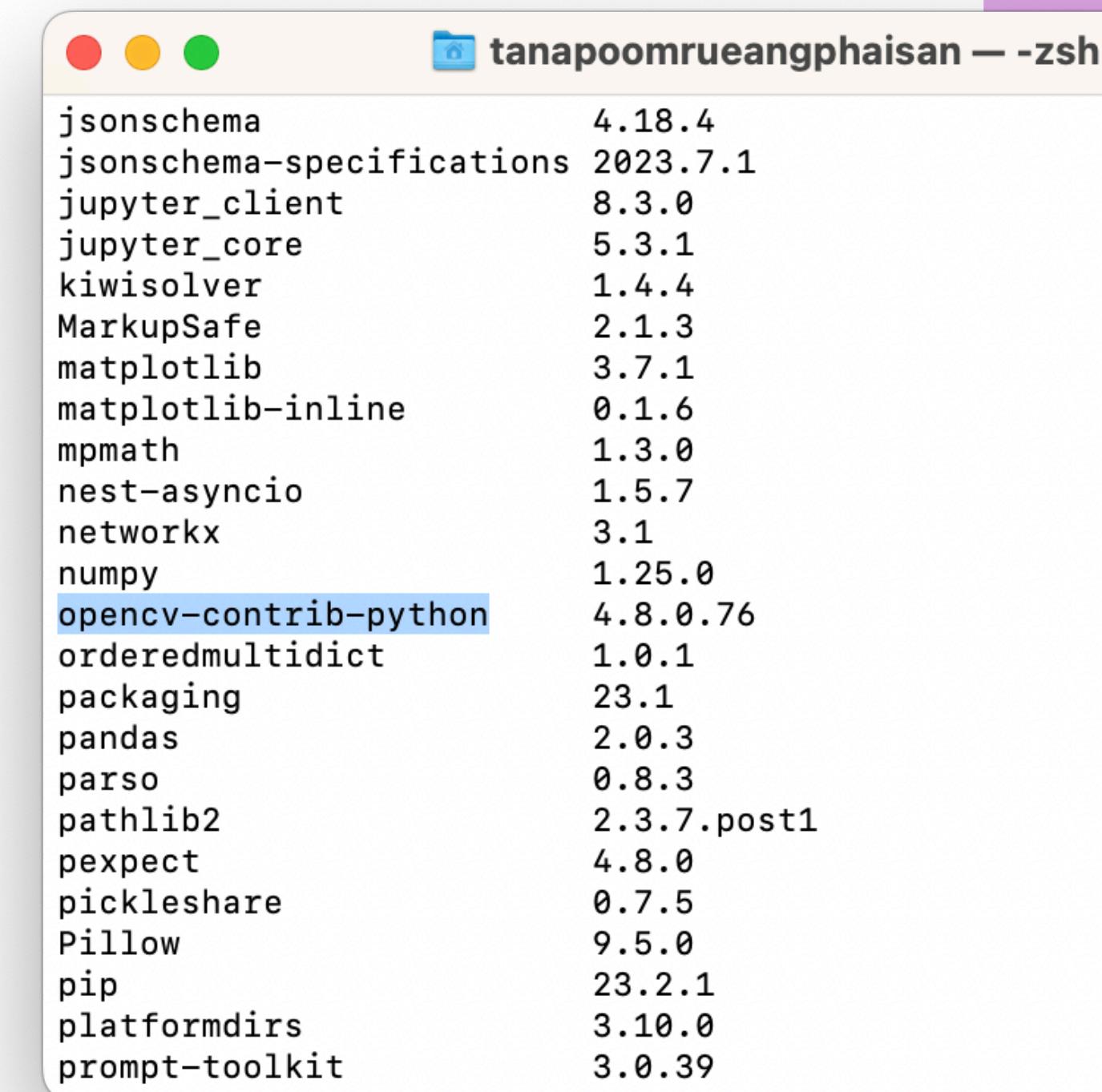


ໂປຣແກຣມຈົດຈໍາໃບໜັ້ງ

- ในการที่เราจะใช้คำสั่งในการ Training ให้นั้นเราต้องใช้ **opencv-contrib-python**
- การลบ opencv
 - pip uninstall opencv-python
- การลง opencv-contrib
 - pip install opencv-contrib-python



A screenshot of a terminal window titled "tanapoomrueangphaisan --zsh". The window displays a list of Python packages and their versions. The package "opencv-contrib-python" is highlighted with a blue selection bar. The list includes:

Package	Version
jsonschema	4.18.4
jsonschema-specifications	2023.7.1
jupyter_client	8.3.0
jupyter_core	5.3.1
kiwisolver	1.4.4
MarkupSafe	2.1.3
matplotlib	3.7.1
matplotlib-inline	0.1.6
mpmath	1.3.0
nest-asyncio	1.5.7
networkx	3.1
numpy	1.25.0
opencv-contrib-python	4.8.0.76
orderedmultidict	1.0.1
packaging	23.1
pandas	2.0.3
parso	0.8.3
pathlib2	2.3.7.post1
pexpect	4.8.0
pickleshare	0.7.5
Pillow	9.5.0
pip	23.2.1
platformdirs	3.10.0
prompt-toolkit	3.0.39

โปรแกรมตรวจจับใบหน้า

- โปรแกรมตรวจจับใบหน้า มีหลักการเขียนโปรแกรมดังต่อไปนี้
 - ตรวจจับหน้า(Face detection) โดยใช้ haarcascades
 - สร้าง dataset หรือ การสร้างไฟล์ภาพในการจัดจำใบหน้า
 - ทำการ Training และสร้างไฟล์ xml เพื่อนำมาตรวจสอบ
 - นำไฟล์ xml ที่ได้ไปทำการตรวจสอบใบหน้า

ตรวจจับหน้า(Face detection)

- ตรวจจับหน้า(Face detection) โดยใช้ haarcascades โดยโปรแกรมนี้พัฒนาจากโปรแกรมเดิมโดยเขียนโปรแกรมแบ่งพังก์ชันการทำงานออกเป็น 3 ส่วน คือ
 - 1.ส่วนในการตรวจจับและวาดเส้น `def draw_boundary(img,text):`
 - 2.นำค่ามาใช้งาน `def detect(img):`
 - 3.ส่วนการแสดงผล `while True :`

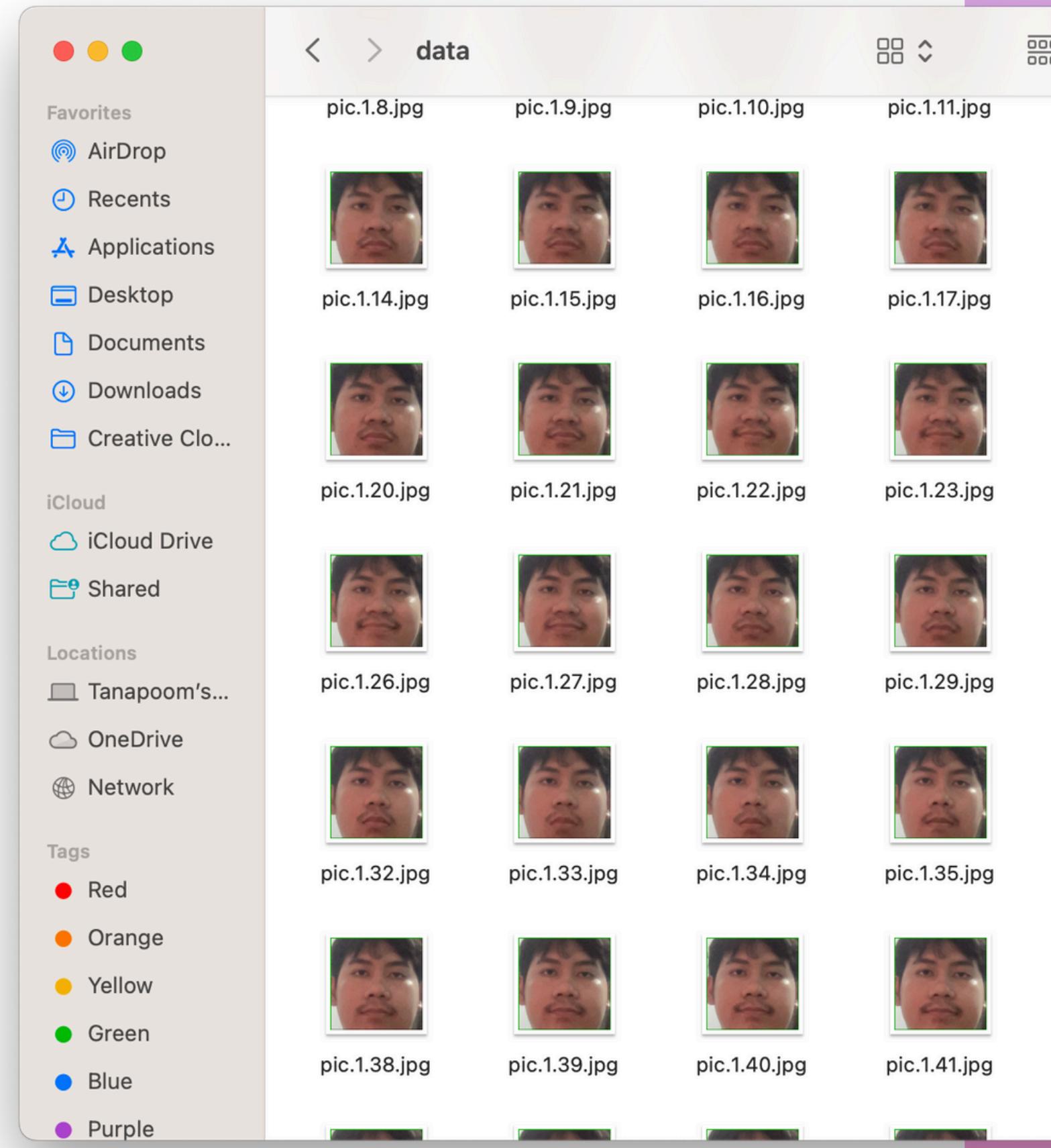


```
1 import cv2 as cv
2
3 cap=cv.VideoCapture(0)
4
5 face_cascade=cv.CascadeClassifier("opencv-master/data/haarcascades/haarcascade_
6 _____
7 def draw_boundary(img,text):
8     gray_img = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
9     face_detect=face_cascade.detectMultiScale(gray_img,1.1,5)
10    for(x,y,w,h) in face_detect :
11        cv.rectangle(img,(x,y),(x+w,y+h),(0,255,0),2)
12        cv.putText(img,text,(x-10,y-10),cv.FONT_HERSHEY_SIMPLEX,2,(0,0,255),3)
13    return img
14 ====
15 def detect(img):
16     img=draw_boundary(img,'face')
17     return img
18 _____
```

```
9  face_cascade = cv.CascadeClassifier('haarcascade_frontalface_default.xml')
10
11 def draw_boundary(img, classifier, color=(255, 0, 0), thickness=3):
12     faces = classifier.detectMultiScale(img, scaleFactor=1.1, minNeighbors=5)
13
14     for(x,y,w,h) in faces:
15         cv.rectangle(img,(x,y),(x+w,y+h),(0,255,0),2)
16         cv.putText(img,"Face", (x-10,y-10),cv.FONT_HERSHEY_SIMPLEX, 2, (0,0,255), 3)
17
18     return img
19
20
21
22
23
24
25
26
27
```

สร้าง dataset

- ทำ dataset หรือการสร้าง ไฟล์ภาพในการจดจำใบหน้าบุคคลที่กำหนด โดยการสร้างฟังก์ชัน `create_dataset` โดยการสร้างไฟล์ภาพขึ้นมาโดยระบุ ค่า 2 ตัวแปร คือ id และ จำนวนของภาพเพื่อมาทำการ train



```
● ● ●
```

```
1 import cv2 as cv
2
3 cap=cv.VideoCapture(0)
4 face_cascade=cv.CascadeClassifier("opencv-master/data/haarcascades/haarcascade_f
5
6 def create_dataset(img,id,img_id):
7     cv.imwrite("data/pic.%s.%s.jpg"%(id,img_id),img)
8 def draw_boundary(img,text):
9     gray_img = cv.cvtColor(img,cv.COLOR_BGR2GRAY)
10    face_detect=face_cascade.detectMultiScale(gray_img,1.1,5)
11    xywh = []
12    for(x,y,w,h) in face_detect :
13        cv.rectangle(img,(x,y),(x+w,y+h),(0,255,0),2)
14        cv.putText(img,text,(x-10,y-10),cv.FONT_HERSHEY_SIMPLEX,2,(0,0,255),3)
15        xywh = [x,y,w,h]
16    return img ,xywh
17
18 def detect(img,img_id):
19     img,xywh = draw_boundary(img,"face")
20     id = 1
```

```
16     return img ,xywh
17
18 def detect(img,img_id):
19     img,xywh = draw_boundary(img,"face")
20     id = 1
21     if len(xywh) == 4 :
22         result = img[xywh[1]:xywh[1]+xywh[3],xywh[0]:xywh[0]+xywh[2]]
23         create_dataset(result,id,img_id)
24     return img
25
26 img_id = 1
27
28 while True :
29     check , frame = cap.read()
30     frame = detect(frame,img_id)
31     cv.imshow("output",frame)
32     img_id += 1
33     if cv.waitKey(40)& 0xff ==ord("q"):
34         break
35
36 cap.release()
37 cv.destroyAllWindows
```

ทำการ Training และสร้างไฟล์ xml

เขียนโปรแกรม Training ใบหน้า เพื่อสร้างไฟล์ xml เพื่อนำมาใช้ในการตรวจสอบ ใบหน้า ที่ได้สร้าง dataset ไว้โดยโปรแกรมนี้จะเขียนแยกจาก โปรแกรมเดิมโดย มีส่วนต่างๆดังนี้

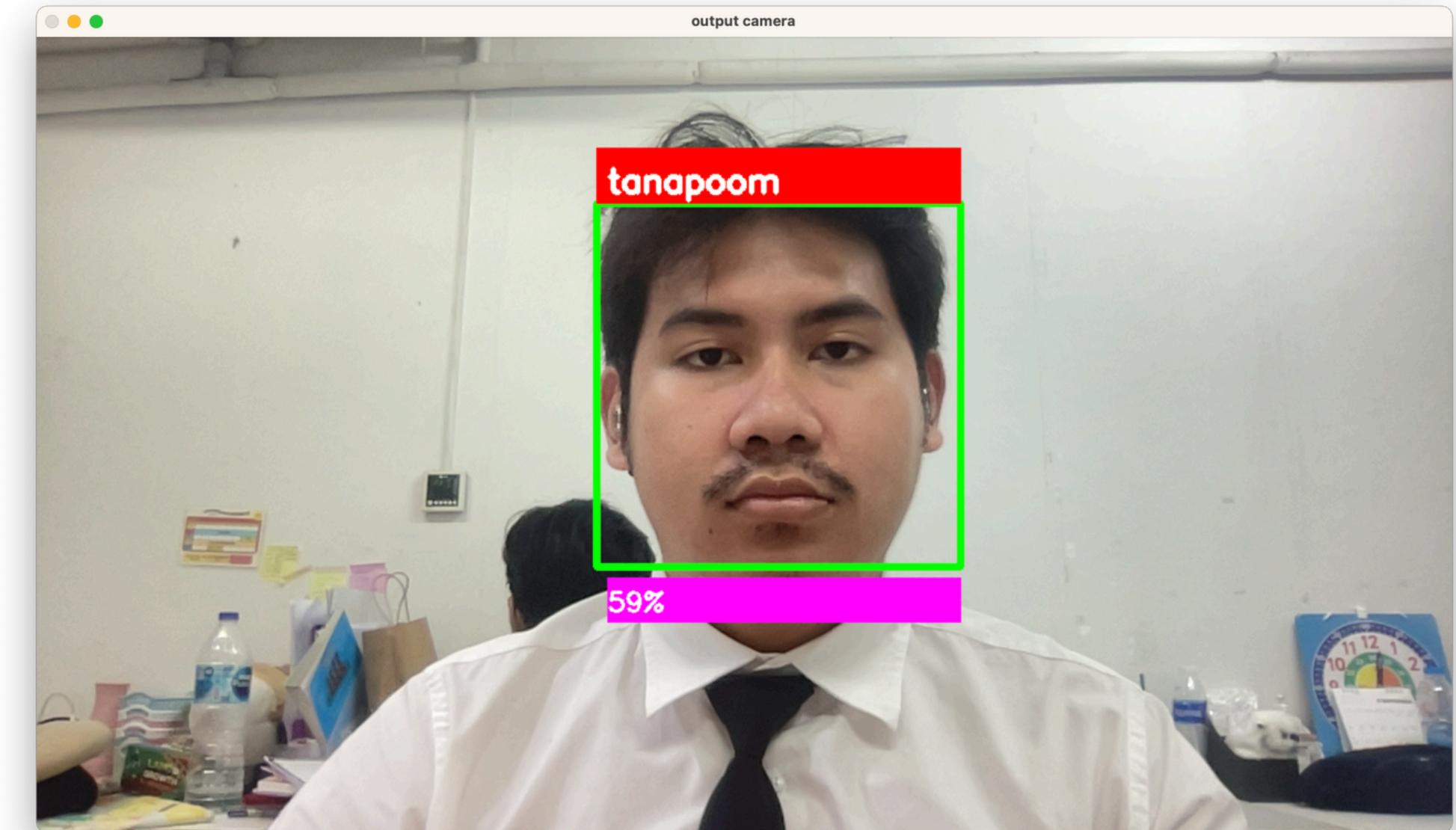
- ส่วนการ import library โดยต้องติดตั้ง library pillown, numpy, opencv-contrib-python ซึ่งเป็น algorithms ในการ train การจดจำใบหน้า
- สร้างฟังก์ชัน ในการ train ใบหน้า โดยการปรับค่าตัวแปร data_dir คือตำแหน่ง การเก็บภาพ
- กำหนด path หรือตัวแปรเส้นทางในตัวแปร list

ทำการ Training และสร้างไฟล์ xml

- การทำงานวนซ้ำ จากข้อมูลใน path โดยให้ทำการเปิดภาพและอ่านภาพเป็น grayscale เพื่อประมวลผลได้ง่าย และแปลงไฟล์ภาพเป็น numpy array เพื่อง่ายต่อการประมวลผลและนำมาเรียงต่อกันใน ตัวแปร faces หรือภาพใบหน้าเรียงต่อกันเป็นชุดข้อมูล รวมถึง ข้อมูล id ที่นำมาต่อกันในตัวแปร ids
- การนำภาพมา train และนำไปเขียนลงในไฟล์ xml ที่ต้องการ

```
1 import numpy as np
2 from PIL import Image
3 import os
4 import cv2
5
6 def train_classifier(data_dir):
7     path = [os.path.join(data_dir,f) for f in os.listdir(data_dir)]
8     faces = []
9     ids = []
10    for image in path:
11        if image == 'data/.DS_Store':
12            continue
13        img = Image.open(image).convert("L")
14        imageNp = np.array(img,'uint8')
15        id = int(os.path.split(image)[1].split(".")[1])
16        faces.append(imageNp)
17        ids.append(id)
18        print(type(ids))
19        ids = np.array(ids)
20    clf = cv2.face.LBPHFaceRecognizer_create()
21    print(clf)
22    clf.train(faces,ids)
23    clf.write("classifier.xml")
24
25 train_classifier("data")
```

นำไฟล์ xml ที่ได้ไปทำการ ตรวจสอบ ใบหน้า



```
● ● ●
```

```
1 def draw_boundary(img,clf):
2     gray_img = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
3     face_detect=face_cascade.detectMultiScale(gray_img,1.1,5)
4     xywh = []
5     for (x,y,w,h) in face_detect :
6         cv2.rectangle(img, (x,y), (x+w, y+ h), (0,255,0),5)
7         cv2.rectangle(img,(x,y-50),(x+w,y),(0,0,255), -1)
8         id, con = clf.predict(gray_img[y:y+h,x:x+w])
9         if con <= 50 :
10             cv2.putText(img, "tanapoom", (x+10, y-10), cv2. FONT_HERSHEY_SIMPLEX, 1, (255,255, 255), 3)
11         else :
12             cv2.putText(img, "unknow", (x+10, y-10), cv2. FONT_HERSHEY_SIMPLEX, 2, (255,255, 255), 3)
13             show_con ="{0}%.format(round(100-con))
14             cv2.rectangle(img, (x+10,y+h+10), (x+w,y+h+50), (255,0,255), -1)
15             cv2.putText(img, show_con, (x+10,y+h+40), cv2.FONT_HERSHEY_SIMPLEX,0.8, (255,255,255), 2)
16             xywh=[x,y,w,h]
17     return img,xywh
```

```
● ● ●  
1 def detect(img,img_id,clf) :  
2     img,xywh=draw_boundary(img, clf)  
3     if len(xywh) == 4 :  
4         result=img[xywh[1]:xywh[1 ]+xywh[3],xywh[0]:xywh[0]+xywh[2]]  
5     return img  
6  
7 clf=cv2.face.LBPHFaceRecognizer_create()  
8 clf.read("classifier.xml")
```



```
1 while (True):
2     check, frame=cap.read()
3     frame=detect(frame,img_id,clf)
4     cv2.imshow("output camera",frame)
5     img_id +=1
6     if cv2.waitKey(1) & 0xFF == ord("g"):
7         break
8 cap.release()
9 cv2.destroyAllWindows()
```

งาน

ให้นักเรียนสร้างโปรแกรมเช็คชื่อเข้าเรียนในวิชา image processing และจะต้องมีนักเรียนในห้องเรียนของโปรแกรมที่นักเรียนทำอย่างน้อย 3 คนและจะต้องทั้งบันทึกใบหน้าตอนที่สแกน และลงเวลาไว้ในรูปด้วย