

```

18. with open("labels.pickle", 'wb') as f:
19.     pickle.dump(label_ids, f)
20.
21. #print(id_)
22. recognizer.train(x_train, np.array(y_labels))
23. recognizer.save("trainer2.yml")

```

The second module, first add eye,face,mouth haarcascade detection which will be used to make border around your face in live webcam. Then add LBPH face recognizer function and make variable to load data train.

```

24. face_cascade = cv2.CascadeClassifier('cascades/data/haarcascade_frontalface_alt2.xml')
25. eye_cascade = cv2.CascadeClassifier('cascades/data/haarcascade_eye.xml')
26. smile_cascade = cv2.CascadeClassifier('cascades/data/haarcascade_smile.xml')
27. recognizer = cv2.face.LBPHFaceRecognizer_create()
28. recognizer2 = cv2.face.LBPHFaceRecognizer_create()
29. recognizer3 = cv2.face.LBPHFaceRecognizer_create()
30. recognizer.read("trainer.yml")
31. recognizer2.read("trainer2.yml")
    recognizer3.read("trainer3.yml")

```

The faces variable is used to detect face when the program detect a face. If the detected face is recognized from data train, the value of the variable will be replaced with the name. This process applied to eye and mouth as well. The difference between them is on the data train file and haarcascade detection.

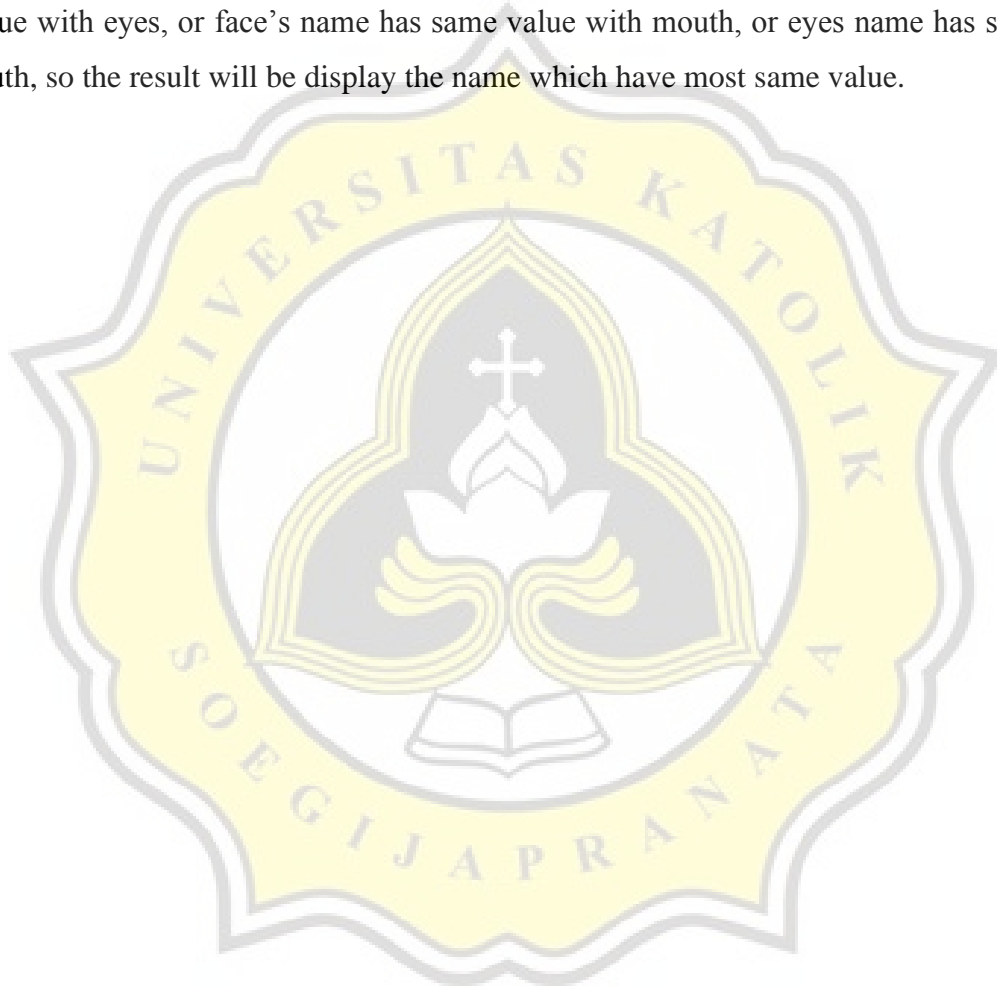
```

32. while(True):
33.     ret, frame = cap.read()
34.     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
35.     faces = face_cascade.detectMultiScale(gray, scaleFactor=1.5,
    minNeighbors=5)
36.     for (x, y, w, h) in faces:
37.         roi_gray = gray[y:y+h, x:x+w]
38.         roi_color = frame[y:y+h, x:x+w]
39.         id_, conf = recognizer.predict(roi_gray)
40.         if conf>=45 and conf <=85:
41.             font = cv2.FONT_HERSHEY_SIMPLEX
42.             name = labels[id_]
43.             color = (255,255,255)
44.             stroke = 2
45.             cv2.putText(frame, name, (x,y), font, 1, color, stroke,
    cv2.LINE_AA)
46.             Face = ('Face : ' + name)
47.             Faces = name
48.             Combine = ('User Detected : ' + name)

```

```
49.  
50.     img_item = '1.png'  
51.     cv2.imwrite(img_item, roi_color)  
52.  
53.     color = (255,0,0) #BGR 0-255  
54.     stroke = 2  
55.     width = x+w  
56.     height = y+h  
     cv2.rectangle(frame, (x,y), (width, height), color, stroke)
```

To count the accuracy, each of detection method is compared. If detected face's name has same value with eyes, or face's name has same value with mouth, or eyes name has same value with mouth, so the result will be display the name which have most same value.



5.2. Results

Result With Low Resolution Dataset

Name	Face	Mouth	Eyes	Detection	Accuracy
Subject A	v	v	x	v	66%
Subject B	x	x	v	X	33%
Subject C	x	x	x	x	0%
Subject D	x	x	x	x	0%
Subject E	x	x	x	x	0%

Table 5.1. Table Result Low Resolution Dataset

Result With High Resolution Dataset

Name	Face	Mouth	Eyes	Detection	Accuracy
Subject A	v	v	v	v	100%
Subject B	x	v	v	v	66%
Subject C	v	v	x	v	66%
Subject D	v	x	v	v	66%
Subject E	x	v	v	v	66%

Table 5.2. Table Result High Resolution Dataset

Obstruction	Face	Mouth	Eye	Detection	Accuracy
Glasses	v	v	x	v	66%
Mask	x	x	x	x	0%
Hoodie	v	v	v	v	100%

Table 5.3. Table Result Obstruction Test

