Training

**####################################################**

**# Modified by pavan kumar #**

**# All right reserved to the respective owner #**

**####################################################**

**# Import OpenCV2 for image processing**

**# Import os for file path**

**import cv2, os**

**# Import numpy for matrix calculation**

**import numpy as np**

**# Import Python Image Library (PIL)**

**from PIL import Image**

**import os**

**def assure\_path\_exists(path):**

**dir = os.path.dirname(path)**

**if not os.path.exists(dir):**

**os.makedirs(dir)**

**# Create Local Binary Patterns Histograms for face recognization**

**recognizer = cv2.face.LBPHFaceRecognizer\_create()**

**# Using prebuilt frontal face training model, for face detection**

**detector = cv2.CascadeClassifier("haarcascade\_frontalface\_default.xml");**

**# Create method to get the images and label data**

**def getImagesAndLabels(path):**

**# Get all file path**

**imagePaths = [os.path.join(path,f) for f in os.listdir(path)]**

**# Initialize empty face sample**

**faceSamples=[]**

**# Initialize empty id**

**ids = []**

**# Loop all the file path**

**for imagePath in imagePaths:**

**# Get the image and convert it to grayscale**

**PIL\_img = Image.open(imagePath).convert('L')**

**# PIL image to numpy array**

**img\_numpy = np.array(PIL\_img,'uint8')**

**# Get the image id**

**id = int(os.path.split(imagePath)[-1].split(".")[1])**

**# Get the face from the training images**

**faces = detector.detectMultiScale(img\_numpy)**

**# Loop for each face, append to their respective ID**

**for (x,y,w,h) in faces:**

**# Add the image to face samples**

**faceSamples.append(img\_numpy[y:y+h,x:x+w])**

**# Add the ID to IDs**

**ids.append(id)**

**# Pass the face array and IDs array**

**return faceSamples,ids**

**# Get the faces and IDs**

**faces,ids = getImagesAndLabels('dataset')**

**# Train the model using the faces and IDs**

**recognizer.train(faces, np.array(ids))**

**# Save the model into trainer.yml**

**assure\_path\_exists('trainer/')**

**recognizer.save('trainer/trainer.yml')**