



ระบบจดจำใบหน้าและระบุตัวตนด้วยกล้อง
DEEP LEARNING FACE RECOGNITION BY CAMERA

จัดทำโดย

นายศิรินทร์ งามดี

นายธรา รอบรู้

โครงการนี้เป็นส่วนหนึ่งของวิชาโครงการ หลักสูตรประกาศนียบัตรวิชาชีพชั้นสูง

สาขาวิชาเทคโนโลยีสารสนเทศ

วิทยาลัยเทคโนโลยีพระยาศรีสุนทร

ปีการศึกษา 2562

ระบบจดจำใบหน้าและระบุตัวตนด้วยกล้อง
DEEP LEARNING FACE RECOGNITION BY CAMERA

จัดทำโดย

นายศิรินทร์	งามดี
นายธรา	รอบรู้

โครงการนี้เป็นส่วนหนึ่งของวิชาโครงการ หลักสูตรประกาศนียบัตรวิชาชีพชั้นสูง
สาขาวิชาเทคโนโลยีสารสนเทศ
วิทยาลัยเทคโนโลยีอรรถวิทย์พัฒนวิชาการ
ปีการศึกษา 2562

COPYRIGHT 2019

COLLEGE OF INFORMATION TECHNOLOGY

ATTAWIT COMMERCIAL TECHNOLOGY COLLEGE



ชื่อโครงการภาษาไทย

ระบบจดจำใบหน้าและระบุตัวตนด้วยกล้อง

ชื่อโครงการภาษาอังกฤษ

Deep Learning Face Recognition by Camera

โดย 1. นายศิรินทร์

งามดี

รหัสประจำตัว 40952

2. นายธรา

รอบรู้

รหัสประจำตัว 41148

คณะกรรมการอนุมัติให้เอกสารโครงการฉบับนี้เป็นส่วนหนึ่งของการศึกษา วิชาโครงการตามหลักสูตรประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.) สาขาวิชาเทคโนโลยีสารสนเทศ วิทยาลัยเทคโนโลยีอรรถวิทย์พัฒนการ (ATC)

(อาจารย์คุณานนท์ สุขเกษม)

อาจารย์ที่ปรึกษา

(อาจารย์คุณานนท์ สุขเกษม)

หัวหน้าสาขาวิชาเทคโนโลยีสารสนเทศ

บทคัดย่อ

หัวข้อโครงการ	ระบบจดจำใบหน้าและระบุตัวตนด้วยกล้อง <i>Deep Learning Face Recognition by Camera</i>		
ผู้จัดทำโครงการ	1. นายศิรินทร์	งามดี	รหัสประจำตัว 40952
	2. นายธรา	รอบรู้	รหัสประจำตัว 41148
อาจารย์ที่ปรึกษา	อาจารย์คุณานนท์ สุขเกษม		
สาขาวิชา	เทคโนโลยีสารสนเทศ		
สถาบัน	วิทยาลัยเทคโนโลยีอรรถวิทย์พัฒนศึกษา ปีการศึกษา 2562		

บทคัดย่อ

ระบบรู้จำใบหน้า (Face Recognition) คือกระบวนการที่ได้นำภาพใบหน้าที่ตรวจจับได้ และประมวลผลแล้วจากขั้นตอนการตรวจจับใบหน้า มาเปรียบเทียบกับฐานข้อมูลของใบหน้าเพื่อระบุว่าใบหน้าที่ตรวจจับได้ตรงกับบุคคลใด เทคโนโลยีจดจำใบหน้านั้นส่วนมากจะใช้ในการรักษาความปลอดภัยแต่เราสามารถประยุกต์ไปใช้ในงานสารสนเทศด้านอื่น ๆ ได้

เราจึงคิดว่าการจดจำใบหน้าที่สามารถระบุชื่อว่าคุณคนนั้นเป็นใครเราจะสามารถแยกแยะบุคคลได้อีกทั้งยังสามารถนำไปใช้เช็คชื่อนักเรียนในห้องเรียนได้, ใช้ในร้านอาหารแยกใบหน้าพนักงานกับลูกค้าได้เพื่อตรวจดูพฤติกรรม การเทคโนโลยีการจดจำใบหน้า นอกจากจะนำมาใช้ในการรักษาความปลอดภัยแล้ว ยังสามารถนำมาใช้งานได้อีกหลากหลายกรณี เช่น ออฟฟิศ ห้างสรรพสินค้า อาคารจอดรถ การมาทำงานของพนักงาน หน่วยงาน องค์กรขนาดใหญ่ อีกทั้งยังสามารถที่จะนำมาใช้งานในด้านการวิเคราะห์การตลาดได้อีกหลายรูปแบบ อาทิ พฤติกรรมลูกค้าที่เข้ามาใช้บริการภายในร้าน

กิตติกรรมประกาศ

โครงการฉบับนี้สำเร็จลุล่วงไปได้ดี ก็เพราะได้รับความเมตตากรุณา ความช่วยเหลือจาก อาจารย์คุณานนท์ สุขเกษม ที่ได้ให้คำแนะนำ และคำปรึกษา เกี่ยวกับการสร้างและออกแบบระบบจดจำใบหน้าและระบุตัวตนด้วยกล้อง Deep Learning Face Recognition by Camera มาโดยตลอด ผู้ทำโครงการรู้สึกซาบซึ้งในความอนุเคราะห์จากท่านเป็นอย่างมาก และกราบขอบพระคุณเป็นอย่างสูง

ผู้ทำโครงการขอกราบขอบพระคุณ บิดา มารดา และเพื่อน ๆ พี่ ๆ ที่ให้กำลังใจและให้โอกาสบุตรได้รับการศึกษาในระดับต่าง ๆ จนกระทั่งได้สำเร็จการศึกษาระดับประกาศนียบัตรวิชาชีพชั้นสูง รวมทั้งคณะอาจารย์ทุกท่านในสาขาวิชาเทคโนโลยีสารสนเทศ ที่ได้ประสิทธิ์ประสาทวิชาความรู้และคำสั่งสอน ให้กับผู้ทำโครงการในการเรียนทุกระดับชั้น

ขอขอบพระคุณครอบครัวที่ให้การช่วยเหลือในทุก ๆ ด้าน จนทำให้ประสบความสำเร็จลุล่วงไปได้ด้วยดี ขอบขอบคุณพี่ ๆ พี่ ๆ ทุกคน ในสาขาเทคโนโลยีสารสนเทศ วิทยาลัยเทคโนโลยีอรรถวิทย์พัฒนชกร ที่คอยให้การช่วยเหลือการทำโครงการฉบับนี้

สุดท้ายความรู้และประโยชน์ที่ได้รับจากโครงการฉบับนี้ ผู้ทำโครงการ ขอมอบความดีความชอบ ที่ได้นี้ให้แก่ผู้มีพระคุณทุกท่าน

ศิรินทร์	งามดี
ธารา	รอบรู้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
กิตติกรรมประกาศ.....	II
สารบัญ.....	III
สารบัญตาราง.....	V
สารบัญรูป.....	VI
สารบัญรูป.....	VI I
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของโครงการ.....	2
1.3 ขอบเขตของโครงการ.....	2
1.4 ขั้นตอนการดำเนินงาน.....	2
1.5 ระยะเวลาการทำโครงการ.....	5
1.6 ประโยชน์ที่คาดว่าจะได้รับ.....	6
1.7 งบประมาณที่ใช้ในการทำโครงการ.....	6
บทที่ 2 แนวคิดและทฤษฎีที่เกี่ยวข้อง	7
2.1 หลักการทำงานของระบบรู้จำใบหน้า (Face Recognition).....	7
2.2 การตรวจจับใบหน้า (Face Detection).....	10
2.3 โปรแกรมใช้ในการแก้ไขและปรับแต่งโค้ด.....	18
2.4 ภาษาที่ใช้ในการเขียนโค้ด.....	21
2.5 ไลบรารีฟังก์ชันต่าง ๆ.....	24
บทที่ 3 วิธีการดำเนินโครงการ	30
3.1 ขั้นตอนการวางแผนและการเตรียมการ.....	30
3.2 การออกแบบ.....	34
3.3 การดำเนินการสร้าง.....	37
บทที่ 4 ผลการศึกษา	38
4.1 ขั้นตอนการออกแบบชิ้นงาน.....	38
4.2 ขั้นตอนการทำงาน.....	39
4.3 ขั้นตอนการทดสอบระบบจดจำใบหน้าและระบุตัวตนด้วยกล้อง.....	50

สารบัญ (ต่อ)

	หน้า
บทที่ 5 สรุปผลและข้อเสนอแนะการดำเนินโครงการ	51
5.1 วัตถุประสงค์ของโครงการ	51
5.2 ประโยชน์ที่คาดว่าจะได้รับ	51
5.3 ปัญหาที่ประสบในการดำเนินโครงการ	52
5.4 ผลการดำเนินโครงการ	52
5.5 อภิปรายผล	52
5.6 ข้อเสนอแนะ	52
บรรณานุกรม	53
ภาคผนวก	54
ภาคผนวก ก แบบเสนอร่างโครงการ	55
ภาคผนวก ข แบบประเมินความก้าวหน้า	64
ภาคผนวก ค คู่มือการติดตั้งและใช้งานระบบจดจำใบหน้าและระบุตัวตน	75
ภาคผนวก ง ประวัติผู้จัดทำ	84

สารบัญตาราง

ตารางที่	หน้า
1.1 ตารางแผนการดำเนินงาน.....	5
3.1 แสดงแผนการดำเนินโครงการ.....	31

สารบัญรูป

รูปที่	หน้า
1.1 รูปแสดง Flow Chart การดำเนินงาน	3
1.1 รูปแสดง Flow Chart การดำเนินงาน (ต่อ)	4
2.1 รูปตัวอย่างของใบหน้าไอเกน (Eigenfaces)	8
2.2 ตัวอย่างของคลาสที่ถูกจัดกลุ่ม (Classified) โดยใช้ LDA	9
2.3 ตัวอย่างของคลาสที่ถูกจัดกลุ่มโดยใช้ LDA (ต่อ)	9
2.4 Examples of the Haar features	11
2.5 The Integral Image trick	11
2.6 ตัวจำแนกประเภทแบบ cascaded (Cascaded classifier)	12
2.7 กระบวนการรู้จำใบหน้าด้วยวิธีทางสถิติ	12
2.8 ขอบเขตการพิจารณาของแต่ละลำดับชั้น	13
2.9 Template Datching methods	15
2.10 การกระจายตัวของข้อมูลตัวอย่าง	16
2.11 ตัวอย่างโครงข่ายประสาทเทียม	17
2.12 Visual Studio Code (วิหวล สตูดิโอ โค้ด)	18
2.13 Visual Studio Code (วิหวล สตูดิโอ โค้ด)	19
2.14 Visual Studio Code (วิหวล สตูดิโอ โค้ด)	20
2.15 Visual Studio Code (วิหวล สตูดิโอ โค้ด)	20
2.16 ภาษา Python	21
2.17 Simple Python	22
2.18 Comment	23
2.19 Statement	23
2.20 OpenCV (Open source Computer Vision)	24
2.21 Deep learning	25
2.22 Machine Learning	26
2.23 Tkinter (ทีเคอินเตอร์)	27
2.24 Vector and Matrix	28
2.25 การสร้างและเข้าถึงข้อมูลใน array	29
2.26 การเข้าถึงข้อมูลใน array หลายมิติ	29
3.1 แผนภาพ Flowchart ขั้นตอนการดำเนินงาน	32

สารบัญรูป

รูปที่	หน้า
3.2 แผนภาพ Flowchart ขั้นตอนการดำเนินงาน (ต่อ).....	33
3.3 หน้าเมนูของระบบรู้จำใบหน้า.....	34
3.4 หน้า Training ให้ระบบรู้จำใบหน้า.....	35
3.5 หน้าต่างระบบรู้จำใบหน้าและระบุตัวตนด้วยกล้อง.....	35
3.6 หน้าต่างกันหลักในขณะรอ.....	36
3.7 หน้าต่างอ้างอิงระบบ.....	36
3.9 การทำงานของระบบ.....	37
4.1 วิวสตูดิโอโค้ด Visual Studio Code.....	38
4.2 Import Library tkinter.....	39
4.3 สร้างหน้าเมนู.....	39
4.4 สร้างปุ่มเพื่อเก็บค่าคำสั่ง.....	40
4.5 ทำการสร้างฟังก์ชันของปุ่ม เกี่ยวกับ>ไลบรารี> ผู้ออกแบบ.....	40
4.6 ออกแบบระบบบันทึกใบหน้า.....	41
4.7 กำหนดที่เก็บของรูปภาพ,ขนาดไซส์ของรูปภาพ,จำนวนรูปภาพที่ต้องการ.....	41
4.8 Datasets > Faces > “ชื่อที่เรากรอกไว้”.....	42
4.9 การเรียกใช้กล้องจากภายนอกคือกล้อง Webcam.....	42
4.10 นับตัวแปร Count มาเปรียบเทียบกับ MAX_CAPTURE.....	43
4.11 Import Library เสริมมาช่วยในการทำงาน.....	44
4.12 ทำการสร้างฟังก์ชันเพื่อระบุ Path และ Out Path.....	44
4.13 สร้างเส้นทางเพื่อใช้เก็บไฟล์ Csv เพื่อเก็บข้อมูล.....	45
4.14 สร้างฟังก์ชันปรับขนาดรูป.....	45
4.15 ทำการประมวลผลภาพเพื่อเก็บลงในไฟล์ processed.csv.....	46
4.16 สร้างฟังก์ชันตรวจจับใบหน้าจากรูปภาพ.....	46
4.17 โมเดลจดจำใบหน้าและเก็บไฟล์รูปรวมใน SVM_MODEL ที่ชื่อว่า faces.lib.....	47
4.18 การเขียนโค้ดรายงานความแม่นยำของรูปภาพ.....	47
4.19 Report เป็นคะแนนยิ่งคะแนนมากจะจับใบหน้าได้ถูกต้องมากขึ้น.....	48
4.20 ระบบตรวจจับอาการหลับง่วงนอนหรือหลับใน.....	48
4.21 ทำการเรียกใช้งานตัวดีกรอบดวงตา.....	49
4.22 กำหนดขนาดของหน้าต่างและอ่านค่าของดวงตา.....	49

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ระบบรู้จำใบหน้า (Face Recognition) ถูกออกแบบมาให้ทำการเปรียบเทียบใบหน้าบุคคลกับฐานข้อมูลใบหน้าที่มีอยู่โดยอัลกอริทึมที่ใช้ในขั้นตอนการสร้างแม่แบบและขั้นตอนการเปรียบเทียบอาจแตกต่างกันไปแล้วแต่การออกแบบระบบของแต่ละระบบ แต่ไม่ว่าจะมี อัลกอริทึมในการทำงานในขั้นตอนการสร้างแม่แบบและขั้นตอนการเปรียบเทียบอย่างไร แต่ขั้นตอนการทำงานโดยรวมของระบบก็ยังคงเหมือนกันอยู่ การตรวจหาใช้วิธีตรวจหาตำแหน่งของ จมูก และ ตา ทั้งสองข้าง ในรูปของเมตริกซ์ แนวตั้งและแนวนอน

โดยทั่วไประบบรู้จำใบหน้าจะประกอบไปด้วย 2 ขั้นตอนหลักคือ การตรวจจับใบหน้า (Face Detection) และ ระบบรู้จำใบหน้า (Face Recognition) 1.1 การตรวจจับใบหน้า คือ กระบวนการค้นหาใบหน้าของบุคคลจากภาพหรือวิดีโอหลังจากนั้นก็จะทำการประมวลผลภาพใบหน้าที่ได้สำหรับขั้นตอนถัดไปเพื่อให้ภาพใบหน้าที่ตรวจจับได้ง่ายต่อการจำแนก และอัลกอริทึมที่ใช้ในการตรวจจับใบหน้าในปัจจุบันก็มีอยู่ด้วยกันหลายวิธีซึ่งอัลกอริทึมในการตรวจจับใบหน้าที่ดีนั้นมีส่วนช่วยในการจำแนก 1.2 ระบบรู้จำใบหน้า (Face Recognition) คือ กระบวนการที่ได้นำภาพใบหน้าที่ตรวจจับได้และประมวลผลแล้วจากขั้นตอนการตรวจจับใบหน้า มาเปรียบเทียบกับฐานข้อมูลของใบหน้าเพื่อระบุว่าใบหน้าที่ตรวจจับได้ตรงกับบุคคลใด เมื่อไม่นานมานี้มีนักพัฒนาได้พัฒนา Library บนภาษา Python ที่ช่วยทำให้ Face Recognition กลายเป็นเรื่องง่ายใช้เพียงแค่โค้ดไม่กี่บรรทัด เพียงเป็นแค่ผู้เริ่มต้นเขียนโปรแกรม ก็สามารถทำ Face Recognition ได้อย่างง่ายดาย

ดังนั้นทางผู้จัดทำเล็งเห็นว่าในอนาคต อุปกรณ์หลายอย่างที่เราใช้ในชีวิตประจำวันรวมทั้ง ทุยแจบ้าน ทุยแจรถ บัตรเอทีเอ็ม หรือแม้กระทั่งการใส่รหัสผ่านต่างๆ จะค่อยๆ หายไปโดยสิ่งที่มาแทนก็คือเทคโนโลยีรู้จำใบหน้า (Face recognition) และเทคโนโลยีจดจำใบหน้านั้นส่วนมากจะใช้ในการรักษาความปลอดภัยแต่เราสามารถประยุกต์ไปใช้ในงานสารสนเทศด้านอื่นๆ ได้เราจึงคิดว่าการจดจำใบหน้าที่สามารถระบุชื่อว่าคุณคนนั้นเป็นใครเราจะสามารถแยกแยะบุคคลได้อีกทั้งยังสามารถนำไปใช้เช็คชื่อนักเรียนในห้องเรียนได้, ใช้ในร้านอาหารแยกใบหน้าพนักงานกับลูกค้าได้เพื่อตรวจสอบพฤติกรรม การ เทคโนโลยีการจดจำใบหน้า นอกจากจะนำมาใช้ในด้าน การรักษาความปลอดภัยแล้ว ยังสามารถนำมาใช้งานได้อีกหลากหลายกรณี เช่น ออฟฟิศ ห้างสรรพสินค้า อาคารจอดรถ การมาทำงานของพนักงาน หน่วยงาน องค์กรขนาดใหญ่ก็ยังสามารถที่จะนำมาใช้งาน ในด้านการวิเคราะห์การตลาดได้อีกหลายรูปแบบ อาทิ พฤติกรรมลูกค้าที่เข้ามาใช้บริการภายในร้าน

1.2 วัตถุประสงค์ของโครงการ

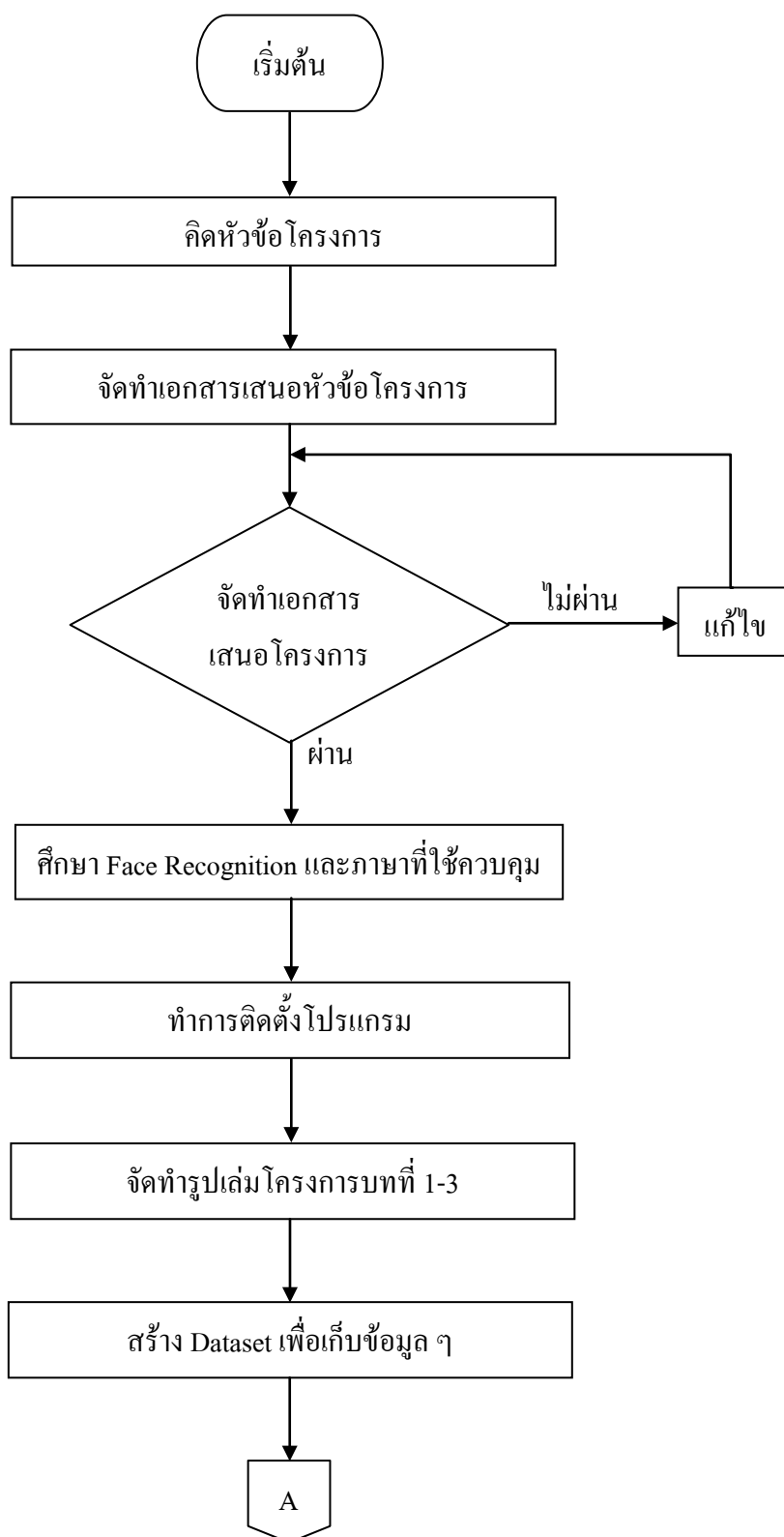
- 1.2.1 เพื่อนำความรู้ในสาขาวิชาเทคโนโลยีสารสนเทศในการจัดทำโครงการ
- 1.2.2 เพื่อศึกษาระบบการจดจำใบหน้า (Face Recognition) เพื่อนำไปต่อยอด
- 1.2.3 เพื่อสร้างความปลอดภัยให้กับสถานที่ เพิ่มความปลอดภัยในชีวิตและทรัพย์สิน

1.3 ขอบเขตของโครงการ

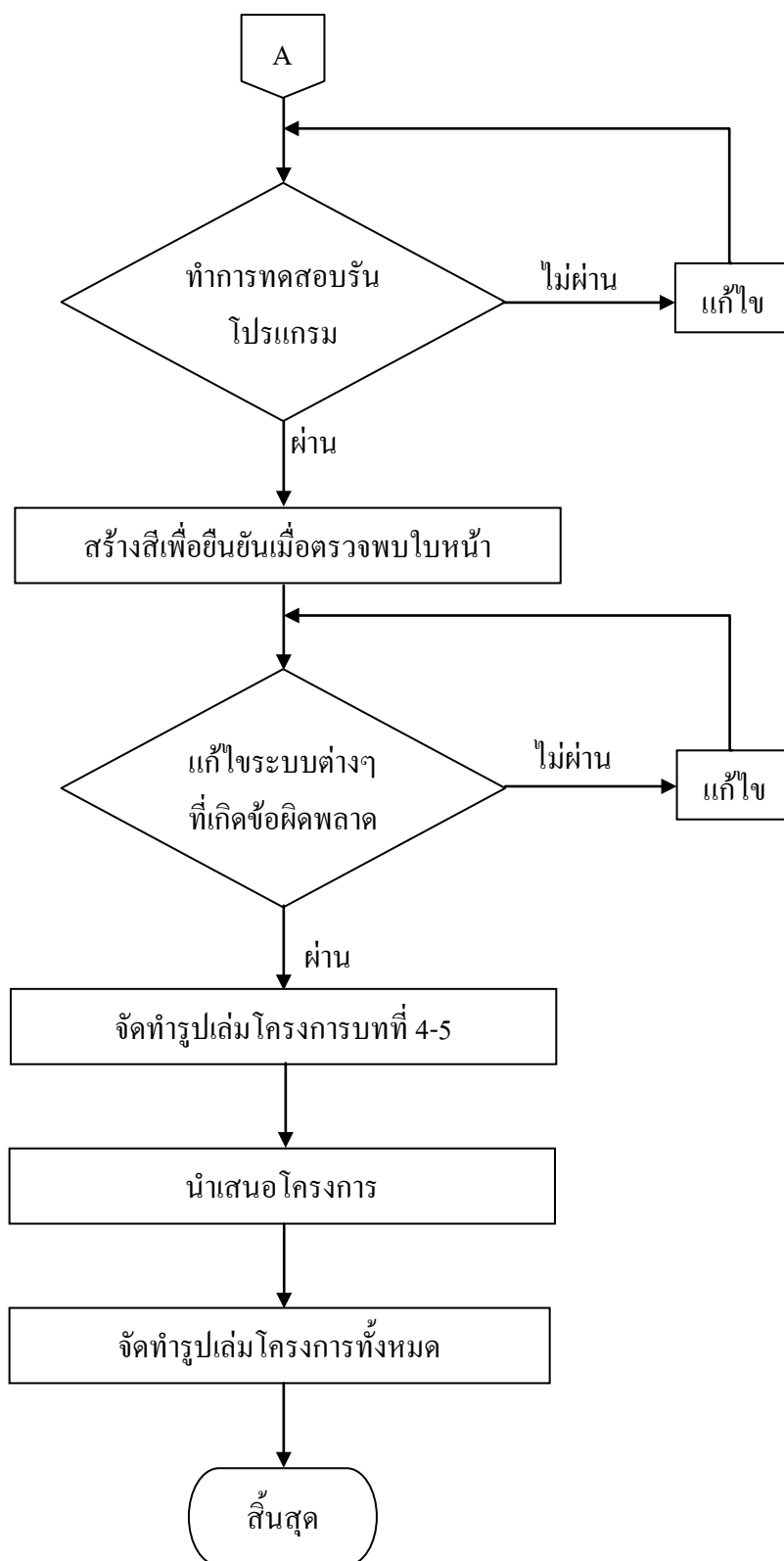
- 1.3.1 สามารถตรวจจับวัตถุใบหน้าคนและแสดงชื่อได้ร้อยละ 80 ขึ้นไป
- 1.3.2 สามารถมีข้อความแสดงผลเมื่อพบวัตถุ
- 1.3.3 ใช้ภาษา Python ในการเขียนโปรแกรม
- 1.3.4 ใช้กล้อง Webcam จำนวน 1 ตัวในการตรวจจับ
- 1.3.5 ระยะในการตรวจจับใบหน้า 1.5 เมตร โดยประมาณ
- 1.3.6 ขนาดของรูปในการตรวจจับใบหน้า 64 x 64 พิกเซล
- 1.3.7 ความเร็วในการตรวจจับใบหน้า 30 FPS ต่อ 1 วินาที
- 1.3.8 พื้นหลังในการตรวจจับต้องเป็นสีขาวแสงไม่มากจนเกินไป

1.4 ขั้นตอนการดำเนินงาน

- 1.4.1 คิดหัวข้อโครงการ
- 1.4.2 จัดทำรูปเล่มเสนอโครงการ
- 1.4.3 เสนอร่างโครงการ
- 1.4.4 แก้ไขเสนอร่างโครงการ
- 1.4.5 ศึกษาเทคโนโลยี Face Recognition และภาษาที่ใช้ควบคุม
- 1.4.6 ทำการติดตั้งโปรแกรม vs code , Python 3.6.8 และ Library ที่ใช้
- 1.4.7 จัดทำรูปเล่มโครงการบทที่ 1-3
- 1.4.8 สร้าง Dataset เพื่อเก็บข้อมูล
- 1.4.9 ทำการทดสอบรันโปรแกรม
- 1.4.10 สร้างเสียงเพื่อยืนยันเมื่อตรวจพบใบหน้าตรงตาม Dataset
- 1.4.11 ปรับปรุงแก้ไขระบบต่างๆที่เกิดข้อผิดพลาด
- 1.4.12 จัดทำรูปเล่มโครงการบทที่ 4-5
- 1.4.13 นำเสนอโครงการ
- 1.4.14 จัดทำรูปเล่มโครงการทั้งหมด



รูปที่ 1.1 แผนภาพ Flowchart ขั้นตอนการดำเนินงาน



รูปที่ 1.1 แผนภาพ Flowchart ขั้นตอนการดำเนินงาน (ต่อ)

1.5 ระยะเวลาการทำโครงการ

ตารางการดำเนินงาน โครงการนี้ใช้ระยะเวลาในการทำ ตั้งแต่เดือน มิถุนายน พ.ศ 2562 ถึง เดือนกุมภาพันธ์ พ.ศ 2563 ดังตารางที่ 1.1

ตารางที่ 1.1 ตารางแผนการดำเนินงาน

ลำดับ	ขั้นตอนการดำเนินงาน	ระยะเวลาการดำเนินงาน									
		พ.ศ.2562							พ.ศ.2563		
		มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	
1	คิดหัวข้อโครงการ	↔									
2	จัดทำรูปเล่มเสนอโครงการ	↔									
3	เสนอร่างโครงการ	↔									
4	แก้ไขเสนอร่างโครงการ	↔									
5	ศึกษาเทคโนโลยีFace Recognition	↔									
6	จัดทำโครงการ ติดตั้งโปรแกรม	↔									
7	จัดทำรูปเล่มบทที่ 1-3	↔									
8	สร้าง Dataset ข้อมูล	↔									
9	ทดสอบรัน โปรแกรม	↔									
10	สร้างสีเพื่อยืนยันเมื่อตรวจพบใบหน้า	↔									
11	แก้ไขระบบต่างๆที่เกิดข้อผิดพลาด	↔									
12	จัดทำรูปเล่มบทที่ 4-5			↔							
13	นำเสนอโครงการ			↔							
14	จัดทำรูปเล่มโครงการ					↔					

1.6 ประโยชน์ที่คาดว่าจะได้รับ

- 1.6.1 เพื่อนำความรู้ในสาขาวิชาเทคโนโลยีสารสนเทศในการจัดทำโครงการ
- 1.6.2 เพื่อศึกษาระบบการจดจำใบหน้า (Face Recognition) เพื่อนำไปต่อยอด
- 1.6.3 เพื่อสร้างความปลอดภัยให้กับสถานที่ เพิ่มความปลอดภัยในชีวิตและทรัพย์สิน

1.7 งบประมาณที่ใช้ในการทำโครงการ

1.7.1 กล้อง (Webcam)	200	บาท
1.7.2 กระดาษ	400	บาท
1.7.3 ค่าแผ่น CD	100	บาท
1.7.4 ค่าทำเล่มเอกสารโครงการ	200	บาท
รวม	<u>900</u>	บาท

บทที่ 2

ทฤษฎีและผลงานวิจัยที่เกี่ยวข้อง

ในการศึกษาผลงานที่เกี่ยวข้องกับการสร้างระบบรู้จำใบหน้า (Face Recognition) สาขาวิชาเทคโนโลยีสารสนเทศ ซึ่งนับว่าเป็นองค์ประกอบหนึ่งที่มีความสำคัญมาก จะทำให้โครงการมีความครบถ้วนสมบูรณ์มากยิ่งขึ้นจนกระทั่งผู้จัดทำได้แบ่งเอกสารและทฤษฎีที่เกี่ยวข้องเป็นหัวข้อดังนี้

2.1 หลักการทำงานของระบบรู้จำใบหน้า (Face Recognition)

2.2 การตรวจจับใบหน้า (Face Detection)

2.3 โปรแกรมใช้ในการแก้ไขและปรับแต่งโค้ด

2.4 ภาษาที่ใช้ในการเขียนโค้ด

2.5 ไลบรารีฟังก์ชันต่าง ๆ

2.5.1 OpenCV (Open source Computer Vision)

2.5.2 Deep learning

2.5.3 Machine Learning

2.5.4 Tkinter

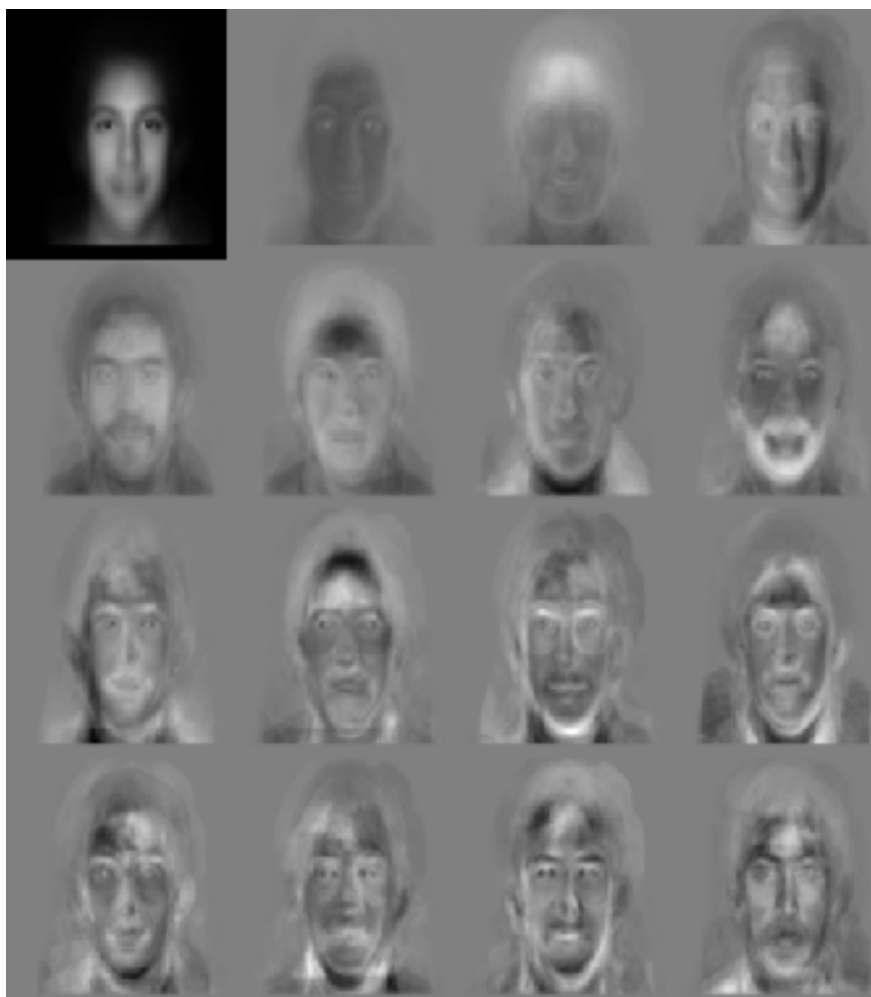
2.5.5 NumPy (Numeric Python)

2.1 หลักการทำงานของระบบรู้จำใบหน้า (Face Recognition)

ระบบรู้จำใบหน้า (Face Recognition) ถูกออกแบบมาเพื่อทำการเปรียบเทียบใบหน้าบุคคลที่สนใจกับฐานข้อมูลใบหน้าที่มีอยู่โดยอัลกอริทึมที่ใช้ในขั้นตอนการสร้างแม่แบบและขั้นตอนการเปรียบเทียบอาจแตกต่างกันไป ขึ้นอยู่กับการออกแบบระบบของแต่ละระบบ แต่ไม่ว่าจะมีอัลกอริทึมใดในการทำงานในขั้นตอนการสร้างต้นแบบและขั้นตอนการเปรียบเทียบมีลักษณะอย่างไร ขั้นตอนการทำงาน ของระบบทั้งหมดยังคงเหมือนกันโดยทั่วไป กระบวนการที่ได้นำภาพใบหน้าที่ตรวจจับได้และประมวลผลเรียบร้อยแล้ว จากขั้นตอนการ ตรวจจับใบหน้ามาเปรียบเทียบกับฐานข้อมูลของใบหน้าที่ระบุยืนยันใบหน้าที่ตรวจจับได้ตรงกับบุคคลใด ตัวอย่างของอัลกอริทึมการรู้จำใบหน้า วิธีการที่ใช้ในการตรวจจับใบหน้าที่มีความสามารถในการประมวลผลได้รวดเร็วและมีอัตรา ความถูกต้องในการตรวจหาสูง โดย Paul Viola และ Michael J. Jones ได้คิดค้นและตีพิมพ์ในปี ค.ศ. 2001 มักจะเรียกว่า Viola-Jones method ซึ่งเป็นอัลกอริทึมที่ใช้วิธีการแทนรูปภาพที่เรียกว่า "Integral Image" จะช่วยให้การคำนวณ feature ทำได้รวดเร็วขึ้น และได้มีการปรับปรุงอัลกอริทึมการเรียนรู้โดยมีพื้นฐานจาก AdaBoost ซึ่งเลือกเอาเฉพาะ critical features (features ที่ให้ classifiers ที่มี ประสิทธิภาพสูงสุด)

2.1.1 Principal Component Analysis (PCA)

PCA หรือ Principal Component Analysis หรือภาษาไทยเรียกว่า การวิเคราะห์องค์ประกอบหลัก คือเทคนิคในการผสม ลักษณะเด่นในเวกเตอร์นำเข้าเพื่อสร้างเวกเตอร์ใหม่ที่อยู่ใน ปริภูมิ(subspace) ที่มีมิติต่ำกว่าเวกเตอร์เดิม โดยการผสมที่เป็นการผสมเชิงเส้นตรงหรือ linear combination คือการนำลักษณะเด่นมาคูณค่าคงที่บางอย่างแล้วบวกค่าเพิ่มเติม การนำ PCA มาใช้ในการพัฒนาระบบรู้จำใบหน้าจะได้โดยการแปลงภาพถ่ายใบหน้า บุคคลสองมิติไปเป็นเวกเตอร์หนึ่งมิติ และเก็บไว้ในฐานข้อมูล และเมื่อต้องการนำรูปภาพใบหน้าบุคคลที่ สนใจมาเปรียบเทียบกับ จะทำการแปลงภาพใบหน้านั้นเป็นเวกเตอร์หนึ่งมิติ แล้วนำเวกเตอร์ไปเปรียบเทียบกับภาพในฐานข้อมูลเพื่อหาผลลัพธ์ โดยภาพที่ 2.4 แสดงตัวอย่างของใบหน้าไอเกน (Eigenfaces)



รูปที่ 2.1 ตัวอย่างของใบหน้าไอเกน (Eigenfaces)

2.1.2 Linear Discriminant Analysis (LDA, a.k.a “fisherfaces”)

LDA มีวิธีการทำงานที่คล้ายกับ PCA ซึ่งใน PCA ต้องหาปริภูมิย่อยที่เมื่อป้อนข้อมูลลงไปแล้ว มีการกระจายตัวสูงสุด แต่ใน LDA ต้องการปริภูมิย่อยที่เมื่อป้อนข้อมูลลงไปแล้ว ข้อมูลจาก class เดียวกันจะเข้าใกล้กันมากขึ้น และข้อมูลจากต่าง class กันจะอยู่ห่างกันมากขึ้นดังในรูปภาพที่ 1.2 แต่ละ บล็อกจะเปรียบเสมือนคลาสหรือภาพบุคคลที่มีความคล้ายคลึงกันถูกจัดให้อยู่ในคลาสเดียวกัน

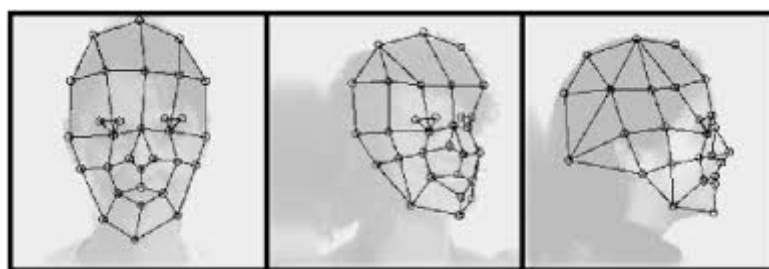


รูปที่ 2.2 ตัวอย่างของคลาสที่ถูกจัดกลุ่ม (Classified) โดยใช้ LDA

2.1.3 Elastic Bunch Graph Matching (EBGM)

EBGM ตั้งอยู่บนพื้นฐานที่ว่ารูปใบหน้าของคนเรามีส่วนที่ไม่เป็นเชิงเส้นอยู่มาก (non-linear) และไม่สามารถวิเคราะห์โดยใช้กระบวนการเชิงเส้นอย่างวิธีที่กล่าวมาก่อนหน้านี้ได้ เช่น ในเรื่องของแสงที่ ตกกระทบใบหน้า, ตำแหน่งของใบหน้า และการแสดงอารมณ์ โดย EBGM จะใช้ Gabor Wavelet และ Gabor filter ในการประมวลผลและสร้างภาพ ใบหน้าโดยการกำหนดจุดที่สนใจบนใบหน้าหลังจากนั้นก็เก็บภาพใบหน้าที่สร้างขึ้นไว้เป็นฐานข้อมูล เมื่อ ต้องการการรู้จำจึงนำภาพเข้าผ่านกระบวนการเดียวกันและเปรียบเทียบระยะห่างของแต่ละจุดของทั้ง สองภาพ ว่ามีความใกล้เคียงเพียงพอมากเพียงใดที่จะสามารถระบุได้ว่าเป็นรูปบุคคลคนเดียวกัน ซึ่งความยากของวิธีการนี้คือการกำหนดจุดที่สนใจบนใบหน้าที่ต้องมีความแม่นยำเป็นอย่างมาก โดยภาพที่

2.3แสดงการกำหนดจุดสนใจบนใบหน้า แบบ Elastic Bunch Map Graphing

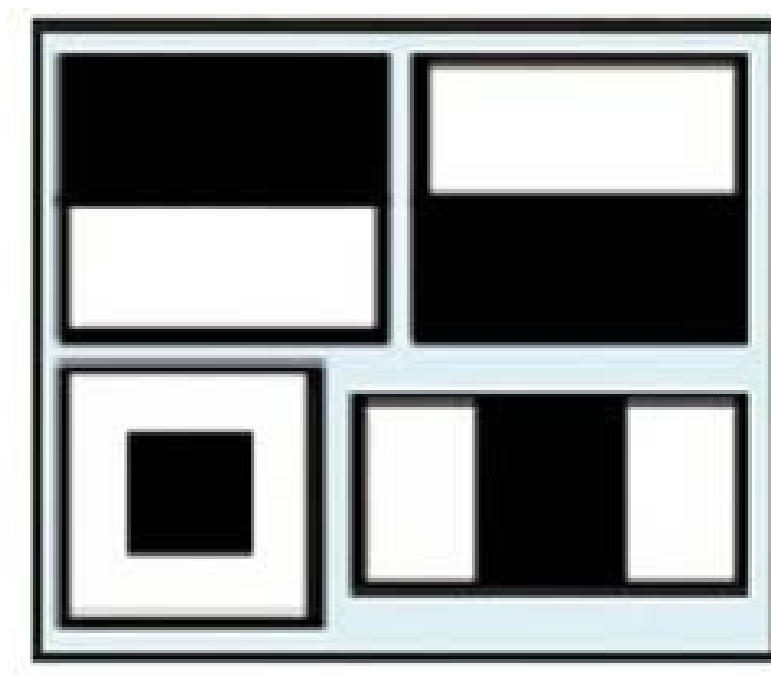


รูปที่ 2.3 ตัวอย่างของคลาสที่ถูกจัดกลุ่ม (Classified) โดยใช้ LDA

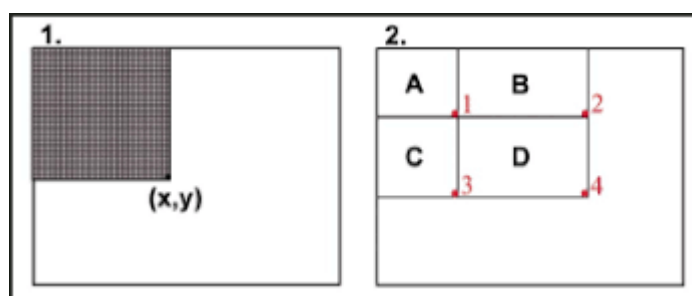
2.2 การตรวจจับใบหน้า (Face Detection)

กระบวนการค้นหาใบหน้าของบุคคลจากภาพหรือวิดีโอ แล้วจะทำการประมวลผลภาพใบหน้าที่ได้สำหรับขั้นตอนถัดไปเพื่อให้ภาพใบหน้าที่ตรวจจับได้ง่ายต่อการจำแนกแยกแยะ และอัลกอริทึมที่ใช้ในการตรวจจับใบหน้าในปัจจุบันมีอยู่ด้วยกันหลากหลายวิธีเนื่องจากเป็นระบบที่ได้รับความนิยมจากนักวิชาการหลายสาขาวิชาจึงทำให้ระบบรู้จำใบหน้า มีผู้คนสนใจศึกษาและพัฒนากันอย่างมากมายอันเนื่องมาจากปัจจัยด้านองค์ความรู้และเทคโนโลยีของอุปกรณ์ต่างๆ ที่พัฒนาขึ้นให้มีความเหมาะสมที่จะนำมาใช้ในระบบจึงทำให้ต้องออกแบบอัลกอริทึมใหม่ให้เหมาะสมซึ่งอัลกอริทึมในการตรวจจับ ใบหน้าที่ดีนั้นมีส่วนช่วยในการจำแนกใบหน้าได้แม่นยำและรวดเร็วขึ้นเป็นอย่างมาก วิธีการที่ใช้ในการตรวจจับใบหน้าที่มีความสามารถในการประมวลผลได้รวดเร็วและมีอัตรา ความถูกต้องในการตรวจหาสูง โดย Paul Viola และ Michael J. Jones ได้คิดค้นและตีพิมพ์ [15] ในปี ค.ศ. 2001 มักจะเรียกว่า Viola-Jones method ซึ่งเป็นอัลกอริทึมที่ใช้วิธีการแทนรูปภาพที่ เรียกว่า "Integral Image" จะช่วยให้การคำนวณ feature ทำได้รวดเร็วขึ้น และได้มีการปรับปรุงอัลกอริทึมการเรียนรู้โดยมีพื้นฐานจาก AdaBoost ซึ่งเลือกเอาเฉพาะ critical features (features ที่ให้ classifiers ที่มี ประสิทธิภาพสูงสุด) นอกจากนี้ยังได้อธิบายถึงการรวม classifiers แบบ cascade ซึ่งช่วยให้ส่วนพื้นหลัง ของภาพถูกปฏิเสธ ได้เร็วและเน้นการคำนวณไปที่บริเวณที่มีลักษณะคล้ายวัตถุที่สนใจมากขึ้น

หลักการพื้นฐานของอัลกอริทึมของ Viola-Jones คือการสแกน sub-window เพื่อตรวจหาใบหน้าจากรูปภาพอินพุต การประมวลผลภาพแบบทั่วไปจะใช้การปรับขนาดภาพขาเข้าที่แตกต่างกัน หลายขนาด และใช้ตัวตรวจหา (Detector) ที่มีขนาดคงที่ค้นหาวัตถุ ซึ่งวิธีนี้ใช้เวลาในการคำนวณมาก เนื่องมาจากการคำนวณบนรูปภาพที่มีขนาดแตกต่างกัน Viola-Jones ได้เสนอแนะวิธีการใหม่โดยการ ปรับขนาดตัวตรวจหา และใช้ตัวตรวจหาค้นหาวัตถุหลายครั้ง (แต่แต่ละครั้งใช้ขนาดแตกต่างกัน) ซึ่งทั้งสอง วิธีจะใช้เวลาในการคำนวณไม่ต่างกันมากนัก แต่ Viola-Jones ได้คิดค้นตัวตรวจหาที่ใช้จำนวนครั้งในการ คำนวณคงที่ แม้จะมีขนาดของภาพที่แตกต่างกัน โดยตัวตรวจหาดังกล่าวนี้นี้สร้างขึ้นโดยใช้ features ของ Haar wavelets ตามตัวอย่างภาพที่ 2.1 และ Integral Image ตามตัวอย่างภาพที่ 2.2

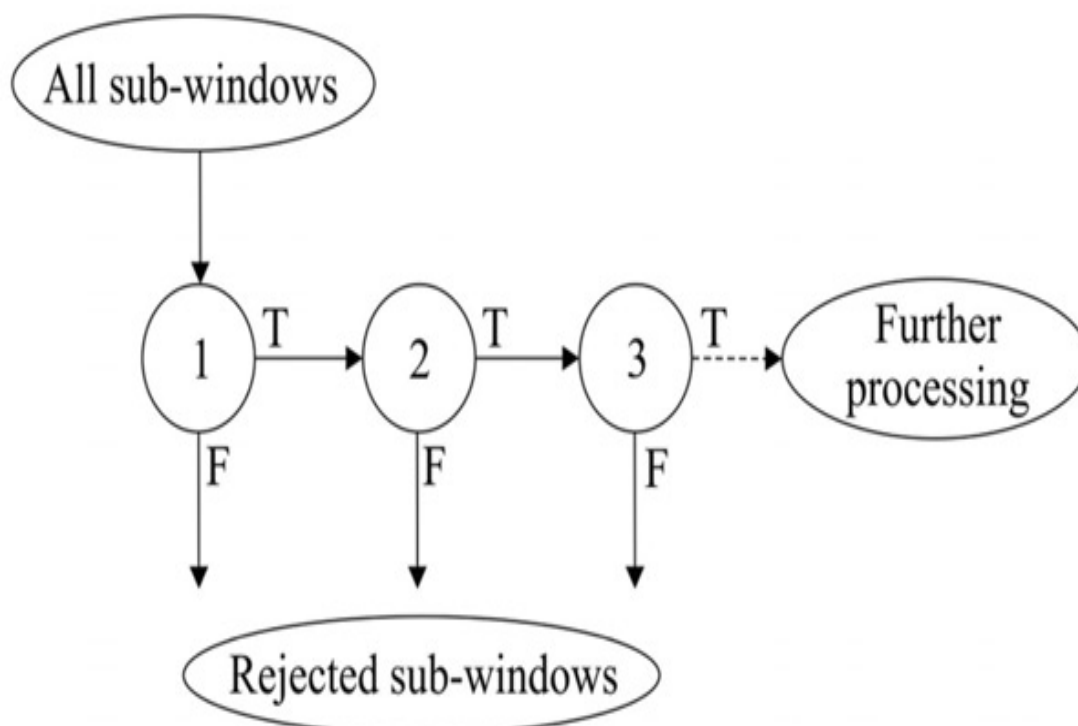


รูปที่ 2.4 Examples of the Haar features



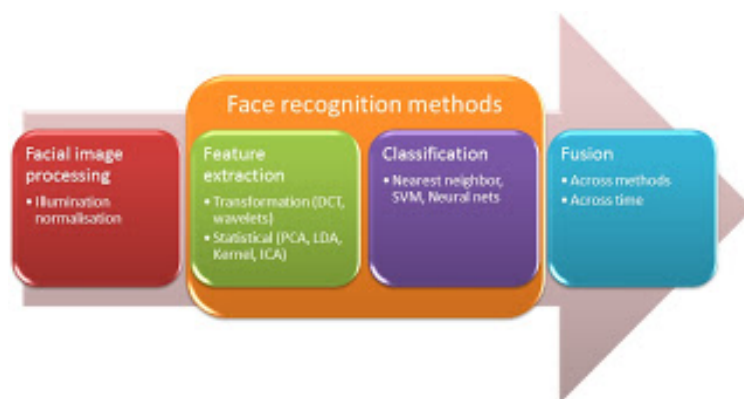
รูปที่ 2.5 The Integral Image trick

หลักการของอัลกอริทึมการค้นหาน้ำของ Viola-Jones คือการใช้ตัวตรวจหาและการสแกนหลายครั้งบนภาพเดิม เนื่องด้วยขนาดที่แตกต่างกัน ถึงแม้ว่า มีใบหน้ามากกว่าหนึ่งหน้า ผลลัพธ์ของ sub-window จำนวนมากยังคงเป็นลบ (negative non-faces) ซึ่งปัญหานี้แก้ไขโดยใช้หลักการ “ปฏิเสธสิ่งที่ไม่ใช่ใบหน้า แทนการค้นหาใบหน้า” เพราะการตัดสินใจรูปภาพที่ปรากฏไม่ใช่ใบหน้า ทำให้เร็วกว่าการค้นหาใบหน้า และได้มีการสร้างตัวจำแนกประเภทแบบ cascaded (Cascaded classifier) คือเป็น Classifier หลายตัวต่อกันเป็นลำดับ ดังแสดงในภาพที่ 2.6 ซึ่งเมื่อ sub-window ถูกจัดประเภท ไม่ใช่ใบหน้า (non-face) จะถูกทำการปฏิเสธทันที แต่ในทางตรงกันข้าม ถ้า sub-window นั้น ถูก จำแนกเป็นประเภทที่มีโอกาสเป็นใบหน้า (maybe-face) จะถูกส่งต่อไปยัง Classifier ตัวถัดไป



รูปที่ 2.6 ตัวจำแนกประเภทแบบ cascaded (Cascaded classifier)

2.2.1 อัลกอริทึมการรู้จำใบหน้า



รูปที่ 2.7 กระบวนการรู้จำใบหน้าด้วยวิธีทางสถิติ

การรู้จำทางสถิติจะมีอยู่สองขั้นตอนหลักนะครับ คือ

- การเรียนรู้ (Training)
- การระบุตัวตน (Identify) หรือ Testing

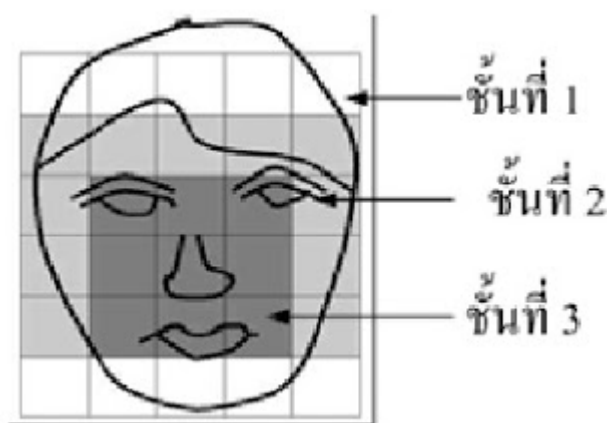
Step 1: เราต้องได้ภาพใบหน้ามาก่อน ไม่ว่าจะได้จากกล้องเว็บแคม กล้องถ่ายภาพธรรมดา หรืออะไรก็ได้แล้วแต่ ต้องมาผ่านกระบวนการตรวจจับใบหน้า (Face Detection) กระบวนการ ตรวจจับใบหน้า ท่านสามารถหา Open source ได้ทั่วไปนะครับ ต่อไปก็ต้องผ่านการปรับแต่งภาพก่อน ไม่ว่าจะเป็นเรื่องของแสงเงา สัญญาณรบกวน ง่ายๆคือ แต่งภาพให้มันดีขึ้นครับ หรือ แต่งให้อยู่ในรูปแบบเดียวกันครับ

Step 2: อันนี้สำคัญครับ คือการดึงลักษณะเฉพาะ (Feature) ของภาพ ผมขอ ยกตัวอย่าง โดยวิธีทางสถิตินะครับ เค้าใช้การลดมิติของภาพ เช่น จากภาพขนาด 100×100 pixel ก็จะมีจำนวน 10000 จุดภาพซึ่งข้อมูลมันเยอะมาก ต้องลดมิติหรือขนาดของข้อมูลลงให้น้อย เช่น ให้เหลือ 20 มิติ หรือ ค่าเอกลักษณ์จำนวน 20 ค่า ด้วยวิธีทางสถิติ วิธีใดวิธีหนึ่ง (ท่านต้องไปศึกษาเพิ่มเติมนะครับ)

Step 3: ส่วนนี้คือการจำแนกกลุ่มของข้อมูล (Classification) ที่เราได้ดึงลักษณะเฉพาะออกมาแล้ว ซึ่งก็มีวิธีอีกมากมาย ยกตัวอย่างเช่น การหาค่าที่ใกล้ที่สุด Nearest neighbor SVM แต่ละวิธีก็ยากง่ายต่างกันรวมถึงความแม่นยำด้วยนะครับ

Step 4: สุดท้ายคือการเก็บลงฐานข้อมูลครับ (ถ้าเป็นขั้นตอนการระบุตัวตนจะทำเหมือนสามขั้นตอนข้างบนแต่จะเรียกว่า Identify)

วิธีเชิงความรู้ (Knowledge-based methods) เป็นวิธีการพิจารณาความสัมพันธ์ระหว่างลักษณะเด่นเฉพาะต่างๆ ใบหน้า ที่มีตำแหน่งและองค์ประกอบหลักพื้นฐานตายตัวบน ใบหน้าตัวอย่างเช่น Yang and Huang (1994) ศึกษาวิธีเชิงความรู้แบบลำดับชั้น (hierarchical) ซึ่งพิจารณาแบ่งเป็น 3 ลำดับชั้น ดังแสดงในรูปที่ 2.8



รูปที่ 2.8 ขอบเขตการพิจารณาของแต่ละลำดับชั้น

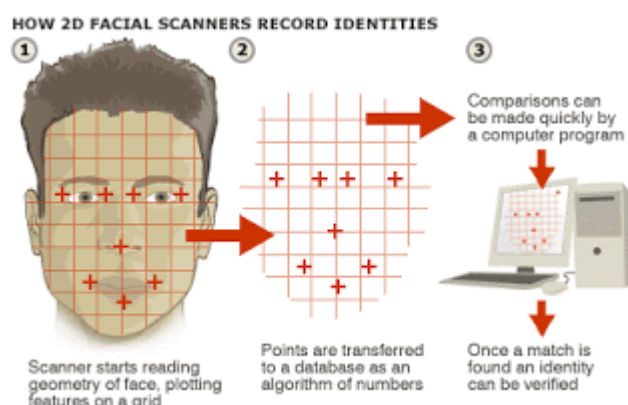
ขั้นแรก พิจารณาหาตำแหน่งความน่าจะเป็นของใบหน้าโดยรวม ขั้นที่ 2 พิจารณาโดยค่าฮิสโตแกรม (histogram) ร่วมกับการหาเส้นขอบ ขั้นที่ 3 พิจารณาลักษณะเด่นภายในของใบหน้าเช่น ความสมมาตรกันของตาทั้ง 2 ข้างตำแหน่งของจมูกและปากในแนวตั้งและแนวนอน ต่อมา Kotropoulos and Pitas (1997) ได้ศึกษาวิธีฐานกฎแบบกำหนดเขต (rule-based localization method) ซึ่งเป็นวิธีการที่คล้ายคลึงกับวิธีของ Yang and Huang แตกต่างกันที่พิจารณาค่าฮิสโตแกรมทั้งใน แนวตั้งและแนวนอน แต่ยังไม่สามารถขจัดปัญหาของภาพ ที่มีพื้นหลังซับซ้อนและปัญหาความหลากหลายของใบหน้าบุคคล รวมทั้งการวางท่าที่แตกต่างกันได้

วิธีเชิงลักษณะ (Feature-based methods) เป็นการใช้อัลกอริทึมพิจารณาลักษณะเด่นและ โครงสร้าง ของใบหน้า รวมทั้งความเปลี่ยนแปลงขององค์ประกอบต่างๆของภาพเช่น แสงเงาตัวอย่างเช่น Chetverikov and Lerch (1993) ใช้ความแตกต่างของแสงเงาและเส้นแนวเป็นแบบจำลองในการค้นหาใบหน้า โดยแบบจำลองนั้นประกอบด้วยจุดความสว่างน้อย (จุดมืด) 2 จุด เพื่อแสดงดวงตาและจุดความสว่างมา (จุดอ่อน) 3 จุด เพื่อแสดงโหนกแก้มและจมูกแล้วหาความสัมพันธ์ของระยะห่างและตำแหน่งของจุดต่างเพื่อคัดเลือกแบบหน้าที่เหมาะสม ระบบนี้มีข้อจำกัดอยู่ที่เมื่อแสงเงาของสภาวะแวดล้อมเปลี่ยนแปลงจะทำให้ประสิทธิภาพในการค้นหาเปลี่ยนไป ต่อมา Graf, Chen, Petajan, and Cosatto (1995) พัฒนาทฤษฎีนี้โดยใช้งานในระบบภาพระดับเทา และนำตัวกรองช่วงผ่าน (band pass filter) ตัวกรองเกาส์เซียน (Gaussian filter) และค่าฮิสโตแกรม มาใช้หาค่าจุดสูงสุดและต่ำสุดเพื่อกำหนดขอบเขตของใบหน้า

ลักษณะเฉพาะอีกประการหนึ่งที่นิยมนำมาพิจารณาหาใบหน้าบุคคลคือ ค่าสีผิวมนุษย์โดยมีการค้นหาในหลายๆปริภูมิสีโดย Crowley and Berard (1997) ทำการเก็บค่าแวกเตอร์ของสีผิวมนุษย์โดยแยกค่าสีออกเป็น R, G และ B เพื่อพิจารณาค่าฮิสโตแกรมของค่าสีผิวมนุษย์ที่ได้จากการสำรวจข้อมูลกลุ่มตัวอย่าง ต่อมา Saxe and Fould (1996) ประยุกต์ใช้วิธีการดังกล่าวกับปริภูมิสี HSV และใช้แนวคิดค่าฮิสโตแกรมตัดผ่าน (histogram intersection) เปรียบเทียบค่าฮิสโตแกรมควบคุมกับค่าฮิสโตแกรมปัจจุบัน ซึ่งทำให้ประสิทธิภาพดีกว่าการใช้ค่าเริ่มเปลี่ยน

Chai and Ngan (1998) พิจารณาค่าสีผิวในปริภูมิสี YCrCb โดยพิจารณาค่าเริ่มเปลี่ยนของค่าสีผิวมนุษย์ที่ได้จากการสำรวจข้อมูลกลุ่มตัวอย่าง ซึ่งต้องระมัดระวังในการเลือกใช้ให้เหมาะสมกับภาพที่ทำการทดลองและเป็นข้อเสียของการคัดแยกสีผิวมนุษย์ด้วยค่าเริ่มเปลี่ยน ต่อมา มีการนำการคัดแยกสีผิวมาใช้ร่วมกับวิธีการอื่นๆในการค้นหาใบหน้าตัวอย่างเช่น Wu, Yokoyama, Pramadihanto, and Yachida(1996) ใช้ทฤษฎีกลุ่มเครือ (fuzzy theory) ในการคัดแยกสีผิวมนุษย์และสีผิวในปริภูมิสี CIE XYZ โดยทำการแปลงจากปริภูมิสี RGB นอกจากนี้ยังมีการนำเอาลักษณะเด่นของใบหน้าอื่นๆมาใช้ในการค้นหาตัวอย่างเช่น (Yokoo and Hagiwara(1996) ใช้คุณสมบัติของโครงหน้าทั่วไปของมนุษย์ที่เป็นรูปไข่สร้างแบบจำลองรูปวงรีเปรียบเทียบพารามิเตอร์

วิธีเทียบเคียงแผ่นแบบ(Template matching methods) เป็นการเปรียบเทียบภาพที่ต้องการค้นหา กับ โครงสร้างแบบจำลองของใบหน้ามาตรฐาน โดยเก็บข้อมูลความสัมพันธ์อย่างอิสระของส่วนต่างๆ บนใบหน้า ได้แก่ โครงสร้างใบหน้า ตา จมูก และปาก ในท่าหน้าตรง ตัวอย่างเช่น Sakai, Nagao and Fujibayashi, (1969) เสนอโครงสร้างแบบจำลองแผ่นแบบย่อย (subtemplates model) โดยใช้ตัวกรองโซเบล (sobel filter) หาเส้นขอบ เพื่อหาตำแหน่งความน่าจะเป็นของส่วนย่อยต่างๆ บนใบหน้าที่สามารถเข้ากันได้ดีที่สุดกับแบบจำลองแผ่นแบบย่อยให้เป็นตำแหน่งของใบหน้าที่ต้องการต่อมา Tsukamoto, Lee, and Tsuji (1994) เสนอแบบจำลองคุณภาพสำหรับรูปแบบหน้า (qualitative model for face pattern : QMF) โดยใช้พารามิเตอร์ของแสงสว่างและเส้นขอบใบหน้าเป็นแบบจำลองของใบหน้า และเทคนิคที่นิยมใช้อีกอันหนึ่งคือการวิเคราะห์องค์ประกอบหลัก (principal component analysis: PCA) โดย Jung, Lee C.W., and Lee Y.C (2002) ศึกษาการค้นหาคำแหน่งที่แน่นอนของใบหน้าโดยใช้หลักการพื้นฐานทางด้านเรขาคณิต วิเคราะห์องค์ประกอบของใบหน้าให้อยู่ในรูปของสมการทางคณิตศาสตร์ต่อมาได้มีการพัฒนาแบบจำลองให้มีความสามารถปรับตัวได้เรียกว่าแผ่นแบบเปลี่ยนรูปได้ (deformable templates) โดยแผ่นแบบสามารถปรับตัวให้รองรับความยืดหยุ่นของโครงสร้างใบหน้าที่เปลี่ยนแปลงไปและลดปัญหาที่มีในแผ่นแบบคงตัวตัวอย่างเช่น Malciu and Preteux(2002) นำแบบจำลองหน้าจากฐานข้อมูล MPEG-4 มาสร้างเป็นแบบจำลองแผ่นแบบพารามิเตอร์ (template parameterization) และพิจารณาความยืดหยุ่นของโครงสร้างใบหน้าเปรียบเทียบเป็นความยืดหยุ่นของสปริง เพื่อหาค่าพลังงานที่สปริงใช้น้อยที่สุดในการจะปรับตัวเพื่อให้ภาพที่หาเส้นขอบนั้นเข้ากับแบบจำลองมากที่สุด

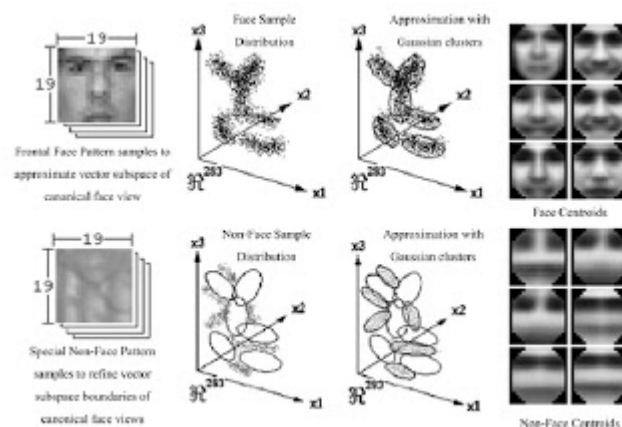


รูปที่ 2.9 Template matching methods

วิธีเชิงลักษณะปรากฏ(Appearance-based methods) เป็นการเปรียบเทียบภาพที่ต้องการค้นหา กับ โครงสร้างแบบจำลองของใบหน้า ที่ทำการเรียนรู้และฝึกสอนให้ระบบจดจำและ

นำความรู้ในฐานข้อมูลมาใช้ในการพิจารณาโดยสามารถแบ่งย่อยได้หลายวิธีด้วยกันในที่นี้จะทำการเสนอเพียงวิธีการที่ได้รับความนิยมใช้งานกันอย่างกว้างขวาง 3 วิธีดังนี้

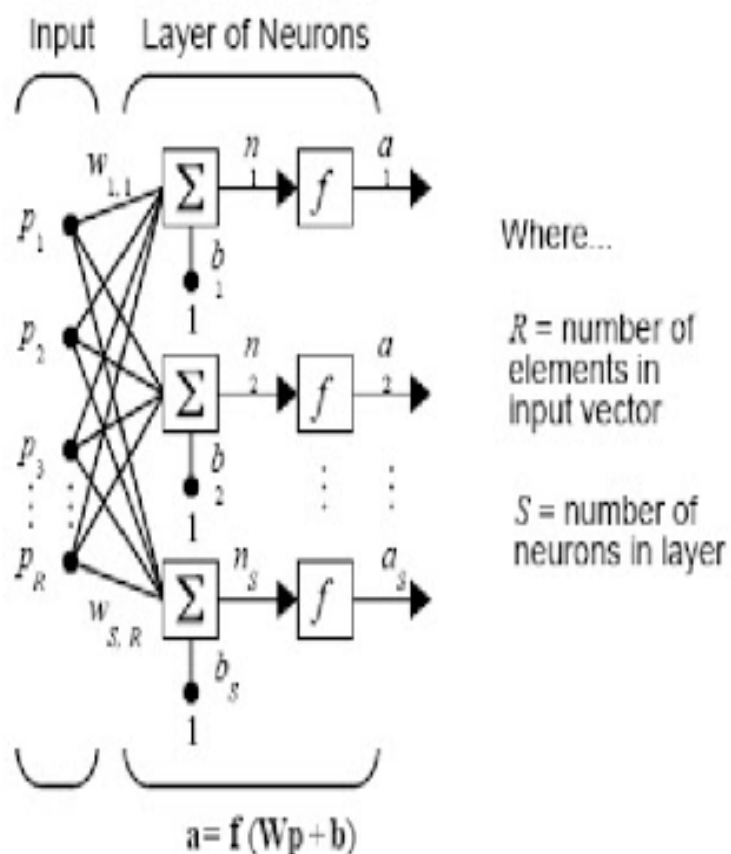
วิธีหน้าลักษณะเฉพาะ (Eigenface Methods) เป็นการวิเคราะห์องค์ประกอบพื้นฐานของใบหน้า ด้วยกระบวนการทางสถิติของใบหน้าที่หลากหลาย โดยหน้าลักษณะเฉพาะคือเซตของเวกเตอร์ลักษณะเฉพาะ (eigen vector) ที่สามารถหาได้จากเมทริกซ์ความแปรปรวนร่วม (covariance matrix) สร้างเป็นแบบจำลองของใบหน้าโดยรวมเอาลักษณะเด่นต่างๆของภาพใบหน้าตัวอย่างมารวมกันเพื่อหาค่าเฉพาะเจาะจงขององค์ประกอบบนใบหน้า ซึ่งในแต่ละบุคคลจะเป็นค่าเฉพาะของบุคคลนั้นๆ ตัวอย่างเช่น Turk and Pentland (1991) นำภาพระดับเทามาแปลงเป็นเวกเตอร์เพื่อหาค่าลักษณะเฉพาะและนำค่าลักษณะเฉพาะของตัวอย่างภาพหน้าบุคคล มาสร้างเป็นแบบจำลองหน้าลักษณะเฉพาะเพื่อค้นหาตำแหน่งของใบหน้าวิธีเชิงการกระจาย (Distribution-Based Methods) เป็นการแสดงการกระจายตัวของรูปแบบข้อมูลตัวอย่างที่มีความเป็นหน้าและความไม่เป็นหน้าเพื่อใช้เป็นบรรทัดฐานในการตัดสินใจตัวอย่างเช่น Sung and Poggio (1998) นำฟังก์ชันเกาส์เซียน (Gaussian function) มาประมาณกลุ่มการกระจายของค่าเฉลี่ยกลุ่มตัวอย่างแสดงในรูปที่ 2.10



รูปที่ 2.10 การกระจายตัวของข้อมูลตัวอย่าง

โครงข่ายประสาทเทียม (Artificial Neural Networks) เป็นการฝึกสอนโครงข่ายด้วยโครงสร้างใบหน้าที่มีความซับซ้อน โดยทำการปรับค่าน้ำหนักประสาทเพื่อให้ค่าความผิดพลาด

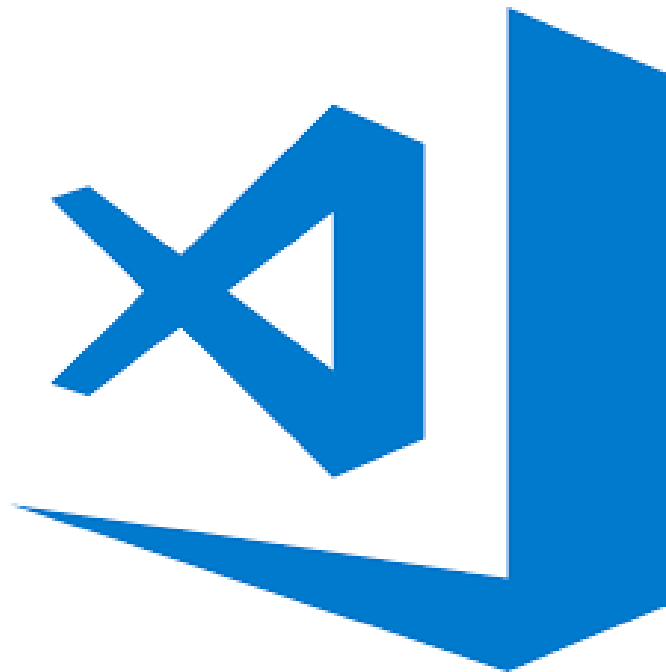
กำลังสองเฉลี่ยมีค่าลดลง ความถูกต้องของโครงข่ายประสาทเทียมขึ้นกับตัวอย่างที่ทำการฝึกสอน จำนวนชั้นนิวรอนและจำนวนนิวรอนที่เหมาะสม ตัวอย่างเช่น Rowley, Baluja, and Kanade (1998) ใช้โครงข่ายประสาทเทียมแบบแพร่กลับในการฝึกสอนข้อมูลตัวอย่าง โดยทำการแบ่งข้อมูลออกเป็นภาพตัวอย่างใบหน้าบุคคลและภาพที่ไม่ใช่ใบหน้าบุคคลอย่างละ 1,048 ภาพ และแบ่งชุดข้อมูลออกเป็น 15 ชุดข้อมูล รวมข้อมูลตัวอย่างทั้งหมดได้ 15,720 ภาพ ซึ่งการฝึกสอนที่มีจำนวนข้อมูลตัวอย่างที่ครอบคลุมและมากเพียงพอจะช่วยลดความผิดพลาดของระบบ



รูปที่ 2.11 ตัวอย่างโครงข่ายประสาทเทียม

2.3 โปรแกรมใช้ในการแก้ไขและปรับแต่งโค้ด

2.3.1 Visual Studio Code (วิซวล สตูดิโอ โค้ด)

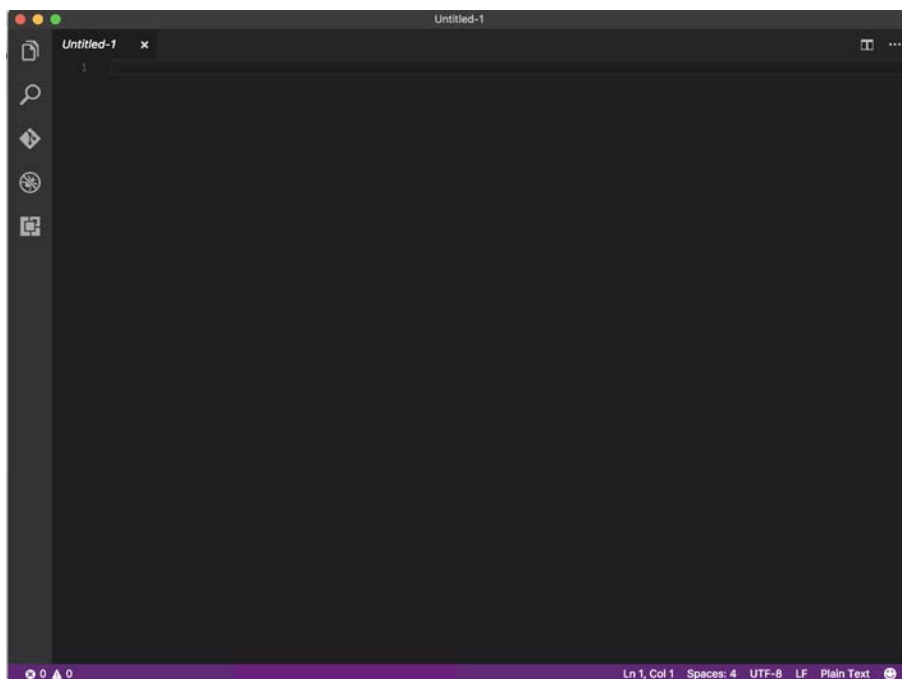


รูปที่ 2.12 Visual Studio Code (วิซวล สตูดิโอ โค้ด)

Visual Studio Code หรือ VS Code เป็นโปรแกรม Code Editor ที่ใช้ในการแก้ไขและปรับแต่งโค้ด จากค่ายไมโครซอฟท์ มีการพัฒนาออกมาในรูปแบบของ OpenSource จึงสามารถนำมาใช้งานได้แบบฟรี ๆ ที่ต้องการความเป็นมืออาชีพซึ่ง Visual Studio Code นั้น เหมาะสำหรับนักพัฒนาโปรแกรมที่ต้องการใช้งานข้ามแพลตฟอร์ม รองรับการใช้งานทั้งบน Windows, macOS และ Linux สนับสนุนทั้งภาษา JavaScript, TypeScript และ Node.js สามารถเชื่อมต่อกับ Git ได้ นำมาใช้งานได้ง่ายไม่ซับซ้อน มีเครื่องมือส่วนขยายต่าง ๆ ให้เลือกใช้อย่างมากมาย ไม่ว่าจะเป็น

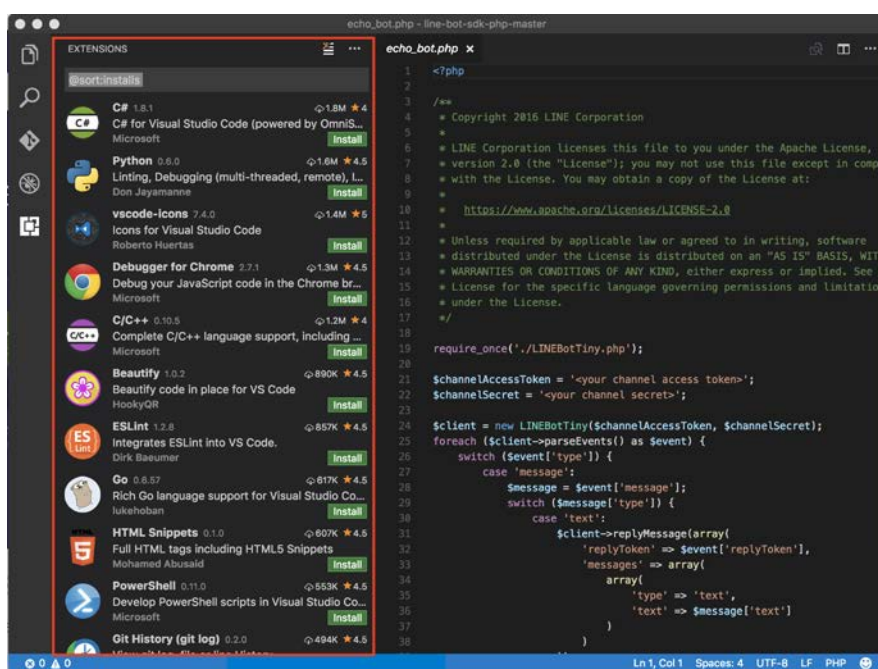
- 1.การเปิดใช้งานภาษาอื่น ๆ ทั้ง ภาษา C++, C#, Java, Python, PHP หรือ Go
- 2.Themes
- 3.Debugger
- 4.Commands

ความแตกต่างระหว่าง VSCode และ Visual Studio คือ VSCode ได้ทำการตัดในส่วน of GUI designer ออกไปเหลือแต่เพียงตัว Editor เท่านั้น จึงทำให้ตัวโปรแกรมนั้นค่อนข้างเบากว่า Visual Studio เป็นอย่างมาก VSCode สามารถนำมาใช้งานได้ฟรี รองรับการทำงานข้ามแพลตฟอร์มสำหรับผู้สนใจใช้งาน สามารถดาวน์โหลดได้ที่ <https://code.visualstudio.com/> ขอท้าวความก่อนว่าเจ้า Editor ตัวนี้นั้นออกมาตั้งแต่ 29 เมษายน ปี 2015 แล้วแหละ พัฒนาขึ้นโดยบริษัทยักษ์ใหญ่นามไมโครซอฟท์ เป็นทั้งตัวแก้ไขและปรับแต่งโค้ด (code optimized editor) ที่ตัดความสามารถมาจาก Visual Studio รุ่นปกติ (พวก GUI designer) ออกไปเหลือแค่ตัว editor



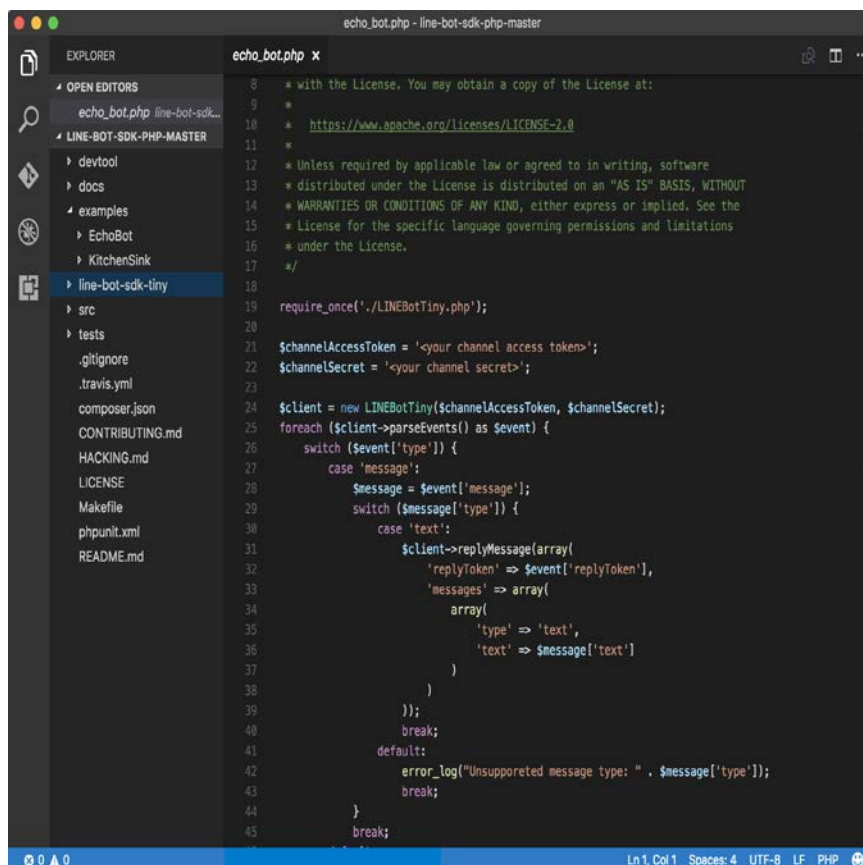
รูปที่ 2.13 Visual Studio Code (วิหาล สตูดิโอ โค้ด)

ความสามารถของเจ้า “vs code” นั้นจะมีความสามารถในการเปิดอ่าน แก้ไข และเขียนโค้ดได้เหมือน editor ตัวอื่นๆ เช่น sublime, Atom, Notepad++ มีนั่นเอง รวมถึงความสามารถในการติดตั้งเครื่องมือเสริม(Extension) โดยรับรองได้ว่ามีซัพพอร์ตไว้เพียบเลยทีเดียว



รูปที่ 2.14 Visual Studio Code (วิซวล สตูดิโอ โค้ด)

แล้วมันแตกต่างกับตัวอื่นยังไงนะหรือ เพราะว่ามันถูกพัฒนามาให้ตอบโจทย์นักพัฒนาอย่างเรามากที่สุด การดีไซน์หน้าตาของมันจึงออกมาแตกต่างจากเจ้าอื่นอย่างเห็นได้ชัดให้เป็นรูปแบบที่เข้าใจและใช้งานได้ง่าย ไม่ต้องศึกษาอะไรเพิ่มเติมก็ใช้งานแบบง่ายๆ ได้เลย



รูปที่ 2.15 Visual Studio Code (วิซวล สตูดิโอ โค้ด)

สิ่งที่ทำให้มันโดดเด่นกว่าตัวอื่นๆ คือการที่ออกแบบให้การค้นหาสิ่งต่างๆ ทำออกมาให้ใช้งานได้ง่ายและดูง่ายกว่าตัวอื่นๆ รวมถึงการที่สร้างให้สามารถเชื่อมต่อกับ Git ได้อย่างรวดเร็วและง่ายดายจนตกใจนี้เอง มีฟังก์ชันในการ commit, push&pull อยู่ในตัว หรือจะดู change ของไฟล์ที่เกิดขึ้นก็ได้แบบง่ายๆ เจ๋งไปเลยสินะ จากการที่ได้ใช้งานมาตลอดหลายเดือน ทำให้รู้สึกได้ว่าชอบเจ้าตัวนี้อย่างจริงจัง อาจจะเพราะว่าตอนใช้งานนั้นรู้สึกว่ามันทำงานได้เร็วกว่าตัวอื่นๆ ใช้งานได้ง่ายกว่า หากคุณเป็นคนหนึ่งที่ยังหา Editor อยู่อีกไม่ได้ก็อยากให้คุณลองโหลดเจ้าตัวนี้มาใช้งานดู โดยสามารถโหลดได้จาก <https://code.visualstudio.com/Download>

2.4 ภาษาที่ใช้ในการเขียนโค้ด

2.4.1 ภาษา Python

Python เป็นภาษาเขียนโปรแกรมระดับสูงที่ใช้กันอย่างกว้างขวางในการเขียนโปรแกรมสำหรับวัตถุประสงค์ทั่วไป ภาษา Python นั้นสร้างโดย Guido van Rossum และถูกเผยแพร่ครั้งแรกในปี 1991 Python นั้นเป็นภาษาแบบ *interprete* ที่ถูกออกแบบโดยมีปรัชญาที่จะทำให้โค้ดอ่านได้ง่ายขึ้น และโครงสร้างของภาษานั้นจะทำให้โปรแกรมเมอร์สามารถเข้าใจแนวคิดการเขียนโค้ดโดยใช้บรรทัดที่น้อยลงกว่าภาษาอย่าง C++ และ Java ซึ่งภาษานั้นถูกกำหนดให้มีโครงสร้างที่ช่วยให้การเขียนโค้ดเข้าใจง่ายทั้งในโปรแกรมเล็กไปจนถึงโปรแกรมขนาดใหญ่

Python นั้นมีคุณสมบัติเป็นภาษาเขียนโปรแกรมแบบไดนามิกส์และมีระบบการจัดการหน่วยความจำอัตโนมัติและสนับสนุนการเขียนโปรแกรมหลายรูปแบบ ที่ประกอบไปด้วยการเขียนโปรแกรมเชิงวัตถุ *imperative* การเขียนโปรแกรมแบบฟังก์ชัน และการเขียนโปรแกรมแบบขั้นตอน มันมีไลบรารีที่ครอบคลุมการทำงานอย่างหลากหลาย ตัวแปรในภาษา Python นั้นมีให้ใช้ในหลายระบบปฏิบัติการ ทำให้โค้ดของภาษา Python สามารถรันในระบบต่างๆ ได้อย่างกว้างขวาง CPython นั้นเป็นการพัฒนาในขั้นต้นของ Python ซึ่งเป็นโปรแกรมแบบ *open source* และมีชุมชนสำหรับเป็นต้นแบบในการพัฒนา เนื่องจากมันได้มีการนำไปพัฒนากระจายไปอย่างหลากหลาย CPython นั้นจึงถูกจัดการโดยองค์กรไม่แสวงหาผลกำไรอย่าง Python Software Foundation



รูปที่ 2.16 ภาษา Python

ภาษา Python นั้นกำเนิดขึ้นในปลายปี 1980 และการพัฒนาของมันนั้นเริ่มต้นใน December 1989 โดย Guido van Rossum ที่ Centrum Wiskunde & Informatica (CWI) ในประเทศเนเธอร์แลนด์ เนื่องในผู้ประสบความสำเร็จในการสร้างภาษา ABC ที่มีความสามารถสำหรับการ *exception handling* และการติดต่อประสานกับระบบปฏิบัติการ Amoeba ซึ่ง Van Rossum นั้นเป็นผู้เขียนหลักการของภาษา Python และเขาทำหน้าที่เป็นกลางในการตัดสินใจสำหรับทิศทางการพัฒนาของภาษา Python เป็นต้น

2.4.2 โครงสร้างของภาษา Python

ในบทนี้ คุณจะได้เรียนรู้และทำความเข้าใจในโครงสร้างของภาษา Python ในภาษาคอมพิวเตอร์นั้นก็มีโครงสร้างของภาษาเช่นเดียวกับภาษามนุษย์ ซึ่งสิ่งเหล่านี้ถูกกำหนดเพื่อเป็นรูปแบบและวิธีในการเขียนโปรแกรมในภาษา Python มันใช้สำหรับควบคุมวิธีที่คุณจะเขียนโค้ดของคุณเพื่อให้เข้าใจโดยตัวแปลภาษาหรือคอมไพเลอร์เพื่อเริ่มต้นการเรียนรู้ในภาษา Python มาดูตัวอย่างของโปรแกรมอย่างง่าย โดยเป็นโปรแกรมที่ถามชื่อผู้ใช้และแสดงข้อความทักทายทางหน้าจอ มาเริ่มเขียนโปรแกรมแรกในภาษา Python ของคุณ ให้คัดลอกโปรแกรมข้างล่างแล้วนำไปรันใน IDE

```
# My first Python program
name = input('What is your name?\n')
print ('Hi, %s.' % name)
print ('Welcome to Python.')
```

รูปที่ 2.17 Simple Python

ในตัวอย่างเป็นโปรแกรมในการรับชื่อและแสดงข้อความทักทายออกทางหน้าจอ ในการรันโปรแกรมคุณสามารถรันได้หลายวิธี แต่ที่แนะนำคือการใช้ Python shell ให้คุณเปิด Python shell ขึ้นมาแล้วกดสร้างไฟล์ใหม่โดยไปที่ File -> New File จะปรากฏกล่อง Text editor ของภาษา Python ขึ้นมาเพื่อรันโปรแกรม Run -> Run Module หรือกด F5 โปรแกรมจะเปลี่ยนกลับไปยัง Python shell และเริ่มต้นทำงานนี้เป็นผลลัพธ์การทำงานในการรันโปรแกรม first.py จาก Python shell ในตัวอย่างเราได้กรอกชื่อเป็น "Mateo" และหลังจากนั้นโปรแกรมได้แสดงข้อความทักทายและจบการทำงาน ในตอนนี้คุณยังไม่ต้องกังวลว่าโปรแกรมในแต่ละบรรทัดนั้นทำงานอย่างไร ซึ่งเราจะอธิบายในต่อไป

2.4.3 Module

ในตัวอย่างโปรแกรมรับชื่อของเราเป็นโปรแกรมแรกของเราในบทเรียน Python นี้ และเราได้บันทึกเป็นไฟล์ที่ชื่อว่า first.py ซึ่งไฟล์ของภาษา Python นั้นจะเรียกว่า Module ซึ่ง Module จะประกอบไปด้วยคลาส ฟังก์ชัน และตัวแปรต่างๆ และนอกจากนี้เรายังสามารถ import โมดูลอื่นเข้ามาในโปรแกรมได้ ซึ่งโมดูลอาจจะอยู่ภายใน package ซึ่งเป็นเหมือน directory ของ Module ในตัวอย่าง first.py จึงเป็นโมดูลของโปรแกรมแรกของเรา

2.4.4 Comment

คอมเมนต์ในภาษา Python นั้นเริ่มต้นด้วยเครื่องหมาย # คอมเมนต์สามารถเริ่มต้นที่ตำแหน่งแรกของบรรทัดและหลังจากนั้นจะประกอบไปด้วย Whitespace หรือ โค้ดของ โปรแกรม หรือคำอธิบาย ซึ่งโดยทั่วไปแล้วคอมเมนต์มักจะใช้สำหรับอธิบายข้อโค้ดที่เราเขียนขึ้นและมันไม่มีผลต่อการทำงานของโปรแกรม นี่เป็นตัวอย่างการคอมเมนต์ในภาษา Python

```
# My first Python program

'''
This is a multiline comment
'''

print ('Hello Python.') # Inline comment
```

รูปที่ 2.18 Comment

ในตัวอย่าง เราได้คอมเมนต์สามแบบด้วยกัน แบบแรกเป็นการคอมเมนต์แบบ single line แบบที่สองเป็นการคอมเมนต์แบบ multiline line และแบบสุดท้ายเป็นการคอมเมนต์แบบ inline หรือการคอมเมนต์ภายในบรรทัดเดียวกัน

2.4.5 Statement

Statement คือคำสั่งการทำงานของโปรแกรม แต่ละคำสั่งในภาษา Python นั้นจะแบ่งแยกด้วยการขึ้นบรรทัดใหม่ ซึ่งจะแตกต่างจากภาษา C และ Java ซึ่งใช้เครื่องหมายเซมิโคลอนสำหรับการจบคำสั่งการทำงาน แต่อย่างไรก็ตาม ในภาษา Python นั้นคุณสามารถมีหลายคำสั่งในบรรทัดเดียวกันได้โดยการใช้เครื่องหมายเซมิโคลอน;

```
name = input('What is your name?\n')
print ('Hi, %s.' % name);
print ('Welcome to Python.');
```

รูปที่ 2.19 Statement

2.5 ไลบรารีฟังก์ชันต่าง ๆ

2.5.1 OpenCV (Open source Computer Vision)



รูปที่ 2.20 OpenCV (Open source Computer Vision)

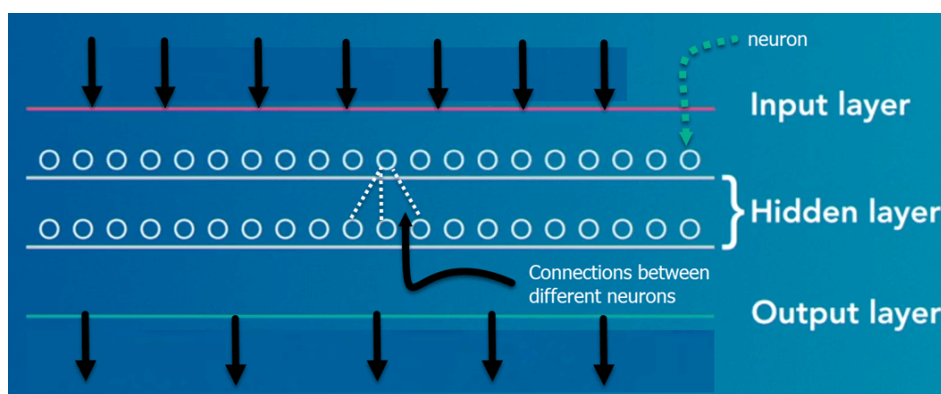
OpenCV (Open source Computer Vision) เป็นไลบรารีฟังก์ชันการเขียนโปรแกรม (Library of Programming Functions) โดยส่วนใหญ่จะมุ่งเป้าไปที่การแสดงผลด้วยคอมพิวเตอร์แบบเรียลไทม์ (Real-Time Computer Vision) เดิมทีแล้วถูกพัฒนาโดย Intel แต่ภายหลังได้รับการสนับสนุนโดย Willow Garage ตามมาด้วย Itseez (ซึ่งต่อมาถูกเข้าซื้อโดย Intel) OpenCV เป็นไลบรารีแบบข้ามแพลตฟอร์ม (Cross-Platform) และใช้งานได้ฟรีภายใต้ลิขสิทธิ์ของ BSD แบบโอเพ่นซอร์ส (Open-Source BSD License) OpenCV ยังสนับสนุนเฟรมเวิร์กการเรียนรู้เชิงลึก (Deep Learning Frameworks) ได้แก่ TensorFlow, Torch/PyTorch และ Caffe

โปรแกรม OpenCV เป็น Library ในภาษา C++ และ Python สำหรับการพัฒนาโปรแกรมที่เกี่ยวข้องกับ Image Processing และ Computer Vision โดยสามารถพัฒนาได้ทั้งในระบบปฏิบัติการวินโดวส์ และ ระบบปฏิบัติการ Linux วิธีใช้งาน OpenCV ใช้งานร่วมกับโปรแกรม Visual Studio C++ ในการนำ OpenCV มาใช้งานร่วมกับ Visual Studio C++ เพื่อใช้พัฒนาโปรแกรมนั้น เราจำเป็นต้องตั้งค่าให้กับโปรแกรม Visual Studio C++ ก่อน เพื่อระบุตำแหน่งของ Library ของ OpenCV ตำแหน่งของไฟล์ที่ต้องใช้ในโปรแกรมและตำแหน่งของ Source File ให้ตัวโปรแกรมทราบ และสามารถดึงมาใช้ได้ Color หรือ Object จะประกอบไปด้วย Histogram ของสีต่างๆ เราจะเก็บลักษณะของ histogram เอาไว้เช็คข้อมูลที่อยู่ในภาพ ที่เปลี่ยนแปลงอยู่ตลอดเวลา หากมีลักษณะที่ใกล้เคียงก็สามารถที่จะระบุตำแหน่งของ object หรือ color เป็นต้น

2.5.2 Deep learning

Deep learning คือ deep learning คือ ซอฟต์แวร์คอมพิวเตอร์ที่เลียนแบบการทำงานของระบบโครงข่ายประสาท (neurons) ในสมองมนุษย์ ถือเป็นขั้นหนึ่งของ machine learning Algorithm ของ deep learning ถูกสร้างขึ้นจากการนำเอา neural network หลายๆ layer มาต่อกัน โดย layer แรกสุดจะทำหน้าที่ในการรับข้อมูล (Input layer) layer สุดท้ายจะทำหน้าที่ส่งผลลัพธ์การประมวลผลออกมา (Output layer) ส่วน layer ระหว่าง layer แรกสุด และ layer สุดท้าย จะถูกเรียกว่า Hidden layer

คำว่า deep learning มีที่มาจากการที่ใช้ layer ของ neural network หลายอันๆมาต่อกัน (มี Hidden layer มากกว่า 2 layer ก็ถือเป็น deep learning แล้ว) เนื่องจาก layer เหล่านี้เป็นโครงสร้างที่ถูกจัดเก็บแบบเป็นกองซ้อน (stack) จึงเปรียบได้ว่า layer ที่เยอะๆ ก็จะทำให้มีโครงสร้างที่ลึก (deep) ยิ่งขึ้นนั่นเอง



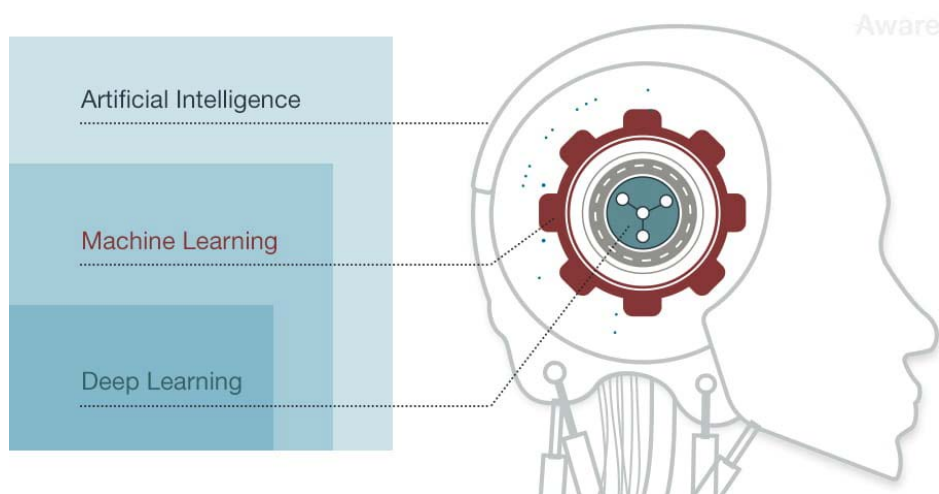
รูปที่ 2.21 Deep learning

กระบวนการทำงานของ deep learning model ที่ใช้ deep learning ให้ความแม่นยำ (accuracy) ที่สูงในหลายๆปัญหา ตั้งแต่การตรวจจับวัตถุ (object detection) ไปจนถึงการรู้จำเสียงพูด (speech recognition) โดยที่เราไม่จำเป็นต้องให้ความรู้พื้นฐานใดๆกับมันไว้ล่วงหน้าเลย เพียงแค่ให้ข้อมูลตัวอย่าง (input data) มันก็จะทำการเรียนรู้จากข้อมูลและสังเคราะห์เป็นองค์ความรู้ ออกมาได้อย่างอัตโนมัติ อาทิเช่น การใช้ deep learning ในวงการเกม เราไม่จำเป็นต้องบอกมันว่า เล่นยังไง แค่ให้มันเรียนรู้จากผู้เล่นที่เก่งๆเป็นจำนวนมาก มันก็เรียนรู้วิธีการเล่นเกมได้อย่างอัตโนมัติเพื่อที่จะเข้าใจหลักการของ deep learning มากยิ่งขึ้น , ให้คุณจินตนาการถึงครอบครัวที่มี พ่อแม่ และ ลูกน้อยวัยละอ่อน , ลูกน้อยคนนั้นมักจะชี้อะไรต่างๆและพูดขึ้นพูดว่า " นี่คือเจ้าแมว" ซึ่งไม่ว่าจะชี้ไปที่ทีวี ตู้เย็น หรือ สุนัข ก็มักจะพูดประโยคนี้ออกมา

2.5.3 Machine Learning

Machine Learning คือ ส่วนการเรียนรู้ของเครื่อง ถูกใช้งานเสมือนเป็นสมองของ *AI (Artificial Intelligence)* เราอาจพูดได้ว่า *AI* ใช้ *Machine Learning* ในการสร้างความฉลาด มักจะใช้เรียกโมเดลที่เกิดจากการเรียนรู้ของปัญญาประดิษฐ์ ไม่ได้เกิดจากการเขียนโดยใช้นุ้ยมนุษย์มีหน้าที่เขียนโปรแกรมให้ *AI (เครื่อง)* เรียนรู้จากข้อมูลเท่านั้น ที่เหลือเครื่องจัดการเอง

Machine Learning เรียนรู้จากสิ่งที่เราส่งเข้าไปกระตุ้น แล้วจดจำเอาไว้เป็นมันสมอง ส่งผลลัพธ์ออกมาเป็นตัวเลข หรือ *code* ที่ส่งต่อไปแสดงผล หรือให้เจ้าตัว *AI* นำไปแสดงการกระทำ *Machine Learning* เองสามารถเอาไปใช้งานได้หลายรูปแบบ ต้องอาศัยกลไกที่เป็นโปรแกรม หรือเรียกว่า *Algorithm* ที่มีหลากหลายแบบ โดยมี *Data Scientist* เป็นผู้ออกแบบ หนึ่งใน *Algorithm* ที่ได้รับความนิยมสูง คือ *Deep Learning* ซึ่งถูกออกแบบมาให้ใช้งานได้ง่าย และประยุกต์ใช้ได้หลายลักษณะงาน อย่างไรก็ตาม ในการทำงานจริง *Data Scientist* จำเป็นต้องออกแบบตัวแปรต่างๆ ทั้งในตัวของ *Deep Learning* เอง และต้องหา *Algorithm* อื่นๆ มาเป็นคู่เปรียบเทียบกับเพื่อมองหา *Algorithm* ที่เหมาะสมที่สุดในการใช้งานจริง



รูปที่ 2.22 Machine Learning

Machine Learning สมองที่เรียนรู้ทั้งหมดเกิดขึ้น วิธีที่ *machine* เรียนรู้เหมือนกับมนุษย์มนุษย์เรียนรู้จากประสบการณ์ ยิ่งพวกเรารู้มาก ยิ่งง่ายต่อการพยากรณ์ว่าสิ่งต่อไปอะไรจะเกิดขึ้น โดยเปรียบเทียบ, เมื่อพวกเราประสบกับเหตุการณ์ที่ไม่เคยเจอมาก่อน, มีความเป็นไปได้ที่ความสำเร็จจะลดลงกว่าเหตุการณ์ที่เคยเจอมาแล้ว *machine* สามารถถูกฝึก(*train*) ได้ในรูปแบบเดียวกัน เพื่อที่จะเพิ่มความแม่นยำในการพยากรณ์ *machine* สามารถมองเห็นตัวอย่างที่เราให้

2.5.4 Tkinter (ทีเคอินเตอร์)

Tkinter เป็น โมดูล อินเตอร์เฟซมาตรฐาน .Tk python Tk GUI Toolkit และ Tkinter สามารถใช้ได้มากที่สุดในแพลตฟอร์ม Unix เดียวกันสามารถนำไปใช้กับทั้งระบบ Windows และ Macintosh, Tk8.0Tkinter มาตรฐานห้องสมุด Python GUI ใช้ Tkinter สามารถสร้างโปรแกรม GUI การสร้างโปรแกรมแบบ GUI โมดูลนำเข้า Tkinter , สร้างการควบคุม , กำหนดต้นแบบการควบคุม ที่เป็นตัวควบคุมที่พวกเขาอยู่ , บอกว่าจีเอ็ม (ผู้จัดการเรขาคณิต) มีการผลิตการควบคุม

โมดูล Tkinter ให้ความหลากหลายของการควบคุมเช่น ปุ่ม ป้ายและกล่องข้อความ , การใช้งานโปรแกรมประยุกต์แบบ GUI การควบคุมเหล่านี้มักจะเรียกว่าการควบคุมหรือส่วนประกอบ

2.5.4.1 จุดเด่น

2.5.4.1.1 ใช้งานกันแพร่หลาย

2.5.4.1.2 มาพร้อมกับ Python ทุกรุ่น (ในกรณีที่มีการรวมมากับ distro)

2.5.4.1.3 มีเอกสารเป็นมาตรฐานอยู่ใน python doc

2.5.4.1.4 ทำงานได้รวดเร็ว

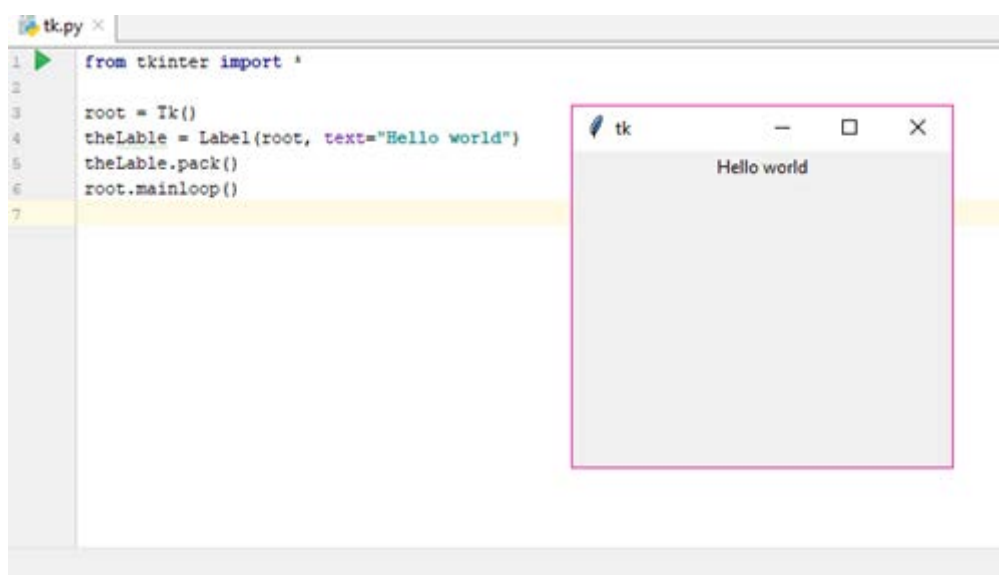
2.5.4.1.5 มี API ง่ายต่อการเรียนรู้เหมาะกับโครงการขนาดเล็ก

2.5.4.2 จุดด้อย

2.5.4.2.1 หน้าตาแต่ละ ไม่เหมือนกัน

2.5.4.2.2 มีความยุ่งยากกับอินเทอร์เฟซที่ซับซ้อน

2.5.4.2.3 ไม่มี GUI builder มาในตัว



รูปที่ 2.23 Tkinter (ทีเคอินเตอร์)

2.5.5 NumPy (Numeric Python)

NumPy (Numeric Python) เป็นโมดูลส่วนเสริมของ Python ที่มีฟังก์ชันนี้เกี่ยวกับคณิตศาสตร์และการคำนวณต่างๆ มาให้ใช้งาน โดยทั่วไปจะ เกี่ยวกับการจัดการข้อมูลชุด (Array) ขนาดใหญ่และเมทริกซ์ NumPy นี้ครอบคลุมการคำนวณมากมายสามารถทำงานได้ใกล้เคียงกับ commercial software เช่น MatLab เลยทีเดียวเนื่องจาก NumPy มีความสามารถมากในรายวิชานี้เราจะเรียนเกี่ยวกับ Vector and matrix mathematics มีหลายวิธีในการ import NumPy เข้ามาใช้งาน โดยทั่วไปวิธีมาตรฐานได้แก่คำสั่ง `import numpy` อย่างไรก็ตามหากมีการเรียกใช้งาน NumPy บ่อยครั้ง แต่แต่ละครั้งจะต้องพิมพ์ `numpy.X` ดังนั้นเพื่อความสะดวก เราจะย่อเวลาเรียกเป็น `np` เราจะเปลี่ยนการ import เป็น `import numpy as np` ทำให้เราเรียกใช้งานฟังก์ชันได้โดยเขียนเป็น `np.x`

2.5.5.1 Arrays

Arrays เป็นคุณสมบัติหลักของ NumPy มีลักษณะคล้ายกับ list ยกเว้นสมาชิกทุกตัวใน array จะต้องเป็นข้อมูลชนิดเดียวกัน โดยทั่วไปแล้วข้อมูล ที่เก็บจะเป็นตัวเลขเช่น `int` หรือ `float` Arrays มีความสามารถในการดำเนินการเกี่ยวกับข้อมูลที่เป็นตัวเลข จำนวนมาก ๆ ได้อย่างรวดเร็วและมีประสิทธิภาพมากกว่า list เราจะนำ Array นี้มาสร้างเป็น Vectors และ Matrices

- Vector เป็นลำดับของตัวเลขที่เขียนในรูป

$$u = \begin{pmatrix} u_1 \\ u_2 \\ \dots \\ u_n \end{pmatrix}$$

ตัวอย่างเช่น $\begin{pmatrix} 1 \\ -2 \end{pmatrix}, \begin{pmatrix} 0.6 \\ -4 \\ 9 \end{pmatrix}$

- Matrix เป็นแถวลำดับสี่เหลี่ยมของตัวเลขซึ่งเขียนในรูป

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix}$$

ตัวอย่างเช่น $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$

รูปที่ 2.24 Vector and Matrix

การสร้าง array เราจะสร้าง Vector และ Matrix ได้โดยการใช้ Array สามารถถูกสร้างได้จาก `list` `import numpy as np > a = np.array([1,2,4,8],float) > print(a) > type(a)` Array มีมิติเทียบได้กับ Vector นั้นเอง การสร้างและเข้าถึงข้อมูลใน array ในการสร้าง array จะรับค่า 2 ค่าได้แก่ list ที่ต้องการเปลี่ยนเป็น array , ชนิดของสมาชิกที่ต้องการสร้างเป็น array สมาชิกของ array จะถูกเข้าถึง แบ่งและจัดการได้เช่นเดียวกับ list เช่น

```
import numpy as np
a = np.array([1,2,4,8], float)
print(a[3])           8.0
a[0] = 5
print(a)              [ 5.  2.  4.  8.]
```

รูปที่ 2.25 การสร้างและเข้าถึงข้อมูลใน array

การเข้าถึงข้อมูลใน array หลายมิติ Array สามารถทำให้เป็นหลายมิติได้ไม่เหมือนกับ list การสามารถเข้าถึง ข้อมูลในแกนที่แตกต่างกันทำได้โดยใช้ comma ภายในปีกกา Array 2 มิติเทียบได้กับ Matrix นั้นเอง

```
import numpy as np
a = np.array([[1,2,3],[4,5,6]],float)
print(a)
[[ 1.  2.  3.]
 [ 4.  5.  6.]]

print(a[0,0])         1.0
print(a[0,1])         2.0
```

รูปที่ 2.26 การเข้าถึงข้อมูลใน array หลายมิติ

บทที่ 3

วิธีการดำเนินโครงการ

ในการดำเนินโครงการระบบรู้จำใบหน้าและระบุตัวตนด้วยกล้อง (Deep Learning Face Recognition by Camera) มีขั้นตอนการสร้างในส่วนต่าง ๆ โดยทางกลุ่มผู้สร้างได้ร่วมกันวางแผนในการปฏิบัติงาน และจัดแบ่งงานตามความเหมาะสม

ขั้นตอนในการดำเนินโครงการ แบ่งออกเป็นดังนี้

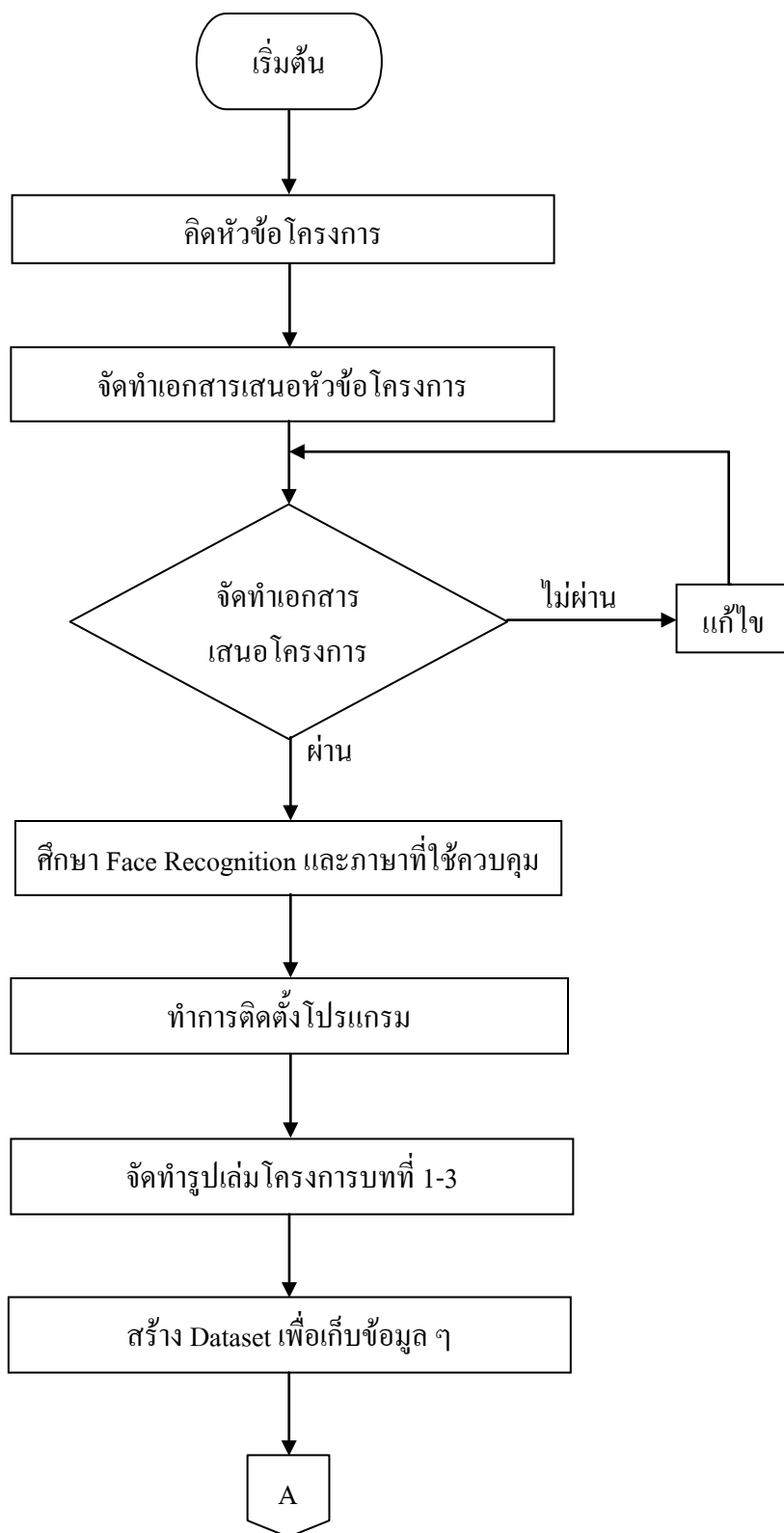
- 3.1 การวางแผนและการเตรียมงาน
- 3.2 การออกแบบ
- 3.3 การดำเนินการสร้าง

3.1 ขั้นตอนการวางแผนและการเตรียมการ

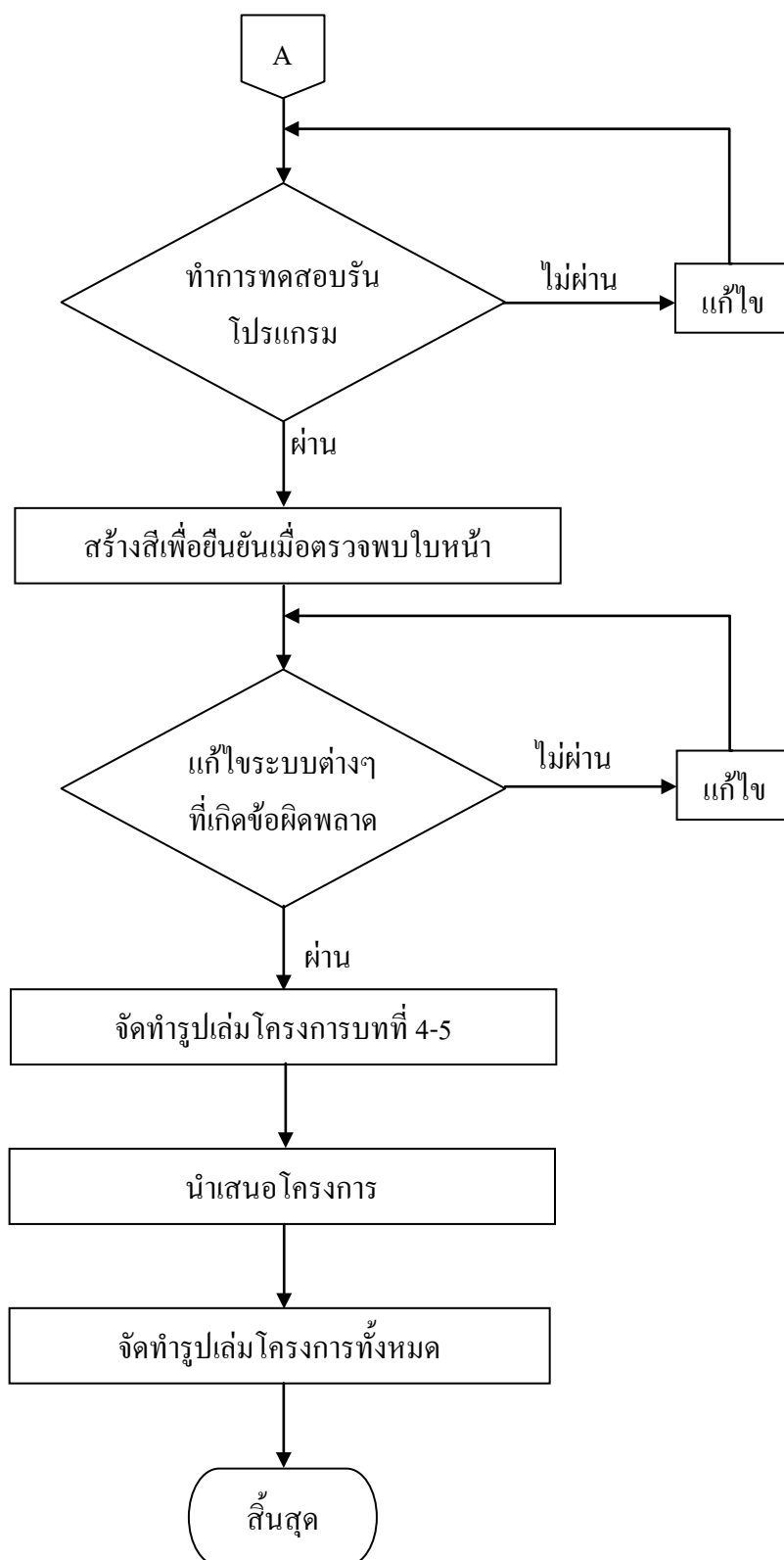
การวางแผนและการดำเนินโครงการ เริ่มเมื่อคณะกรรมการพิจารณาโครงการให้เสนอหัวข้อโครงการในภาคเรียนที่ 1 ทางกลุ่มผู้จัดทำได้เสนอหัวข้อโครงการระบบรู้จำใบหน้าและระบุตัวตนด้วยกล้อง

ซึ่งมีลำดับขั้นตอนต่าง ๆ ในการดำเนินการดังตารางที่ 3.1

- 3.1.1 การวางแผนการทำโครงการ
 - 3.1.1.1 คิดหัวข้อโครงการ
 - 3.1.1.2 จัดทำรูปเล่มเสนอโครงการ
 - 3.1.1.3 เสนอร่างโครงการ
 - 3.1.1.4 แก้ไขเสนอร่างโครงการ
 - 3.1.1.5 ศึกษาเทคโนโลยี Face Recognition และภาษาที่ใช้ควบคุม
 - 3.1.1.6 ทำการติดตั้งโปรแกรม vs code , Python 3.6.8 และ Library ที่ใช้
 - 3.1.1.7 จัดทำรูปเล่มโครงการบทที่ 1-3
 - 3.1.1.8 สร้าง Dataset เพื่อเก็บข้อมูล
 - 3.1.1.9 ทำการทดสอบรันโปรแกรม
 - 3.1.1.10 สร้างเสียงเพื่อยืนยันเมื่อตรวจพบใบหน้าตรงตาม Dataset
 - 3.1.1.11 ปรับปรุงแก้ไขระบบต่างๆที่เกิดข้อผิดพลาด
 - 3.1.1.12 จัดทำรูปเล่มโครงการบทที่ 4-5
 - 3.1.1.13 นำเสนอโครงการ
 - 3.1.1.14 จัดทำรูปเล่มโครงการทั้งหมด



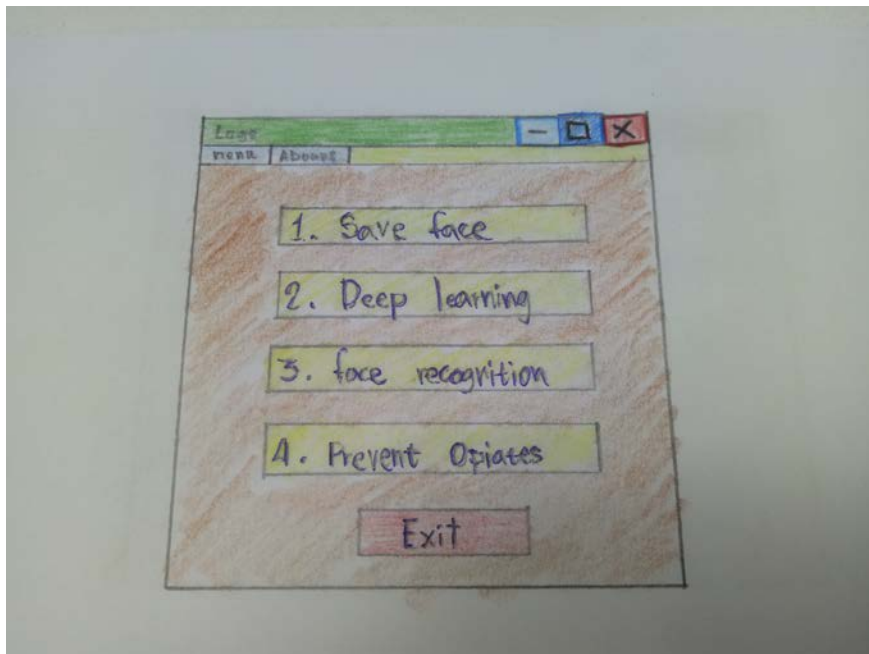
รูปที่ 3.1 แผนภาพ Flowchart ขั้นตอนการดำเนินงาน



รูปที่ 3.2 แผนภาพ Flowchart ขั้นตอนการดำเนินงาน (ต่อ)

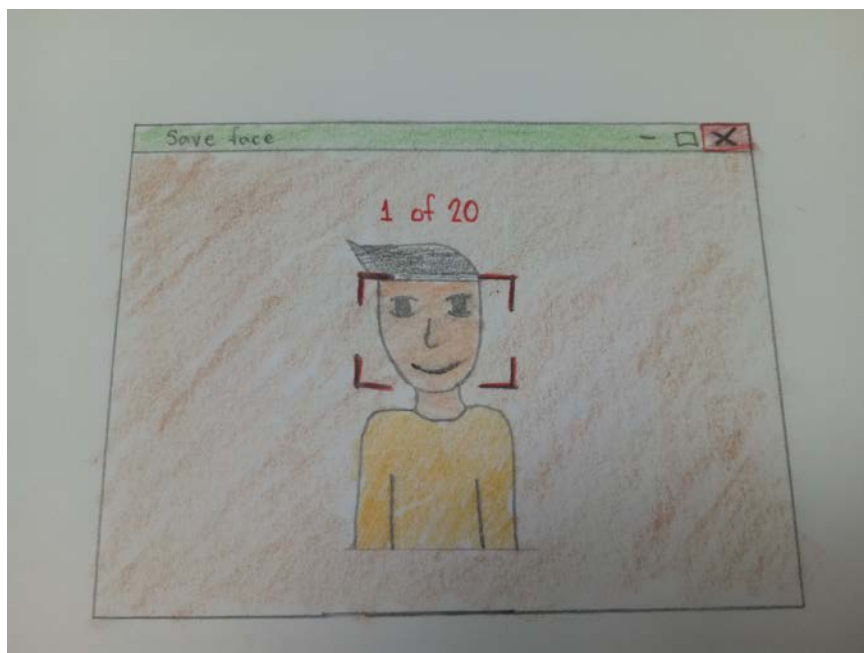
3.2 การออกแบบ

3.2.1 รูปหน้าต่างเมนูของระบบรู้จำใบหน้าและระบุตัวตนด้วยกล้อง



รูปที่ 3.3 หน้าเมนูของระบบรู้จำใบหน้า

3.2.2 Button ที่ 1 บันทึกใบหน้า Save face



รูปที่ 3.3 หน้าบันทึกใบหน้า Save face

3.2.3 Button ที่ 2 ทำการ Training ให้ระบบรู้จำใบหน้า

```

PROBLEMS 118 OUTPUT DEBUG CONSOLE TERMINAL
Accuracy Score: 99.67 %
Confusion Matrix:
[[ 99  1  0  0  0  0]
 [  0 100  0  0  0  0]
 [  0  1 99  0  0  0]
 [  0  0  0 100  0  0]
 [  0  0  0  0 100  0]
 [  0  0  0  0  0 100]]
Classification Report:
              precision    recall  f1-score   support

   Aon         1.00        0.99        0.99        100
   First       0.98        1.00        0.99        100
   Home        1.00        0.99        0.99        100
   Koy         1.00        1.00        1.00        100
   Nick        1.00        1.00        1.00        100
   Pu          1.00        1.00        1.00        100

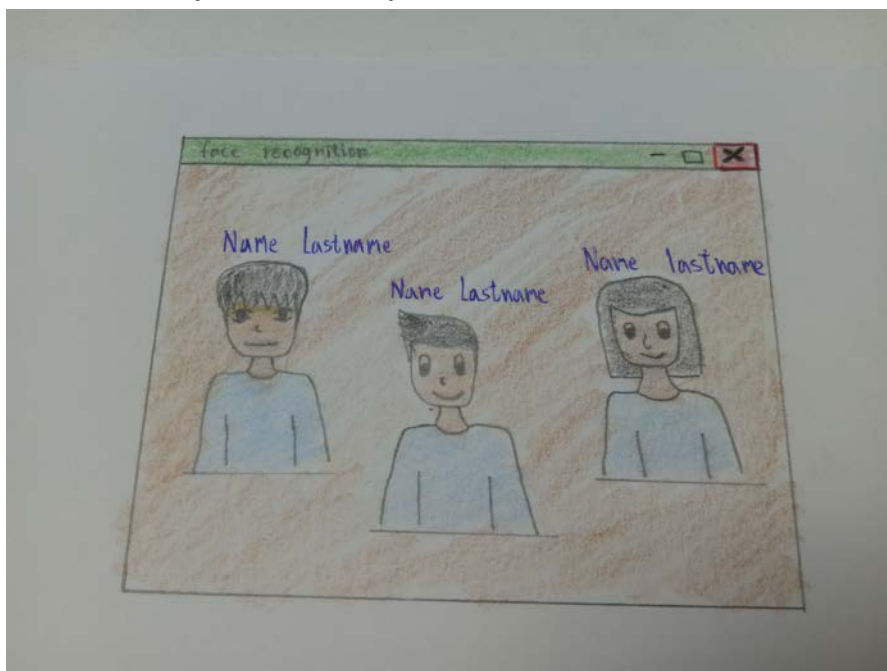
 accuracy          1.00
macro avg          1.00        1.00        1.00        600
weighted avg       1.00        1.00        1.00        600

PS D:\FaceReconition>

```

รูปที่ 3.4 หน้า Training ให้ระบบรู้จำใบหน้า

3.2.4 Button ที่ 3 รูปหน้าต่างระบบรู้จำใบหน้าและระบุตัวตนด้วยกล้อง



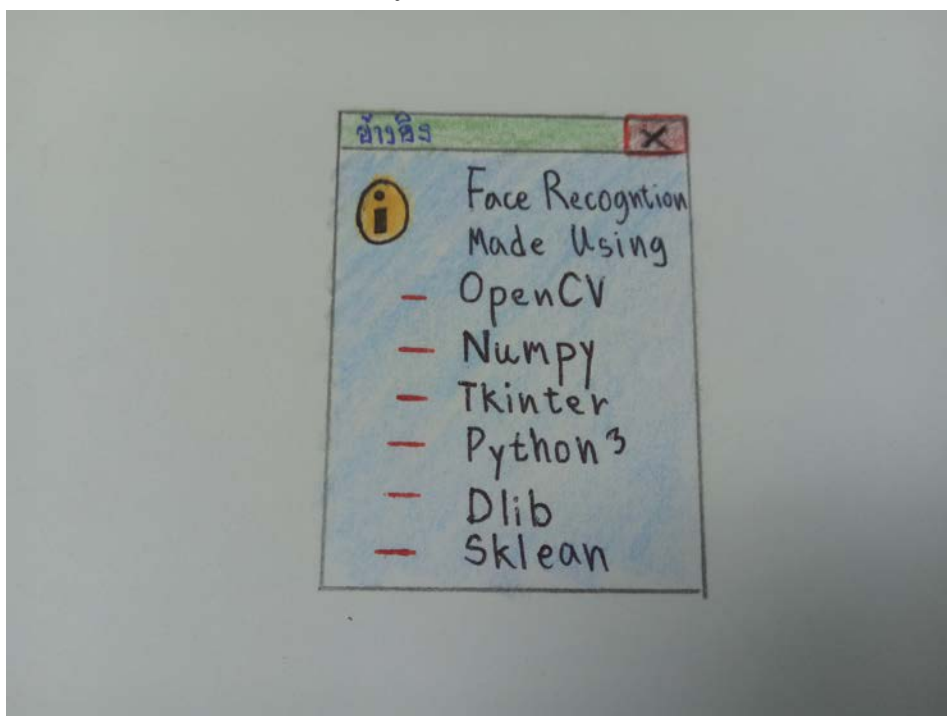
รูปที่ 3.5 หน้าต่างระบบรู้จำใบหน้าและระบุตัวตนด้วยกล้อง

3.2.5 Button ที่ 4 หน้าต่างกันหลักในขณะรถ Prevent Opiates



รูปที่ 3.6 หน้าต่างกันหลักในขณะรถ

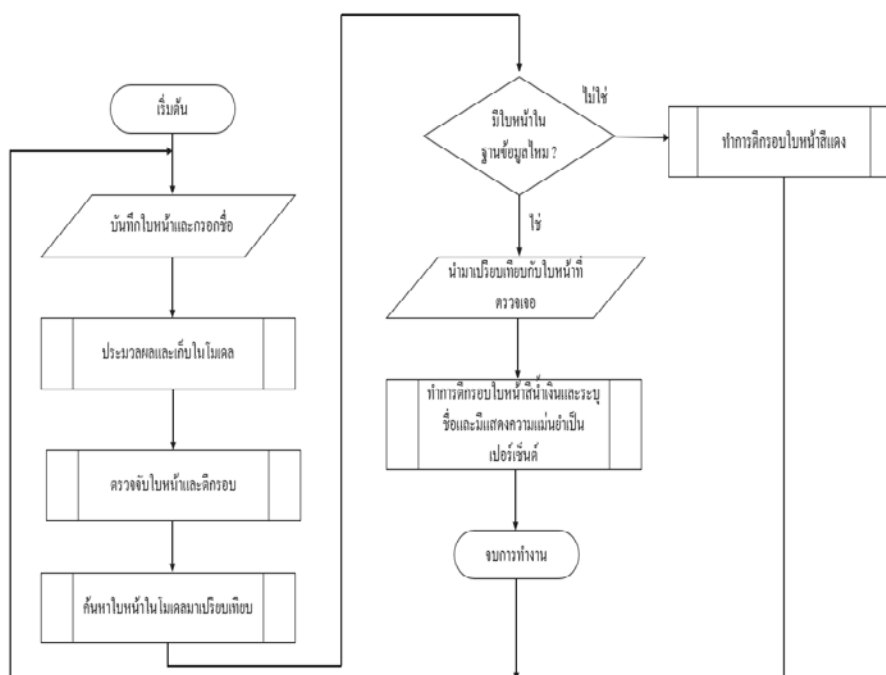
3.2.6 หน้าต่างอ้างอิงระบบระบุจำใบหน้าและระบุตัวตน



รูปที่ 3.7 หน้าต่างอ้างอิงระบบ

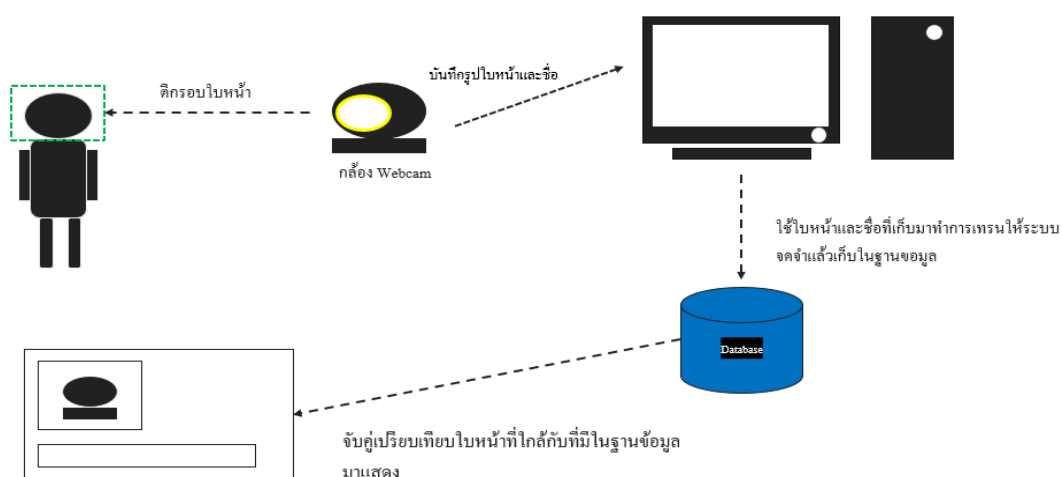
3.3 การดำเนินการสร้าง

3.3.1 Flow Chart โครงการระบบรู้จำใบหน้าและระบุตัวตนด้วยกล้อง



รูปที่ 3.8 Flow Chart ระบบรู้จำใบหน้าและระบุตัวตนด้วยกล้อง

3.3.2 แผนผังการทำงานของระบบ



รูปที่ 3.9 การทำงานของระบบ

บทที่ 4

ผลการศึกษา

ผลการศึกษาทำให้เราได้รู้ถึงการทำงานร่วมกันเป็นกลุ่มต้องสามัคคีกันในการวิเคราะห์และออกแบบตลอดจนถึงขั้นตอนการทำงานแต่ละขั้นตอนว่ามีอะไรบ้าง ในที่นี้ทางคณะผู้จัดทำจะกล่าวถึงผลของการศึกษาข้อมูล และผลที่ได้รับอย่างละเอียด แบ่งออกเป็นดังนี้

4.1 ขั้นตอนการออกแบบชิ้นงาน

4.2 ขั้นตอนการทำงาน

4.3 ขั้นตอนการทดสอบระบบจดจำใบหน้าและระบุตัวตนด้วยกล้อง

4.1 ขั้นตอนการออกแบบชิ้นงาน

ขั้นตอนนี้คณะผู้จัดทำได้ทำการประชุมกลุ่มกันออกความคิดเห็นและเพื่อแบ่งหน้าที่ของแต่ละคนเพื่อให้ขั้นตอนในการทำงานได้อย่างรวดเร็ว และมีการกระจายหน้าที่การทำงานให้เท่าเทียมกันเพื่อให้ทุกคนได้แสดงความคิดเห็นได้อย่างอิสระ และให้ทุกคนมีส่วนร่วมในการทำงาน และรับความคิดเห็นของผู้อื่นด้วย ขั้นตอนการออกแบบมีดังนี้

4.1.1 แนวคิด แนวคิดที่ทางคณะผู้จัดทำได้เริ่มจากการทำ โครงการระบบจดจำใบหน้าและระบุตัวตนด้วยกล้อง (Face Recognition) ทางคณะผู้จัดทำจึงมีความเห็นให้นำ Visual Studio Code (Vs Code) มาใช้ให้เป็นชุดควบคุมการเขียนคำสั่งโค้ดของภาษาไพธอน (Python 3) ในการพัฒนาระบบจดจำใบหน้าและระบุตัวตนด้วยกล้อง เพื่อจะได้เป็นต้นแบบให้กับนักศึกษา สาขาเทคโนโลยีสารสนเทศ ต่อไป



รูปที่ 4.1 วิวทศตูดิโอโค้ด Visual Studio Code

4.1.2 การเริ่มต้นคิดแบบ ในการเริ่มต้นคิดแบบนี้ทางคณะผู้จัดทำได้เริ่มต้นกันค้นหาข้อมูลต่าง ๆ ตามเว็บไซต์ และค้นหาอุปกรณ์ที่จะได้ใช้เพื่อสอดคล้องกับ ชุดคำสั่งของภาษาไพธอน (Python 3) ในการพัฒนาระบบจดจำใบหน้าและระบุตัวตนด้วยกล้อง อุปกรณ์ที่ทางคณะผู้จัดทำคิดไว้ มีดังนี้

4.1.2.1 คอมพิวเตอร์ (Computer) , โน้ตบุ๊ก (Notebook)

4.1.2.2 สายเชื่อมต่อ (USB)

4.1.2.3 ลำโพง (loudspeaker)

4.1.2.4 กล้องเว็บแคม (Webcam Camera)

4.2 ขั้นตอนการทำงาน

ขั้นตอนการทำงานนั้นกล่าวถึงขั้นตอนในการทำงานตั้งแต่ต้นว่ามีอะไรบ้าง

4.2.1 เริ่มขั้นตอนแรกโดยเราทำการ Import Library tkinter เข้ามา

```
from tkinter import *
import tkinter.messagebox
from tkinter import messagebox
```

รูปที่ 4.2 Import Library tkinter

4.2.2 สร้างเมนู> ออก และ เกี่ยวกับ>ไลบรารี> ผู้ออกแบบ

```
menu = Menu(root)
root.config(menu=menu)

subm1 = Menu(menu)
menu.add_cascade(label="เมนู", menu=subm1)
subm1.add_command(label="ออก", command=exit)

subm2 = Menu(menu)
menu.add_cascade(label="เกี่ยวกับ", menu=subm2)
subm2.add_command(label="ไลบรารี", command=anotherWin)
subm2.add_command(label="ผู้ออกแบบ", command=Contri)
```

รูปที่ 4.3 สร้างหน้าเมนู

4.2.3 สร้างปุ่มเพื่อเก็บค่าคำสั่งเมื่อกดไปแล้วเปิดเป็นระบบต่าง ๆ

```

but1=Button(frame,padx=5,pady=5,width=30,bg='white',fg='black',relief=GROOVE,command=sa,text='ระบบบันทึกใบหน้า',font=('helvetica 15 bold'))
but1.place(x=55,y=110)

but2=Button(frame,padx=5,pady=5,width=30,bg='white',fg='black',relief=GROOVE,command=ta,text='ระบบรู้จำใบหน้า',font=('helvetica 15 bold'))
but2.place(x=55,y=180)

but3=Button(frame,padx=5,pady=5,width=30,bg='white',fg='black',relief=GROOVE,command=fa,text='ระบบจดจำใบหน้าและระบุตัวตน',font=('helvetica 15 bold'))
but3.place(x=55,y=250)

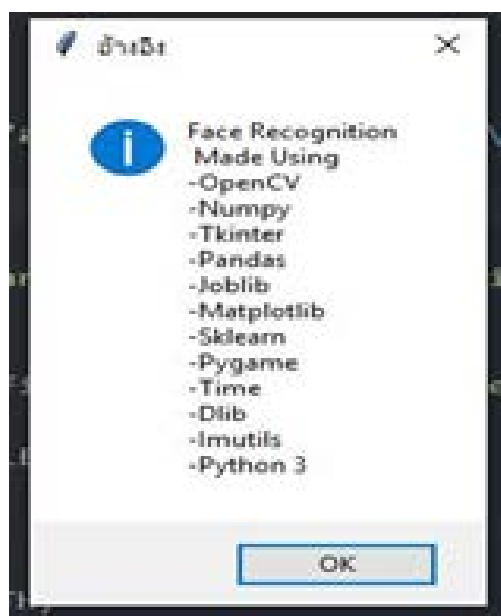
but3=Button(frame,padx=5,pady=5,width=30,bg='white',fg='black',relief=GROOVE,command=dd,text='ระบบตรวจจัดการหลับใหล',font=('helvetica 15 bold'))
but3.place(x=55,y=320)

but5=Button(frame,padx=5,pady=5,width=5,bg='white',fg='black',relief=GROOVE,text='aan',command=exitt,font=('helvetica 12 bold'))
but5.place(x=210,y=478)

```

รูปที่ 4.4 สร้างปุ่มเพื่อเก็บค่าคำสั่ง

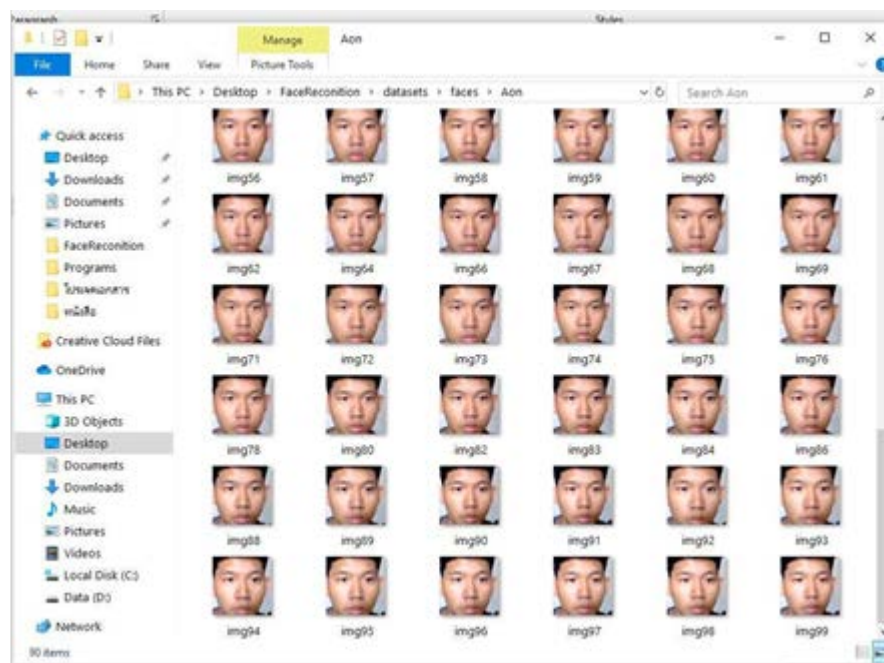
4.2.4 ให้ทำการสร้างฟังก์ชันของปุ่ม เกี่ยวกับ>ไลบรารี> ผู้ออกแบบ เพื่อให้ได้หน้าต่าง
เช่นเมื่อเรากดปุ่ม ไลบรารีก็จะโชว์หน้าต่างดังนี้



รูปที่ 4.5 ทำการสร้างฟังก์ชันของปุ่ม เกี่ยวกับ>ไลบรารี> ผู้ออกแบบ

4.2.5 ออกแบบระบบบันทึกใบหน้าโดยระบบบันทึกใบหน้านี้จะทำงานโดยการไปสร้างโฟลเดอร์> โฟลเดอร์ชื่อที่เราทำการกรอกเข้ามาก่อนทำการบันทึกใบหน้า > รูปภาพจะอยู่ในโฟลเดอร์ที่เราบันทึกเข้ามาก่อนหน้านี้เช่นพอร์นโปรแกรมระบบจะให้Input Name : เราก็ทำการกรอกชื่อในที่นี้เราทำการกรอกว่า Aon จากนั้นระบบก็จะทำการบันทึกใบหน้าของเราโดยที่เรามองไปที่กล้องเพื่อให้ใบหน้าที่ชัดเจนตัวอย่าง Path ที่บันทึก

C:\Users\thara\Desktop\FaceReconition\datasets\faces\Aon



รูปที่ 4.6 ออกแบบระบบบันทึกใบหน้า

4.2.6 ให้เรากำหนดที่เก็บของรูปภาพ,ขนาดไซส์ของรูปภาพ,จำนวนรูปภาพที่ต้องการบันทึก,โมเดลตัวติกรอบใบหน้า,กำหนดสีของกรอบที่วงรอบ ๆ ใบหน้าเรา ดังนี้

```
OUTPUT_PATH = 'datasets/faces'
SIZE = (64,64)
MAX_CAPTURE = 100
detector = cv2.CascadeClassifier('C:\\Users\\thara\\Desktop\\FaceReconition\\model-haar\\haarcascade_frontalface_default.xml')

font = cv2.FONT_HERSHEY_SIMPLEX
color = (0,255,0)
```

รูปที่ 4.7 กำหนดที่เก็บของรูปภาพ,ขนาดไซส์ของรูปภาพ,จำนวนรูปภาพที่ต้องการ

4.2.7 ในขั้นตอนนี้เราจะทำการกรอกชื่อของบุคคลนั้นก่อนแล้วจะนำไปสร้าง Folder ใน Datasets > Faces > “ชื่อที่เรากรอกไว้”

```
label = input('กรุณากรอกชื่อ : ')
output_path = Path(OUTPUT_PATH)
if not output_path.exists():
    output_path.mkdir()
output_face_path = Path(OUTPUT_PATH + '/' + label)
if not output_face_path.exists():
    output_face_path.mkdir()
```

รูปที่ 4.8 Datasets > Faces > “ชื่อที่เรากรอกไว้”

4.2.8 ต่อมาเราจะทำการเรียกใช้กล้องจากภายนอกคือกล้อง Webcam แล้วก็กำหนดตัวแปรไว้จำนวนรูปคือ Count = 0 จะได้ดังนี้ตรง Capture = cv2.VideoCapture(1) ตรง (1) ก็เป็นกล้องจากต่อภายนอกถ้าเป็นกล้องของโน้ตบุ๊กหรือกล้องติดมากับเครื่องจะใช้(0)ถ้าเป็นกล้องนอกจะเริ่มที่(1) ไป เรียกว่าเรามีกล้องกี่ตัวก็นับไปครับแล้วนำมาใส่ในวงเล็บ

```
count = 0
capture = cv2.VideoCapture(1)
```

รูปที่ 4.9 การเรียกใช้กล้องจากภายนอกคือกล้อง Webcam

4.2.9 ขั้นตอนนี้เราจะทำการนับตัวแปร Count มาเปรียบเทียบกับ MAX_CAPTURE โดยใช้คำสั่ง While Loop() โดยถ้า Count น้อยกว่า MAX_CAPTURE จะทำการเข้าสู่ Loop จากนั้นจะทำการจับภาพแบบ Frame ต่อ Frame ต่อมาก็จะแสดงภาพแบบขาวดำผ่าน Function cvtColor แล้วแสดง Frame ที่เป็นขาวดำพอได้แล้วให้ทำการใช้ faces เพื่อตรวจจับหน้าเสร็จแล้วเราจะทำการใช้ if len(faces) เพื่อตรวจหาใบหน้าเพื่อติกรอบ สีเหลี่ยมแล้วจะทำการเช็คขนาดของใบหน้าอีกทีถ้าใบหามีขนาดมากหรือเท่ากับ Size ที่เรากำหนดไว้จะทำการเข้า Loop if เพื่อทำการปรับขนาดใหม่และสร้าง Path และชื่อของ รูปภาพโดยตั้งว่า /img + str(count) เพื่อได้ไฟล์รูปและตัวเลขเรียงลำดับและตั้งนามสกุลของ ไฟล์ภาพด้วย .jpg จากนั้นจะใช้ Function cv2.imwrite เพื่อสร้างรูปมาใหม่ตามที่เรากำหนดไว้ และใช้ Function cv2.rectangle เพื่อกำหนดขนาดและสีของกรอบที่ติใบหน้าแล้วใช้ Function cv2.putText เพื่อนับจำนวนรูปภาพโดย Str(count)+str(MAX_CAPTURE) thickness = 2 คือความหนาของกรอบ แล้วจะทำการนับ Count +=1 แล้วก็กลับไปวนจนครบจำนวนที่กำหนดไว้แล้วทำการโชว์หน้าต่างของระบบโดยใช้ cv2.imshow("ชื่อหัวข้อ", frame) แล้วจะใช้คำสั่งเมื่อกด q จะทำการปิดโปรแกรม และล้างค่าที่ตกค้างโดยใช้ Capture.release() และ Function cv2.destroyAllWindows() ไว้สำหรับปิดการทำงานดังนี้

```
while count < MAX_CAPTURE:
    ret, frame = capture.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = detector.detectMultiScale(gray, 1.2, 4)
    if len(faces) > 0:
        for (x, y, w, h) in faces:
            if w >= SIZE[0]:
                face = frame[y:y+h, x:x+w]
                output_name = OUTPUT_PATH + '/' + label + '/img' + str(count) + '.jpg'
                face_cropped = face
                face_resized = cv2.resize(face_cropped, SIZE, interpolation=cv2.INTER_LINEAR)
                cv2.imwrite(output_name, face_resized)
                cv2.rectangle(frame, (x,y), (x+w,y+h), color, 2)
                cv2.putText(frame, 'all = ' + str(count) + ' / ' + str(MAX_CAPTURE), (x,y-10), font, 0.6, color, thickness=2)
                count += 1
    cv2.imshow('Face-Auto-Save', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
    capture.release()
cv2.destroyAllWindows()
```

รูปที่ 4.10 นับตัวแปร Count มาเปรียบเทียบกับ MAX_CAPTURE

4.2.10 ออกแบบระบบเทรนใบหน้าให้จดจำใบหน้าในระบบนี้ทางเราจะทำการนำรูปภาพที่บันทึกมาทำให้สามารถรู้จำใบหน้าของแต่ละคนได้ด้วยการนำรูปภาพใบหน้าของแต่ละคนมารวบรวมไว้ใน SVM_MODEL และเก็บรูปที่ทำการเทรนแล้วมาไว้ในโฟลเดอร์ Datasets > processed และทำการบันทึกข้อมูลลงในไฟล์ faces.csv และ ไฟล์ processed.csv เพื่อเก็บข้อมูลเป็นตาราง

4.2.10.1 ขั้นแรกให้ทำการ Import Library เสริมมาช่วยในการทำงาน

```
1 import cv2
2 import numpy as np
3 import pandas as pd
4 import matplotlib.pyplot as plt
5 from joblib import dump, load
6 from pathlib import Path
7
8 from sklearn.decomposition import PCA
9 from sklearn.svm import SVC
10 from sklearn.model_selection import cross_val_predict
11 from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

รูปที่ 4.11 Import Library เสริมมาช่วยในการทำงาน

4.2.10.2 ขั้นตอนนี้ทำการสร้างฟังก์ชันเพื่อระบุ Path และ Out Path เพื่อดึงและบันทึก ข้อมูลดังนี้

```
def ta():
    HAAR_MODEL = 'C:\\Users\\thara\\Desktop\\FaceReconition\\model-haar\\haarcascade_frontalface_default.xml'
    INPUT_IMAGE_PATH = 'C:\\Users\\thara\\Desktop\\FaceReconition\\datasets\\faces'
    OUTPUT_CSV_FILE = 'C:\\Users\\thara\\Desktop\\FaceReconition\\datasets\\faces.csv'
    PROCESSED_IMAGE_PATH = 'C:\\Users\\thara\\Desktop\\FaceReconition\\datasets\\processed'
    PROCESSED_CSV_FILE = 'C:\\Users\\thara\\Desktop\\FaceReconition\\datasets\\cropped.csv'
    DETECTED_FACE_PATH = 'C:\\Users\\thara\\Desktop\\FaceReconition\\datasets\\cropped'
    DETECTED_CSV_FILE = 'C:\\Users\\thara\\Desktop\\FaceReconition\\datasets\\cropped.csv'
    OUTPUT_MODEL_NAME = 'C:\\Users\\thara\\Desktop\\FaceReconition\\model-svm\\faces.lib'
```

รูปที่ 4.12 ทำการสร้างฟังก์ชันเพื่อระบุ Path และ Out Path

4.2.10.3 ทำการสร้างเส้นทางเพื่อใช้เก็บไฟล์ Csv เพื่อเก็บข้อมูลลง ตารางดังนี้

```
def create_csv(dataset_path, output_csv):
    root_dir = Path(dataset_path)
    items = root_dir.iterdir()

    filenames = []
    labels = []
    print('อ่านรูปภาพจากไฟล์ ... ')
    for item in items:
        if item.is_dir():
            for file in item.iterdir():
                if file.is_file():
                    print(str(file))
                    filenames.append(file)
                    labels.append(item.name)
    raw_data = {'filename': filenames, 'label': labels}
    df = pd.DataFrame(raw_data, columns=['filename', 'label'])
    df.to_csv(output_csv)
    print(len(filenames), 'อ่านไฟล์รูปภาพ')
    # input("Press [ENTER] key to continue...")
```

รูปที่ 4.13 สร้างเส้นทางเพื่อใช้เก็บไฟล์ Csv เพื่อเก็บข้อมูล

4.2.10.4 สร้างฟังก์ชันปรับขนาดรูปเพื่อทำการประมวลผลอีกครั้ง

```
def resize(image, width=None, height=None):
    (h, w) = image.shape[:2]
    if width is None:
        r = height/float(h)
        dim = (int(w*r), height)
    else:
        r = width/float(w)
        dim = (width, int(h*r))
    return cv2.resize(image, dim, interpolation=cv2.INTER_LINEAR)
```

รูปที่ 4.14 สร้างฟังก์ชันปรับขนาดรูป

4.2.10.5 ทำการประมวลผลภาพเพื่อเก็บลงในไฟล์ processed.csv แบบ Row & Columns และเก็บลงในโฟลเดอร์ processed อีกครั้งโดยรวมไฟล์รูปภาพที่อ่านแล้วมาไว้ในโฟลเดอร์เดียวและปรับขนาดความกว้างเท่ากับ 64 ดังนี้

```
def process_image(input_csv, output_csv, output_path_name):
    dataset = pd.read_csv(input_csv, sep=',')
    ids = dataset.values[:,0]
    names = dataset.values[:,1]
    labels = dataset.values[:,2]

    output_path = Path(output_path_name)
    if not output_path.exists():
        output_path.mkdir()

    filenames = []
    print('ประมวลผลภาพหน้า ... ')
    for item in names:
        input_path = Path(item)
        if input_path.is_file():
            output_name = output_path_name + '/image' + str(ids[len(filenames)]) + input_path.suffix
            print(input_path, '->', output_name)
            image = cv2.imread(str(input_path))
            image = resize(image, width=64)
            cv2.imwrite(output_name, image)
            filenames.append(output_name)
    prc_data = {'filename': filenames, 'label': labels}
    df = pd.DataFrame(prc_data, columns=['filename', 'label'])
    df.to_csv(output_csv)
    print(len(filenames), 'ประมวลผลไฟล์ภาพแล้ว')
    # input("Press [ENTER] key to continue...")
```

รูปที่ 4.15 ทำการประมวลผลภาพเพื่อเก็บลงในไฟล์ processed.csv

4.2.10.6 ต่อมาจะเป็นการสร้างฟังก์ชันตรวจจับใบหน้าจากรูปภาพที่เราบันทึกมา โดยอ่านรูปภาพแล้วจะทำการบันทึกข้อมูลลงในไฟล์ faces.csv แบบ Row & Columns อีกครั้งดังนี้

```
def detect_face(input_csv, output_csv, output_path_name):
    dataset = pd.read_csv(input_csv, sep=',')
    ids = dataset.values[:,0]
    names = dataset.values[:,1]
    labels = dataset.values[:,2]

    output_path = Path(output_path_name)
    if not output_path.exists():
        output_path.mkdir()

    clf = cv2.CascadeClassifier(HAAR_MODEL)
    face_filenames = []
    face_labels = []
    count = 0
    face_count = 0
    print('ตรวจจับใบหน้า ... ')
    for item in names:
        image = cv2.imread(item)
        face_label = labels[count]

        gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        faces = clf.detectMultiScale(gray, 1.3, 5)
        for (x, y, w, h) in faces:
            cropped = image[y:y+h, x:x+w]
            output_file = output_path_name + '/face' + str(len(face_filenames)) + '.jpg'
            cv2.imwrite(output_file, cropped)

            face_filenames.append(output_file)
            face_labels.append(face_label)
        print(item, '->', len(faces), ' face(s) detected')
        face_count += len(faces)
        count += 1
    crp_data = {'filename': face_filenames, 'label': face_labels}
    df = pd.DataFrame(crp_data, columns=['filename', 'label'])
    df.to_csv(output_csv)
    print('Total of', face_count, 'face(s) detected')
    # input("Press [ENTER] key to continue...")
```

รูปที่ 4.16 สร้างฟังก์ชันตรวจจับใบหน้าจากรูปภาพ

4.2.10.7 ต่อไปจะเป็นระบบทำให้โมเดลจดจำใบหน้าและเก็บไฟล์รูปรวมใน SVM_MODEL ที่ชื่อว่า faces.lib ในขั้นตอนนี้อีกเราเทรนใบหน้าบ่อย ๆ ระบบจะ สามารถจำได้ ยิ่งขึ้นเป็นการเรียนรู้ให้ระบบรู้จักใบหน้ามากยิ่งขึ้น

```
def train_model(train_csv, output_model_name):
    dataset = pd.read_csv(train_csv, sep=',')
    ids = dataset.values[:,0]
    names = dataset.values[:,1]
    labels = dataset.values[:,2]

    images = []
    print('อบรมโมเดลให้รู้จักใบหน้า ...')
    for item in names:
        image = cv2.imread(str(item), 0)
        resized = cv2.resize(image, (64,64), interpolation=cv2.INTER_LINEAR)
        images.append(np.ravel(resized))

    clf = SVC(kernel='linear', probability=True)
    clf.fit(images, labels)
    dump(clf, output_model_name)
    print('สร้างโมเดลใน...', output_model_name)
    # input("Press [ENTER] key to continue...")
```

รูปที่ 4.17 โมเดลจดจำใบหน้าและเก็บไฟล์รูปรวมใน SVM_MODEL ที่ชื่อว่า faces.lib

4.2.10.7 ขั้นตอนสุดท้ายคือขั้นตอนการรายงานความแม่นยำของรูปภาพที่ทำการเทรนมาแล้วโดยสรุปเป็น Report เป็นคะแนนยิ่งคะแนนมากจะจับใบหน้าได้ถูกต้องมากขึ้น ดังนี้

```
def validate_model(validate_csv, model_name):
    dataset = pd.read_csv(validate_csv, sep=',')
    ids = dataset.values[:,0]
    names = dataset.values[:,1]
    labels = dataset.values[:,2]

    images = []
    print('ตรวจสอบรูปแบบการรู้จักใบหน้า ...')
    for item in names:
        image = cv2.imread(str(item), 0)
        resized = cv2.resize(image, (64,64), interpolation=cv2.INTER_LINEAR)
        images.append(np.ravel(resized))

    clf = load(model_name)
    y_p = cross_val_predict(clf, images, labels, cv=3)
    print('คะแนนความละเอียด:', '{0:.4g}'.format(accuracy_score(labels,y_p) * 100), '%')
    print('ความเข้ากันของเมทริกซ์:')
    print(confusion_matrix(labels,y_p))
    print('รายงานการจำแนกประเภท:')
    print(classification_report(labels,y_p))

create_csv(INPUT_IMAGE_PATH, OUTPUT_CSV_FILE)
process_image(OUTPUT_CSV_FILE, PROCESSED_CSV_FILE, PROCESSED_IMAGE_PATH)
train_model(PROCESSED_CSV_FILE, OUTPUT_MODEL_NAME)
validate_model(PROCESSED_CSV_FILE, OUTPUT_MODEL_NAME)
```

รูปที่ 4.18 การเขียนโค้ดรายงานความแม่นยำของรูปภาพ

```

570 ประมาณผลให้ภาพแล้ว
ลบโมเดลผู้ใช้จำนอง...
สร้างโมเดลใน... C:\Users\thara\Desktop\FaceReconition\model-svm\faces.lib
ตรวจสอบรูปแบบการรู้จำโมเดล...
คะแนนความละเอียด: 99.82 %
ความเข้ากันได้ของเมทริกซ์:
[[ 91  0  0  0  0  0]
 [  0 79  0  0  0  0]
 [  1  0 99  0  0  0]
 [  0  0  0 100  0  0]
 [  0  0  0  0 100  0]
 [  0  0  0  0  0 100]]
รายงานการจำแนกประเภท:

```

	precision	recall	f1-score	support
Aon	0.99	1.00	0.99	91
First	1.00	1.00	1.00	79
Home	1.00	0.99	0.99	100
Koy	1.00	1.00	1.00	100
Nick	1.00	1.00	1.00	100
Pu	1.00	1.00	1.00	100
accuracy			1.00	570
macro avg	1.00	1.00	1.00	570
weighted avg	1.00	1.00	1.00	570

รูปที่ 4.19 Report เป็นคะแนนยิ่งคะแนนมากจะจับใบหน้าได้ถูกต้องมากขึ้น

4.2.11 ออกแบบระบบตรวจจับอาการหลับง่วงนอนหรือหลับในทำการสร้างฟังก์ชันเพื่อเก็บรายละเอียดของดวงตาเพื่อนำมาเปรียบเทียบ

```

def eye_aspect_ratio(eye):
    A = distance.euclidean(eye[1], eye[5])
    B = distance.euclidean(eye[2], eye[4])
    C = distance.euclidean(eye[0], eye[3])
    ear = (A + B) / (2.0 * C)
    return ear

```

รูปที่ 4.20 ระบบตรวจจับอาการหลับง่วงนอนหรือหลับใน

4.2.12 ทำการเรียกใช้งานตัวตรวจจับดวงตาและตัววาดจุดสำคัญของดวงตาและกำหนดเวลาที่ใช้ในการหลับตาพร้อมกำหนดค่าของดวงตา ดังนี้

```
thresh = 0.25
frame_check = 10
detect = dlib.get_frontal_face_detector()
predict = dlib.shape_predictor("D:\FaceReconition\model-haar\shape_predictor_68_face_landmarks.dat")
```

รูปที่ 4.21 ทำการเรียกใช้งานตัวตรวจจับดวงตา

4.2.13 ทำการเข้าสู่ While เพื่อกำหนดขนาดของหน้าต่างและอ่านค่าของดวงตาพอได้ แล้วจะเข้าสู่ For เพื่อทำการแปลงค่าดวงตาให้อ่านค่าที่ละเอียดเฟรมและจะเข้าสู่ If ก็ต่อเมื่อดวงตาหลับนานกว่าที่เรากำหนดคือ 10 วิแล้วจะทำการส่งเสียงแจ้งเตือนเพื่อให้เรา รู้สึกตัวแต่ถ้าไม่เข้าสู่ If ก็จะทำให้การนับค่าใหม่ไปเรื่อย ๆ ถ้าจะออกโปรแกรมออกได้โดย การกด Q ถ้ากดตัวอื่นโปรแกรมจะไม่ออกและกลับมาทำงานเหมือนเดิม

```
while True:
    ret, frame=cap.read()
    frame = imutils.resize(frame, width=720)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    subjects = detect(gray, 0)
    for subject in subjects:
        shape = predict(gray, subject)
        shape = face_utils.shape_to_np(shape)#converting to Numpy Array
        leftEye = shape[15:17]
        rightEye = shape[22:24]
        leftEAR = eye_aspect_ratio(leftEye)
        rightEAR = eye_aspect_ratio(rightEye)
        ear = (leftEAR + rightEAR) / 2.0
        leftEyeHull = cv2.convexHull(leftEye)
        rightEyeHull = cv2.convexHull(rightEye)
        cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)
        cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)
        if ear < thresh:
            flag += 1
            print (flag)
            if flag >= frame_check:
                cv2.putText(frame, playsound('D:\\FaceReconition\\Sound\\nasty.mp3'), (10, 30),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
            else:
                flag = 0
        cv2.imshow("Drowsiness-Detection", frame)
        key = cv2.waitKey(1) & 0xFF
        if key ==ord("q"):
            break
    cv2.destroyAllWindows()
    cap.release()
```

รูปที่ 4.22 กำหนดขนาดของหน้าต่างและอ่านค่าของดวงตา

4.3 ขั้นตอนการทดสอบระบบจดจำใบหน้าและระบุตัวตนด้วยกล้อง

จากการทดสอบระบบจดจำใบหน้าและระบุตัวตนด้วยกล้อง ข้อกังวลข้อแรกของการใช้เทคโนโลยีดังกล่าวคือปัญหาด้านความแม่นยำถูกต้อง ระบบเหล่านี้พัฒนาขึ้นจากฐานข้อมูลรูปภาพไม่ต่างจากคนเรา ที่มักจะไม่สามารถแยกแยะหน้าตาของกลุ่มคนต่างเชื้อชาติที่เราไม่คุ้นเคยได้ ถ้าฐานข้อมูลที่ใช้ในการฝึกสอนไม่ครอบคลุมหรือมีกลุ่มประชากรบางประเภทน้อย ก็อาจทำให้ระบบขาดความแม่นยำได้ สถานะดาบสองคมของเทคโนโลยีจดจำหน้านั้นไม่ต่างจากเทคโนโลยีอื่น ๆ ในมุมหนึ่งเราได้เห็นศักยภาพทั้งด้านธุรกิจ ด้านความปลอดภัย และด้านการบังคับใช้กฎหมาย แต่ในอีกด้านหนึ่ง เทคโนโลยีการจดจำหน้านั้นสุ่มเสี่ยงต่อการก่อให้เกิดการละเมิดความเป็นส่วนตัวในระดับที่ไม่เคยมีมาก่อน เมื่อมีปัจจัยเร่งด่วนด้านความมั่นคงปลอดภัยของสังคม ความเป็นส่วนตัวจะเป็นสิ่งแรกที่โดนสละทิ้งไปก่อน ดังนั้นเราไม่ควรจะรอให้เกิดสภาวะดังกล่าว แล้วจึงค่อยหันมาสนใจประเด็นเกี่ยวข้องกับเทคโนโลยีใกล้ตัวนี้

บทที่ 5

สรุปผลและข้อเสนอแนะการดำเนินโครงการ

ในการจัดทำโครงการดำเนินการสร้างระบบจดจำใบหน้าและระบุตัวตนด้วยกล้อง (Deep Learning Face Recognition by Camera) นี้สามารถบรรลุวัตถุประสงค์ตามที่ผู้จัดทำได้วางแผนไว้ ซึ่งระบบจดจำใบหน้าและระบุตัวตนด้วยกล้องสามารถทำงานได้ตามที่ต้องการ และการประสบปัญหาต่าง ๆ ทางคณะผู้จัดทำได้ปรับปรุงแก้ไขในส่วนต่าง ๆ เพื่อให้ระบบจดจำใบหน้าและระบุตัวตนด้วยกล้องเกิดความผิดพลาดได้น้อยที่สุด สามารถสรุปผลการดำเนินโครงการและข้อเสนอแนะดังนี้

- 5.1 วัตถุประสงค์โครงการ
- 5.2 ประโยชน์ที่คาดว่าจะได้รับ
- 5.3 ปัญหาที่ประสบในการดำเนินโครงการ
- 5.4 ผลการดำเนินโครงการ
- 5.5 อภิปรายผล
- 5.6 ข้อเสนอแนะ

5.1 วัตถุประสงค์โครงการ

- 5.1.1 เพื่อนำความรู้ในสาขาวิชาเทคโนโลยีสารสนเทศในการจัดทำโครงการ
- 5.1.2 เพื่อศึกษาระบบการจดจำใบหน้า (Face Recognition) เพื่อนำไปต่อยอด
- 5.1.3 เพื่อสร้างความปลอดภัยให้กับสถานที่ เพิ่มความปลอดภัยในชีวิตและทรัพย์สิน

5.2 ประโยชน์ที่คาดว่าจะได้รับ

- 5.2.1 เพื่อนำความรู้ในสาขาวิชาเทคโนโลยีสารสนเทศในการจัดทำโครงการ
- 5.2.2 เพื่อศึกษาระบบการจดจำใบหน้า (Face Recognition) เพื่อนำไปต่อยอด
- 5.2.4 เพื่อสร้างความปลอดภัยให้กับสถานที่ เพิ่มความปลอดภัยในชีวิตและทรัพย์สิน

5.3 ปัญหาที่ประสบในการดำเนินโครงการ

การดำเนินการจัดทำโครงงานนั้นผู้จัดทำได้ประสบปัญหาการดำเนินโครงการหลายอย่าง ทางคณะผู้จัดทำจะอธิบายสาเหตุและวิธีการแก้ปัญหาเป็นข้อๆ ดังนี้

5.3.1 ตรวจจับใบหน้าผิดพลาดเกิดจากแสงในห้อง

5.3.2 การแจ้งเตือนด้วยเสียงมีการกระตุก

5.3.3 แสดงข้อผิดพลาดในการตรวจจับ

5.4 ผลการดำเนินโครงการ

ในการดำเนินโครงการจัดทำระบบจดจำใบหน้าและระบุตัวตนด้วยกล้อง เริ่มจากการนำเสนอโครงการต่อคณะกรรมการพิจารณาทางคณะกรรมการได้เสนอแนะส่วนต่าง ๆ และคณะผู้จัดทำได้ทำตามข้อเสนอแนะของคณะกรรมการ จนได้รับอนุมัติการทำโครงการแล้ว ทางคณะผู้จัดทำได้ได้ศึกษาข้อมูลรายละเอียดต่าง ๆ เกี่ยวกับการ จัดทำระบบจดจำใบหน้าและระบุตัวตนด้วยกล้อง โดยได้ทำการออกแบบและดำเนินการ จัดทำตามที่วางแผนไว้จนสำเร็จ

ผลการดำเนินโครงการระบบจดจำใบหน้าและระบุตัวตนด้วยกล้อง ได้ทำการตรวจสอบการใช้งานและตรวจสอบแล้วได้ผลว่าตัวระบบจดจำใบหน้า นั้นทำงานได้ปกติตามที่ผู้จัดทำคาดหวังไว้

5.5 อภิปรายผล

ในการทดสอบระบบจดจำใบหน้าและระบุตัวตนด้วยกล้อง ทางผู้จัดทำได้ทำการทดสอบการทำงานของระบบแล้วว่าทำงานได้อย่างปกติ

5.6 ข้อเสนอแนะ

กล่าวถึงข้อเสนอแนะในการพัฒนาระบบจดจำใบหน้าและระบุตัวตนด้วยกล้อง

5.6.1 ควรทำการเพิ่มเสียงในการตรวจจับใบหน้าเพื่อเป็นการแจ้งเตือนเมื่อพบใบหน้า

5.6.2 การนำระบบมาพัฒนาต่อยอดโดยการตรวจจับดวงตาป้องกันการอากรง่วงนอน

5.6.3 ควรมีการจัดทำโครงการให้หลากหลายให้ครบต่อกลุ่มสาระการเรียนรู้

5.6.4 ทำการรีด UI ให้ง่ายต่อการใช้งาน

บรรณานุกรม

- Atthiwat wanmai (2557). **Deep learning**. ค้นข้อมูล 4 ตุลาคม 2562,
จาก <https://medium.com/@athivvat/deep-learning>
- Kong Ruksiamza (2561). **ตรวจจับวัตถุในรูปภาพ**. ค้นข้อมูล 10 พฤษภาคม 2562,
จาก <https://medium.com/@kongruksiamza>
- Mikki Pastel (2560). **Python GUI ด้วย Tkinter**. ค้นข้อมูล 20 พฤษภาคม 2562,
จาก <https://mikipastel.blogspot.com/>
- Pattapol Srirak (2561). **มาลองเล่น Face Recognition**. ค้นข้อมูล 25 พฤษภาคม 2562,
จาก <https://medium.com/@P.Srirak>
- Somkiat yodtipan (2557). **การวิเคราะห์ ด้วย Pandas**. ค้นข้อมูล 27 พฤษภาคม 2562,
จาก <http://www.somkiat.cc/>
- Vithan Minaphinant (2559). **Machine learning**. ค้นข้อมูล 1 ธันวาคม 2562,
จาก <https://blog.finnomena.com/machine-learning>
- Wannapong panyaporn (2558). **เริ่มต้น Opencv 3 กับ Python**. ค้นข้อมูล 3 ธันวาคม 2562,
จาก <https://python3.wannaphong.com/>
- Wannapong panyaporn (2558). **ฟังก์ชันคำสั่งไลบรารี CSV**. ค้นข้อมูล 4 ธันวาคม 2562,
จาก <https://python3.wannaphong.com/>
- มินโฮ คานาซะ (2562). **เทคโนโลยีจดจำใบหน้า**. ค้นข้อมูล 5 ธันวาคม 2562,
จาก <https://www.patanasongsivilai.com/>
- วิทยาลัยเซาร็อัสบางกอก (2561). **คู่มือการใช้งาน Opencv**. ค้นข้อมูล 10 ธันวาคม 2562,
จาก <http://www.theerapone.com/sbc/courses/openCV>

ภาคผนวก

ภาคผนวก ก แบบเสนอร่างโครงการ

ภาคผนวก ข แบบประเมินความก้าวหน้า

ภาคผนวก ค คู่มือการติดตั้งและใช้งานเว็บแอปพลิเคชัน คลินิกการ

ภาคผนวก ง ประวัติผู้จัดทำ

ภาคผนวก ก

แบบเสนอร่างโครงการ



สาขาวิชาเทคโนโลยีสารสนเทศ
แบบเสนอร่างโครงการ

เรื่อง

ระบบจดจำใบหน้าและระบุตัวตนด้วยกล้อง

DEEP LEARNING FACE RECOGNITION BY CAMERA

โดย

นายศิรินทร์	งามดี	รหัสประจำตัว 40952
นายธรา	รอบรู้	รหัสประจำตัว 41148

ภาคเรียนที่ 1/2562

สาขาวิชาเทคโนโลยีสารสนเทศ
วิทยาลัยเทคโนโลยีอรรถวิทย์พัฒนวิชาการ

แบบเสนอร่างโครงการ
สาขาวิชาเทคโนโลยีสารสนเทศ

ชื่อโครงการ ระบบจดจำใบหน้าและระบุตัวตนด้วยกล้อง

Deep Learning Face Recognition by Camera

ชื่อผู้เสนอโครงการ 1. นายศิขรินทร์ งามดี รหัสประจำตัว 40952 (หัวหน้ากลุ่มโครงการ)

ชื่อผู้ร่วมโครงการ 2. นายธรา รอบรู้ รหัสประจำตัว 41148

นักศึกษาสาขาวิชาเทคโนโลยีสารสนเทศ รอบ ป้าย*

ชื่ออาจารย์ที่ปรึกษาโครงการ อาจารย์คุณานนท์ สุขเกษม

มีความประสงค์ขออนุมัติหัวข้อโครงการ เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ประกาศนียบัตรวิชาชีพชั้นสูง ในวิชาโครงการ จำนวน 4 หน่วยกิต

จึงเรียนมาเพื่อโปรดพิจารณา ดังรายละเอียดโครงการที่แนบมาด้วย

ลงชื่อ นายศิขรินทร์ งามดี (หัวหน้ากลุ่มโครงการ)

ลงชื่อ นายธรา รอบรู้ (สมาชิกกลุ่มโครงการ)

..... / /

ความเห็นอาจารย์ที่ปรึกษาโครงการ	ความเห็นผู้รับผิดชอบโครงการ สาขาวิชา
.....	
.....	
.....	
ลงนาม	ลงนาม
..... / /	 / /
ลงนาม	
..... / /	

หมายเหตุ พร้อมแนบโครงการ ตามแบบฟอร์มที่กำหนดมาด้วยแล้ว

1. ชื่อโครงการ

ระบบจดจำใบหน้าและระบุตัวตนด้วยกล้อง (Deep Learning Face Recognition by Camera)

2. ความเป็นมาและความสำคัญของปัญหา

ระบบรู้จำใบหน้า (Face Recognition) ถูกออกแบบมาให้ทำการเปรียบเทียบใบหน้าบุคคลกับฐานข้อมูลใบหน้าที่มีอยู่โดยอัลกอริทึมที่ใช้ในขั้นตอนการสร้างแม่แบบและขั้นตอนการเปรียบเทียบอาจแตกต่างกันไปแล้วแต่การออกแบบระบบของแต่ละระบบ แต่ไม่ว่าจะมี อัลกอริทึมในการทำงานในขั้นตอนการสร้างแม่แบบและขั้นตอนการเปรียบเทียบอย่างไร แต่ขั้นตอนการทำงานโดยรวมของระบบก็ยังคงเหมือนกันอยู่ การตรวจหาใช้วิธีตรวจหาตำแหน่งของ จมูก และ ตาทั้งสองข้าง ในรูปของเมตริกซ์ แนวตั้งและแนวนอน

โดยทั่วไประบบรู้จำใบหน้าจะประกอบไปด้วย 2 ขั้นตอนหลักคือ การตรวจจับใบหน้า (Face Detection) และ ระบบรู้จำใบหน้า (Face Recognition) 1.1 การตรวจจับใบหน้า คือกระบวนการค้นหาใบหน้าของบุคคลจากภาพหรือวิดีโอหลังจากนั้นก็ทำการประมวลผลภาพใบหน้าที่ได้สำหรับขั้นตอนถัดไปเพื่อให้ภาพใบหน้าที่ตรวจจับได้ง่ายต่อการจำแนก และ อัลกอริทึมที่ใช้ในการตรวจจับใบหน้าในปัจจุบันก็มีอยู่ด้วยกันหลายวิธีซึ่งอัลกอริทึมในการตรวจจับใบหน้าที่ดีนั้นมีส่วนช่วยในการจำแนก 1.2 ระบบรู้จำใบหน้า (Face Recognition) คือกระบวนการที่ได้นำภาพใบหน้าที่ตรวจจับได้และประมวลผลแล้วจากขั้นตอนการตรวจจับใบหน้า มาเปรียบเทียบกับฐานข้อมูลของใบหน้าเพื่อระบุว่าใบหน้าที่ตรวจจับได้ตรงกับบุคคลใด เมื่อไม่นานมานี้มีนักพัฒนาได้พัฒนา Library บนภาษา Python ที่ช่วยทำให้ Face Recognition กลายเป็นเรื่องง่ายใช้เพียงแค่โค้ดไม่กี่บรรทัด เพียงแค่ผู้เริ่มต้นเขียนโปรแกรม ก็สามารถทำ Face Recognition ได้อย่างง่ายดาย

ดังนั้นทางผู้จัดทำเล็งเห็นว่าในอนาคต อุปกรณ์หลายอย่างที่เราใช้ในชีวิตประจำวันรวมทั้ง ทุยแจบ้าน ทุยแจรถ บัตรเอทีเอ็ม หรือแม้กระทั่งรหัสผ่านต่าง ๆ จะค่อยๆ หายไปโดยสิ่งที่มาแทนก็คือเทคโนโลยีรู้จำใบหน้า(Face recognition) และเทคโนโลยีจดจำใบหน้านั้นส่วนมากจะใช้ในการรักษาความปลอดภัยแต่เราสามารถประยุกต์ไปใช้ในงานสารสนเทศด้านอื่นๆ ได้เราจึงคิดว่าการจดจำใบหน้าที่สามารถระบุชื่อว่าคุณคนนั้นเป็นใครเราจะสามารถแยกแยะบุคคลได้อีกทั้งยังสามารถนำไปใช้เช็คชื่อนักเรียนในห้องเรียนได้, ใช้ในร้านอาหารแยกใบหน้าพนักงานกับลูกค้าได้เพื่อตรวจสอบพฤติกรรม การ เทคโนโลยีการจดจำใบหน้า นอกจากจะนำมาใช้ในการรักษาความปลอดภัยแล้ว ยังสามารถนำมาใช้งานได้อีกหลากหลายกรณี เช่น ออฟฟิศ ห้างสรรพสินค้า อาคารจอดรถ การมาทำงานของพนักงาน หน่วยงาน องค์กรขนาดใหญ่ก็ยังสามารถที่จะนำมาใช้งานในด้านการวิเคราะห์การตลาดได้อีกหลายรูปแบบ อาทิ พฤติกรรมลูกค้าที่เข้ามาใช้บริการภายในร้าน

3. วัตถุประสงค์ของโครงการ

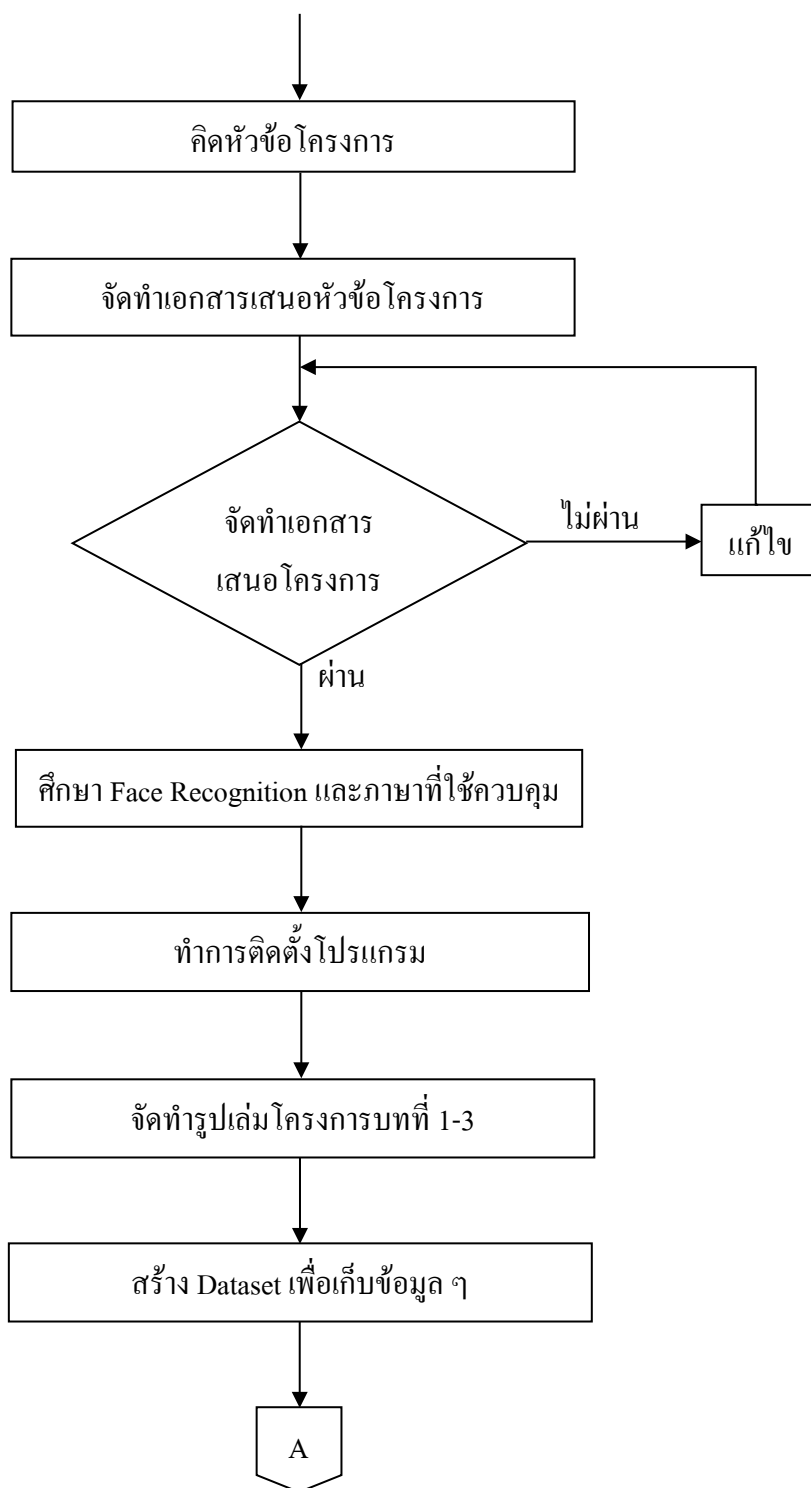
- 3.1 เพื่อนำความรู้ในสาขาวิชาเทคโนโลยีสารสนเทศในการจัดทำโครงการ
- 3.2 เพื่อศึกษาระบบการจดจำใบหน้า (Face Recognition) เพื่อนำไปต่อยอด
- 3.3 เพื่อสร้างความปลอดภัยให้กับสถานที่ เพิ่มความปลอดภัยในชีวิตและทรัพย์สิน

4. ขอบเขตของโครงการ

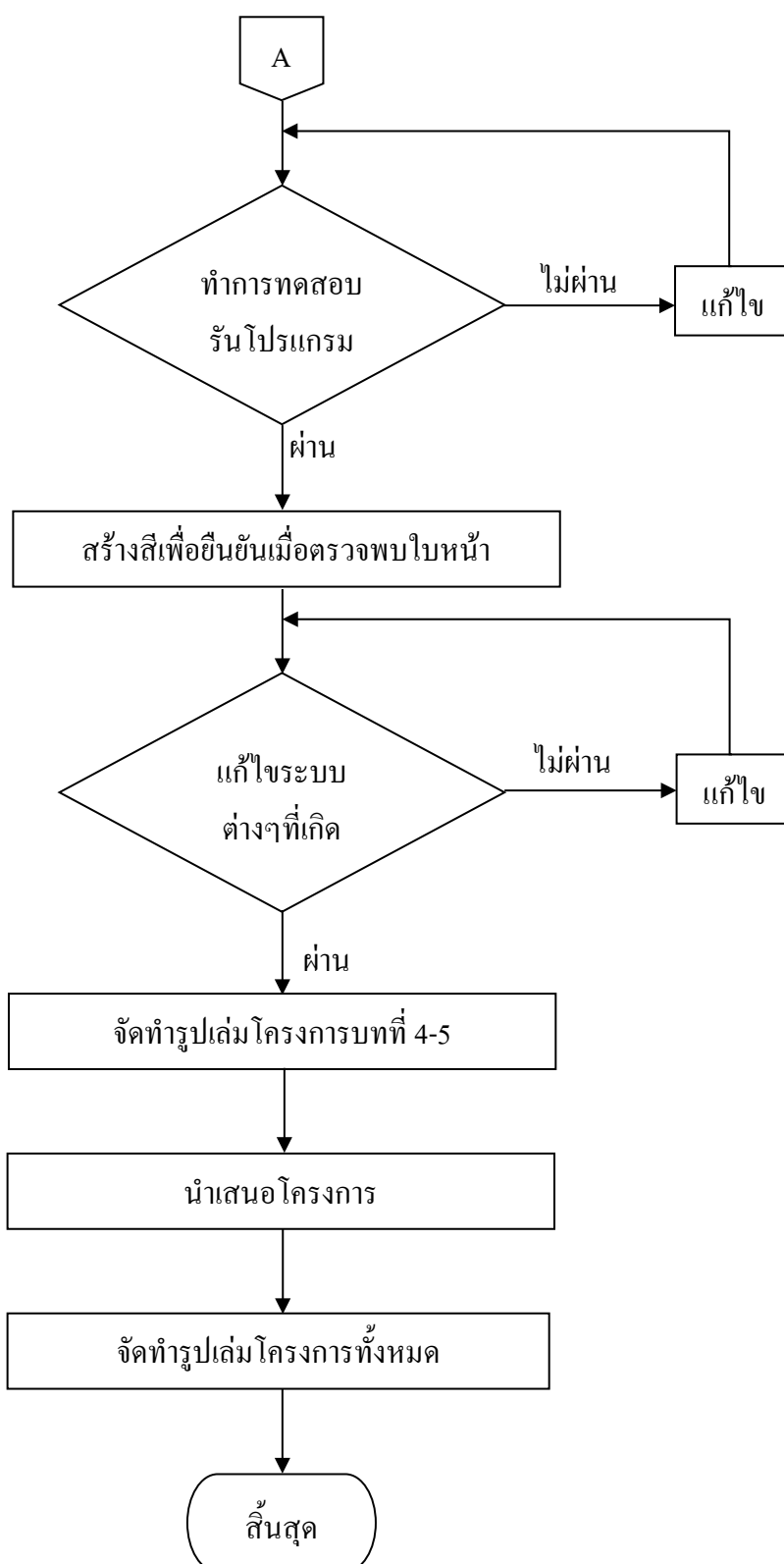
- 4.1 สามารถตรวจจับวัตถุใบหน้าคนและแสดงชื่อได้ร้อยละ 80 ขึ้นไป
- 4.2 สามารถมีข้อความแสดงผลเมื่อพบวัตถุ
- 4.3 ใช้ภาษา Python ในการเขียนโปรแกรม
- 4.4 ใช้กล้อง Webcam จำนวน 1 ตัวในการตรวจจับ
- 4.5 ระยะในการตรวจจับใบหน้า 1.5 เมตรโดยประมาณ
- 4.6 ขนาดของรูปในการตรวจจับใบหน้า 64 x 64 พิกเซล
- 4.7 ความเร็วในการตรวจจับใบหน้า 30 FPS ต่อ 1 วินาที
- 4.8 พื้นหลังในการตรวจจับต้องเป็นสีขาวแสงไม่มากจนเกินไป

5. ขั้นตอนการดำเนินงาน

- 5.1 คิดหัวข้อโครงการ
- 5.2 คิดหัวข้อโครงการ
- 5.3 เสนอโครงการ
- 5.4 แก้ไขเสนอร่างโครงการ
- 5.5 ศึกษาเทคโนโลยี Face Recognition และภาษาที่ใช้ควบคุม
- 5.6 ทำการติดตั้งโปรแกรม vs code , Python 3.6.8 และ Library ที่ใช้
- 5.7 จัดทำรูปเล่มโครงการบทที่ 1-3
- 5.8 สร้าง Dataset เพื่อเก็บข้อมูล
- 5.9 ทำการทดสอบรันโปรแกรม
- 5.10 สร้างเสียงเพื่อยืนยันเมื่อตรวจพบใบหน้าตรงตาม Dataset
- 5.11 ปรับปรุงแก้ไขระบบต่างๆที่เกิดข้อผิดพลาด
- 5.12 จัดทำรูปเล่มโครงการบทที่ 4-5
- 5.13 นำเสนอโครงการ
- 5.14 จัดทำรูปเล่มโครงการทั้งหมด



รูปที่ 1.1 แผนภาพ Flowchart ขั้นตอนการดำเนินงาน



รูปที่ 1.1 แผนภาพ Flowchart ขั้นตอนการดำเนินงาน (ต่อ)

6. ระยะเวลาการทำโครงการ

ตารางการดำเนินงานโครงการนี้ใช้ระยะเวลาในการทำ ตั้งแต่เดือน มิถุนายน พ.ศ 2562 ถึงเดือน กุมภาพันธ์ พ.ศ 2563 ดังตารางที่ 1.1

ตารางที่ 1.1 ตารางแผนการดำเนินงาน

ลำดับ	ขั้นตอนการดำเนินงาน	ระยะเวลาการดำเนินงาน								
		พ.ศ.2562							พ.ศ.2563	
		มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.
1	คิดหัวข้อโครงการ	↔								
2	จัดทำรูปเล่มเสนอโครงการ	↔								
3	เสนอร่างโครงการ	↔								
4	แก้ไขเสนอร่างโครงการ	↔→								
5	ศึกษาเทคโนโลยีFace Recognition	↔→								
6	จัดทำโครงการ ติดตั้งโปรแกรม	←→	→							
7	จัดทำรูปเล่มบทที่ 1-3	←→	→							
8	สร้าง Dataset ข้อมูล	←→	→							
9	ทดสอบรัน โปรแกรม	←→	→							
10	สร้างสีเพื่อยืนยันเมื่อตรวจพบใบหน้า	←→	→							
11	แก้ไขระบบต่างๆที่เกิดข้อผิดพลาด	←→	→							
12	จัดทำรูปเล่มบทที่ 4-5			←→	→					
13	นำเสนอโครงการ			←→	→				→	
14	จัดทำรูปเล่มโครงการ					←→	→			→

7. ประโยชน์ที่คาดว่าจะได้รับ

- 7.1 เพื่อนำความรู้ในสาขาวิชาเทคโนโลยีสารสนเทศในการจัดทำโครงการ
- 7.2 เพื่อศึกษาระบบการจดจำใบหน้า (Face Recognition) เพื่อนำไปต่อยอด
- 7.3 เพื่อสร้างความปลอดภัยให้กับสถานที่ เพิ่มความปลอดภัยในชีวิตและทรัพย์สิน

8. งบประมาณที่ใช้ในการทำโครงการ

8.1 กล้อง (Webcam)	200	บาท
8.2 กระดาษ	400	บาท
8.3 ค่าแผ่น CD	100	บาท
8.4 ค่าทำเล่มเอกสาร โครงการ	200	บาท
รวม	<u>900</u>	บาท

9. เอกสารอ้างอิง

https://bottomlineis.co/Tech_Face_Recognition

<https://www.bangkokbiznews.com/blog/detail/646837>

ภาคผนวก ข

แบบประเมินความก้าวหน้า



แบบประเมินความก้าวหน้าโครงการ

ระบบจดจำใบหน้าและระบุตัวตนด้วยกล้อง
DEEP LEARNING FACE RECOGNITION BY CAMERA

ชื่อผู้จัดทำ

นายศิรินทร์	งามดี	รหัสประจำตัว 40952
นายธรา	รอบรู้	รหัสประจำตัว 41148

ภาคเรียนที่ 1/2562

สาขาวิชาเทคโนโลยีสารสนเทศ

วิทยาลัยเทคโนโลยีอรรถวิทย์พัฒนวิชาการ

แบบฟอร์มประเมินความก้าวหน้าโครงการ

ชื่อโครงการ	ระบบจดจำใบหน้าและระบุตัวตนด้วยกล้อง
ชื่อโครงการ	Deep Learning Face Recognition by Camera
ปีการศึกษา	2562

ชื่อผู้จัดทำโครงการ	(1) นายศิขรินทร์ งามดี	รหัสประจำตัว	409520	ชั้น	ปวส.2
	(2) นายธारा รอบรู้	รหัสประจำตัว	41148	ชั้น	ปวส.2

ชื่ออาจารย์ที่ปรึกษาโครงการ อาจารย์คุณานนท์ สุขเกษม

โครงการนี้จัดอยู่ในกลุ่มของ

- | | |
|---|--|
| ☐ Web Programming | ☐ Computer Multimedia |
| ☐ Computer Programming | ☐ Database System |
| ☐ Hardware Computer | ☐ |

ขอบเขตของโครงการทั้งหมด

- 1.1 สามารถตรวจจับวัตถุใบหน้าคนและแสดงชื่อได้ร้อยละ 80 ขึ้นไป
- 1.2 สามารถมีข้อความแสดงผลเมื่อพบวัตถุ
- 1.3 ใช้ภาษา Python ในการเขียนโปรแกรม
- 1.4 ใช้กล้อง Webcam จำนวน 1 ตัวในการตรวจจับ
- 1.5 ระยะในการตรวจจับใบหน้า 1.5 เมตรโดยประมาณ
- 1.6 ขนาดของรูปในการตรวจจับใบหน้า 64 x 64 พิกเซล
- 1.7 ความเร็วในการตรวจจับใบหน้า 30 FPS ต่อ 1 วินาที
- 1.8 พื้นที่หลังในการตรวจจับต้องเป็นสีขาวแสงไม่มากจนเกินไป

รายงานผลความก้าวหน้าของโครงการ 25%

รายละเอียดของหัวข้อที่รายงานผลความก้าวหน้า	ผ่าน	ไม่ผ่าน	เหตุผล (กรณีไม่ผ่าน)
1. รวบรวมข้อมูล			
2. ศึกษาข้อมูล - Face Recognition - Python - OpenCV			
3. เขียนโปรแกรม Face Recognition			
4. โครงการ บทที่ 1			
5. โครงการ บทที่ 2 (บางส่วน)			

การประเมินผลความก้าวหน้าของโครงการโดยอาจารย์ที่ปรึกษา

☐ ผ่าน

☐ ไม่ผ่าน

ข้อชี้แจงของอาจารย์ที่ปรึกษา (กรณีที่อาจารย์ที่ปรึกษาประเมินไม่ผ่าน)

.....
.....

ลายเซ็น

(อาจารย์คุณานนท์ สุขเกษม)

อาจารย์ที่ปรึกษา

วันที่ / / 25....

การประเมินผลความก้าวหน้าของโครงการโดยกรรมการ☐ ผ่าน☐ ไม่ผ่าน

ความคิดเห็นของกรรมการ (กรณีที่กรรมการประเมินไม่ผ่าน)

.....

.....

ลายเซ็น

()

กรรมการตรวจสอบ

วันที่ / / 25....

รายงานผลความก้าวหน้าของโครงการ 50%

รายละเอียดของหัวข้อที่รายงานผลความก้าวหน้า	ผ่าน	ไม่ผ่าน	เหตุผล (กรณีไม่ผ่าน)
ส่วนผู้ดูแลระบบ			
6. หน้า Login			
7. หน้าเพิ่ม ลบ แก้ไขข้อมูลผู้ใช้งาน			
8. หน้าตรวจสอบการใช้งานของผู้ใช้งาน			
ส่วนเจ้าหน้าที่			
9. หน้าเพิ่ม แก้ไขข้อมูลผู้ใช้งาน			
10. หน้าเพิ่ม แก้ไข ข้อมูลการเบิก - จ่าย			
11. หน้าตรวจสอบข้อมูลวงเงินคงเหลือ			
12. สถิติการเบิก -จ่ายค่ารักษาพยาบาลของผู้เบิก - กราฟสถิติเปรียบเทียบการเบิก - จ่ายของ แต่ละภาควิชา			
13. จัดทำเอกสารโครงการบทที่ 2 ในส่วน ที่เหลือ			
14. จัดทำเอกสารโครงการบทที่ 3			

การประเมินผลความก้าวหน้าของโครงการโดยอาจารย์ที่ปรึกษา

☐ ผ่าน

☐ ไม่ผ่าน

ข้อชี้แจงของอาจารย์ที่ปรึกษา (กรณีที่อาจารย์ที่ปรึกษาประเมินไม่ผ่าน)

.....

ลายเซ็น

(อาจารย์คุณานนท์ สุขเกษม)

อาจารย์ที่ปรึกษา

วันที่ / / 25....

การประเมินผลความก้าวหน้าของโครงการโดยกรรมการ

☐ ผ่าน

☐ ไม่ผ่าน

ความคิดเห็นของกรรมการ (กรณีที่กรรมการประเมินไม่ผ่าน)

.....

ลายเซ็น

()

กรรมการตรวจสอบ

วันที่ / / 25....

รายงานผลความก้าวหน้าของโครงการ 75%

รายละเอียดของหัวข้อที่รายงานผลความก้าวหน้า	ผ่าน	ไม่ผ่าน	เหตุผล (กรณีไม่ผ่าน)
ส่วนบุคลากร			
15. หน้าตรวจสอบข้อมูลส่วนตัว			
16. หน้าตรวจสอบข้อมูลการเบิก - จ่ายค่า รักษาพยาบาล			
17. หน้าตรวจสอบข้อมูลสิทธิการเบิก - จ่าย ประจำปีงบประมาณ			
ส่วนรายงาน			
18. หน้ารายงานตรวจสอบสิทธิการเบิก – จ่ายค่า รักษาพยาบาลประจำวัน			
19. หน้ารายงานตรวจสอบสิทธิการเบิก – จ่ายค่า รักษาพยาบาลประจำเดือน			
20. หน้ารายงานตรวจสอบสิทธิการเบิก – จ่ายค่า รักษาพยาบาลประจำปีงบประมาณ			
21. ทดสอบการใช้ระบบงาน			
22. แก้ไขระบบงาน			
23. จัดทำโครงการบทที่ 4			

การประเมินผลความก้าวหน้าของโครงการโดยอาจารย์ที่ปรึกษา

☒ ผ่าน

☐ ไม่ผ่าน

ข้อชี้แจงของอาจารย์ที่ปรึกษา (กรณีที่อาจารย์ที่ปรึกษาประเมินไม่ผ่าน)

.....

ลายเซ็น

(อาจารย์คุณานนท์ สุขเกษม)

อาจารย์ที่ปรึกษา

วันที่ / / 25....

การประเมินผลความก้าวหน้าของโครงการโดยกรรมการ

☐ ผ่าน

☐ ไม่ผ่าน

ความคิดเห็นของกรรมการ (กรณีที่กรรมการประเมินไม่ผ่าน)

.....

.....

ลายเซ็น

()

กรรมการตรวจสอบ

วันที่ / / 25....

รายงานผลความก้าวหน้าของโครงการ 100%

รายละเอียดของหัวข้อที่รายงานผลความก้าวหน้า	หมายเหตุ
24. ทดสอบระบบสารสนเทศงานสวัสดิการการเบิก – จ่ายค่ารักษาพยาบาล	
25. แก้ไขและสรุประบบงาน	
26. รูปเล่มโครงการฉบับสมบูรณ์	
27. ขอสอบโครงการ	

การประเมินผลความก้าวหน้าของโครงการโดยอาจารย์ที่ปรึกษา

☐ ผ่าน

☐ ไม่ผ่าน

ข้อชี้แจงของอาจารย์ที่ปรึกษา (กรณีที่อาจารย์ที่ปรึกษาประเมินไม่ผ่าน)

.....

ลายเซ็น

(อาจารย์คุณานนท์ สุขเกษม)

อาจารย์ที่ปรึกษา

วันที่ / / 25....

บันทึกการเข้าพบอาจารย์ที่ปรึกษา[illegible]

ภาคผนวก ค

คู่มือการติดตั้งและใช้งานระบบจดจำใบหน้าและระบุตัวตน

คู่มือการติดตั้งและใช้งานโปรแกรม

ขั้นตอนการติดตั้งแบ่งออกเป็น 2 ขั้นตอนคือ

1. ติดตั้งโปรแกรม Visual Studio Code (Vscode)
2. ติดตั้ง Python

1. ติดตั้งโปรแกรม Visual Studio Code (Vscode)

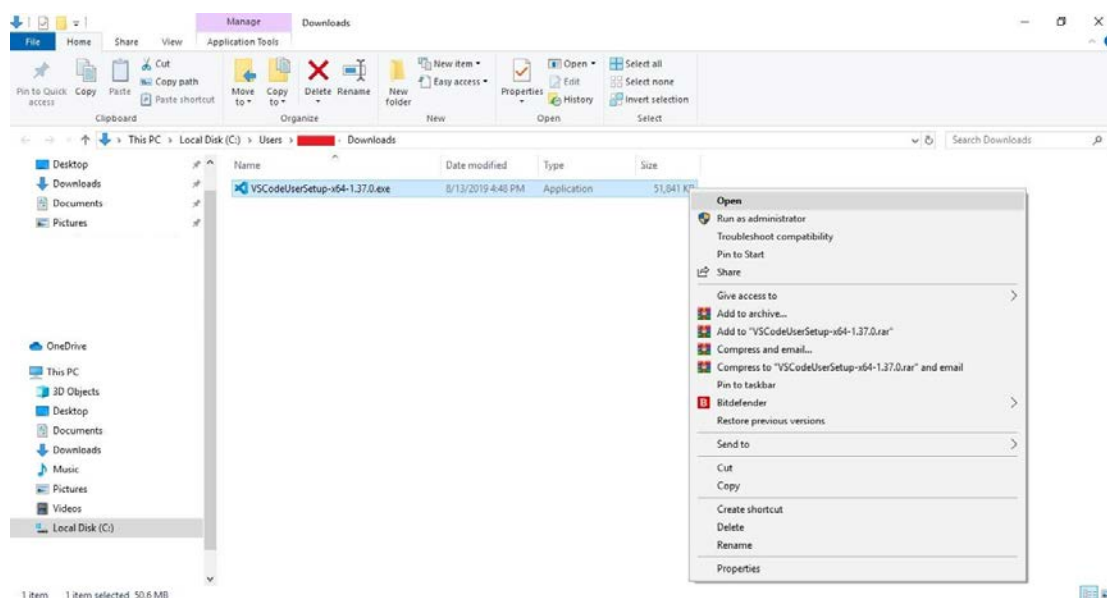
Visual Studio Code คือโปรแกรมที่ไว้ใช้แก้ไขโค้ดหรือก็คือ IDE นั้นเองและยังเป็นโปรแกรมที่พัฒนาโดย Microsoft และยังใช้ได้หลายระบบ

วิธีการติดตั้งโปรแกรม Visual Studio Code (Vscode)

จัดเตรียมโปรแกรมเพื่อติดตั้ง ดาวน์โหลดโปรแกรม Visual Studio Code (Vscode) จากเว็บไซต์ที่ไว้ <https://code.visualstudio.com> และ Download มาติดตั้งใช้งานตามระบบปฏิบัติการของเรา

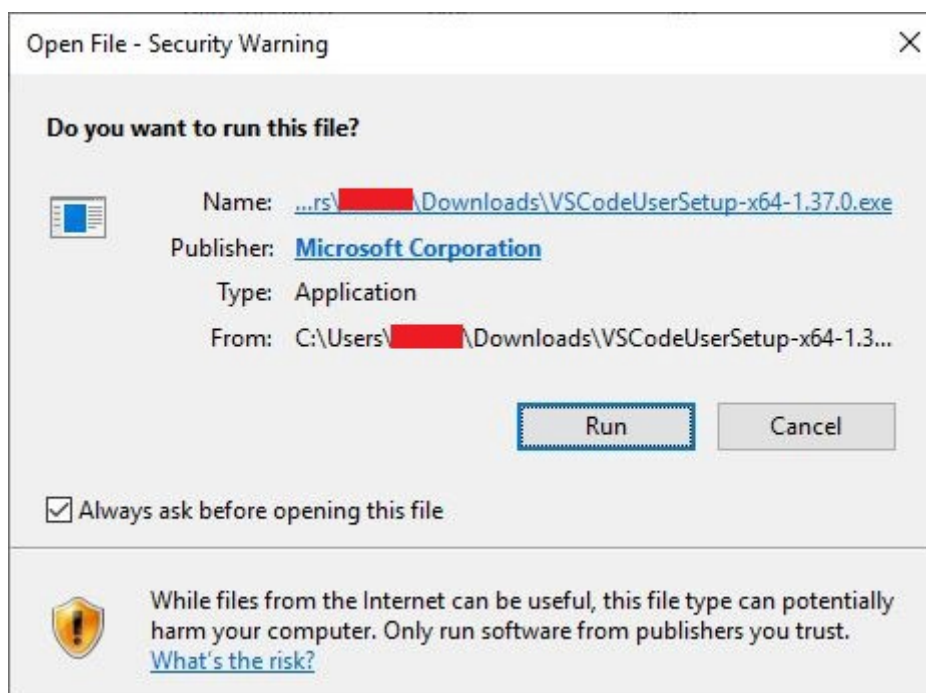
ขั้นตอนการติดตั้ง Visual Studio Code

1. ดับเบิลคลิก หรือคลิกขวาและกด “Open” โปรแกรมที่ดาวน์โหลดมา



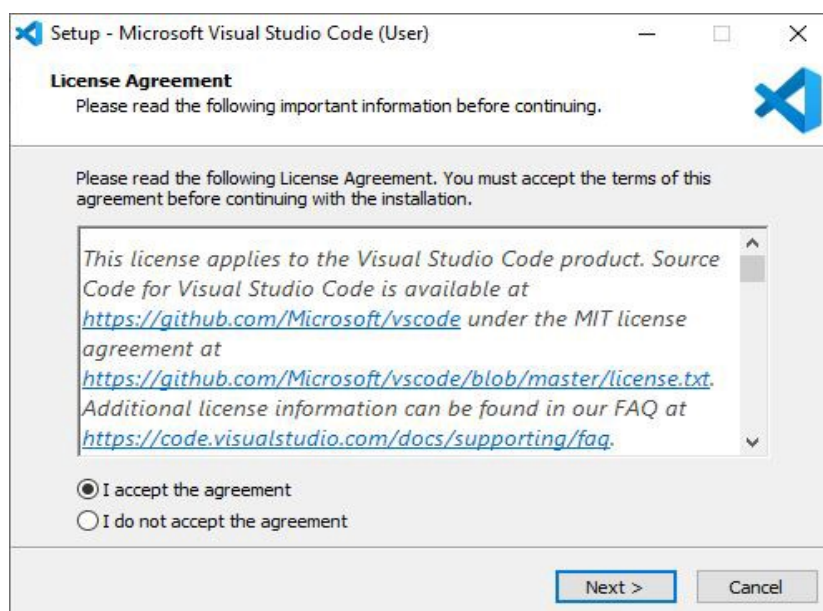
รูปที่ 1 ขั้นตอนการติดตั้งโปรแกรม Vscode

2. คลิกปุ่ม “Run”



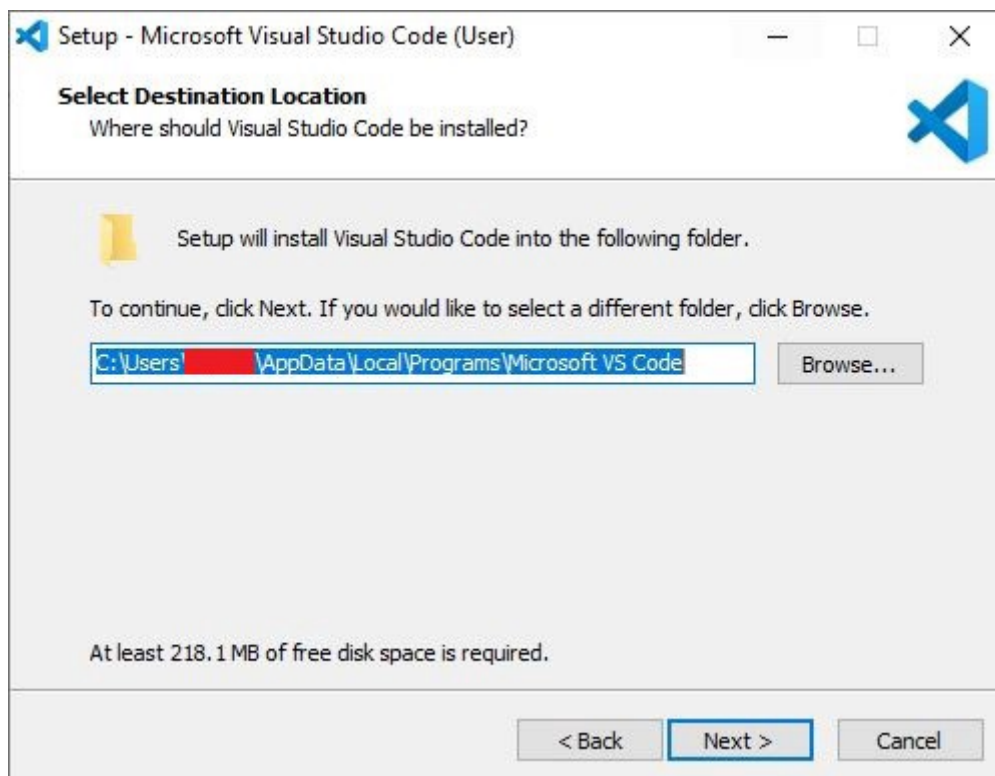
รูปที่ 2 แสดงรายละเอียดเงื่อนไข

3. เลือก “I accept the agreement” และคลิกปุ่ม “Next >”



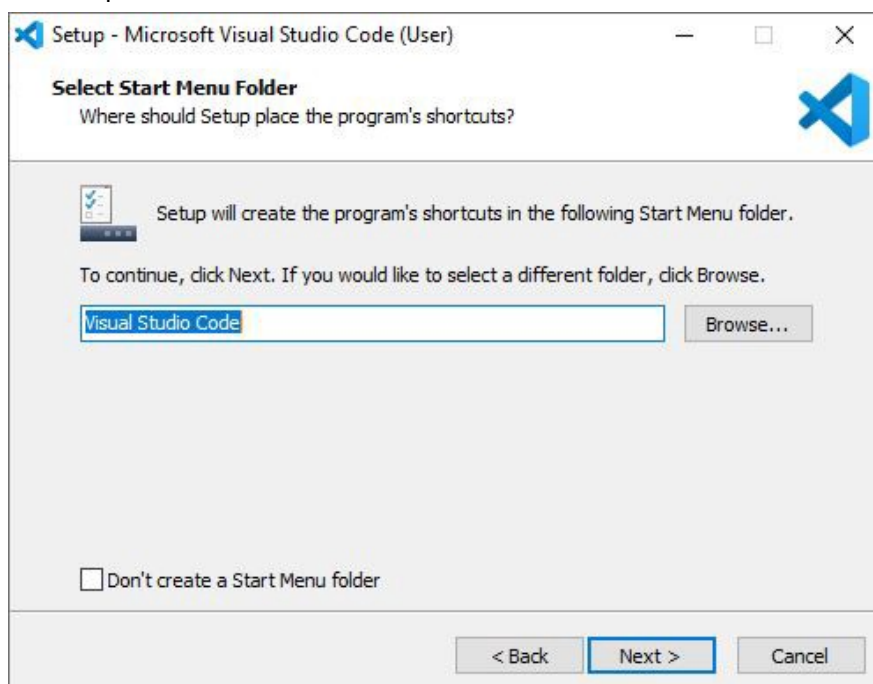
รูปที่ 3 เลือกการยินยอมเพื่อติดตั้ง

4. เลือกพื้นที่ในการจัดเก็บโปรแกรม (แนะนำให้ใช้ Default ที่ให้มา) และคลิกปุ่ม “Next >”



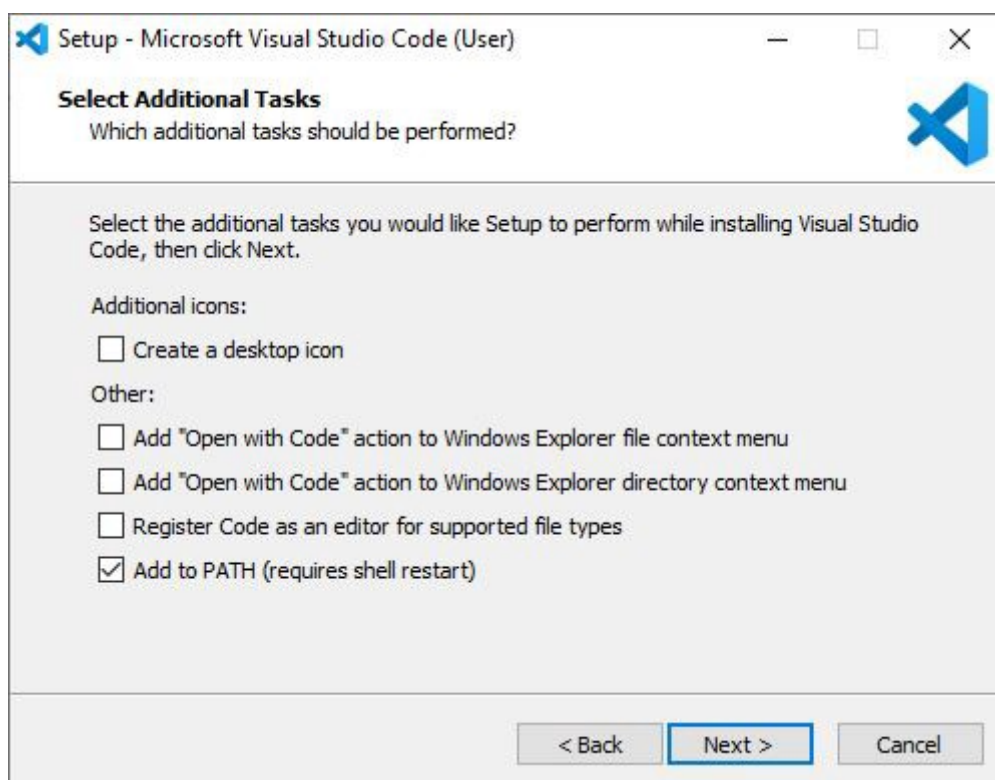
รูปที่ 4 เลือกตำแหน่งที่ต้องการติดตั้ง

5. คลิกปุ่ม “Next >”



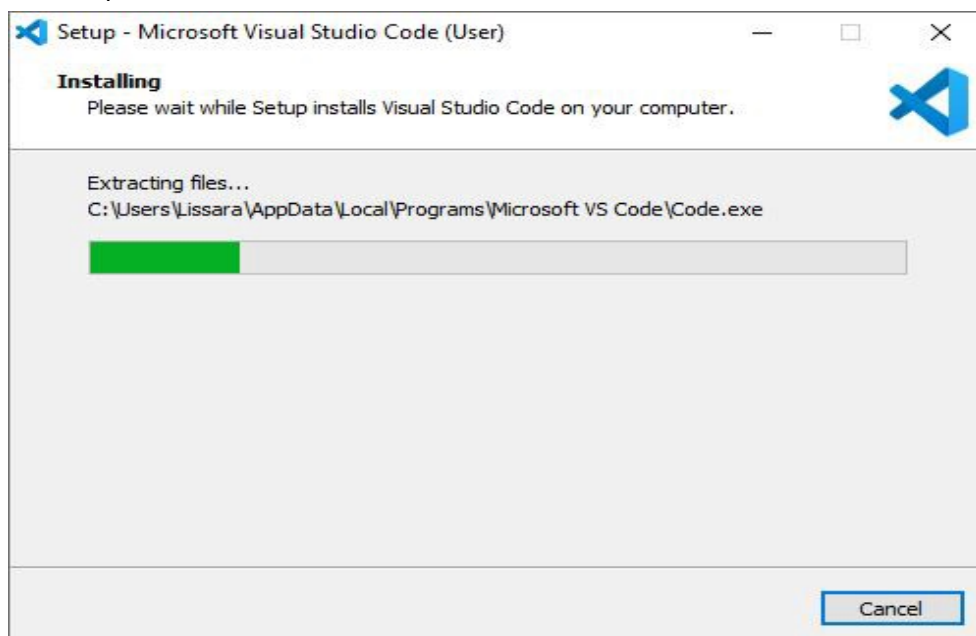
รูปที่ 5 แสดงการกำหนดให้แสดง Icon Link ที่ Start Menu

6. เลือกส่วนเพิ่มงานให้เลือก Create a desktop icon และ Add to PATH (requires shell restart) จากนั้นให้คลิกปุ่ม “Next >”



รูปที่ 6 แสดงการ กำหนดค่าต่างๆของโปรแกรม

7. คลิกปุ่ม “Install” เพื่อติดตั้งโปรแกรม





รูปที่ 7 แสดงหน้าจอขั้นตอนสิ้นสุดการติดตั้งโปรแกรม

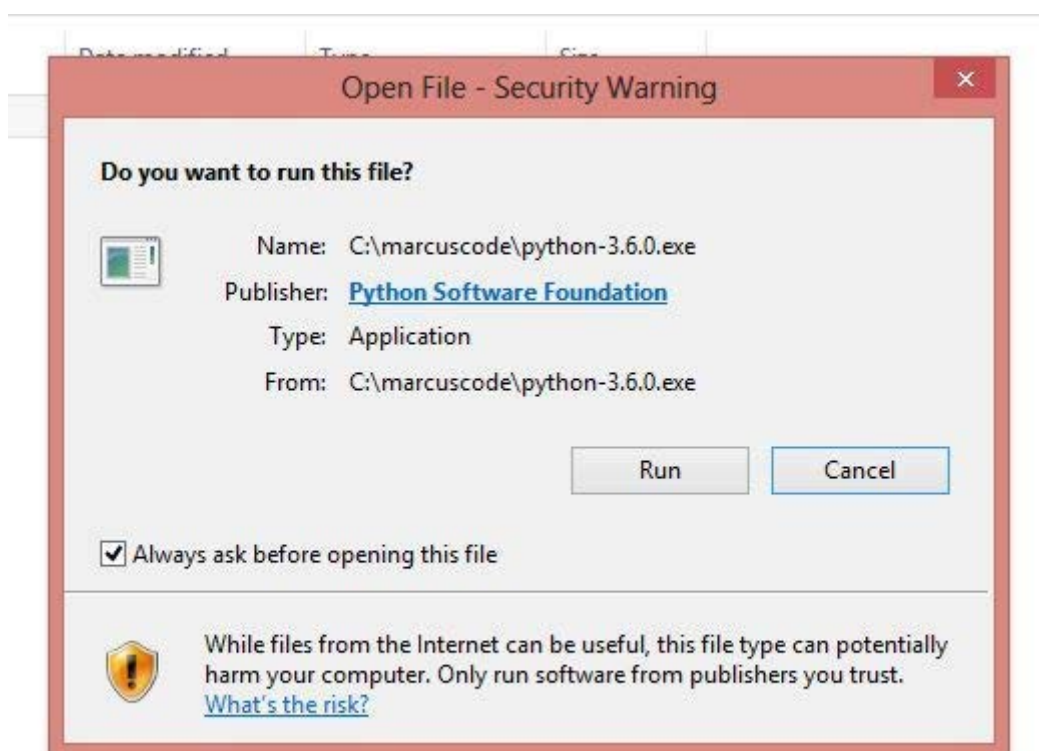
2. การติดตั้ง Python

2.1 ขั้นตอนการติดตั้งโปรแกรม Python นั้นเป็นแพ็คเกจของภาษาเพื่อให้สามารถใช้งานโค้ดของภาษา Python ได้และยังเป็นโปรแกรมที่ฟรีสามารถใช้งานได้โดยสามารถโหลดได้จากเว็บไซต์ <https://www.python.org/downloads/> และทำการ Download มาโดยแนะนำลดเวอร์ชัน 3.7.5 เป็นต้นไป



รูปที่ 8 Download โปรแกรม Python

2.1.2 ให้ไปที่สถานที่ที่คุณดาวน์โหลดภาษา Python ไว้และคลิกที่ไฟล์ "python-3.6.0.exe" และคลิกที่ "Run"



รูปที่ 9 เลือก Run เพื่อดำเนินการต่อ

2.1.3 คลิกเลือกที่ "Add Python 3.6 to PATH" เพื่อให้ระบบทำการกำหนด PATH เพื่อให้ภาษา Python สามารถทำงานได้กับ Command line อัตโนมัติ



รูปที่ 10 ตรวจสอบ Path ก่อนติดตั้งโปรแกรม

2.1.4 คลิกที่ "Install now " เพื่อเริ่มการติดตั้งภาษา Python





รูปที่ 11 ติดตั้งโปรแกรมสำเร็จ

ภาคผนวก ง

ประวัติผู้จัดทำ



ประวัติผู้เขียน

ชื่อ – สกุล	นายศิขรินทร์ งามดี
วันเดือนปี	19 ธันวาคม พ.ศ. 2541
สถานที่เกิด	กรุงเทพมหานคร
สถานที่อยู่ปัจจุบัน	162/82 ซอย มังกรขันดี ถนน เทพารักษ์ ตำบล แพรกษา อำเภอ เมือง จังหวัดสมุทรปราการ 10280
สถานที่ศึกษา	วิทยาลัยเทคโนโลยีอรรถวิทย์พัฒนวิชาการ บางนา
ประวัติการศึกษา	
พ.ศ. 2559	มัธยมศึกษาตอนปลาย การศึกษานอกระบบ สมุทรปราการ
พ.ศ. 2560 – ปัจจุบัน	ประกาศนียบัตรวิชาชีพชั้นสูง สาขาวิชาเทคโนโลยีสารสนเทศ วิทยาลัยเทคโนโลยีอรรถวิทย์พัฒนวิชาการ



ประวัติผู้เขียน

ชื่อ – สกุล	นายธรา รอบรู้
วันเดือนปี	25 กันยายน พ.ศ. 2542
สถานที่เกิด	สมุทรปราการ
สถานที่อยู่ปัจจุบัน	20/9 หมู่ 6 ซอย อยู่ดี ตำบล บางโหลง อำเภอบางพลี จังหวัดสมุทรปราการ 10540
สถานที่ศึกษา	วิทยาลัยเทคโนโลยีอรรถวิทย์พัฒนการ บางนา
ประวัติการศึกษา	
พ.ศ. 2557	มัธยมศึกษาตอนต้น บำบ่อวิทยาคม
พ.ศ. 2560 – ปัจจุบัน	ประกาศนียบัตรวิชาชีพชั้นสูง สาขาวิชาเทคโนโลยีสารสนเทศ วิทยาลัยเทคโนโลยีอรรถวิทย์พัฒนการ