder. The process stops when 30 images are captured or the 'Esc' key is pressed.