



รายงานสัมมนาระบบสารสนเทศ

การจดจำใบหน้า(Face recognition by opencv python libraby)

จัดทำโดย

นายกฤษณะ อ่อนแก้ว 6010513040

นายเกริกเกียรติ เหมะรักษ์ 6010513041

อาจารย์ที่ปรึกษา

ดร.จันทวรรณ ปิยะวัฒน์

รายงานฉบับนี้เป็นส่วนหนึ่งของวิชา สัมมนาระบบสารสนเทศ

477-404 Seminar in Information System

ภาคเรียนที่ 1 ปีการศึกษา 2563

สาขาระบบสารสนเทศทางธุรกิจ ภาควิชาบริหารธุรกิจ

คณะวิทยาการจัดการ มหาวิทยาลัยสงขลานครินทร์

คำนำ

รายงานสัมมนา เรื่อง การจดจำใบหน้า (Face recognition by opencv python library) จัดทำขึ้นเพื่อศึกษาหาความรู้เกี่ยวกับส่วนของส่วนขยายของภาษาไพธอน เพื่อทดสอบความแม่นยำของการใช้ OpenCV ในการตรวจสอบใบหน้า คณะผู้จัดทำจึงหวังว่ารายงานสัมมนาเล่มนี้จะมีประโยชน์แก่ผู้ที่สนใจและต้องการศึกษาในเรื่องดังกล่าว ไม่มากก็น้อย

ข้าพเจ้าหวังว่าเนื้อหาในรายงานฉบับนี้จะเป็นประโยชน์แก่ผู้ที่ศึกษา หากรายงานฉบับนี้มีข้อผิดพลาดประการใด ต้องขออภัยมา ณ ที่นี้ด้วย

คณะผู้จัดทำ

สิงหาคม 2563

สารบัญ

หน้า

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญ.....	1
1.2 จุดประสงค์ของการศึกษา.....	1
1.3 ขอบเขตการค้นคว้า.....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2

บทที่ 2 แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....3

2.1 แนวคิดเกี่ยวกับ Face recognition.....	3
2.1.1 Face recognition คืออะไร.....	3
2.1.2 หลักการทำงานของ Face recognition.....	3
2.1.3 ความแตกต่าง Face DetectionและFace Recognition.....	4
2.1.4 ประโยชน์ของ face recognition.....	5
2.2 แนวคิดเกี่ยวกับ OpenCV.....	6
2.2.1 OpenCV คืออะไร.....	6
2.2.2 ประโยชน์ของ OpenCV.....	6
2.2.3 การใช้ OpenCV กับ Python.....	7
2.3 แนวคิดเกี่ยวกับ Numpy.....	8
2.3.1 Numpy คืออะไร.....	8
2.3.2 ประโยชน์ของ Numpy.....	9
2.3.3 อาร์เรย์ นัมไพ (Numpy) คืออะไร.....	9
2.3.4 ความแตกต่างระหว่าง PythonและNumpy Array.....	10

สารบัญ(ต่อ)

หน้า

2.3.5 การImport Numpy module.....	10
2.3.6 การใช้อาร์เรย์ (ndarray) เบื้องต้น.....	10
บทที่ 3 อุปกรณ์ในการใช้งาน.....	11
3.1 Python.....	11
3.2 OpenCV.....	15
3.3 Numpy.....	17
บทที่ 4 การดำเนินงาน.....	19
4.1 การอ่านค่ารูปภาพ วิดีโอ.....	19
4.1.1 การอ่านค่ารูปภาพ.....	19
4.1.2 การอ่านค่าวิดีโอ.....	20
4.2 พื้นฐานการจับใบหน้า.....	21
บทที่ 5 สรุปผลการดำเนินงาน.....	31
บรรณานุกรม.....	33

สารบัญรูปภาพ

หน้า

บทที่ 2

รูปที่ 2.1 หลักการการตรวจจับใบหน้า.....	4
รูปที่ 2.2 ความแตกต่าง Face DetectionและFace recognition.....	5
รูปที่ 2.3 สัญญาณลักษณะของ Opencv.....	6
รูปที่ 2.4 สัญญาณลักษณะของนัมไพ (Numpy).....	8

บทที่ 3

รูปที่ 3.1 ดาวน์โหลด Python 3.8.5.....	13
รูปที่ 3.2 หน้าต่างการตั้งค่า เมื่อดาวน์โหลดเสร็จ.....	13
รูปที่ 3.3 หน้าต่างการตั้งค่าสำหรับที่เก็บไฟล์.....	14
รูปที่ 3.4 ภาพเมื่อโปรแกรมติดตั้งสำเร็จ.....	14
รูปที่ 3.5 รูปภาพเมื่อทำการค้นหาโปรแกรมหลังจากการติดตั้ง.....	15
รูปที่ 3.6 รูปภาพกดเรียกใช้ Command Prompt.....	16
รูปที่ 3.7 ติดตั้ง OpenCV.....	16
รูปที่ 3.8 ติดตั้ง OpenCv สำเร็จ.....	17
รูปที่ 3.9 รูปภาพกดเรียกใช้ Command Prompt.....	18
รูปที่ 3.10 ติดตั้ง Numpy.....	18
รูปที่ 3.11 ติดตั้ง Numpy สำเร็จ.....	18

บทที่ 4

รูปที่ 4.1 รูปภาพโค้ดการอ่านค่าของรูปภาพ.....	19
รูปที่ 4.2 รูปภาพผลลัพธ์ของการอ่านค่าของรูปภาพ.....	19

สารบัญรูปภาพ(ต่อ)

หน้า

รูปที่ 4.3 รูปภาพโค้ดการอ่านค่าของวีดีโอ.....	20
รูปที่ 4.4 รูปภาพผลลัพธ์ของการอ่านค่าของรูปภาพ.....	20
รูปที่ 4.5 สร้างดาต้าเซตให้การจับใบหน้า (1).....	21
รูปที่ 4.6 สร้างดาต้าเซตให้การจับใบหน้า (2).....	21
รูปที่ 4.7 โค้ดการอ่านค่าใบหน้าและสร้างกล่องบนใบหน้า.....	21
รูปที่ 4.8 ผลลัพธ์ของการสร้างกล่องบนใบหน้า.....	22
รูปที่ 4.9 โค้ดการอ่านค่าดวงตาและสร้างพื้นที่ของดวงตา.....	22
รูปที่ 4.10 ผลลัพธ์ของการสร้างที่บนดวงตา.....	23
รูปที่ 4.11 โค้ดการสร้างพื้นที่บนปาก.....	23
รูปที่ 4.12 ผลลัพธ์การสร้างพื้นที่บนปาก.....	24
รูปที่ 4.13 โค้ดของการรวมใบหน้าดวงตาและปาก (1).....	24
รูปที่ 4.14 โค้ดของการรวมใบหน้าดวงตาและปาก (2).....	25
รูปที่ 4.15 ผลลัพธ์ของการรวมใบหน้า ดวงตาและปาก.....	25
รูปที่ 4.16 รูปโพลเดอร์ที่ใช้เพื่อรันโปรแกรม.....	26
รูปที่ 4.17 รูปโค้ดการทำดาต้าเซต (1).....	26
รูปที่ 4.18 รูปโค้ดการทำดาต้าเซต (2).....	27
รูปที่ 4.19 ผลสร้างดาต้าเซตของบุคคลที่ 1.....	27
รูปที่ 4.20 ภาพที่ถูกบันทึกลงในโพลเดอร์ดาต้าเซต.....	27
รูปที่ 4.21 โค้ดการเทรนการจดจำใบหน้า (1).....	28
รูปที่ 4.22 โค้ดการเทรนการจดจำใบหน้า (2).....	28

สารบัญรูปภาพ(ต่อ)

หน้า

รูปที่ 4.23 โพลเดอร์เทรนเนอร์.....	29
รูปที่ 4.24 โค้ดของการทำการจดจำใบหน้า (1).....	29
รูปที่ 4.25 โค้ดของการทำการจดจำใบหน้า (2).....	30
รูปที่ 4.26 ผลลัพธ์ของการทำจดจำใบหน้าและบอกเปอร์เซ็นต์ความใกล้เคียงกับด้าเซต.....	30

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

เนื่องจากในปัจจุบันยังคงมีการพัฒนาเกี่ยวกับการตรวจจับใบหน้าอยู่ระบบนี้ยังคงเป็นสิ่งสำคัญในยุคปัจจุบันเนื่องจากสามารถที่จะนำไปสร้าง AI และประยุกต์ใช้กับความต้องการของผู้ที่ต้องการจะใช้งานต่าง ๆ ตรวจจับอุณหภูมิ ตรวจจับผู้ร้าย เทียบใบหน้าของคน การหาสิ่งของที่มีความคล้ายกันโดยให้การตรวจจับเป็นตัวช่วยหา โดยมีโปรแกรมเมอร์หลายท่านนำไปใช้งานในฟังก์ชันต่าง ๆ ตามที่ตนสนใจ ตอนนี้มีการพัฒนามาอย่างแพร่หลาย ทางผู้จัดทำเล็งเห็นว่าระบบนี้มีการทำงานที่หลากหลายจึงต้องการที่จะนำเสนอรูปแบบของการจดจำใบหน้า โดยมีจุดประสงค์หลักคือศึกษาการตรวจจับและการทำงานของระบบการตรวจจับใบหน้า

การสืบค้นในปัจจุบันมีอย่างแพร่หลายทั้งการค้นหาในห้องสมุด การค้นหาออนไลน์แต่สื่อที่ทำออกมาในประเทศไทยเกี่ยวกับการจดจำใบหน้านั้นค่อนข้างมีน้อยแต่ก็ไม่ใช่ไม่มีเลย ซึ่งผู้จัดทำได้เล็งเห็นประโยชน์ที่มากมายของการจดจำใบหน้า เพื่อส่งเสริมให้โปรแกรมเมอร์ได้ใช้งานฟังก์ชันนี้อย่างแพร่หลาย ยิ่งในยุคปัจจุบันที่มีการแพร่ขยายของ COVID-19 ส่วนใหญ่จึงมีการจัดทำโปรแกรมการตรวจจับอุณหภูมิขึ้นมาเพื่อวัดอุณหภูมิของผู้คน นั่นก็ถือเป็นส่วนหนึ่งของการตรวจจับใบหน้า

ทางผู้จัดทำเล็งเห็นว่าโปรแกรมเมอร์เริ่มให้ความสนใจเรื่องการตรวจจับใบหน้ามากยิ่งขึ้น ทางผู้จัดทำจึงทำพื้นฐานหรือสิ่งที่ควรรู้เกี่ยวกับการออกแบบโปรแกรมหรือการทำตรวจจับใบหน้าโดยใช้ OpenCV เพื่อเป็นสื่อการเรียนรู้หรือเป็นการสร้างพื้นฐานให้กับคนที่สนใจที่จะทำเรื่องเกี่ยวกับการตรวจจับใบหน้า โดยการออกแบบรายงานให้เป็นแบบการสอนเพื่อนำไปใช้งานได้จริง โดยการอธิบายหลักขั้นต้นให้เข้าใจง่ายและผู้ที่พึงสนใจในด้านโปรแกรมเมอร์ก็สามารถที่จะเขียนโค้ดและทำโปรแกรมการตรวจจับใบหน้าได้ด้วยการอ่านรายงานเล่มนี้

1.2 จุดประสงค์ของการศึกษา

1.2.1 เพื่อศึกษาการใช้งาน python library openCV

1.2.2 เพื่อทำความเข้าใจเกี่ยวกับ face recognition

1.2.3 เพื่อเป็นแนวทางการศึกษาให้กับผู้ที่สนใจเรื่อง opencv และ face recognition

1.3 ขอบเขตของการค้นคว้า

- 1.3.1 ศึกษาการใช้งาน python library opencv ตั้งแต่วิธีการติดตั้งจนถึงการ Run code
- 1.3.2 ศึกษาการทำงานของ face recognition และ python library opencv
- 1.3.3 ศึกษาการทำงานของ numpy ใช้งานร่วมกับ opencv เพื่อทำ face recognition

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1.4.1 ได้เรียนรู้การใช้งานเกี่ยวกับ python library opencv
- 1.4.2 ได้เรียนรู้การทำ face recognition
- 1.4.3 ผู้ใช้งานได้รู้จัก python library opencv และ face recognition

บทที่2

ทฤษฎีที่เกี่ยวข้อง

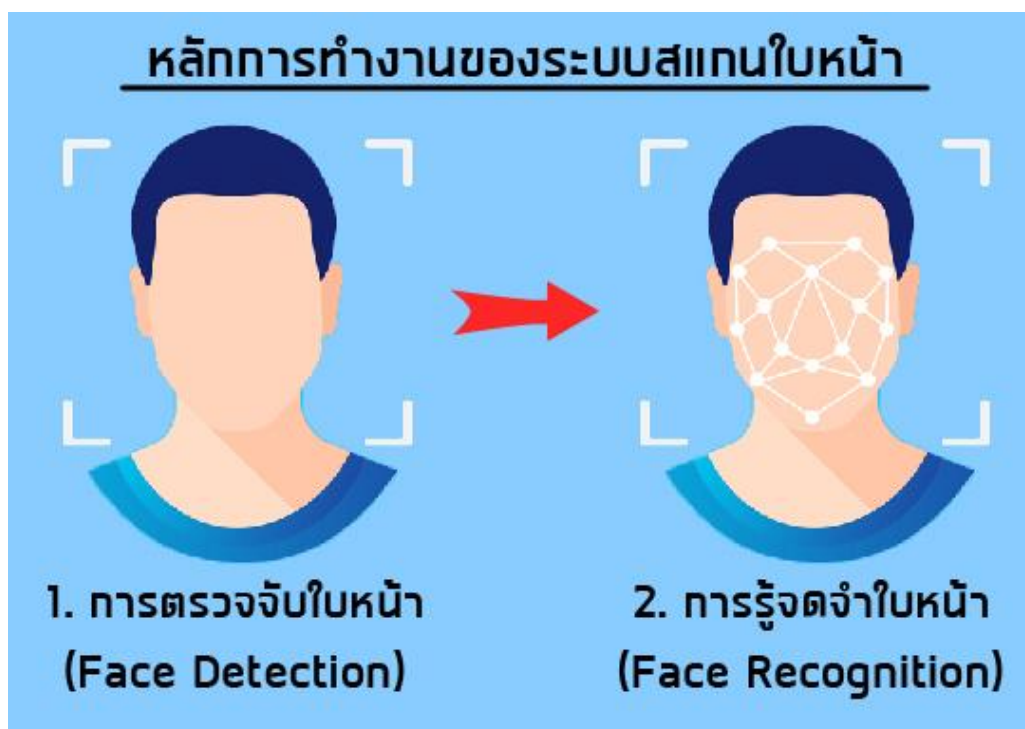
2.1 แนวคิดเกี่ยวกับ Face recognition

2.1.1 Face recognition คืออะไร

ดร.กอบเกียรติ สระอุบล(2563) กล่าวว่าเทคโนโลยีที่ถูกสร้างขึ้นมาเพื่อเรียนรู้และจดจำโครงสร้างใบหน้าของมนุษย์ แล้วนำข้อมูลใบหน้าที่จดจำหรือตรวจจับได้ส่งไปให้ระบบ เพื่อนำไปใช้วิเคราะห์หรือประมวลผลในการทำงานในส่วนขั้นตอนอื่น ๆ อีกต่อไป ซึ่งเทคโนโลยีที่นำระบบการเรียนรู้จดจำใบหน้า ไปใช้งานมากที่สุดคือ เทคโนโลยีที่เกี่ยวข้องกับระบบความปลอดภัย ไม่ว่าจะเป็น ระบบ Access Control ระบบกล้องวงจรปิด หรือ ระบบรักษาความปลอดภัยในมือถือความก้าวหน้าของการพัฒนาศักยภาพการทำงานเอไอ ด้วยระบบ แมชชีน เลิร์นนิง (Machine Learning) สร้างเป็นระบบประมวลผลที่สามารถตรวจจับใบหน้า (detect) และระบุตัวตน (identify) ได้ทั้งลักษณะภาพนิ่ง และภาพเคลื่อนไหวที่ถูกบันทึกไว้ ตลอดจนการเคลื่อนไหวแบบเรียลไทม์ (real time) ปัจจุบันมีการนำเทคโนโลยี Face Recognition มาใช้ด้านต่าง ๆ ทั้งระบบรักษาความปลอดภัย และเพิ่มประสิทธิภาพงานบริการ เพื่อสร้างประสบการณ์และต่อยอดสู่การตลาดในรูปแบบต่าง ๆ ที่เหมาะสมต่อไป

2.1.2 หลักการทำงานของ Face recognition

ดร.กอบเกียรติ สระอุบล(2563) กล่าวว่าหลักการทำงานของ Face Recognition คือ การสร้างโมเดลการอ้างอิง ที่เรียกว่า “faceprint” ขึ้นมา โดยระบบจะวิเคราะห์จากลักษณะเฉพาะต่าง ๆ บนใบหน้า เช่น โครงหน้า ความกว้างของจมูก ระยะห่างระหว่างตาทั้งสองข้าง ขนาดของโหนกแก้ม ความลึกของเบ้าตา รวมถึงพื้นผิวบนใบหน้า (facial texture) เป็นต้น จากนั้น ระบบจะทำการสร้างจุดเชื่อมโยงบนใบหน้า (nodal points) เพื่อเปรียบเทียบกับรูปภาพที่ถูกเก็บไว้ในฐานข้อมูล (data base) ทั้งในลักษณะภาพนิ่งและภาพเคลื่อนไหว เพื่อความแม่นยำในการระบุตัวตนของผู้ที่ต้องเข้าสู่กระบวนการตรวจสอบหรือหลักการทำงานของ Face Recognition คือ การสร้างโมเดลการอ้างอิง ที่เรียกว่า “faceprint” ขึ้นมา โดยระบบจะวิเคราะห์จากลักษณะเฉพาะต่าง ๆ บนใบหน้า เช่น โครงหน้า ความกว้างของจมูก ระยะห่างระหว่างตาทั้งสองข้าง ขนาดของโหนกแก้ม ความลึกของเบ้าตา รวมถึงพื้นผิวบนใบหน้า (facial texture) เป็นต้น จากนั้น ระบบจะทำการสร้างจุดเชื่อมโยงบนใบหน้า (nodal points) เพื่อเปรียบเทียบกับรูปภาพที่ถูกเก็บไว้ในฐานข้อมูล (data base) ทั้งในลักษณะภาพนิ่งและภาพเคลื่อนไหว เพื่อความแม่นยำในการระบุตัวตนของผู้ที่ต้องเข้าสู่กระบวนการตรวจสอบ



รูปที่ 2.2 ความแตกต่าง Face Detection และ Face recognition

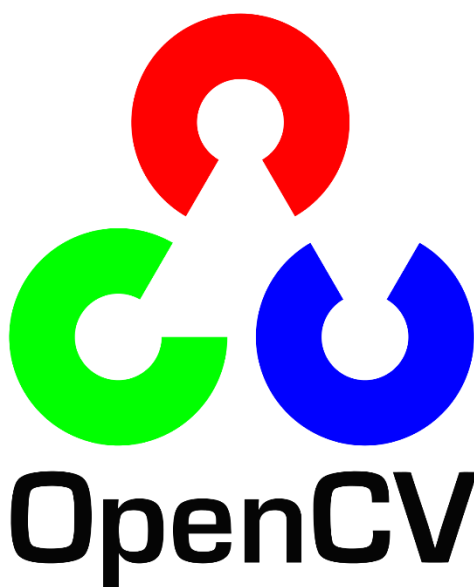
2.1.4 ประโยชน์ของ face recognition

ดร.กอบเกียรติ สระอุบล(2563) ประโยชน์ของ face recognition มีมากมายแล้วแต่การทำงานของโปรแกรมเมอร์ว่าต้องการนำไปใช้ทำอะไร ส่วนใหญ่นำไปทำเพื่อการตรวจสอบ เช่น การตรวจสอบความปลอดภัย ตรวจสอบอุณหภูมิ เป็นต้น ทั้งนี้ ตัวระบบเองสามารถปรับเปลี่ยนได้ตามที่โปรแกรมเมอร์ต้องการ หากต้องการวัดอุณหภูมิสามารถแก้ไขให้ตรวจจับใบหน้าและตรวจวัดอุณหภูมิในส่วนที่ตรวจจับได้ ระบบการตรวจสอบใบหน้าการเข้าสู่สมาร์ทโฟนของระบบ ios ก็ใช้ระบบนี้เช่นกันในการเปรียบเทียบหน้าจากหน้าที่บ้านที่ก๊อไป ไปตรวจสอบจากการจับกล้องของสมาร์ทโฟนกับตาต้าที่ได้บันทึกไว้ก่อนหน้านี้เพื่อเข้าสู่เนื้อหาทั้งหมดนี้แล้วแต่มาจากการทำ face recognition ทั้งสิ้นและยังมีระบบอีกมากมายที่ใช้งานระบบของ face recognition เช่น สะแกนลายนิ้วมือ มีหลักการตรวจจับเดียวกันแต่ปรับเปลี่ยนให้เข้ากับการตรวจจับลายนิ้วมือบนนิ้วมือแทนการตรวจจับใบหน้า

2.2 แนวคิดเกี่ยวกับ Opencv

2.2.1 Opencv คืออะไร

Opencv (Open source computer vision) เป็น Library ฟังก์ชันการเขียนโปรแกรมโดยมีเป้าหมายที่การแสดงผลด้วยคอมพิวเตอร์แบบเรียลไทม์ (Real-Time) โดย opencv ถูกพัฒนาขึ้นโดย Intel โดย opencv เป็น library ข้ามแพลตฟอร์มและสามารถใช้งานได้ฟรี ลิขสิทธิ์ของ BSD และยังรองรับ Frameworks การเรียนรู้เชิงลึก Deep Learning Frameworks เช่น Torch, PyTorch เป็นต้น



รูปที่ 2.3 สัญลักษณ์ของ Opencv

2.2.2 ประโยชน์ของ Opencv

เพื่อให้การพัฒนาโปรแกรมทางด้าน การมองเห็นของคอมพิวเตอร์ (Computer Vision) คือสามารถประมวลผลภาพดิจิทัลได้ทั้งภาพนิ่ง และภาพเคลื่อนไหวเช่น ภาพจากกล้องVDO หรือ VDO File เป็นไปได้ อย่างเป็นสะดวก มีฟังก์ชันสำเร็จรูปสำหรับการจัดการข้อมูลภาพ และการประมวลผลภาพพื้นฐานเช่น การหาขอบภาพ การกรองข้อมูลภาพ

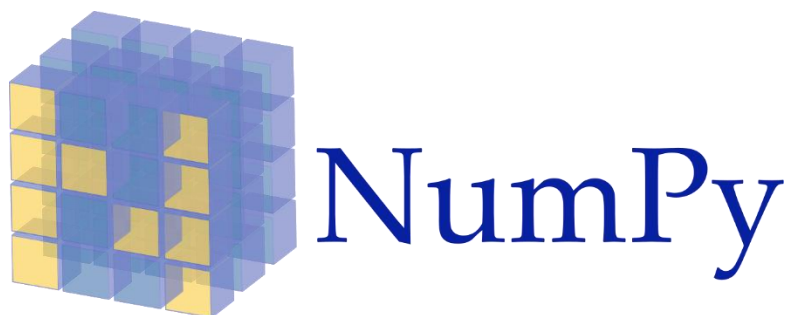
2.2.3 การใช้ Opencv กับ Python

Nuttakan Chuntra(2561) กล่าวว่า opencv เขียนขึ้นจากภาษาซีและทำงานในภาษาซีเป็นหลัก แต่สามารถเรียกใช้ผ่านภาษาโปรแกรมระดับสูงที่เขียนง่ายๆเช่นไพธอนได้ การเขียนในไพธอนนั้นข้อดีก็คือเขียนง่ายกว่าเขียนในภาษาซีโดยตรง แม้จะมีข้อเสียคือทำงานช้าไปบ้าง แต่การทำงานของคำสั่งต่าง ๆ ใน opencv นั้นโดยหลัก ๆ แล้วจะไปเรียกใช้ฟังก์ชันซึ่งเขียนในภาษาซี ไม่ได้เขียนด้วยไพธอนโดยตรง เมื่อการทำงานเบื้องหลังเป็นภาษาซีจึงทำงานรวดเร็ว ไม่ต้องห่วงว่าใช้ในไพธอนแล้วจะช้าลงกว่าใช้ในภาษาซีโดยตรงมากนัก แต่ก็เพราะ opencv ยึดพื้นฐานจากภาษาซีเป็นหลัก จึงทำให้มีลักษณะการใช้งานที่ออกจะเป็นสไตล์ภาษาซี แม้จะใช้ผ่านไพธอนก็ตาม ซึ่งคนที่เขียนไพธอนเป็นหลักอาจรู้สึกไม่คุ้นชินอยู่บ้างอีกทั้งภาษาไพธอนมีมอดูลอื่น ๆ หลากหลายซึ่งสามารถนำมาใช้ร่วมกันเพื่อทำงานได้ยืดหยุ่นกว้างขวางขึ้นเรื่องหนึ่งที่สร้างความยุ่งยากในการใช้ opencv-python ก็คือเรื่องของชนิดตัวแปร เพราะในภาษาไพธอนมักจะแบ่งแค่ว่าเป็นจำนวนเต็ม (int) และจำนวนจริง (float) แต่มันยังสามารถแปลงไปมาโดยอัตโนมัติตามความเหมาะสมโดยไม่ต้องคอยใส่ใจตลอดด้วยแต่ฟังก์ชันต่าง ๆ ของ opencv จะมีกำหนดชนิดของตัวแปรที่ใช้งานได้อยู่ ส่วนใหญ่จะไม่มีมีการเปลี่ยนชนิดตัวแปรให้โดยอัตโนมัติ อีกทั้งตัวแปรก็ไม่ได้แบ่งแค่ int กับ float แต่ int ก็มี int หลายชนิด float หลายชนิด ซึ่งต่างกันไปจำนวนบิตขนาดข้อมูลที่ใช้เก็บค่า ถ้าใช้ชนิดที่อาจทำงานได้ไม่ถูกต้องข้อมูลรูปภาพเป็นตัวเลขตั้งแต่ 0 ถึง 255 โดยทั่วไปจึงใช้ข้อมูลชนิด uint8 ซึ่งเป็นจำนวนเต็มบวกขนาด ๘ บิต ฟังก์ชันส่วนใหญ่ของ opencv ก็ใช้กับตัวแปรชนิดนี้อย่างไ้ก็ตาม uint8 ไม่สามารถเก็บข้อมูลติดลบได้ บางฟังก์ชันที่ต้องทำงานกับค่าลบอาจใช้ int32 (จำนวนเต็ม ๓๒ บิต) เป็นหลัก และถ้าต้องทำงานกับจำนวนที่ไม่ใช่จำนวนเต็มแต่มีทศนิยมก็จะใช้ float32 (จำนวนจริง ๓๒ บิต) ปกติในไพธอน int และ float จะหมายถึง int64 และ float64 ซึ่งก็สมารถใช้แทนเมื่อทำงานใน opencv ได้เช่นกันในกรณีส่วนใหญ่ แต่ไม่ใช่ทั้งหมด บางครั้งจึงจำเป็นต้องเปลี่ยนเป็น int32 หรือ float32 ด้วย รายละเอียดเรื่องชนิดตัวแปรจะไปเขียนถึงตอนอธิบายวิธีใช้ฟังก์ชันอีกที ถ้าฟังก์ชันนั้นมีความจำเพาะเรื่องชนิดข้อมูลที่ใช้งาน โดยรวมแล้ว opencv มีการใช้งานหลายอย่างที่ดูแปลกๆไปจากมอดูลที่เขียนด้วยไพธอนทั่วไป บางทีอาจรู้สึกไม่เข้าใจว่าทำไมต้องเป็นอย่างนั้น ซึ่งตรงนี้ที่จริงก็เป็นเพราะเดิมทีเป็นมอดูลที่ใช้ในภาษาซีนั่นเอง

2.3 แนวคิดเกี่ยวกับ Numpy

2.3.1 Numpy คืออะไร

wannaphong(2558) กล่าวว่า Numpy (num ย่อมาจาก numeric) เป็นโมดูลส่วนขยาย python ที่ใช้การทำงานที่มีประสิทธิภาพในอาร์เรย์ของข้อมูลที่เป็นเนื้อเดียวกัน Numpy อนุญาตให้ Python ทำหน้าที่เป็นภาษาระดับสูงสำหรับการจัดการข้อมูลตัวเลข Numpy เป็นโมดูลที่สำคัญมากที่สุดใน Python เลยก็ว่าได้ เพราะใช้สำหรับการคำนวณทางคณิตศาสตร์ที่มีความซับซ้อน เช่นการเขียนชุดตัวเลขให้อยู่ในรูปเมทริกซ์ ซึ่งทำให้สะดวกต่อการคำนวณ numpy ทำให้สามารถใช้ออบเจ็กต์พิเศษชนิดหนึ่งที่ชื่อว่า อาร์เรย์(ndarray) ซึ่งเป็นออบเจ็กต์ที่เก็บข้อมูลเป็นกลุ่มแถว สามารถเก็บเป็นหลายมิติได้และสามารถนำข้อมูลภายในมาคำนวณได้อย่างรวดเร็วจริงอยู่ว่าภาษาไพธอนมีข้อมูลชนิดลิสต์ (list) ซึ่งสามารถใช้เขียนแถวตัวเลขเรียงกันหลาย ๆ ตัว และยังสามารถทำเป็นสองหรือสามหรือหลายมิติได้ ดังนั้นการใช้ numpy นั้นนอกจากจะทำให้เขียนง่ายขึ้นแล้วยังทำให้โปรแกรมเร็วขึ้นกว่าเดิม เรียกได้ว่าไม่มีข้อเสียใด ๆ ในการใช้งาน ต่อให้เขียนออบเจ็กต์หรือฟังก์ชันที่ช่วยในการคำนวณเมทริกซ์ขึ้นมาเองก็ไม่ อาจจะเร็วไปกว่าการใช้ numpy ตราบใดที่ไม่ได้เขียนบนภาษาซี นอกจากนี้ยังมีมอดูลสำคัญอีกตัวที่สร้างขึ้นมาสืบมาจาก numpy ก็คือ scipy ซึ่งเป็นมอดูลที่ประกอบไปด้วยคำสั่งต่าง ๆ ที่ช่วยในการคำนวณทางคณิตศาสตร์ เพื่อใช้ในด้านวิทยาศาสตร์แขนงต่าง ๆ ที่เฉพาะทางขึ้นไปอีก การใช้งาน scipy นั้นค่อนข้างเป็นเฉพาะ



รูปที่ 2.4 สัญลักษณ์ของนัมไพ(Numpy)

2.3.2 ประโยชน์ของ Numpy

wannaphong(2558) กล่าวว่า NumPy (Numpy) มีประโยชน์เพิ่มเติมใดบ้างที่จะช่วยให้การเขียนโปรแกรมลดลงโดยเฉพาะอย่างยิ่งสิ่งที่เกี่ยวข้องกับการคำนวณทางคณิตศาสตร์

1. NumPy ใช้หน่วยความจำน้อยกว่ามากในการจัดเก็บข้อมูลอาร์เรย์ NumPy ใช้จำนวนหน่วยความจำน้อยกว่าอย่างมากเมื่อเปรียบเทียบกับ python นอกจากนี้ยังมีกลไกในการระบุประเภทข้อมูลของเนื้อหาซึ่งจะช่วยเพิ่มประสิทธิภาพของการทำงานต่อไป
2. การดำเนินการทางคณิตศาสตร์ในอาร์เรย์ NumPy ตามที่ระบุไว้ก่อนหน้านี้ NumPy ไม่เพียง แต่จะเกี่ยวกับการจัดเก็บข้อมูลที่มีประสิทธิภาพเท่านั้น แต่ยังทำให้การดำเนินการทางคณิตศาสตร์เป็นไปอย่างง่ายดาย
3. ขนาด โครงสร้างข้อมูลแบบ Numpy ใช้พื้นที่น้อยกว่า
4. ประสิทธิภาพ มีความเร็วในการใช้งาน
5. ฟังก์ชัน NumPy มีฟังก์ชันที่ปรับให้เหมาะสมเช่นการดำเนินงานพีชคณิตเชิงเส้นในตัว
6. หน่วยความจำประโยชน์หลักของการใช้อาร์เรย์ NumPy มีการใช้หน่วยความจำน้อยกว่าและมีพฤติกรรมการใช้งานจริงที่ค่อนข้างดี
7. ความเร็ว แสดงจำนวนการปฏิบัติงานระหว่าง Python และ Numpy ค้นพบว่า Numpy มีประสิทธิภาพที่ดีกว่า Python

2.3.3 อาร์เรย์ NumPy (Numpy) คืออะไร

อาร์เรย์ NumPy เป็นอาร์เรย์หลายมิติของวัตถุชนิดเดียวกันทั้งหมด ในหน่วยความจำมันเป็นวัตถุที่ชี้ไปยังบล็อกของหน่วยความจำของข้อมูลที่เก็บไว้ในหน่วยความจำนั้นแถมมันยังช่วยติดตามจำนวนของมิติอีกด้วย

2.3.4 ความแตกต่างระหว่าง Python และ Numpy Array

นัมไพ (Numpy) เป็นห้องสมุดหลักสำหรับการคำนวณทางวิทยาศาสตร์ใน Python มีวัตถุอาเรย์หลายมิติที่มีประสิทธิภาพสูงและเครื่องมือสำหรับการทำงานกับอาเรย์เหล่านี้ อาเรย์ numpy เป็นตารางของค่าชนิดเดียวกันทั้งหมด จำนวนมิติคืออันดับของอาเรย์ คำถามเริ่มต้นที่พบบ่อยคือความแตกต่างที่แท้จริงคืออะไร คำตอบคือประสิทธิภาพ โครงสร้างข้อมูล Numpy ทำงานได้ดีขึ้น

2.3.5 การ Import Numpy module

หลายวิธีในการ import NumPy เข้ามาใช้งาน โดยทั่วไปวิธีมาตรฐาน ได้แก่ คำสั่ง `import numpy` อย่างไรก็ตามหากมีการเรียกใช้งาน NumPy บ่อยครั้ง แต่ละครั้งจะต้อง พิมพ์ `numpy.X` ดังนั้นเพื่อความสะดวก จะย่อเวลาเรียกเป็น `np` จะเปลี่ยนการ import เป็น `import numpy as np` ทำให้เรียกการใช้งานฟังก์ชันได้โดยเขียนนี้เป็น `np.X`

2.3.6 การใช้อาเรย์ (ndarray) เบื้องต้น

numpy เป็นมอดูลที่ทำให้เราสามารถใช้ออบเจกต์ชนิดที่เรียกว่า ndarray ซึ่งหมายถึงอาเรย์หลายมิติ บางครั้งก็เรียกว่าอาเรย์เฉยๆ คำว่า "อาเรย์" เป็นคำที่ถูกใช้ในภาษาอื่น ๆ อีกหลายภาษา เช่น ภาษาซี, php, จาวาสคริปต์, รูปี เป็นต้น แต่ว่าสำหรับในภาษาไพธอนสิ่งที่มีคุณสมบัติเทียบเท่ากับอาเรย์ในภาษาอื่นนั้น กลับ เรียกว่าลิสต์ ส่วนอาเรย์จะหมายถึง ndarray ของ numpy ในมอดูลมาตรฐานของภาษาไพธอนก็มีสิ่งที่เรียกว่า อาเรย์อยู่ แต่ก็ไม่ใช่ที่นิยมใช้ ดังนั้นพอพูดถึงอาเรย์ในภาษาไพธอนแล้วจึงมักหมายถึง ndarray ของ numpy นั่นเอง ก่อนที่จะเริ่มใช้งานสิ่งที่ต้องทำเป็นอย่างแรกก็คือทำการ import เรียกใช้ขึ้นมาก่อน

บทที่ 3

อุปกรณ์ในการใช้งาน

3.1 Python

Sarayut Nonsiri(2560) กล่าวว่าPython คือชื่อภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่ง ซึ่งถูกพัฒนาขึ้นมาโดยไม่ยึดติดกับแพลตฟอร์ม กล่าวคือสามารถรันภาษา Python ได้ทั้งบนระบบ Unix, Linux, Windows NT, Windows 2000, Windows XP หรือแม้แต่ระบบ FreeBSD อีกอย่างหนึ่งภาษาตัวนี้เป็น OpenSource เหมือนอย่าง PHP ทำให้ทุกคนสามารถที่จะนำ Python มาพัฒนาโปรแกรมของเราได้ฟรี ๆ โดยไม่ต้องเสียค่าใช้จ่ายและความเป็น OpenSource ทำให้มีคนเข้ามาช่วยกันพัฒนาให้ Python มีความสามารถสูงขึ้นและใช้งานได้ครอบคลุมกับทุกลักษณะงาน

ประวัติความเป็นมาของภาษา Python ไพทอน ภาษาไพทอน (Python programming language) เป็นภาษาโปรแกรมแบบอินเทอร์พรีเตอร์ที่สร้างโดย กีโด ฟาน รอสซัม (Guido van Rossum) ในพ.ศ. 2533 ปัจจุบันดูแลโดย มูลนิธิซอฟต์แวร์ไพทอน

จุดเด่นของภาษาไพทอน ไพทอนเป็นภาษาสคริปต์ ทำให้ใช้เวลาในการเขียนและคอมไพล์ไม่มากทำให้เหมาะ กับงานด้านการดูแลระบบ (System administration) เป็นอย่างยิ่ง ได้มีการสนับสนุนภาษาไพทอน โดยเป็นส่วนหนึ่งของระบบปฏิบัติการยูนิกซ์, ลินุกซ์และสามารถติดตั้งให้ทำงานเป็นภาษาสคริปต์ของวินโดวส์ผ่านระบบ Windows Script Host ได้อีกด้วยและPythonเองก็ได้ถูกนำมาพัฒนา Web application อย่างแพร่หลาย ซึ่งมี Framework สำหรับทำเว็บของ Python ที่ได้รับความนิยมอย่างมากคือ Flask

ไวยากรณ์อ่านง่าย ไวยากรณ์ของไพทอนได้กำจัดการใช้สัญลักษณ์ที่ใช้ในการแบ่งบล็อกของโปรแกรม และใช้การย่อหน้าแทน ทำให้สามารถอ่านโปรแกรมที่เขียนได้ง่าย นอกจากนั้นยังมีการสนับสนุนการเขียน docstring ซึ่งเป็นข้อความสั้นๆ ที่ใช้อธิบายการทำงานของฟังก์ชัน, คลาส, และโมดูลอีกด้วย

ความเป็นภาษากาว ไพทอนเป็นภาษากาว (Glue Language) ได้อย่างดีเนื่องจากสามารถเรียกใช้ภาษาโปรแกรมอื่น ๆ ได้หลายภาษาทำให้เหมาะที่จะใช้เขียนเพื่อประสานงานโปรแกรมที่เขียนในภาษาต่างกันได้

ไลบรารีในไพทอน การเขียนโปรแกรมในภาษาไพทอนโดยใช้ไลบรารีต่าง ๆ เป็นการลดภาระของโปรแกรมเมอร์ได้เป็นอย่างดีทำให้โปรแกรมเมอร์ไม่ต้องเสียเวลากับการเขียนคำสั่งที่ซ้ำ ๆ เช่นการแสดงผลข้อมูลออกสู่หน้าจอหรือการรับค่าต่าง ๆ ไพทอนมีชุดไลบรารีมาตรฐานมาให้ตั้งแต่ติดตั้งอินเตอร์พรีเตอร์ นอกจากนั้นยังมี ผู้พัฒนาจากทั่วโลกดำเนินการพัฒนาไลบรารีซึ่งช่วยอำนวยความสะดวกในด้านต่าง ๆ โดยจะ

เผยแพร่ในรูปแบบของแพ็คเกจต่าง ๆ ซึ่งสามารถติดตั้งเพิ่มเติมได้อีกด้วย สุดท้ายคือภาษาไพทอน ทำงานเร็วที่สุดเมื่อเทียบกับภาษา script ด้วยกัน เช่น php, jsp, asp

คุณลักษณะเด่นของภาษา Python

คุณลักษณะเด่นของภาษา Python
สนับสนุนแนวคิด Object oriented หรือ OOP (Object Oriented Programming)
เป็น Open Source
โค้ดที่เขียนด้วย Python สามารถนำไปรันบนระบบปฏิบัติการได้หลากหลาย
สนับสนุนเทคโนโลยี COM ของ Ms-windows
Python รวมมาตรฐานการ Interface Tkinter ซึ่งสนับสนุนบนระบบ X windows, Ms-windows และ Macintosh การใช้คำสั่ง Tkinter API ช่วยให้โปรแกรมเมอร์ไม่ต้องแก้ไขโค้ดเมื่อนำไปรันบนระบบปฏิบัติการอื่น ๆ
เป็น Dynamic typing คือสามารถเปลี่ยนชนิดข้อมูลได้ง่ายและสะดวก
มี Built-in Object Types คือโครงสร้างของข้อมูลที่สามารถใช้ได้ ใน Python ประกอบด้วย list, dictionary, string ที่ง่ายต่อการใช้งานและมีประสิทธิภาพสูง
มีเครื่องมือต่าง ๆ มากมาย เช่น การประมวลผล Text file การเรียงข้อมูล การเชื่อมต่อ string การตรวจสอบเงื่อนไขของข้อความการแทนค่า เป็นต้น
มีมอดูลสำหรับการ Regular Expression
มีมอดูลที่สร้างขึ้นจากนักพัฒนามากมาย ได้แก่ COM, Image, CORBA, ORBs, XML เป็นต้น
จัดการหน่วยความจำอย่างอัตโนมัติสามารถจัดการพื้นที่หน่วยความจำที่ไม่ต่อเนื่องให้ทำงานได้อย่างมีประสิทธิภาพ
อนุญาตให้ฝังชุดคำสั่งของ Python เอาไว้ภายในโค้ดภาษา C/C++ ได้
อนุญาตให้โปรแกรมเมอร์สร้าง Dynamic Link Library (DLL) เพื่อใช้ร่วมกับ Python
มีมอดูลสนับสนุนเกี่ยวกับ network process trade regular, expression, xml, GUI และอื่นๆ
ประกอบด้วย module สำหรับสร้าง Internet Script และติดต่อกับอินเทอร์เน็ตผ่าน Sockets, และทำหน้าที่เป็น CGI Script ตลอดจนใช้งานคำสั่ง FTP, Gopher, XML และอื่น ๆ อีกมาก
สามารถประมวลผลทางด้านวิทยาศาสตร์และวิศวกรรมศาสตร์ได้อย่างมีประสิทธิภาพ
มีฟังก์ชันสนับสนุนฐานข้อมูลเช่น MySQL, Sybase, Oracle, Informix, ODBC และอื่น ๆ
มี library สนับสนุนด้านการสร้างภาพกราฟฟิก เช่น ทำภาพเบลอหรือภาพชัดหรือเขียนข้อความบนภาพ ตลอดจนบันทึกไฟล์ในรูปแบบต่าง ๆ ได้อย่างสะดวกและมีประสิทธิภาพ
มี library สนับสนุนด้านปัญญาประดิษฐ์

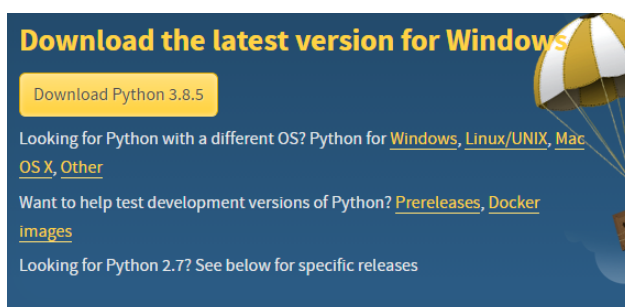
มีไลบรารีสำหรับสร้างเอกสาร PDF โดยไม่ต้องติดตั้ง Acrobat Writer

มีไลบรารีสำหรับสร้าง Shockwaves Flash (SWF) โดยไม่ต้องติดตั้ง Macromedia Flash

ขั้นตอนการติดตั้ง Python

1.ดาวน์โหลดไฟล์ติดตั้งของpythonจาก <https://www.python.org/downloads/>

2.เลือก “Python 3.8.5” จากนั้นกด “Download”



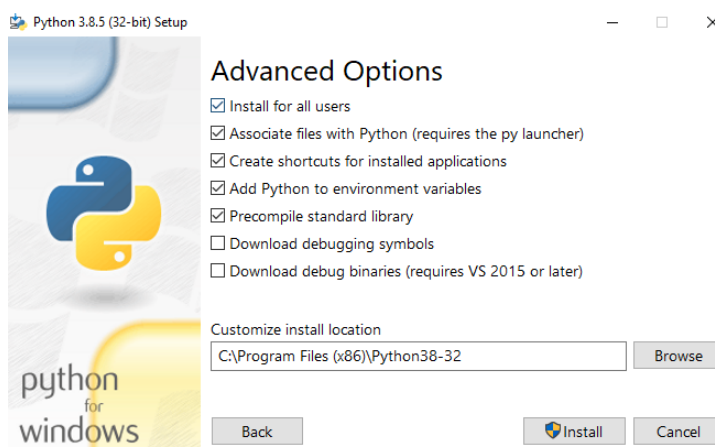
รูปที่ 3.1 ดาวน์โหลด Python 3.8.5

3.เมื่อโปรแกรมดาวน์โหลดเสร็จ จะขึ้นหน้าต่างการติดตั้ง ให้เลือก “Add Python 3.8 to PATH” และกด “Customize installation”



รูปที่ 3.2 หน้าต่างการตั้งค่า เมื่อดาวน์โหลดเสร็จ

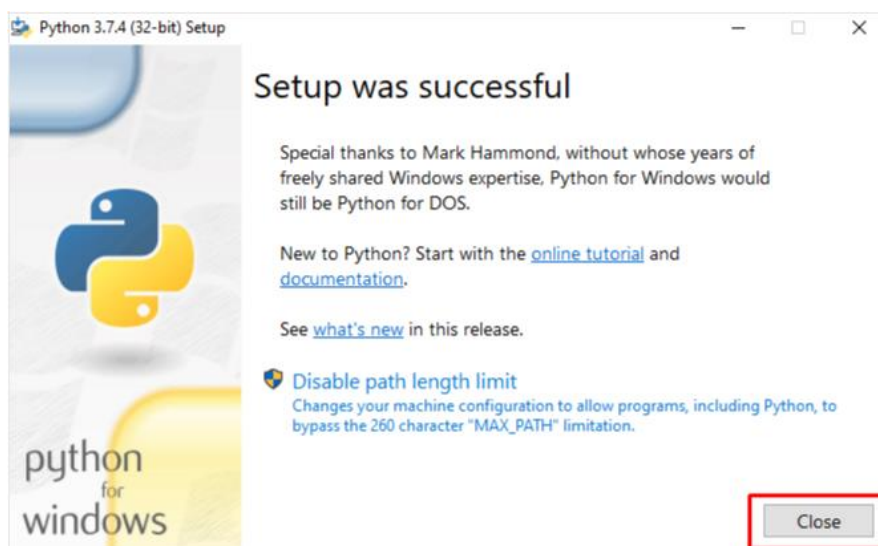
4. หลังจากนั้นให้คลิกตรงคำว่า “Install for all users”



รูปที่ 3.3 หน้าต่างการตั้งค่าสำหรับที่เก็บไฟล์

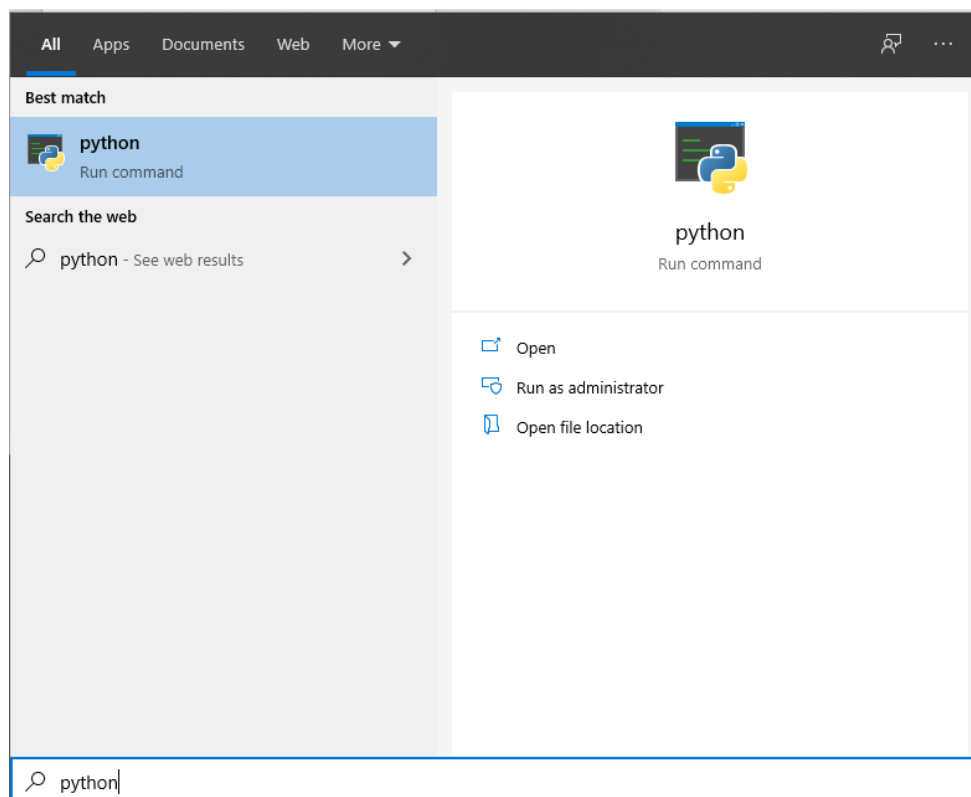
5. หลังจากนั้นรอให้โปรแกรมทำการติดตั้ง

6. เมื่อโปรแกรมติดตั้งเสร็จแล้ว จะขึ้นหน้าต่างว่า “Setup was successful” เป็นการติดตั้งอย่างสมบูรณ์ จากนั้นให้เรากด “Close”



รูปที่ 3.4 ภาพเมื่อโปรแกรมติดตั้งสำเร็จ

8. หลังจากนั้น ทดลองหาโปรแกรม Python ถ้าขึ้นดังนี้ก็ถือว่าติดตั้งโปรแกรมเสร็จสมบูรณ์



รูปที่ 3.5 รูปภาพเมื่อทำการค้นหาโปรแกรมหลังจากการติดตั้ง

3.2 OpenCV

Nuttakan Chuntra(2561) กล่าวว่า OpenCV ถูกเขียนขึ้นด้วยภาษา C++ มีการรองรับ Python, Java และ MATLAB/OCTAVE — API สำหรับอินเทอร์เน็ตเหล่านี้สามารถพบได้ในเอกสารออนไลน์ ซึ่งมีการรวมไว้หลากหลายภาษา เช่น C#, Perl, Ch, Haskell และ Ruby ได้รับการพัฒนาเพื่อส่งเสริมการนำมาใช้งานโดยผู้ใช้ที่เพิ่มขึ้น

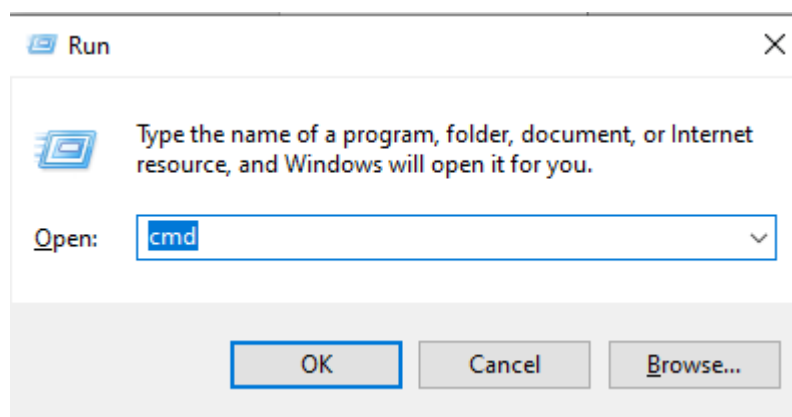
OpenCV (Open source Computer Vision) เป็นไลบรารีฟังก์ชันการเขียนโปรแกรม (Library of Programming Functions) โดยส่วนใหญ่จะมุ่งเป้าไปที่การแสดงผลด้วยคอมพิวเตอร์แบบเรียลไทม์ (Real-Time Computer Vision) เดิมทีแล้วถูกพัฒนาโดย Intel แต่ภายหลังได้รับการสนับสนุนโดย Willow Garage ตามมาด้วย Itseez (ซึ่งต่อมาถูกเข้าซื้อโดย Intel) OpenCV เป็นไลบรารีแบบข้ามแพลตฟอร์ม (Cross-Platform) และใช้งานได้ฟรีภายใต้ลิขสิทธิ์ของ BSD แบบโอเพ่นซอร์ส (Open-Source BSD

License) OpenCV ยังสนับสนุนเฟรมเวิร์กการเรียนรู้เชิงลึก (Deep Learning Frameworks) ได้แก่ TensorFlow, Torch/PyTorch และ Caffe

เพื่อให้การพัฒนาโปรแกรมทางด้าน การมองเห็นของคอมพิวเตอร์ (Computer Vision) คือสามารถประมวลผลภาพดิจิทัลได้ทั้งภาพนิ่ง และภาพเคลื่อนไหวเช่น ภาพจากกล้องVDO หรือ VDO File เป็นไปได้ อย่างสะดวก มีฟังก์ชันสำเร็จรูปสำหรับจัดการข้อมูลภาพ และการประมวลผลภาพพื้นฐานเช่น การหาขอบภาพ การกรองข้อมูลภาพ

ขั้นตอนการติดตั้ง OpenCV

1.ทำการเปิด cmd ของ window โดยการกด Alt + Window พิมพ์คำว่า cmd ในช่องว่าง กด OK เพื่อเรียกใช้งาน



รูปที่ 3.6 รูปภาพกดเรียกใช้ Command Prompt

2.เมื่อเรียกใช้งาน Command Prompt แล้วให้พิมพ์ pip install-python เพื่อติดตั้ง opencv

```
C:\Users\ProgrammingKnowledge>python --version
Python 3.7.0
C:\Users\ProgrammingKnowledge>pip --version
pip 10.0.1 from c:\python\python37\lib\site-packages\pip (python 3.7)
C:\Users\ProgrammingKnowledge>pip install opencv-python
```

รูปที่ 3.7 ติดตั้ง OpenCV

3.เมื่อติดตั้งเสร็จสิ้นจะมีลักษณะดังภาพ

```
C:\Users\ProgrammingKnowledge>python --version
Python 3.7.0

C:\Users\ProgrammingKnowledge>pip --version
pip 10.0.1 from c:\python\python37\lib\site-packages\pip (python 3.7)

C:\Users\ProgrammingKnowledge>pip install opencv-python
Collecting opencv-python
  Downloading https://files.pythonhosted.org/packages/cc/3c/e9b46d4ff65d4dfcca1789e32113f38d5fc1804840a4aa0bc2437ecef860/opencv_python-4.0.0.21-cp37-cp37m-win_amd64.whl (30.4MB)
    100% |#####| 30.4MB 195kB/s
Requirement already satisfied: numpy>=1.14.5 in c:\python\python37\lib\site-packages (from opencv-python) (1.16.0)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.0.0.21
You are using pip version 10.0.1, however version 19.0.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Users\ProgrammingKnowledge>
```

รูปที่ 3.8 ติดตั้ง OpenCv สำเร็จ

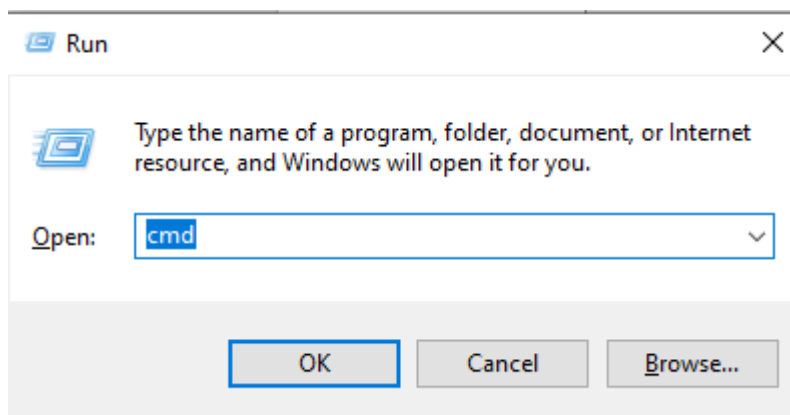
4.เมื่อติดตั้งเสร็จสิ้นสามารถเรียกใช้งาน OpenCV ได้โดยการใช้ import cv หรือ Import cv2

3.3 Numpy

wannaphong(2558) กล่าวว่าNumpy (num ย่อมาจาก numeric) เป็นโมดูลส่วนขยาย python ที่ใช้การทำงานที่มีประสิทธิภาพในอาร์เรย์ของข้อมูลที่เป็นเนื้อเดียวกัน Numpy อนุญาตให้ Python ทำหน้าที่เป็นภาษาระดับสูงสำหรับการจัดการข้อมูลตัวเลข Numpy เป็นโมดูลที่สำคัญมากที่สุดใน Python เลยก็ว่าได้ เพราะใช้สำหรับการคำนวณทางคณิตศาสตร์ที่มีความซับซ้อน เช่นการเขียนชุดตัวเลขให้อยู่ในรูปแมทริกซ์ ซึ่งทำให้สะดวกต่อการคำนวณ numpy ทำให้สามารถใช้ออบเจกต์พิเศษชนิดหนึ่งที่ชื่อว่า อาร์เรย์(ndarray) ซึ่งเป็นออบเจกต์ที่เก็บข้อมูลเป็นกลุ่มแถว สามารถเก็บเป็นหลายมิติได้และสามารถนำข้อมูลภายในมาคำนวณได้อย่างรวดเร็วจริงอยู่ว่าภาษาไพธอนมีข้อมูลชนิดลิสต์ (list) ซึ่งสามารถใช้เขียนแถวตัวเลขเรียงกันหลาย ๆ ตัว และยังทำเป็นสองหรือสามหรือหลายมิติได้ ดังนั้นการใช้ numpy นั้นนอกจากจะทำให้เขียนง่ายขึ้นแล้วยังทำให้โปรแกรมเร็วขึ้นกว่าเดิม เรียกได้ว่าไม่มีข้อเสียใด ๆ ในการใช้งาน ต่อให้เขียนออบเจกต์หรือฟังก์ชันที่ช่วยในการคำนวณแมทริกซ์ขึ้นมาเองก็ไม่ อาจจะเร็วไปกว่าการใช้ numpy ตราบใดที่ไม่ได้เขียนบนภาษาซี นอกจากนี้ยังมีมอดูลสำคัญอีกตัวที่สร้างขึ้นมาจาก numpy ก็คือ scipy ซึ่งเป็นมอดูลที่ประกอบไปด้วยคำสั่งต่าง ๆ ที่ช่วยในการคำนวณทางคณิตศาสตร์ เพื่อใช้ในด้านวิทยาศาสตร์แขนงต่าง ๆ ที่เฉพาะทางขึ้นไปอีก การใช้งาน scipy นั้นค่อนข้างเป็นเฉพาะ

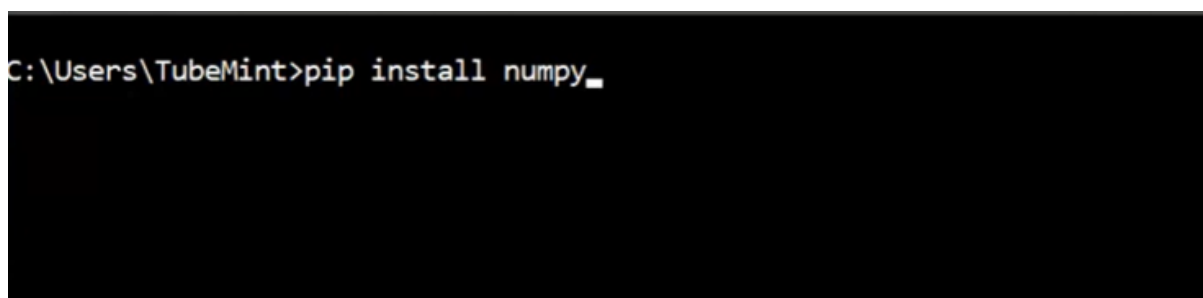
ขั้นตอนการติดตั้ง Numpy

1.ทำการเปิด cmd ของ window โดยการกด Alt + Window พิมพ์คำว่า cmd ในช่องว่าง กด OK เพื่อเรียกใช้งาน



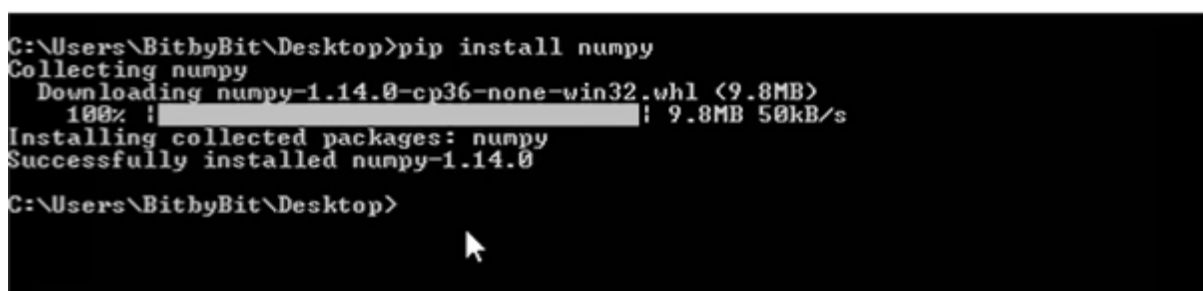
รูปที่ 3.9 รูปภาพกดเรียกใช้ Command Prompt

2.เมื่อเรียกใช้งาน Command Prompt แล้วให้พิมพ์ pip install numpy เพื่อติดตั้ง numpy



รูปที่ 3.10 ติดตั้ง Numpy

3.เมื่อติดตั้งเสร็จสิ้นจะมีลักษณะดังภาพ



รูปที่ 3.11 ติดตั้ง Numpy สำเร็จ

บทที่ 4

การดำเนินงาน

4.1 การอ่านค่า รูปภาพ วีดีโอ

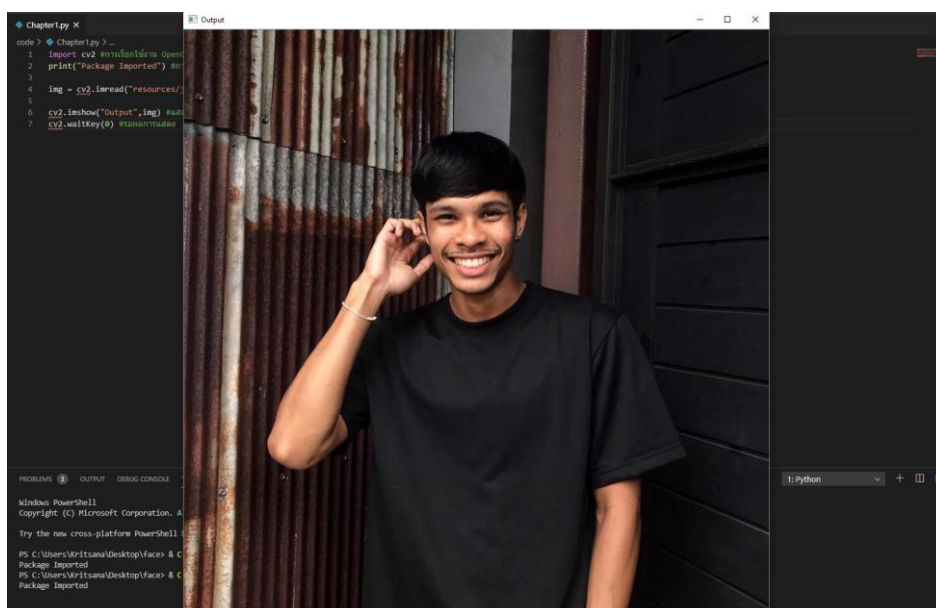
4.1.1 การอ่านค่ารูปภาพ

โค้ดการอ่านค่าของรูปภาพ

```
Chapter1.py X
code > Chapter1.py > ...
1 import cv2 #การเรียกใช้งาน OpenCv
2 print("Package Imported") #การสั่งให้แสดงค่าของPackageที่เรียกใช้
3
4 img = cv2.imread("resources/job.jpg") #เรียกใช้งานรูปภาพที่ต้องการอ่านค่า
5
6 cv2.imshow("Output",img) #แสดงรูปภาพ
7 cv2.waitKey(0) #รอผลการแสดง
```

รูปที่ 4.1 รูปภาพโค้ดการอ่านค่าของรูปภาพ

ผลลัพธ์ของการอ่านค่าที่แสดงผลจากโค้ด



รูปที่ 4.2 รูปภาพผลลัพธ์ของการอ่านค่าของรูปภาพ

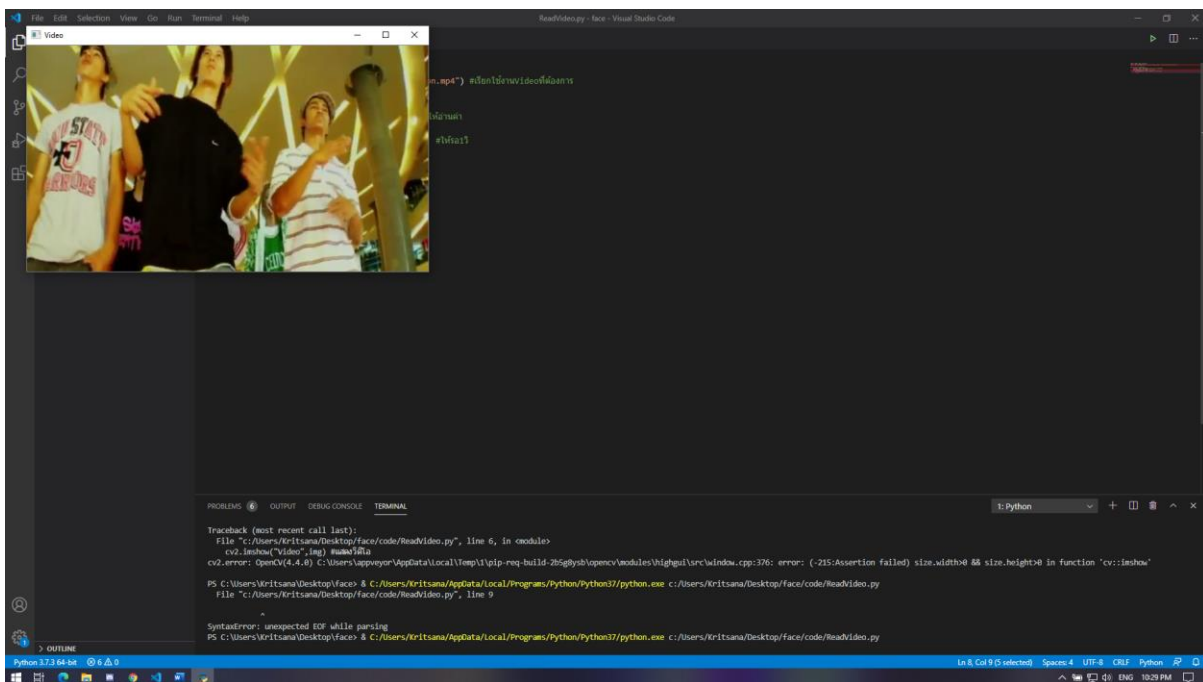
4.1.2 การอ่านค่า วิดีโอ

โค้ดการอ่านค่าของวิดีโอ

```
code > ReadVideo.py > ...
1 import cv2 #เรียกใช้OpenCV
2 cap = cv2.VideoCapture("resources/cartoon.mp4") #เรียกใช้งานVideoที่ต้องการ
3
4 while True: #กำหนดตัวแปล
5     success, img = cap.read() #หากถูกต้องให้อ่านค่า
6     cv2.imshow("Video",img) #แสดงวิดีโอ
7     if cv2.waitKey(1) & 0xFF ==ord('q'): #ให้รอ1
8         break
```

รูปที่ 4.3 รูปภาพโค้ดการอ่านค่าของวิดีโอ

ผลลัพธ์ของการอ่านค่าที่แสดงผลจากโค้ด



รูปที่ 4.4 รูปภาพผลลัพธ์ของการอ่านค่าของรูปภาพ

4.2 พื้นฐานการจับใบหน้า

สร้างดาต้าเซตให้แก่ใบหน้าโดยการนำดาต้าเซตจาก git hub

<https://github.com/Itseez/opencv/tree/master/data/haarcascades> เพื่อสร้างพื้นที่ใบหน้าโดยตั้งชื่อโฟลเดอร์ Cascades

Cascades	2/24/2018 2:52 AM	File folder	
faceDetection.py	11/1/2020 3:44 PM	Python File	2 KB
faceEyeDetection.py	11/1/2020 3:44 PM	Python File	2 KB
faceSmileDetection.py	11/1/2020 3:44 PM	Python File	2 KB
faceSmileEyeDetection.py	11/1/2020 3:45 PM	Python File	2 KB

รูปที่ 4.5 สร้างดาต้าเซตให้การจับใบหน้า (1)

โหลดดาต้าเก็บไว้เพื่อเรียกใช้งาน

haarcascade_eye.xml	2/24/2018 2:52 AM	XML Document	334 KB
haarcascade_frontalface_default.xml	2/24/2018 2:52 AM	XML Document	909 KB
haarcascade_smile.xml	2/24/2018 2:52 AM	XML Document	185 KB

รูปที่ 4.6 สร้างดาต้าเซตให้การจับใบหน้า (2)

โค้ดการอ่านค่าของใบหน้าและสร้างกล่องบนใบหน้าเพื่อจดจำ

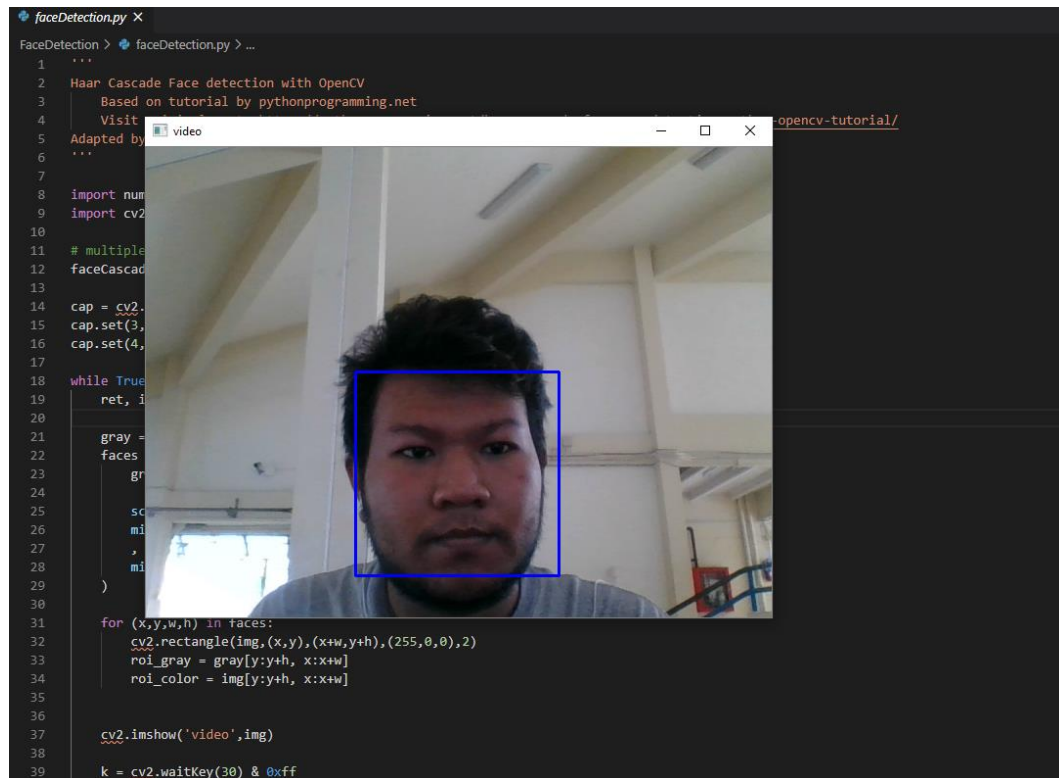
```

1  # faceDetection.py
2  """
3  Haar Cascade Face detection with OpenCV
4  Based on tutorial by pythonprogramming.net
5  Visit original post: https://pythonprogramming.net/haar-cascade-face-eye-detection-python-opencv-tutorial/
6  ...
7  ...
8  ...
9  ...
10 ...
11 # multiple cascades: https://github.com/Itseez/opencv/tree/master/data/haarcascades
12 faceCascade = cv2.CascadeClassifier('cascades/haarcascade_frontalface_default.xml')
13
14 cap = cv2.VideoCapture(0)
15 cap.set(3,640) # width of frame
16 cap.set(4,480) # height of frame
17
18 while True:
19     ret, img = cap.read()
20
21     gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
22     faces = faceCascade.detectMultiScale(
23         gray,
24         scaleFactor=1.2,
25         minNeighbors=5,
26         minSize=(20, 20)
27     )
28
29     for (x,y,w,h) in faces:
30         cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
31         roi_gray = gray[y:y+h, x:x+w]
32         roi_color = img[y:y+h, x:x+w]
33
34     cv2.imshow('video',img)
35
36     k = cv2.waitKey(30) & 0xff
37     if k == 27: # esc - ปิดหน้าต่าง
38         break
39
40 cap.release()
41 cv2.destroyAllWindows()
42

```

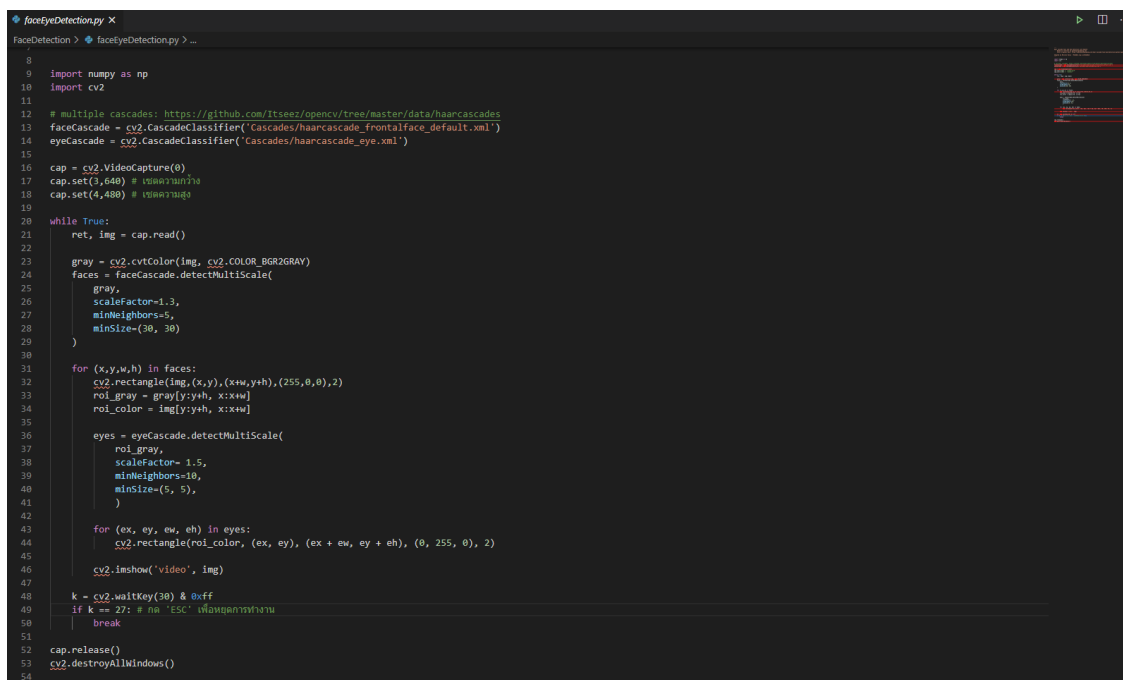
รูปที่ 4.7 โค้ดการอ่านค่าใบหน้าและสร้างกล่องบนใบหน้า

ผลลัพธ์ของการสร้างกล่องบนใบหน้า



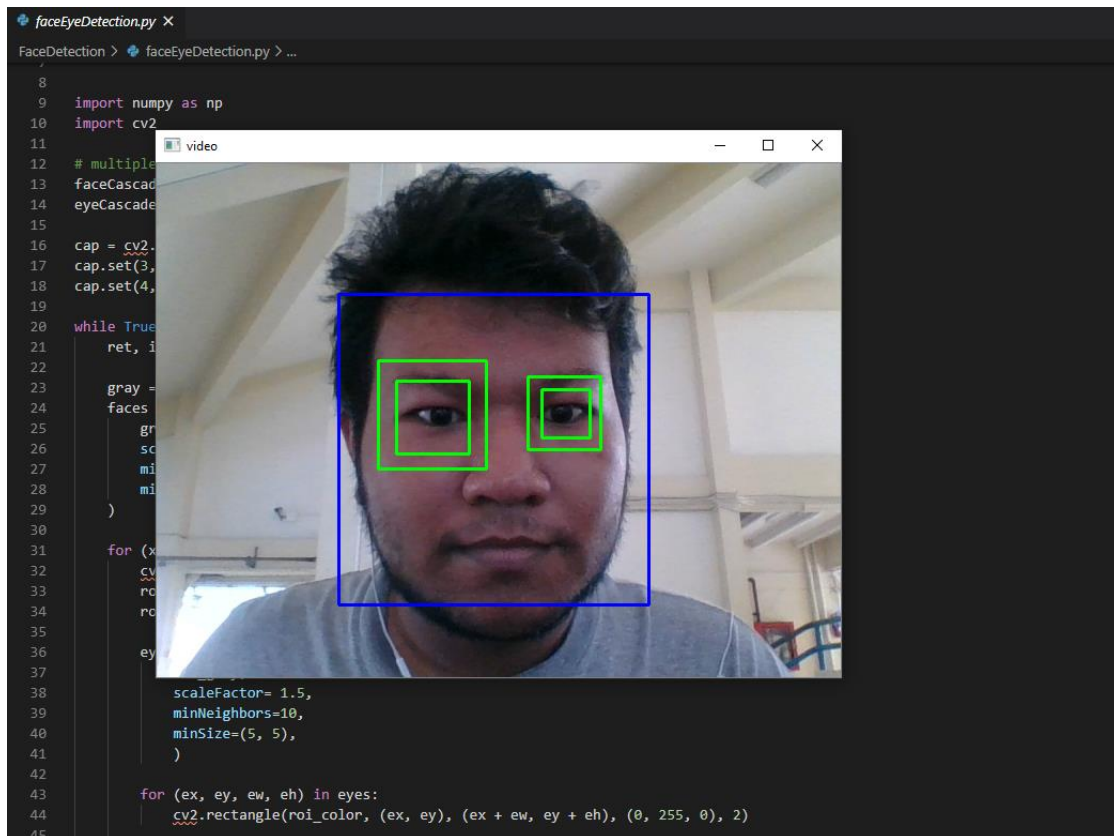
รูปที่ 4.8 ผลลัพธ์ของการสร้างกล่องบนใบหน้า

โค้ดการอ่านค่าของดวงตาและสร้างพื้นที่ของดวงตา



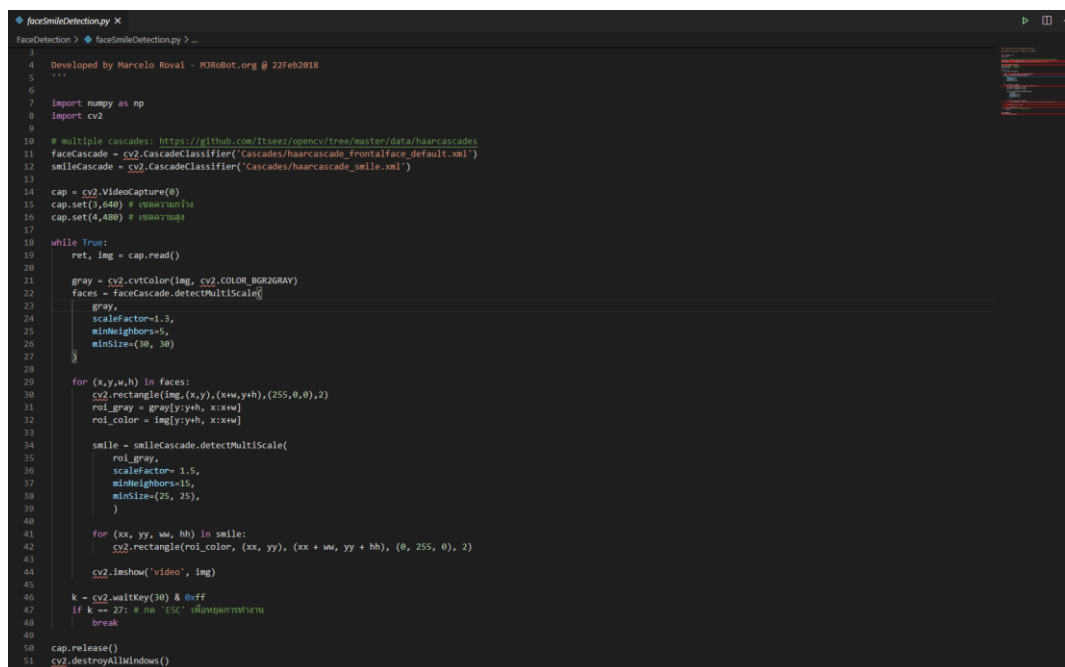
รูปที่ 4.9 โค้ดการอ่านค่าดวงตาและสร้างพื้นที่ของดวงตา

ผลลัพธ์ของการสร้างพื้นที่บนดวงตา



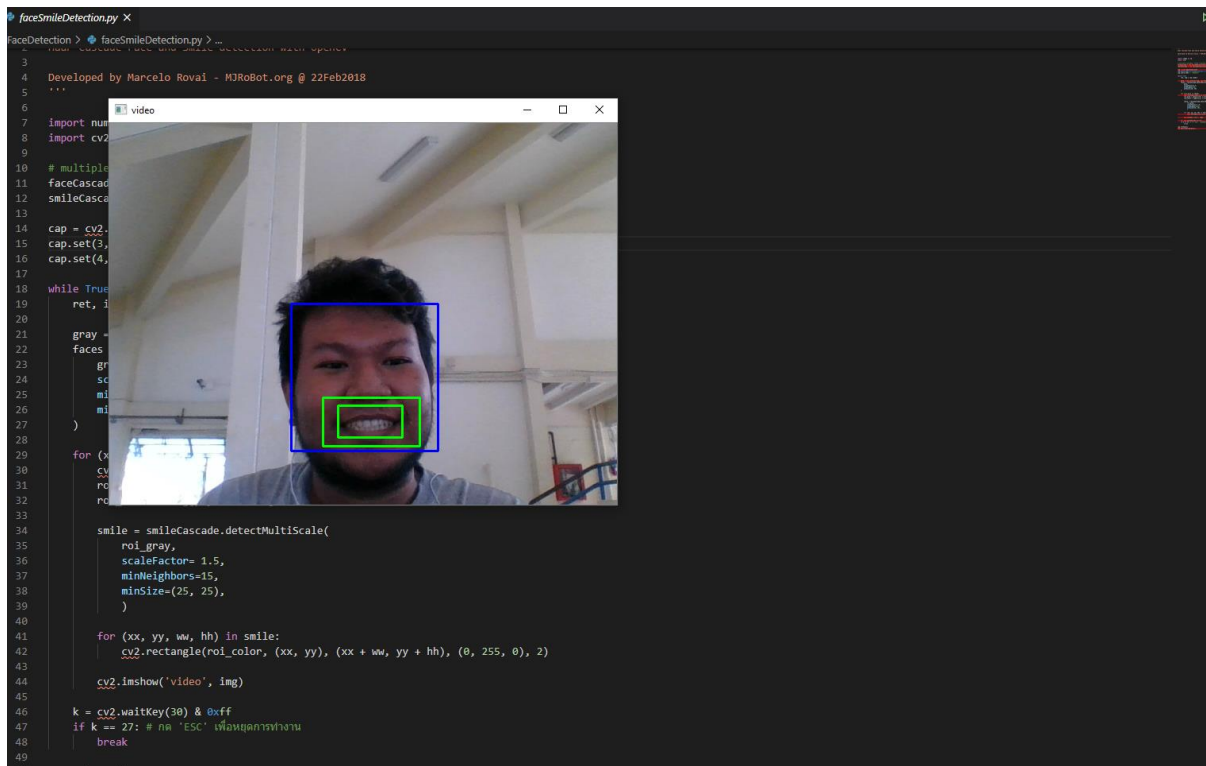
รูปที่ 4.10 ผลลัพธ์ของการสร้างพื้นที่บนดวงตา

โค้ดการสร้างพื้นที่บนปาก



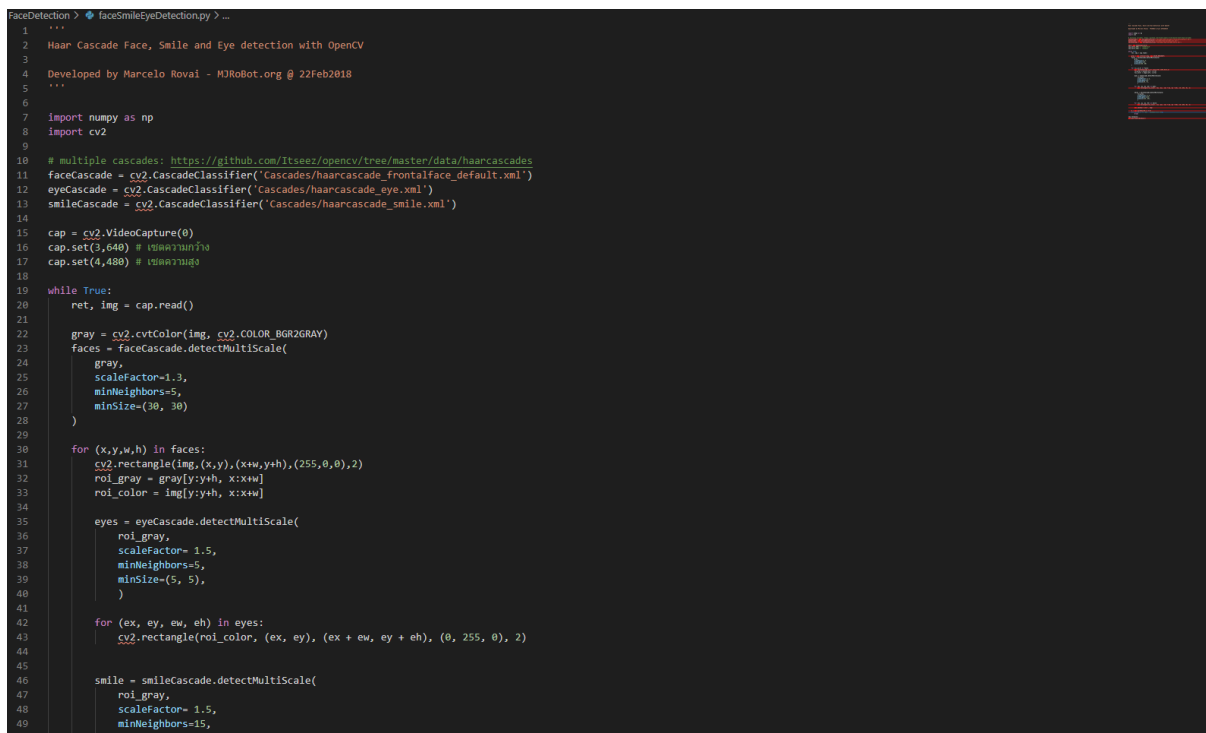
รูปที่ 4.11 โค้ดการสร้างพื้นที่บนปาก

ผลลัพธ์การสร้างพื้นที่บนปาก



รูปที่ 4.12 ผลลัพธ์การสร้างพื้นที่บนปาก

โค้ดผลรวมของการสร้างพื้นที่ของใบหน้า ดวงตาและปากเข้าด้วยกัน



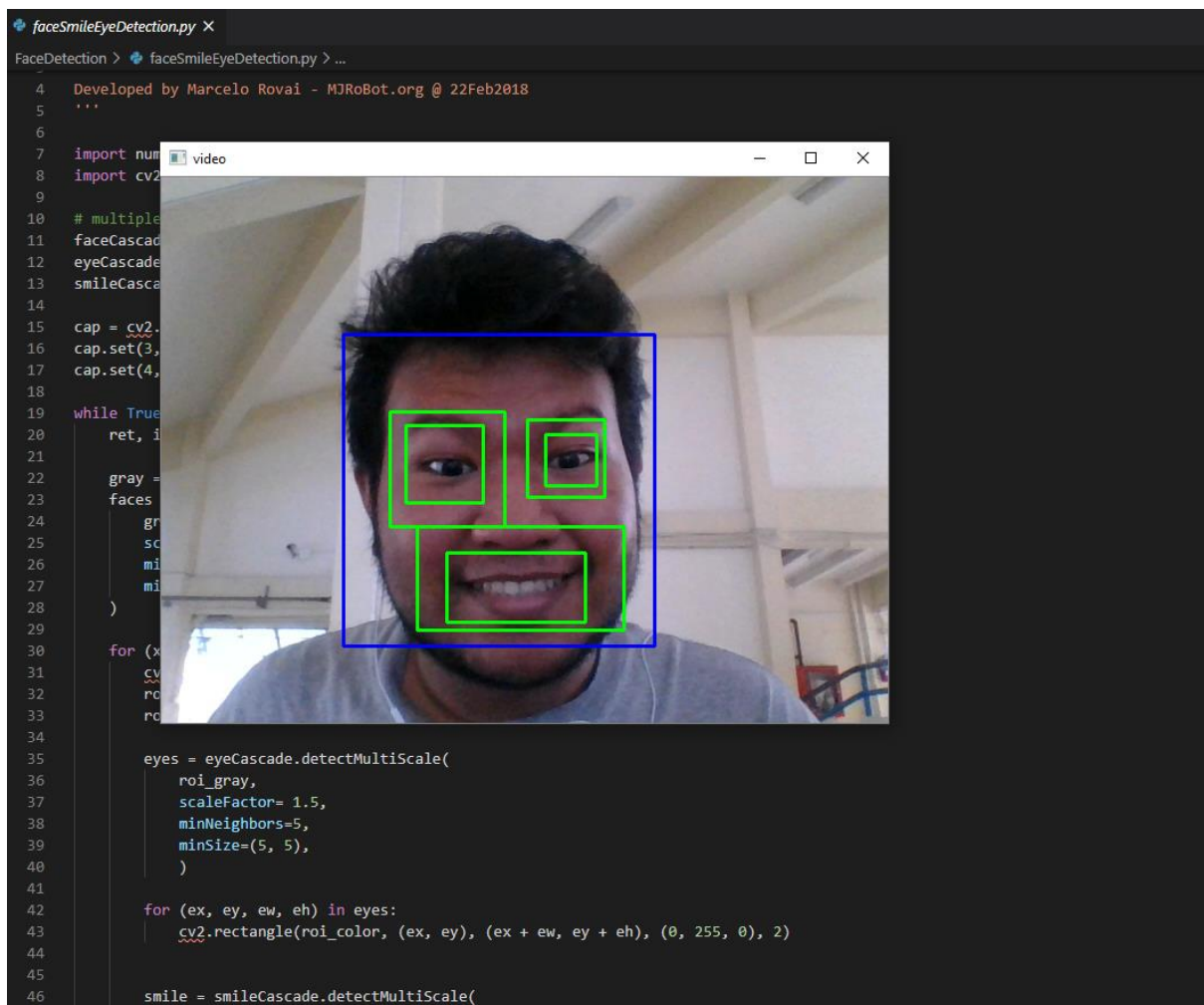
รูปที่ 4.13 โค้ดของการรวมใบหน้าดวงตาและปาก (1)


```

faceSmileEyeDetection.py X
FaceDetection > faceSmileEyeDetection.py > ...
35     eyes = eyeCascade.detectMultiScale(
36         roi_gray,
37         scaleFactor= 1.5,
38         minNeighbors=5,
39         minSize=(5, 5),
40     )
41
42     for (ex, ey, ew, eh) in eyes:
43         cv2.rectangle(roi_color, (ex, ey), (ex + ew, ey + eh), (0, 255, 0), 2)
44
45     smile = smileCascade.detectMultiScale(
46         roi_gray,
47         scaleFactor= 1.5,
48         minNeighbors=15,
49         minSize=(25, 25),
50     )
51
52     for (xx, yy, ww, hh) in smile:
53         cv2.rectangle(roi_color, (xx, yy), (xx + ww, yy + hh), (0, 255, 0), 2)
54
55     cv2.imshow('video', img)
56
57     k = cv2.waitKey(30) & 0xff
58     if k == 27: # กด 'ESC' เพื่อหยุดการทำงาน
59         break
60
61 cap.release()
62 cv2.destroyAllWindows()
63
64

```

รูปที่ 4.14 โค้ดของการรวมใบหน้าดวงตาและปาก (2)



รูปที่ 4.15 ผลลัพธ์ของการรวมใบหน้า ดวงตาและปาก

4.1.4 การทำFace recognitionแบบสร้างdatasetของการตรวจจับและ train ให้กับโปรแกรมการตรวจจับใบหน้าโดยอัตโนมัติ

สร้างโฟลเดอร์ datasetและtrainerเพื่อรองรับการทำงานของโค้ดที่ตั้งไว้

Name	Date modified	Type	Size
dataset	10/25/2020 10:23 PM	File folder	
trainer	10/25/2020 12:12 AM	File folder	
01_face_dataset.py	11/1/2020 4:36 PM	Python File	2 KB
02_face_training.py	2/24/2018 2:52 AM	Python File	2 KB
03_face_recognition.py	10/25/2020 12:19 AM	Python File	3 KB
haarcascade_frontalface_default.xml	2/24/2018 2:52 AM	XML Document	909 KB

รูปที่ 4.16 รูปโฟลเดอร์ที่ใช้เพื่อรันโปรแกรม

โค้ดการทำดาต้าเซต

```

FacialRecognition > 01_face_dataset.py > ...
1
2
3 Capture multiple Faces from multiple users to be stored on a DataBase (dataset directory)
4 ==> Faces will be stored on a directory: dataset/ (if does not exist, pls create one)
5 ==> Each face will have a unique numeric integer ID as 1, 2, 3, etc
6
7 Based on original code by Anirban Kar: https://github.com/thecodacus/Face-Recognition
8 Developed by Marcelo Rovai - MJRoBot.org @ 21Feb18
9
10
11
12 import cv2
13 import os
14
15 cam = cv2.VideoCapture(0)
16 cam.set(3, 640) # ปรับความกว้าง
17 cam.set(4, 480) # ปรับความสูง
18
19 face_detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
20
21 # ล็อกหน้าจอในหน้าต่างเดียวและกด
22 face_id = input('\n enter user id end press <return> ==> ')
23
24 print("\n [INFO] Initializing face capture. Look the camera and wait ...")
25 # เริ่มบันทึกภาพของใบหน้า
26 count = 0
27
28 while(True):
29
30     ret, img = cam.read()
31
32     gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
33     faces = face_detector.detectMultiScale(gray, 1.3, 5)
34
35     for (x,y,w,h) in faces:
36
37         cv2.rectangle(img, (x,y), (x+w,y+h), (255,0,0), 2)
38         count += 1
39
40         # บันทึกข้อมูลลงในโฟลเดอร์ที่จัดตั้งไว้
41         cv2.imwrite('dataset/User.' + str(face_id) + '.' + str(count) + ".jpg", gray[y:y+h,x:x+w])
42
43         cv2.imshow('image', img)
44
45     k = cv2.waitKey(100) & 0xff # กด 'ESC' เพื่อออกเล็ก
46     if k == 27:
47         break
48     elif count >= 30: # บันทึกข้อมูลอย่าง 30 ใบหน้าแล้วหยุด
49         break

```

รูปที่ 4.17 รูปโค้ดการทำดาต้าเซต (1)

```

44
45     k = cv2.waitKey(100) & 0xff # กด 'ESC' เพื่อยกเลิก
46     if k == 27:
47         break
48     elif count >= 30: # บันทึกตัวอย่าง30ใบหน้าแล้วหยุด
49         break
50
51 # ทำการล้างข้อมูล
52 print("\n [INFO] Exiting Program and cleanup stuff")
53 cam.release()
54 cv2.destroyAllWindows()
55
56
57

```

รูปที่ 4.18 รูปโค้ดการทำดาต้าเซต (2)

ผลของการทำงานของโค้ดในการทำดาต้าเซต

```

enter user id end press <return> ==> 1

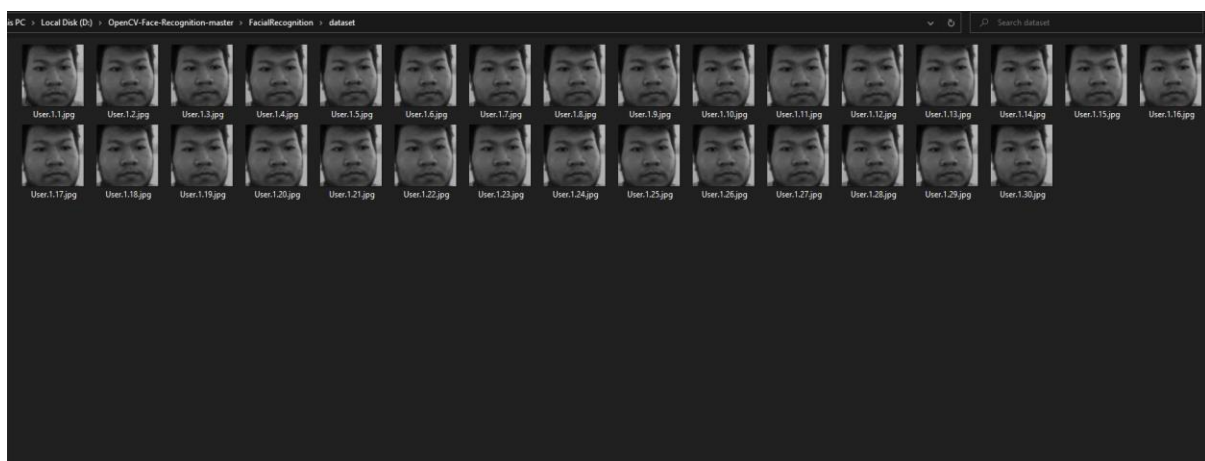
[INFO] Initializing face capture. Look the camera and wait ...

[INFO] Exiting Program and cleanup stuff
>>>

```

รูปที่ 4.19 ผลสร้างดาต้าเซตของบุคคลที่ 1

ไฟล์รูปที่ทำดาต้าเซตจะถูกจัดเก็บไว้ในโฟลเดอร์ของdataset



รูปที่ 4.20 ภาพที่ถูกบันทึกลงในโฟลเดอร์ดาต้าเซต

โค้ดการเทรนเพื่อให้ระบบทำการรันใบหน้า

```

FacialRecognition > O2_face_training.py > ...
1  ....
2  Training Multiple Faces stored on a DataBase:
3  ==> Each face should have a unique numeric integer ID as 1, 2, 3, etc
4  ==> LBPH computed model will be saved on trainer/ directory. (if it does not exist, pls create one)
5  ==> for using PIL, install pillow library with "pip install pillow"
6
7  Based on original code by Anirban Kar: https://github.com/thecodacus/Face-Recognition
8
9  Developed by Marcelo Roval - MJRoBot.org @ 21Feb18
10
11  ...
12
13  import cv2
14  import numpy as np
15  from PIL import Image
16  import os
17
18  # กำหนดชื่อไฟล์ฐานจากโฟลเดอร์
19  path = 'dataset'
20
21  recognizer = cv2.face.LBPHFaceRecognizer_create()
22  detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml");
23
24  # ฟังก์ชันการรับภาพ
25  def getImagesAndLabels(path):
26
27      imagePath = [os.path.join(path,f) for f in os.listdir(path)]
28      faceSamples=[]
29      ids = []
30
31      for imagePath in imagePath:
32
33          PIL_img = Image.open(imagePath).convert('L') # แปลงเป็นสีเทา
34          img_numpy = np.array(PIL_img,'uint8')
35
36          id = int(os.path.splitext(imagePath)[-1].split(".")[1])
37          faces = detector.detectMultiScale(img_numpy)
38
39          for (x,y,w,h) in faces:
40              faceSamples.append(img_numpy[y:y+h,x:x+w])
41              ids.append(id)
42
43      return faceSamples,ids
44
45  print ("\n [INFO] Training faces. It will take a few seconds. Wait ...")
46  faces,ids = getImagesAndLabels(path)
47  recognizer.train(faces, np.array(ids))

```

รูปที่ 4.21 โค้ดการเทรนการจดจำใบหน้า (1)

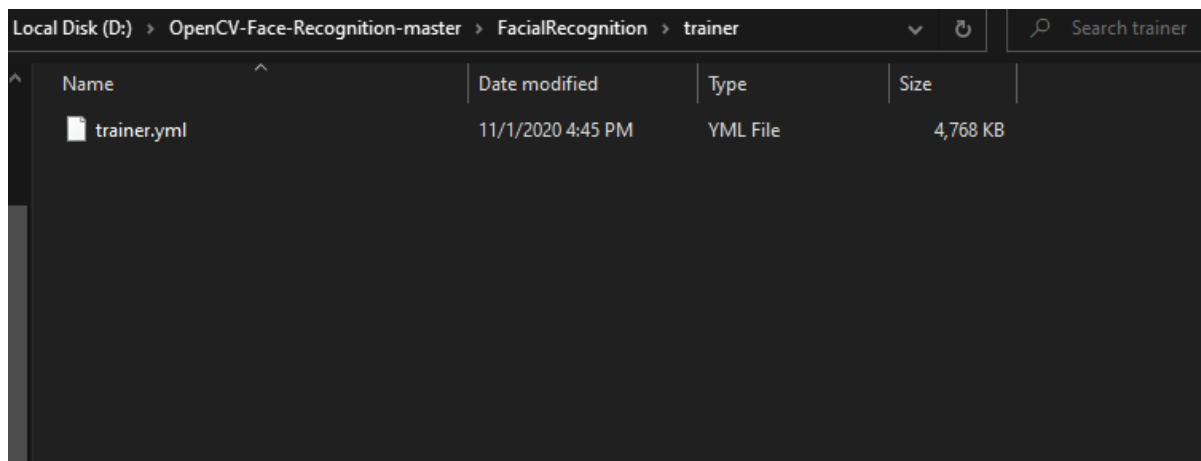
```

27  imagePath = [os.path.join(path,f) for f in os.listdir(path)]
28  faceSamples=[]
29  ids = []
30
31  for imagePath in imagePath:
32
33      PIL_img = Image.open(imagePath).convert('L') # แปลงเป็นสีเทา
34      img_numpy = np.array(PIL_img,'uint8')
35
36      id = int(os.path.splitext(imagePath)[-1].split(".")[1])
37      faces = detector.detectMultiScale(img_numpy)
38
39      for (x,y,w,h) in faces:
40          faceSamples.append(img_numpy[y:y+h,x:x+w])
41          ids.append(id)
42
43      return faceSamples,ids
44
45  print ("\n [INFO] Training faces. It will take a few seconds. Wait ...")
46  faces,ids = getImagesAndLabels(path)
47  recognizer.train(faces, np.array(ids))
48
49  # บันทึกโมเดลลงใน trainer
50  recognizer.write('trainer/trainer.yml') # recognizer.save() worked on Mac, but not on Pi
51
52  # ตัวเลขที่ต้องการจะเทรนแล้วจบการทำงาน
53  print("\n [INFO] {0} faces trained. Exiting Program".format(len(np.unique(ids))))
54

```

รูปที่ 4.22 โค้ดการเทรนการจดจำใบหน้า (2)

เมื่อการรันโค้ดผ่านจะทำการสร้างไฟล์ yml ลงในโฟลเดอร์เทรน



รูปที่ 4.23 โฟลเดอร์เทรนเนอร์

โค้ดของการตรวจจับใบหน้าและบอกเปอร์เซ็นต์ของความเหมือนที่เทรนและดัด้าเซตไว้

```

FacialRecognition > 03_face_recognition.py > ...
1
2 Real Time Face Recognition
3   -> Each face stored on dataset/ dir, should have a unique numeric integer ID as 1, 2, 3, etc
4   -> LBPH computed model (trained faces) should be on trainer/ dir
5 Based on original code by Anirban Kar: https://github.com/thecodacus/Face-Recognition
6
7 Developed by Marcelo Rovai - MJRoBot.org @ 21Feb18
8
9 ...
10
11 import cv2
12 import numpy as np
13 import os
14
15 recognizer = cv2.Face_LBPHFaceRecognizer_create()
16 recognizer.read('trainer/trainer.yml')
17 cascadePath = "haarcascade_frontalface_default.xml"
18 faceCascade = cv2.CascadeClassifier(cascadePath);
19
20 font = cv2.FONT_HERSHEY_SIMPLEX
21
22 # ชื่อไฟล์
23 id = 0
24
25 # ชื่อที่เกี่ยวข้องกับรหัสตัวอย่าง เช่น id1=game , id2=job
26 names = ['None', 'Game', 'Job', 'Ilza', 'Z', 'W']
27
28 # เริ่มต้นการจับภาพและวิธีโอแบบเรียลไทม์
29 cam = cv2.VideoCapture(0)
30 cam.set(3, 640) # เซตขนาดของดตามกว้าง
31 cam.set(4, 480) # เซตขนาดของดตามสูง
32
33 # ปรับขนาดหน้าต่างการจับภาพ
34 minW = 0.1*cam.get(3)
35 minH = 0.1*cam.get(4)
36
37 while True:
38
39     ret, img =cam.read()
40
41
42     gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
43
44     faces = faceCascade.detectMultiScale(
45         gray,
46         scaleFactor = 1.2,
47         minNeighbors = 5,
48         minSize = (int(minW), int(minH)),
49     )
50

```

รูปที่ 4.24 โค้ดของการทำการจดจำใบหน้า (1)

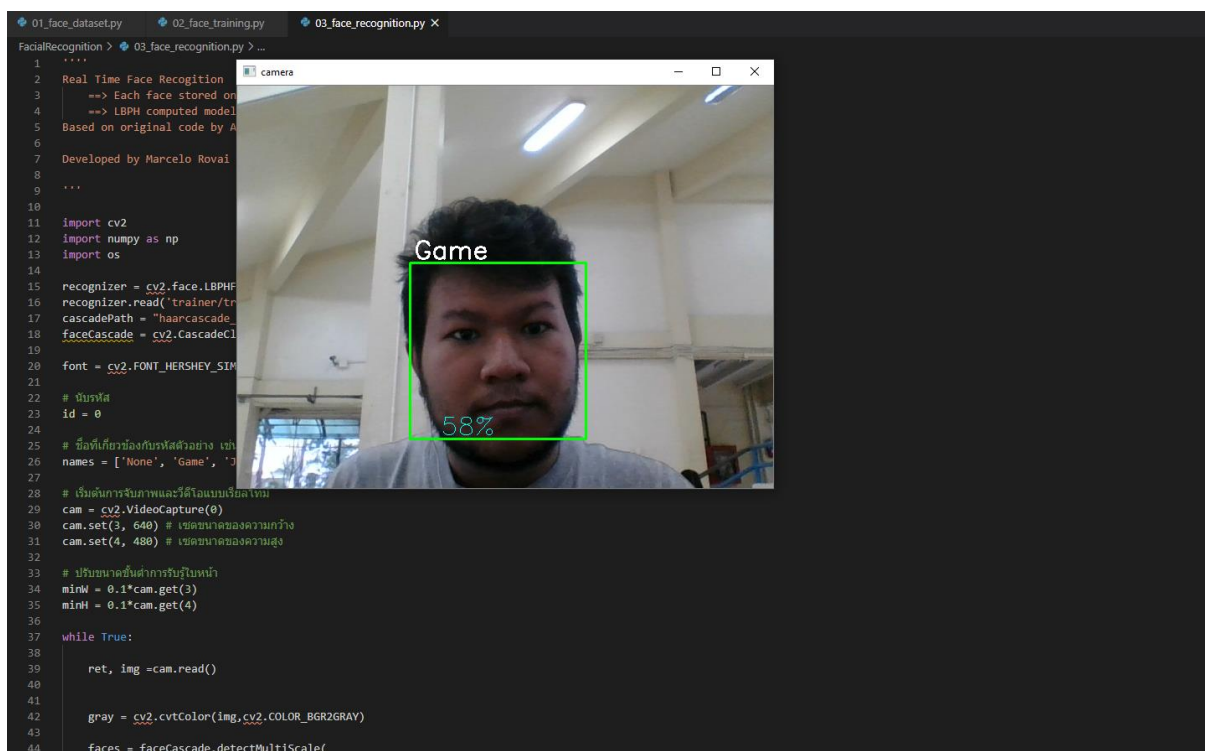
```

50
51 for(x,y,w,h) in faces:
52
53     cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)
54
55     id, confidence = recognizer.predict(gray[y:y+h,x:x+w])
56
57     # ตรวจสอบความเหมือนของใบหน้าก็เปลี่ยน
58     if (confidence < 100):
59         id = names[id]
60         confidence = " {0}%".format(round(100 - confidence))
61     else:
62         id = "unknown"
63         confidence = " {0}%".format(round(100 - confidence))
64
65     cv2.putText(img, str(id), (x+5,y-5), font, 1, (255,255,255), 2)
66     cv2.putText(img, str(confidence), (x+5,y+h-5), font, 1, (255,255,0), 1)
67
68     cv2.imshow('camera',img)
69
70     k = cv2.waitKey(10) & 0xff # กด 'ESC' เพื่อจบการทำงาน
71     if k == 27:
72         break
73
74 # เพื่อล้างข้อมูล
75 print("\n [INFO] Exiting Program and cleanup stuff")
76 cam.release()
77 cv2.destroyAllWindows()
78

```

รูปที่ 4.25 โค้ดของการทำการจดจำใบหน้า (2)

ผลลัพธ์ของการทำจดจำใบหน้าและบอกเปอร์เซ็นต์ความเหมือนของดาด้าเซตที่ได้จัดทำไว้



รูปที่ 4.26 ผลลัพธ์ของการทำจดจำใบหน้าและบอกเปอร์เซ็นต์ความใกล้เคียงกับดาด้าเซต

บทที่ 5

สรุปผลการดำเนินงาน

ประวัติโดยย่อของ OpenCV OpenCV เป็นโครงการริเริ่มการวิจัยของอินเทลเพื่อแนะนำแอปพลิเคชันที่ใช้ CPU มาก ได้เปิดตัวอย่างเป็นทางการเมื่อปี 2542 นปีพ. ศ. 2549 เป็นฉบับแรก OpenCV 1.0 ได้รับการปล่อยออกมา ในเดือนตุลาคมปี 2552 OpenCV 2 ได้รับการเผยแพร่เป็นรุ่นที่สอง ในเดือนสิงหาคม 2555 OpenCV ดำเนินการโดยองค์กร OpenCV องค์กรไม่แสวงหาผลกำไร

จากการทำงานของ OpenCV เป็นไปได้ตามเนื้อหาที่สามารถทำได้ตามการทำงานของ Computer Vision หมายถึงการหมายถึงการอธิบายถึงการมองเห็นและการวิเคราะห์ที่ได้จากสิ่งที่คอมพิวเตอร์เห็น การทำความเข้าใจภาพแบบ 3D จากภาพแบบ 2D ในแง่ของคุณสมบัติของโครงสร้างที่มีอยู่ในภาพ และมีความเกี่ยวข้องกับการสร้างแบบจำลองและเทียบกับวิสัยทัศน์ของมนุษย์โดยใช้ซอฟต์แวร์และฮาร์ดแวร์คอมพิวเตอร์ สำคัญคือ Image Processing หรือการประมวลผลภาพ เน้นการจัดการรูปภาพ Pattern Recognition อธิบายถึงการแยกแยะว่าเป็นวัตถุ Photogrammetry เป็นเรื่องเกี่ยวกับการวัดสิ่งต่าง ๆ จากภาพ

การประยุกต์ใช้ OpenCV ในการทำสิ่งต่าง ๆ เช่น การกำหนดตำแหน่งหุ่นยนต์โดยอัตโนมัติ, การนำทาง, การหลบหลีก การเลี้ยงอุปสรรค, การประกอบกร, การจัดการ (เช่น หุ่นยนต์ PUMA), Human Robot Interaction (HRI) หุ่นยนต์อัจฉริยะเพื่อโต้ตอบกับและรับใช้ผู้คน, การจำแนกและการตรวจหา (เช่นการจำแนกโรคหรือเซลล์และการตรวจหาเนื้องอก), การแยกแยะ การแบ่งเป็นส่วนๆ 2D / 3D, การฟื้นฟูวัยมนุษย์ 3D (MRI หรืออัลตราซาวด์), การผ่าตัดหุ่นยนต์ด้วยการมองของเครื่อง, การตรวจสอบทางอุตสาหกรรม (การตรวจจับข้อบกพร่อง), การประกอบชิ้นส่วน, การอ่านฉลากบาร์โค้ดและแพคเกจ, การเรียงลำดับวัตถุ, การรู้จำ ความเข้าใจในเอกสาร (เช่น OCR), Biometrics (ม่านตาพิมพ์ลายนิ้วมือจดจำใบหน้า), การเฝ้าระวัง การตรวจจับกิจกรรมหรือพฤติกรรมที่น่าสงสัยบางอย่าง, รถยนต์ไร้คนขับ, ความปลอดภัย เช่น การตรวจสอบความคืบหน้าของโปรแกรมควบคุม

การประมวลผลภาพโมดูลนี้ครอบคลุมการดำเนินการประมวลผลภาพต่าง ๆ เช่นการกรอกรูปภาพการแปลงรูปทรงเรขาคณิตการแปลงพื้นที่สีฮิสโตแกรมเป็นต้นในไลบรารี Java ของ OpenCV โมดูลนี้จะรวมอยู่ในชุดที่มีชื่อ org.opencv.imgproc, วิดีโอโมดูลนี้ครอบคลุมแนวคิดการวิเคราะห์วิดีโอเช่นการประมาณค่าการเคลื่อนที่การลบพื้นหลังและการติดตามวัตถุ ในไลบรารี Java ของ OpenCV โมดูลนี้จะรวมอยู่ในแพ็คเกจที่มีชื่อ org.opencv.video, วิดีโอ I / O โมดูลนี้อธิบายถึงตัวจับสัญญาณวิดีโอและตัวแปลงสัญญาณที่ใช้ไลบรารี OpenCV ในไลบรารี Java ของ OpenCV โมดูลนี้จะรวมอยู่ในแพ็คเกจที่มีชื่อ org.opencv.videoio calib3d โมดูลนี้ประกอบด้วยอัลกอริทึมเกี่ยวกับอัลกอริทึมเรขาคณิต มุมมองพื้นฐานการสอบเทียบกล้องถ่ายรูปเดี่ยว

และสเตอริโอการประมาณค่าของวัตถุการติดต่อแบบสเตอริโอและองค์ประกอบของการสร้างภาพ 3D ในไลบรารี Java ของ OpenCV โมดูลนี้จะรวมอยู่ในแพ็คเกจที่มีชื่อ `org.opencv.calib3d`, `features2d` โมดูลนี้มีแนวคิดในการตรวจจับคุณลักษณะและคำอธิบาย ในไลบรารี Java ของ OpenCV โมดูลนี้จะรวมอยู่ในชุดที่มีชื่อ `org.opencv.features2d`, `Objdetect` โมดูลนี้รวมถึงการตรวจจับวัตถุและอินสแตนซ์ของคลาสที่กำหนดไว้ล่วงหน้า เช่นใบหน้าดวงตาแกว้คนรถยนต์ เป็นต้นในไลบรารี Java ของ OpenCV โมดูลนี้จะรวมอยู่ในชุดที่มีชื่อ `org.opencv.objdetect`, `Highgui` นี่คือนินเตอร์เฟซที่ง่ายต่อการใช้งานที่มีความสามารถแบบ UI ที่เรียบง่าย ในไลบรารี Java ของ OpenCV คุณลักษณะของโมดูลนี้จะรวมอยู่ในแพ็คเกจที่แตกต่างกันสองแบบคือ `org.opencv.imgcodecs` และ `org.opencv.videoio`

ลิงค์โค้ดของทางกลุ่มได้จัดทำไว้ : https://github.com/theriosza555/Face_Recognition

บรรณานุกรม

- [1] Navapat Jesadapatrakul. (2563,สิงหาคม 3). Image Processing [Online]. แหล่งที่มา:
<https://medium.com/tni-university/image-processing-981c65c26289>
- [2] Nuttakan Chuntra. (2563,สิงหาคม 3). OpenCV คืออะไร? [Online]. แหล่งที่มา:
<https://medium.com/@nut.ch40/opencv-คืออะไร-8771e2a4c414>
- [3] ดร.กอบเกียรติ สระอุบล. (2563,มิถุนายน 1). Data ScienceและAI: Machine Learningด้วยPython
 กรุงเทพฯ: หสม มีเดีย เนทเวิร์ค
- [4] Codequs. (2563,สิงหาคม 3). Python Tutorial: Image processing with Python (Using OpenCV)
 [Online]. แหล่งที่มา: <https://codequs.com/p/S1PCba3DV/python-tutorialimage-processing-with-python-using-opencv#Convert-image-to-grayscale-Black-White>
- [5] Undermind. (2563,กันยายน 15). ข้อดีของ opencv และ จะเขียนได้อย่างไร[Online]. แหล่งที่มา:
http://imageopencv.blogspot.com/2010/02/opencv_11.html
- [6] ProgrammingKnowledge. (2563,กันยายน 15). OpenCV Python Tutorial For Beginners 2 -
 How to Install OpenCV for Python on Windows 10[Online]. แหล่งที่มา:
https://www.youtube.com/watch?v=d3AT9EGp4iw&ab_channel=ProgrammingKnowledge
- [7] Murtaza's Workshop - Robotics and AI. (2563,กันยายน 18). LEARN OPENCV in 3 HOURS with
 Python | Including 3x Example Projects (2020)[Online]. แหล่งที่มา:
https://www.youtube.com/watch?v=WQeoO7MI0Bs&fbclid=IwAR30jMOUqc4KRob8w5UsP7oel oelaMGSjhQFgG5qYHdH2hdmlpK73inm_oQ&ab_channel=Murtaza%27sWorkshop-RoboticsandAI
- [8] phyblas. (2563,ตุลาคม 10). opencv-python เบื้องต้น บทที่ ๑: บทนำ[Online]. แหล่งที่มา:
<https://phyblas.hinaboshi.com/oshi01>
- [9] likegeeks. (2563,ตุลาคม 12). Python Image Processing Tutorial (Using OpenCV)[Online].
 แหล่งที่มา: <https://likegeeks.com/python-image-processing/>

บรรณานุกรม(ต่อ)

- [10] Sarayut Nonsiri. (2563,ตุลาคม 12). ภาษาโปรแกรม Python คืออะไร ?[Online]. แหล่งที่มา: <https://www.9experttraining.com/articles/python-คืออะไร>
- [11] Developer Team. (2563,ตุลาคม 14). พื้นฐานการใช้ NumPyใน Python 3[Online]. แหล่งที่มา: <https://www.borntodev.com/2020/04/16/พื้นฐานการใช้-numpy-ใน-python-3/>
- [12] Asst. Prof. Banyapon Poolsawas (2563,ตุลาคม 14). การทำ Face Detection บน Video ด้วย Python และ OpenCV[Online]. แหล่งที่มา: <http://www.daydev.com/face-recognition/face-detection-on-recorded-videos-using-opencv-in-python.html>
- [13] alalek. (2563,ตุลาคม 17). opencv/data/haarcascades/[Online]. แหล่งที่มา: <https://github.com/opencv/opencv/tree/master/data/haarcascades>