การเขียนโปรแกรมวิเคราะห์ใบหน้า ด้วย OpenCV



Face Recognition

ระบบการรู้จำใบหน้าหรือระบบการจดจำใบหน้า คือ ระบบการตรวจหาใบหน้าของมนุษย์และปรับภาพใบหน้าโดยอัตโนมัติ กรอบจะ ปรากฏขึ้นบนใบหน้าที่ถูกตรวจจับ และโฟกัส สี และค่าการวัดแสงจะถูกปรับโดยอัตโนมัติ นอกจากนั้นเมื่อบันทึกด้วยคุณภาพแบบ HD เทคโนโลยีการบีบอัดจะจัดสรรความจุของข้อมูลให้ลดลง แต่ได้ข้อมูลที่เป็นประโยชน์มากขึ้นเพื่อปรับคุณภาพของภาพ ข้อมูลที่ได้จะถูก นำไปเปรียบเทียบกับข้อมูลตัวอย่างที่เก็บบันทึกไว้ อาจจะทั้งใบหน้าหรือเพียงบางส่วน ขึ้นอยู่กับชนิดของวิธีแยกเอกลักษณ์ใบหน้า ระบบ การรู้จำใบหน้าเป็นส่วนหนึ่งของเทคโนโลยีปัญญาประดิษฐ์ ในส่วนเนื้อหาของเรื่อง การรับรู้ของเครื่อง (Machine perception)

ระบบวิเคราะห์ใบหน้าถือว่าเป็นหนึ่งในระบบที่ใช้ในการพิสูจน์ยืนยันตัวตนบุคคล โดยใช้คุณลักษณะจำเพาะทางสรีระ (Biometric) โดยระบบรู้จำใบหน้าจะทำงานโดยการเปรียบเทียบใบหน้าจากภาพถ่ายดิจิตอล หรือภาพจากกล้องวิดีโอของบุคคลที่เราสนใจกับฐาน ข้อมูลใบหน้าที่มีอยู่ และเมื่อเปรียบเทียบเสร็จก็จะแสดงผลใบหน้าที่อยู่ในฐานข้อมูล ที่มีใบหน้าเหมือนกับภาพที่นำมาเปรียบเทียบออกมา ระบบรู้จำใบหน้านั้นได้ถูกพัฒนามาอย่างต่อเนื่องเป็นเวลามากกว่าสิบปีมาแล้ว

เทคโนโลยีการเรียนรู้จดจำใบหน้า (Face Recognition) คือ เทคโนโลยีที่ถูกสร้างขึ้นมาเพื่อเรียนรู้และจดจำโครงสร้างใบหน้าของมนุษย์ แล้วนำข้อมูลใบหน้าที่จดจำหรือตรวจจับได้ส่งไปให้ระบบ เพื่อนำไปใช้วิเคราะห์หรือประมวลผลในการทำงานในส่วนขั้นตอนอื่น ๆ อีก ต่อไป ซึ่งเทคโนโลยีที่นำระบบการเรียนรู้จดจำใบหน้า ไปใช้งานมากที่สุดคือ เทคโนโลยีที่เกี่ยวข้องกับระบบความปลอดภัย ไม่ว่าจะเป็น ระบบ Access Control ระบบกล้องวงจรปิด หรือ ระบบรักษาความปลอดภัยในมือถือ เป็นระบบที่ช่วยเพิ่มประสิทธิภาพการระบุและ ยืนยันอัตลักษณ์บุคกลด้วยความรวดเร็ว แม่นยำ และสามารถเพิ่มโอกาสในการต่อยอดความสำเร็จให้หลากหลายแวควงธุรกิจได้ด้วย

หลักการทำงานของ Face Recognition คือ การสร้างโมเคลการอ้างอิง ที่เรียกว่า "faceprint" ขึ้นมา โดยระบบจะวิเคราะห์จากลักษณะ เฉพาะต่าง ๆ บนใบหน้า เช่น โครงหน้า ความกว้างของจมูก ระยะห่างระหว่างตาทั้งสองข้าง ขนาดของโหนกแก้ม ความลึกของเบ้าตา รวม ถึงพื้นผิวบนใบหน้า (facial texture) เป็นต้น จากนั้น ระบบจะทำการสร้างจุดเชื่อมโยงบนใบหน้า (nodal points) เพื่อเปรียบเทียบกับ รูปภาพที่ถูกเก็บไว้ในฐานข้อมูล (data base) ทั้งในลักษณะภาพนิ่งและภาพเคลื่อนไหว เพื่อความแม่นยำในการระบุตัวตนของผู้ที่ต้องเข้าสู่ กระบวนการตรวจสอบ

ดังนั้น หลักการทำงานของระบบรู้จำใบหน้า ที่ถูกออกแบบมาให้ทำการเปรียบเทียบใบหน้าบุคคลที่เราต้องการจะตรวจสอบ กับฐาน ข้อมูลใบหน้าที่มีอยู่ โดยอัลกอริทึมที่ใช้ในขั้นตอนการสร้างแม่แบบและขั้นตอนการเปรียบเทียบอาจมีความแตกต่างกันไป แล้วแต่วิธีการ ออกแบบระบบของแต่ละระบบ แต่ไม่ว่าจะมีอัลกอริทึมในการทำงานในขั้นตอนการสร้างแม่แบบและขั้นตอนการเปรียบเทียบยังไง แต่ ขั้นตอนการทำงานโดยรวมของระบบกี่ยังกงเหมือนกันอยู่

ขั้นตอนในการทำ Face Recognition

 การตรวจจับใบหน้า (Face Detection) คือ กระบวนการค้นหาใบหน้าของบุคคลจากภาพหรือวิดีโอ หลังจากนั้นก็จะทำการประมวล ผลภาพใบหน้าที่ได้สำหรับขั้นตอนถัดไป เพื่อให้ภาพใบหน้าที่ตรวจจับได้ง่ายต่อการจำแนก และ อัลกอริทึมที่ใช้ในการตรวจจับใบหน้า ในปัจจุบันก็มีอยู่ด้วยกันหลายวิธี ซึ่งอัลกอริทึมในการตรวจจับใบหน้าที่ดีนั้น มีส่วนช่วยในการจำแนกใบหน้าได้แม่นยำและรวดเร็วขึ้น เป็นอย่างมาก

2. การรู้จดจำใบหน้า (Face Recognition) คือ กระบวนการที่ได้นำภาพใบหน้าที่ตรวจจับได้ และประมวลผลแล้วจากขั้นตอนการตรวจ จับใบหน้า มาเปรียบเทียบกับฐานข้อมูลของใบหน้าเพื่อระบุว่าใบหน้าที่ตรวจจับได้ตรงกับบุคคลใด แล้วจึงนำผลลัพธ์ที่ได้ ส่งไปให้ระบบ หรือโปรแกรมเพื่อประมวลผลอื่นๆ ต่อไป

OpenCV (Open source Computer Vision) คือ ใลบรารีฟังก์ชันการเขียนโปรแกรม (Library of Programming Functions) โดยส่วน ใหญ่จะมุ่งเป้าไปที่การแสดงผลด้วยคอมพิวเตอร์แบบเรียลไทม์ (Real-Time Computer Vision) เดิมทีแล้วถูกพัฒนาโดย Intel แต่ภายหลัง ใด้รับการสนับสนุนโดย Willow Garage ตามมาด้วย Itseez (ซึ่งต่อมาถูกเข้าซื้อโดย Intel) OpenCV เป็นไลบรารีแบบข้ามแพลตฟอร์ม (Cross-Platform) และใช้งานได้ฟรีภายใต้ลิขสิทธิ์ของ BSD แบบโอเฟนซอร์ส (Open Source BSD License) OpenCV ยังสนับสนุนเฟรม เวิร์กการเรียนรู้เชิงลึก (Deep Learning Frameworks) ได้แก่ TensorFlow, Torch/PyTorch และ Caffe

Haar Cascades เป็น Object Detection Algorithm ใช้ในการตรวจจับวัตถุแบบ real-time อ้างอิงจากงานวิจัยของ Paul Viola และ Michael Jones ชื่อ "Rapid Object Detection using a Boosted Cascade of Simple Features" published in 2001 โดยหลักการทำงานของ Haar Cascades จะอ่านข้อมูล pixel ของรูป และนำมาเปรียบเทียบกับ feature ที่มีอยู่บนใบหน้าซึ่ง Haar Cascades สามารถปรับแต่งเพื่อตรวจจับวัตถุได้หลายอย่าง และเก็บข้อมูลในการตรวจ และโปรแกรมที่ถูกปรับแต่งเพื่อค้นหาวัตถุต่าง ๆ เก็บไว้ใน รูปของไฟล์ xml ซึ่งหากเราต้องการตรวจจับใบหน้า เราต้องนำไฟล์ cascades ที่ปรับแต่งสำหรับการจับใบหน้านั่นเอง



1. ดาวน์โหลดไฟล์ haarcascade_frontalface_default.xml

 1.1 สร้างโฟลเดอร์ Face_recognition ไว้ในโฟลเดอร์ /home/pi โดยคลิก <u>File Manager</u> แล้วคลิกขวาตรงพื้นที่ว่างเลือก <u>New</u> <u>Folder...</u> ตั้งชื่อโฟลเดอร์ว่า "Face_recognition"

		File N Trash	Nanager				
	↑ /home/p	oi					
					II		
	Adafruit_D- HT	Arduino	Bookshelf	Desktop	Documents		
					0		
	snack	Snack Machine Vending	Templates	test	Videos		
			New F	Folder			
			New F	ile			
			Paste	+ AII	_		
			Invert	Selection			
		Creat	ting New Fo	older	~	^	×
Enter	a name for	the newly	created fol	der:			
Face	e_recognitio	n					
				Cance	I Oł	•	

1.2 กลิกเปิด <u>Web Browser</u> พิมพ์ข้อกวามก้นหาว่า "haarcascade_frontalface_default.xml" จากนั้นกด <u>Enter</u>



1.3 คลิกเปิคเว็บไซต์ github.com คังรูป



1.4 คลิกปุ่ม <u>Download</u>

 Q opencv/data/haarcasca: × + ← → C a github.com/opencv/blob/4.x/data/haarcascades_cuda/h < ☆ ★ □ a 	~
← → C @ github.com/opencv/blob/4.x/data/haarcascades_cuda/h_ < ☆ ♣ □ ▲	
	. :
General opency (Public)	
♡ Sponsor 🗘 Notifications 💱 Fork 55.4k 🛱 Star 69.7k 👻	
<> Code 💿 Issues 2.4k 👔 Pull requests 96 💿 Actions 🖽 Wiki 🛈 Security 😶	
Image: second	
StevenPuttemans fixing CUDA cascade Latest commit aea328e on Feb 27, 2015 SHistory	
Ax 0 contributors	
1.2 MB Download Ü	

1.5 คลิก<u>ขุด 3 ขุด</u>ด้านบนขวาของเพจ จากนั้นเลือกเมนู <u>More Tools</u> --> <u>Save page as ...</u>

O opencv/data/haarcascad × +	~
← → C	< 🔅 🛊 🖬 🛎 🗄
G opencv / opencv Public	
Sponsor 🗘 Notifications 💱 Fork 55.4k	☆ Star 69.7k -
<> Code Issues 2.4k Pull requests 96 Actions 	iki 🕐 Security 😶
12 4.x + opencv / data / haarcascades_cuda / haarcascade_frontalface_default.xml	Go to file
(StevenPuttemans fixing CUDA cascade Latest commit aea328e on F	b 27, 2015 🕥 History
At 0 contributors	
1.2 MB	Download
later de la companya	unte l ferene al e ferral transl

		Save File			~ ^ X
Name: haarcascade	e_frontalface,	_default <mark>.xml</mark>			
ଢ Home	∢ û pi	Face_recognitio	n	—	5
🖿 Desktop	Name		* 8	Size Type	Modified
Documents					
↓ Downloads					
J Music					
Dictures					
Videos					
+ Other Locations					
				*	.XML,*.xml 🔻
				Cancel	Save 🕅

2. เขียนโปรแกรมสำหรับบันทึกใบหน้า

2.1 เสียบสายกล้องเว็บแคมเข้ากับพอร์ต USB ของบอร์ค Raspberry Pi



2.2 เปิดโปรแกรม Thonny Python IDE ขึ้นมาเพื่อเขียนโด้ดคำสั่งด้วยภาษา Python โดยคลิกที่สัญลักษณ์ 🐞 เลือกเมนู Programming แล้วคลิกเลือกโปรแกรม <u>Thonny Python IDE</u>



2.3 เขียน โค้คคำสั่ง คังนี้

```
1 import cv2 #นำเข้าไลบรารี cv2
 2
   import os #นำเข้าโมดูล OS เพื่อทำงานกับระบบปฏิบัติการ
 3
 4 cam = cv2.VideoCapture(0) #object ที่ประกาศขึ้นมาเพื่อใช้ในการ Capture Video
 5 cam.set(3, 640) #กำหนดความกว้างของหน้าจอกล้อง
 6 cam.set(4, 480) #กำหนดความสูงของหน้าจอกล้อง
 8 #เรียกใช้ไฟล์ cascade
 9
   face_detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
11 #รับค่าตัวเลขทางคีย์บอร์ดใส่ในตัวแปร face id
12 face_id = input('\n enter user id end press <return> ==> ')
13
14 #แสดงข้อความให้มองกล้อง
15 print("\n [INFO] Initializing face capture. Look the camera and wait ...")
16
17 count = 0 #กำหนดตัวแปร count = 0 เพื่อใช้ในการวนลูป
18
19 while(True): #วนลูป
20
        ret, img = cam.read() #เปิดภาพวิดีโอ
21
        img = cv2.flip(img, 1) #พลิกภาพวิดีโอ
        gray = cv2.cvtColor(img, cv2.COLOR BGR2GRAY) #ปรับภาพวิดีโอเป็นโหมดขาวดำ
23
        faces = face detector.detectMultiScale(gray, 1.3, 5)
24
        for (x,y,w,h) in faces:
26
            #สร้างกรอบสีเหลียมล้อมรอบใบหน้า
27
            cv2.rectangle(img, (x,y), (x+w,y+h), (255,0,0), 2)
28
            count += 1 #ปรับค่าตัวแปร count เพิ่มขึ้นทีละ 1
29
30
            #บันทึกรูปภาพใบหน้าไว้ในโฟลเดอร์ dataset
31
            cv2.imwrite("dataset/User." + str(face id) + '.' + str(count)
32
                          + ".jpg", gray[y:y+h,x:x+w])
34
            cv2.imshow('image', img) #เปิดโชว์ภาพโดยมีการกำหนดหัวเรืองและแสดงรูป
35
        k = cv2.waitKey(100) & 0xff #หากกด Esc หรือวนครบ 30 รอบให้ปิดกล้อง
36
37
        if k == 27:
38
            break
39
        elif count >= 30:
40
              break
41
42 print("\n [INFO] Exiting Program and cleanup stuff") #แสดงข้อความออกจากโปรแกรม
43 cam.release()
44 cv2.destroyAllWindows() #ปิดหน้าต่าง
```

2.4 บันทึกไฟล์ โดยกลิกที่เมนู <u>File</u> จากนั้นเลือกโฟลเดอร์ Face_recognition พิมพ์ชื่อไฟล์ว่า "addface" ในช่อง Name แล้วกลิกปุ่ม

<u>OK</u>

	New	Load Save Run Debug Over In
	2	import os
	3	<pre>cam = cv2.VideoCapture(0)</pre>
	5	<pre>cam.set(3, 640) # set video width cam.set(4, 480) # set video beight</pre>
l	0	com.set(4, 400) # set video height
		Save as 🗸 🗸 🗙
Name:	addface •	—
ம் Home	÷	
🖿 Deskt	ор	Name 🔻 Size Type Modified
🗅 Docur	ments	
↓ Down	loads	
J Music	;	
D Pictur	res	
► Video	s	
+ Other	Locations	6
		Python files 🕶
		Cancel

2.5 คลิก <u>Run</u> เพื่อรันโปรแกรม



ผลลัพธ์ที่ได้

1. หน้าต่าง Shell จะปรากฏข้อความ "Enter ID and press Enter ==>" เพื่อรับชื่อจากคีย์บอร์ด

New	È Load	Save	Run) Debug	Over	Into	Out	O Stop	کر Zoom	Quit	Switch 1 reguli mod	
addface.py×												
1 2 3	import import	cv2 os	#นำเข้าไล #นำเข้าโเ	าบรารี cา มดูล 0S	/2 เพื่อทำง	านกับระบ	ບປฏิบัติ	การ			İ	
4 5 6 7	4 cam = cv2.VideoCapture(0) #object ที่ประกาศขึ้นมาเพื่อใช้ในการ Capture 5 cam.set(3, 640) #กำหนดความกว้างของหน้าจอกล้อง 6 cam.set(4, 480) #กำหนดความสูงของหน้าจอกล้อง											
8	#เรียกใช้ไฟล์ cascade face_detector = cv2.CascadeClassifier('haarcascade_frontalfa											
Shell												
<pre>>>> %Run addface.py enter user id end press <return> ==> 1</return></pre>												
										Pyt	thon 3.9.2	

2. เมื่อพิมพ์ตัวเลข ID ลงไป กล้องเว็บแคมจะเปิดขึ้นมา โดยจะปรากฏกรอบสี่เหลี่ยมสีน้ำเงินรอบใบหน้า และจะทำการถ่ายรูป จำนวน 30 รูป





3. รูปภาพ 30 รูปจะถูกบันทึกไว้ในโฟลเคอร์ dataset โคยมีรูปแบบชื่อไฟล์รูปภาพ ดังรูป

Shell

/home/pi/l	Face_recognit	tion/dataset							
-	-	-	**	-	-	**	-	-	-
User.1.1.jpg	User.1.2.jpg	User.1.3.jpg	User.1.4.jpg	User.1.5.jpg	User.1.6.jpg	User.1.7.jpg	User.1.8.jpg	User.1.9.jpg	User
									1.10.jpg
<u> </u>	**	-	÷	-	-	-	÷	<u> </u>	-
User	User	User	User	User	User	User	User	User	User
1.11.jpg	1.12.jpg	1.13.jpg	1.14.jpg	1.15.jpg	1.16.jpg	1.17.jpg	1.18.jpg	1.19.jpg	1.20.jpg
1.21.jpg	1.22.jpg	1.23.jpg	1.24.jpg	1.25.jpg	1.26.jpg	1.27.jpg	1.28.jpg	1.29.jpg	1.30.jpg



2.1 ดาวน์โหลดไฟล์ trainer.yml

1. กลิกเปิด <u>Web Browser</u> พิมพ์ข้อกวามก้นหาว่า "trainer.yml" จากนั้นกด <u>Enter</u>



2. คลิกลิงก์เว็บไซต์ github.com คังรูป



3. คลิกปุ่ม <u>Download</u>



4. คลิกสัญลักษณ์ <u>งุค 3 งุค</u> จากนั้นคลิกคำสั่ง <u>More tools</u> --> <u>Save page as...</u>

https://ra								
f https://raw.git	thubuserco × +			•	×.			
\leftrightarrow \rightarrow x $\hat{\bullet}$	raw.githubusercontent.com/nole23/face-recog	nition/master/trai	< \$	* 0	1 🗖 🗄			
%YAML:1.0		New tab	New tab					
radius: 1 neighbors: 8		New window	New window Ctrl+N					
grid_x: 8 grid_y: 8		New Incognit	to window	Ctrl+S	Shift+N			
histograms: - !!opencv-mat	rix	History	•					
rows: 1 cols: 16384		Downloads	Ctrl+J					
dt: f data: [1.4	6092044e-03, 3.65230115e-03, 0., 0., 2	Bookmarks			•			
0., 2.2 5.11322 4.29276	04420050-02, 0., 0., 0., 0., 0., 0., 0 1700-03, 7.304602190-04, 0., 0., 2.921 1200-02 0 0 0 1 2140228260-0	Zoom	- 10	+ %00	53			
4.38270	960e-02, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.	Print			Ctrl+P			
0., 0.,	0., 7.30460219e-04, 2.92184087e-03, 0	Cast		/				
0., 0.,	0., 0., 0., 7.30460219e-04, 0., 0., 0	Find			Ctrl+F			
1.46	Save page as Ctrl+S	More tools	-		×			
0.,	Create shortcut	Edit	Cut	Сору	Paste			
0.,	Name window 🔪	Settings						
0.,	Clear browsing data Ctrl+Shift+Del	About Chrom	About Chromium					
	Extensions	Exit						

5. บันทึกไฟล์ไว้ในโฟลเดอร์ <u>Face recognition</u> จากนั้นคลิกปุ่ม <u>Save</u>



6. เปิดโปรแกรม Thonny Python IDE ขึ้นมาเพื่อเขียนโค้ดคำสั่งด้วยภาษา Python โดยคลิกที่สัญลักษณ์ 👔 เลือกเมนู

<u>Programming</u> แล้วคลิกเลือกโปรแกรม <u>Thonny Python IDE</u>



7. เขียน โค้คคำสั่ง ดังนี้

```
import cv2
   import numpy as np
 3
   from PIL import Image
 4
   import os
 6
   path='dataset'
 7
 8
   recognizer = cv2.face.LBPHFaceRecognizer create()
 9
   detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
11
12
   def getImagesAndLabels(path):
13
        imagePaths = [os.path.join(path,f) for f in os.listdir(path)]
14
        faceSamples = []
15
        ids = []
16
17
        for imagePath in imagePaths:
            PIL_img = Image.open(imagePath).convert('L')
18
            img_numpy = np.array(PIL_img,'uint8')
19
20
            id = int(os.path.split(imagePath)[-1].split('.')[1])
            faces = detector.detectMultiScale(img numpy)
            for (x,y,w,h) in faces:
24
                faceSamples.append(img_numpy[y:y+h,x:x+w])
25
                ids.append(id)
26
        return faceSamples, ids
27
28 print('\n [INFO] Training faces It will take a few seconds, wait....')
29 faces,ids = getImagesAndLabels(path)
30 recognizer.train(faces,np.array(ids))
31 recognizer.save('trainer.yml')
32 print("\n [INFO] {0} Face trained. Exit Program".format(len(np.unique(ids))))
```

8. บันทึกไฟล์ โดยคลิกที่เมนู <u>File</u> จากนั้นเลือกโฟลเดอร์ Face_recognition พิมพ์ชื่อไฟล์ว่า "face_training" ในช่อง Name แล้วคลิก ปุ่ม <u>OK</u>



		Save as			~ ^ X
Name: face_training					
☆ Home	∢ û pi	Face_recognition	+		C2
🛅 Desktop	Name	-	Size	Туре	Modified
 Documents Downloads Music Pictures Videos Other Locations 	 addface Bank_Fa dataset 	py ace_Recognition	1.2 kB	Text	Fri Thu Fri
			Cance	Pytho	on files 🕶 OK

9. คลิก <u>Run</u> เพื่อรันโปรแกรม



ผลลัพธ์ที่ได้

1. หน้าต่าง Shell จะปรากฏข้อความ "[INFO] Training faces It will take a few seconds, wait...." เพื่อแจ้งว่า กำลังเรียนรู้รูปภาพ

ใบหน้าอยู่ โปรครอสักครู่

2. หน้าต่าง Shell จะปรากฏข้อความ "[INFO] [INFO] ตัวเลข Face trained. Exit Program" เพื่อบอกจำนวนรูปภาพใบหน้าทั้งหมด ในโฟลเดอร์ dataset ที่ทำการเรียนรู้ (Train)





1. คลิก <u>New</u> เพื่อสร้างไฟล์ใหม่

	Ĥ	Ś.	0				E	0	۲	8
New	Load	Save	Run	Debug	Over	Into	Out	Stop	Zoom	Quit
Shell										
>>>										î
									P١	thon 3.9.2

2. พิมพ์โค้คคำสั่ง คังนี้

```
1 import cv2 #import library
 2 import numpy as np
3 import os
4 from PIL import Image
6 if not os.path.isfile("trainer.yml"):
 7
     print("Please train the data first")
8
     exit(0)
10 #iniciate id counter
11 id = 0
12 font = cv2.FONT_HERSHEY_SIMPLEX
13
14 face cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
15 cap = cv2.VideoCapture(0)
16 cap.set(3, 640) # set video width
17 cap.set(4, 480) # set video height
18
19 # Define min window size to be recognized as a face
20 minW = 0.1 \times cap.get(3)
21 minH = 0.1*cap.get(4)
22
23 recognizer = cv2.face.LBPHFaceRecognizer_create()
24 recognizer.read("trainer.yml")
25
26 #กำหนดชื่อให้ตรงกับ face ID ของใบหน้า
27 names = ['None', 'vivy', 'NAME2', 'NAME3', 'NAME4', 'NAME5', 'NAME6']
28
29 while True:
30
        ret, img =cap.read()
        img = cv2.flip(img, 1) # Flip vertically
32
        gray = cv2.cvtColor(img,cv2.COLOR BGR2GRAY)
33
        faces = face_cascade.detectMultiScale(gray, 1.3, 5)
34
35
        for(x,y,w,h) in faces:
36
            cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)
37
            id, confidence = recognizer.predict(gray[y:y+h,x:x+w])
38
39
            # Check if confidence is less them 100 ==> "0" is perfect match
40
            if (confidence > 65):
                id = names[id]
confidence = " {0}%".format(round(100 - confidence))
41
42
43
            else:
44
                id = "unknown"
45
                confidence = "
                                 {0}%".format(round(100 - confidence))
46
47
            cv2.putText(img, str(id), (x+5,y-5), font, 1, (255,255,255), 2)
48
            cv2.putText(img, str(confidence),(x+5,y+h-5),font, 1, (255,255,0),1)
49
50
        cv2.imshow('camera',img)
51
52
        k = cv2.waitKey(10) & 0xff # Press 'ESC' for exiting video
53
        if k == 27:
54
            break
55
56 # Do a bit of cleanup
57 print("\n [INFO] Exiting Program and cleanup stuff")
58 cap.release()
59 cv2.destroyAllWindows()
```

 บันทึกไฟล์ โดยคลิกที่เมนู <u>File</u> จากนั้นเลือกโฟลเดอร์ Face_recognition พิมพ์ชื่อไฟล์ว่า "face_recognition" ในช่อง Name แล้วคลิก ปุ่ม <u>OK</u>



4. คลิก <u>Run</u> เพื่อรันโปรแกรม



ผลลัพธ์ที่ได้

จะปรากฏหน้าต่างกล้องขึ้นมา จากนั้นจะจับใบหน้าเพื่อวิเคราะห์ว่าใบหน้าตรงกับรูปภาพในโฟลเดอร์ dataset หรือไม่ โดยจะมี กรอบสี่เหลี่ยมสีเขียวขึ้นมากลุมรอบใบหน้า หากวิเคราะห์ออกมาว่าใบหน้าตรงกับรูปภาพในโฟลเดอร์ dataset ก็จะปรากฏชื่อขึ้นมา



หากวิเคราะห์ออกมาว่าใบหน้าไม่ตรงกับรูปภาพในโฟลเดอร์ dataset จะปรากฏชื่อว่า unknownขึ้นมาตรงกรอบสี่เหลี่ยมสีเขียว



0

