Part1:

*import* cv2  
*import* numpy *as* np  
face\_classifier=cv2.CascadeClassifier('C:/python37/Lib/site-packages/cv2/data/haarcascade\_frontalface\_default.xml')  
  
  
*def* face\_extractor(*img*):  
  
 gray = cv2.cvtColor(*img*, cv2.COLOR\_BGR2GRAY)  
 faces = face\_classifier.detectMultiScale(gray,1.3,5)  
  
 *if* faces *is*():  
 *return None  
  
 for* (x,y,w,h) *in* faces:  
 cropped\_face = *img*[y:y+h, x:x+w]  
  
 *return* cropped\_face  
  
cap = cv2.VideoCapture(0)  
  
count = 0  
*while True*:  
 ret, frame = cap.read()  
 *if* face\_extractor(frame) *is not None*:  
 count += 1  
 face = cv2.resize(face\_extractor(frame), (200, 200))  
 face = cv2.cvtColor(face, cv2.COLOR\_BGR2GRAY)  
  
 file\_name\_path = 'E:/face data/user' + str(count) + '.jpg'  
 cv2.imwrite(file\_name\_path,face)  
  
 cv2.putText(face,str(count),(50,50),cv2.FONT\_HERSHEY\_COMPLEX,1,(0,255,0),2)  
 cv2.imshow('Face Cropper',face)  
  
 *else*:  
 print("Face not found")  
 *pass  
 if* cv2.waitKey(1)==13 *or* count==100:  
 *break*cap.release()  
cv2.destroyAllWindows()  
print('collecting samples complete!!!')

part2:

*import* cv2  
*import* numpy *as* np  
*from* os *import* listdir # use to fetch data from directory.  
*from* os.path *import* isfile, join  
  
data\_path = 'E:/face data/'  
onlyfiles = [f *for* f *in* listdir(data\_path) *if* isfile(join(data\_path,f))]  
  
training\_data, labels = [], []  
  
*for* i ,files *in* enumerate(onlyfiles):  
 image\_path = data\_path + onlyfiles[i]  
 images = cv2.imread(image\_path, cv2.IMREAD\_GRAYSCALE)  
 training\_data.append(np.asarray(images, dtype=np.uint8))  
 labels.append(i)  
  
labels = np.asarray(labels, dtype=np.int32)  
  
model = cv2.face.LBPHFaceRecognizer\_create()  
  
model.train(np.asarray(training\_data), np.asarray(labels))  
  
print('Model Training Complete!!!!')  
  
face\_classifier=cv2.CascadeClassifier('C:/python37/Lib/site-packages/cv2/data/haarcascade\_frontalface\_default.xml')  
  
*def* face\_detector(*img*, size = 0.5):  
 gray = cv2.cvtColor(*img*, cv2.COLOR\_BGR2GRAY)  
 faces = face\_classifier.detectMultiScale(gray,1.3,5)  
  
 *if* faces *is*():  
 *return img*,[]  
  
 *for*(x,y,w,h) *in* faces:  
 cv2.rectangle(*img*, (x,y),(x+w,y+h),(0,255,255),2)  
 roi = *img*[y:y+h, x:x+w]  
 roi = cv2.resize(roi, (200,200))  
  
 *return img*,roi  
  
cap = cv2.VideoCapture(0)  
*while True*:  
  
 ret, frame = cap.read()  
  
 image, face = face\_detector(frame)  
  
 *try*:  
 face = cv2.cvtColor(face, cv2.COLOR\_BGR2GRAY)  
 result = model.predict(face)  
  
 *if* result[1] < 500:  
 confidence = int(100\*(1-(result[1])/300))  
 display\_string = str(confidence)+'% Confidence it is user'  
 cv2.putText(image, display\_string,(100,120), cv2.FONT\_HERSHEY\_COMPLEX,1,(250,125,255), 2)  
  
 *if* confidence > 75:  
 cv2.putText(image, "Unlocked", (250, 450), cv2.FONT\_HERSHEY\_COMPLEX, 1, (0, 255, 0), 2)  
 cv2.imshow('Face Cropper', image)  
  
 *else*:  
 cv2.putText(image, "Locked", (250, 450), cv2.FONT\_HERSHEY\_COMPLEX, 1, (0, 0, 255), 2)  
 cv2.imshow('Face Cropper', image)  
  
  
 *except*:  
 cv2.putText(image, "Face Not Found", (250, 450), cv2.FONT\_HERSHEY\_COMPLEX, 1, (255, 0, 0), 2)  
 cv2.imshow('Face Cropper', image)  
 *pass  
  
 if* cv2.waitKey(1)==13:  
 *break*cap.release()  
cv2.destroyAllWindows()