การวิเคราะห์การเคลื่อนไหวดวงตา
EYE TRACKING AND EYE MOVEMENT

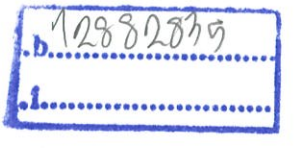
กันตินันท์ ไทยตรง
ธงชัย น้อยหิรัญ
อติคุณ นาคนวลนิ่ม

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

การวิเคราะห์การเคลื่อนไหวดวงตา
EYE TRACKING AND EYE MOVEMENT

กันตินันท์ ไทยตรง
ธงชัย น้อยหิรัญ
อติคุณ นาคนวลนิ่ม



ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

EYE TRACKING AND EYE MOVEMENT

KANTINUN THAITRONG
THONGCHAI NOIHIRAN
ATIKHUN NAKNAUNNIM

A SPEACIAL PROBLEM SUBMITTED IN PRATIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE
IN COMPUTER DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2015

หัวข้อปัญหาพิเศษ การวิเคราะห์การเคลื่อนไหวดวงตา

ชื่อนักศึกษา นายกันตินันท์ ไทยตรง 55050215

นายธงชัย น้อยหิรัญ 55050314

นายอติคุณ นาคนวลนิ่ม 55050524

ปริญญา วิทยาศาสตรบัณฑิต วิทยาการคอมพิวเตอร์

ภาควิชา วิทยาการคอมพิวเตอร์

ปีการศึกษา 2558

อาจารย์ที่ปรึกษา ผศ.ธีระ ศิริธีรากุล

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตาม หลักสูตรปริญญาวิทยาศาสตรบัณฑิต วิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2558

คณะกรรมการสอบ	ลายมือชื่อ
อ.สันธนะ อู่อุดมยิ่ง ประธานกรรมการ	
ดร.สายชล ใจเย็น กรรมการ	
ผศ.ธีระ ศิริธีรากุล กรรมการและอาจารย์ที่ปรึกษา	

ลิขสิทธิ์ของคณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อปัญหาพิเศษ	การวิเคราะห์การเคลื่อนไหวดวงตา	
ชื่อนักศึกษา	นายกันตินันท์ ไทยตรง	55050215
	นายธงชัย น้อยหิรัญ	55050314
	นายอติคุณ นาคนวลนิ่ม	55050524
ปริญญา	วิทยาศาสตรบัณฑิต วิทยาการคอมพิวเตอร์	
ภาควิชา	วิทยาการคอมพิวเตอร์	
ปีการศึกษา	2558	
อาจารย์ที่ปรึกษา	ผศ.ธีระ ศิริธีรากุล	

บทคัดย่อ

งานวิจัยฉบับนี้จะอธิบายถึงการพัฒนาของ Eye Tracking ซึ่งเป็นระบบทดสอบการใช้งานผ่านการวิเคราะห์การเคลื่อนไหวของดวงตา โดยระบบนี้เกิดจากแนวคิดการพัฒนาการปฏิสัมพันธ์ระหว่างผู้ใช้งานกับคอมพิวเตอร์เพื่อตรวจจับการเคลื่อนไหวของดวงตาของผู้ใช้เพื่อลดช่องว่างระหว่างมนุษย์และคอมพิวเตอร์ โดยจะจำลองการทำงานและการรับรู้ของมนุษย์ออกมาเป็นโมเดลเพื่อให้คอมพิวเตอร์เข้าใจผู้ใช้งานได้ ระบบการวิเคราะห์การเคลื่อนไหวดวงตาคือซอฟต์แวร์ที่นำเอาหลักการเชิงเทคนิคของหลักจิตวิทยา และกายศาสตร์มาผนวกกันเพื่อนำไปพัฒนาระบบนี้ขึ้นมา โดยระบบจะทำการค้นหาใบหน้าของผู้ทดสอบเพื่อทำการค้นหาดวงตาบนใบหน้าจากนั้นระบบจะทำการจับไปยังม่านตาทั้งซ้ายและขวาของผู้ทดสอบจากนั้นระบบจะแสดงค่าแกน x และแกน y ออกมาเพื่อให้รู้ว่าดวงตานั้นอยู่ตำแหน่งที่เท่าไหร่เพื่อทำการหาค่าเฉลี่ยแต่ละตำแหน่งของแต่ละคนเพื่อนำมาวิเคราะห์ตำแหน่งที่คนจะมองมากที่สุด

คำสำคัญ: eye tracking, eye movement

Title	Eye Tracking And Eye Movement	
Students	Mr. Kantinun Thaitrong	55050215
	Mr. Thongchai Noihiran	55050314
	Mr. Atikhun Naknaunnim	55050524
Degree	Bachelor of Science Major Program Computer Science	
Department	Computer Science	
Academic Year	2015	
Advisor	Asst. Prof. Teera Siriteerakul	

Abstract

This research is explain about development of eye tracking, which is a test system process by movement of the eye. By this system start from develop the idea of interaction between user and computer to detect user eye movement to reduce the gap between human and computer by imitate human sense and work to make a model and make computer understand the user. Eye movement process system is use a technical of psychology and anatomy to develop this system. By system try to searching the face of tester to find the eye of tester then system. Then system detect the position of left and right of the iris of the eye to show value of X and Y axis for position of user eyes. To calculate for average of any user position to analyze for where is the most focus point of human.

Keyword: eye tracking, eye movement

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี ด้วยความช่วยเหลือของผู้ช่วยศาสตราจารย์ธีระ ศิริธีรากุล อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งท่านได้ให้คำแนะนำและข้อคิดเห็นต่างๆ อันเป็นประโยชน์อย่างยิ่งในการทำวิจัย อีกทั้งยังช่วยแก้ปัญหาต่างๆ ที่เกิดขึ้นระหว่างการดำเนินงานอีกด้วย ขอขอบคุณ คุณณัฐวุฒิ ลีกุลพิทักษ์ สำหรับข้อแนะนำและความช่วยเหลือในทุกๆ ด้านในการทำวิจัย นอกจากนี้ขอขอบคุณเพื่อนๆ ในห้องปฏิบัติการคอมพิวเตอร์ทุกคนที่เป็นกำลังใจ และให้ความช่วยเหลือในการทำวิทยานิพนธ์เรื่องนี้

สุดท้ายนี้ ผู้วิจัยขอขอบพระคุณบิดามารดา และครอบครัว ซึ่งเปิดโอกาสให้ได้รับการศึกษาเล่าเรียน ตลอดจนคอยช่วยเหลือและให้กำลังใจผู้วิจัยเสมอมาจนสำเร็จการศึกษา

นายกันตินันท์ ไทยตรง

นายธงชัย น้อยหิรัญ

นายอติคุณ นาคนวลนิ่ม

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก
บทคัดย่อภาษาอังกฤษ ..... ข
กิตติกรรมประกาศ ..... ค
สารบัญ ..... ง
สารบัญตาราง ..... ฉ
สารบัญรูป ..... ช
บทที่ 1 บทนำ 1
1.1 ที่มาและความสำคัญของโครงงาน 1
1.2 วัตถุประสงค์ 2
1.3 ขอบเขตของปัญหา 2
1.4 ประโยชน์ที่คาดว่าจะได้รับ 2
1.5 ขั้นตอนการดำเนินงาน 3
1.6 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ 3
บทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง 4
2.1 การประมวลภาพ 4
2.2 OpenCV 10
2.3 Haar like-feature 13
2.4 Region of interest (ROI) 14
2.5 Circle Detection 15
2.6 การติดตามลักษณะเด่นของใบหน้า 16
2.7 Gaussian Blur 18

สารบัญ(ต่อ)

หน้า

บทที่ 3 การวิเคราะห์ออกแบบและพัฒนาระบบ 20

3.1 การออกแบบระบบ 20

3.2 กระบวนการทำงาน 22

3.3 การดำเนินงานในส่วน Face Localization 23

3.3 การดำเนินงานในส่วน Eye Detection 23

บทที่ 4 การทดลองและผลการทดลอง 26

4.1 การทดสอบ 26

4.2 ผลการทดสอบ 27

4.3 ผลการทดลอง 32

บทที่ 5 สรุปผลการทดลอง 34

5.1 การทดสอบ 34

5.2 ปัญหาอุปสรรคและแนวทางการแก้ไข 34

5.3 แนวทางในการพัฒนาต่อ 35

บรรณานุกรม 36

ภาคผนวก 37

ภาคผนวก ก 38

สารบัญตาราง

หน้า

ตารางที่ 1 แสดง Use Case Description ของการทดสอบระบบ 21

ตารางที่ 2 แสดง Use Case Description ของการวิเคราะห์และสรุปผลการทดลอง.....22

ตารางที่ 3 ตารางการทดสอบการใช้งานระบบตรวจจับดวงตา.....32

สารบัญรูป

หน้า

รูปที่ 2.1 ภาพที่ใช้ในการทดสอบระดับเทา 5
รูปที่ 2.2 ระดับเทาของ pixel ในแถวแนวนอนที่ 3 และแถวแนวตั้งที่ 2 6
รูปที่ 2.3 การตรวจหาดวงตา 10
รูปที่ 2.4 โครงสร้างของ OpenCV 11
รูปที่ 2.5 ระบบปฏิบัติการและสถาปัตยกรรมที่ OpenCV สนับสนุน 12
รูปที่ 2.6 รูปแบบของรูปเหลี่ยมสำหรับการตรวจจับลักษณะแบบต่างๆ 13
รูปที่ 2.7 ตัวอย่างการใช้รูปเหลี่ยมตรวจจับลักษณะต่างๆ 13
รูปที่ 2.8 การคำนวณแบบ Integral image 14
รูปที่ 2.9 ตัวอย่างการกำหนดบริเวณที่สนใจ 15
รูปที่ 2.10 ส่วนประกอบของดวงตา 16
รูปที่ 2.11 ขั้นตอนวิธีการติดตามดวงตา 17
รูปที่ 2.12 ตัวอย่างขั้นตอนวิธีการติดตามดวงตา 18
รูปที่ 2.13 ภาพที่ใช้ Gaussian blur 19
รูปที่ 3.1 Use Case Diagram ของระบบ 20
รูปที่ 3.2 ภาพที่ได้จากการค้นหาส่วนของใบหน้า 24
รูปที่ 3.3 ภาพที่ค้นหาใบหน้าไม่พบ 24
รูปที่ 3.4 ภาพที่ได้จากการค้นหาส่วนของดวงตา 25
รูปที่ 3.5 ภาพที่ได้จากการค้นหาจุดกึ่งกลางดวงตา 25
รูปที่ 4.1 ตารางจำลองหน้าจอโน้ตบุ๊ค 26
รูปที่ 4.2 ทิศทางในการมองบนหน้าจอโน้ตบุ๊ค 27
รูปที่ 4.3 ผู้ทดสอบมองไปยังตำแหน่งที่ 1 27

สารบัญรูป(ต่อ)

หน้า

รูปที่ 4.4 ผู้ทดสอบมองไปยังตำแหน่งที่ 2 28
รูปที่ 4.5 ผู้ทดสอบมองไปยังตำแหน่งที่ 3 28
รูปที่ 4.6 ผู้ทดสอบมองไปยังตำแหน่งที่ 4 29
รูปที่ 4.7 ผู้ทดสอบมองไปยังตำแหน่งที่ 5 29
รูปที่ 4.8 ผู้ทดสอบมองไปยังตำแหน่งที่ 6 30
รูปที่ 4.9 ผู้ทดสอบมองไปยังตำแหน่งที่ 7 30
รูปที่ 4.10 ผู้ทดสอบมองไปยังตำแหน่งที่ 8 31
รูปที่ 4.11 ผู้ทดสอบมองไปยังตำแหน่งที่ 9 31
รูปที่ ก.1 หน้าต่างของเว็บไซต์ www.opencv.org/downloads 38
รูปที่ ก.2 แสดงการสร้างโปรเจค 39
รูปที่ ก.3 แสดงหน้าต่างการตั้งค่า 39
รูปที่ ก.4 แสดงหน้าต่างการตั่งค่า 40
รูปที่ ก.5 แสดงหน้าต่างการคลิกเลือก 40
รูปที่ ก.6 แสดงหน้าต่างการคลิกเลือก Configuration Manager 41
รูปที่ ก.7 แสดงหน้าต่าง Configuration Manager 41
รูปที่ ก.8 แสดงการตั้งค่าใน Configuration Manager 42
รูปที่ ก.9 แสดงการตั้งค่าใน Configuration Manager 42
รูปที่ ก.10 แสดงหน้าต่างเมื่อคลิกปุ่ม New ที่ Win32 43
รูปที่ ก.11 แสดงการตั้งค่าใน Configuration Manager 43

สารบัญรูป(ต่อ)

หน้า

รูปที่ ก.12 แสดงขั้นตอนการตั้งค่าใน Property.....45

รูปที่ ก.13 แสดงขั้นตอนการตั้งค่าใน Property.....45

รูปที่ ก.14 แสดงขั้นตอนการตั้งค่าใน Property.....46

รูปที่ ก.15 แสดงขั้นตอนสุดท้ายของการตั้งค่าใน Property.....46

รูปที่ ก.16 ตัวอย่างการใช้งานไลบรารี OpenCV เพื่อแสดงรูปภาพ.....47

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

การตรวจหาและติดตามดวงตา คือ หนึ่งในความท้าทายของศาสตร์ด้านวิทยาการคอมพิวเตอร์ในศตวรรษที่ 21 เพราะตำแหน่งของการมอง และข้อมูลที่ได้รับจากการติดตามนั้นสามารถนำไปประยุกต์ใช้กับงานต่างๆ อย่างแพร่หลาย ดังนั้นทางคณะผู้วิจัยมีความสนใจในการติดตาม และการเคลื่อนที่ของใบหน้าและดวงตาของมนุษย์ โดยได้ทดลองการควบคุมกล้องเว็บแคมให้สามารถเปิดและสามารถจับใบหน้าและดวงตา

การพัฒนาระบบการวิเคราะห์การเคลื่อนไหวของดวงตาจะต้องทำการประมวลผลภาพเพื่อหาตำแหน่งของใบหน้าและดวงตาซึ่งมีหลายงานวิจัยที่ได้ประมวลผลภาพเพื่อหาอวัยวะบนใบหน้ามาก่อนหน้านี้ เช่น การประมวลผลภาพใบหน้าเพื่อระบุอารมณ์ของผู้ใช้, การประมวลผลภาพเพื่อหาตำแหน่งใบหน้าของผู้ใช้ เป็นต้น ซึ่งทางคณะผู้วิจัยได้นำการประมวลผลภาพใบหน้าเพื่อนำมาพัฒนาเพื่อค้นหาตำแหน่งและติดตามการเคลื่อนไหวของดวงตา ซึ่งกระบวนการเหล่านี้เป็นการประมวลผลโดยเทคนิคการประมวลผลภาพดิจิทัล (Image processing) กระบวนการเหล่านี้สามารถประมวลผลได้เป็นอย่างดี แต่เมื่อนำมาพัฒนาใช้งานร่วมกับอุปกรณ์จริงยังไม่เป็นที่นิยมมากนักเนื่องด้วยความซับซ้อนของเทคนิคที่จำเป็นต้องใช้ระยะเวลาในการศึกษาและพัฒนาเป็นเวลานานซึ่งต่อมาได้มีผู้พัฒนาอัลกอลิทึ่มในการประมวลผลภาพที่สามารถทำงานร่วมกับกล้องเว็บแคมได้อย่าง OpenCV ขึ้นมา

โครงการปัญหาพิเศษนี้นำทฤษฎีประมวลภาพมาประยุกต์ใช้งานเพื่อการศึกษาวิเคราะห์คุณลักษณะของใบหน้า ดวงตา และตาดำจากภาพวิดีโอกล้องเว็บแคมซึ่งใช้เทคนิคการค้นหาใบหน้า โดยโครงงานนี้ได้เลือกใช้ไลบรารี่ของ OpenCV เข้ามาช่วยในการพัฒนาโปรแกรมร่วมกับการเขียนโปรแกรมภาษา C++

1.2 วัตถุประสงค์

1.2.1 สร้างระบบที่สามารถตรวจจับใบหน้า ดวงตา และม่านตา

1.2.2 เป็นการนำเอาความรู้ Computer Vision & Image Processing มาพัฒนาเพื่อต่อยอดเทคโนโลยีในปัจจุบัน

1.2.3 ลดค่าใช้จ่ายในการสร้างเทคโนโลยี

1.3 ขอบเขตของปัญหา

1.3.1 ใช้งานได้บนโน้ตบุ๊ค หรือเครื่องคอมพิวเตอร์ที่มีกล้องเว็บแคมเท่านั้น

1.3.2 ขณะใช้งานโปรแกรมใบหน้าของผู้ใช้งานต้องอยู่ในเฟรมของกล้องเว็บแคมเท่านั้น

1.3.3 สามารถใช้งานโปรแกรมได้ทีละ 1 User เท่านั้น

1.3.4. ถ้ามีสิ่งใดมาบดบังใบหน้าโปรแกรมจะไม่สามารถ Detect ได้

1.3.5. การใช้งานต้องอาศัยการเครื่องที่ของดวงตาที่ถูกต้องตามที่ผู้พัฒนากำหนดไว้ ไม่เช่นนั้นโปรแกรมจะไม่ตอบสนอง

1.3.6 บริเวณที่ใช้งานจะต้องมีแสงสว่างเพียงพอที่จะให้กล้องเห็นดวงตาและใบหน้าของผู้ใช้งาน

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1. ได้เรียนรู้เทคนิคและขั้นตอนวิธีในการตรวจจับและติดตามการเคลื่อนที่ของดวงตา และตาดำมนุษย์

1.4.2. ได้เรียนรู้เทคนิคในการประยุกต์ใช้คำสั่งต่างๆ ที่มีอยู่ มาใช้งานร่วมกับอัลกอลิทึมในการตรวจจับและติดตาม การเคลื่อนที่ของดวงตามนุษย์ให้สามารถนำไปใช้งานได้จริง

1.4.3. สามารถนำไปต่อยอดในการออกแบบและพัฒนาอุปกรณ์ช่วยเหลืออื่นๆ สำหรับผู้พิการหรือผู้ป่วยที่ไม่สามารถเคลื่อนไหวร่างกายในส่วนต่างๆ ได้ในอนาคต

1.5 ขั้นตอนการดำเนินงาน

1.5.1 กำหนดขอบเขตในการพัฒนา

1.5.2 ศึกษาวิธีการพัฒนาระบบตรวจจับและติดตามการเคลื่อนที่ของใบหน้า และดวงตา

1.5.3 ศึกษาวิธีการใช้งานอุปกรณ์และทฤษฎีการตรวจจับภาพ

1.5.4 ออกแบบโครงสร้าง และการทำงานของระบบ

1.5.5 ทดสอบการทำงาน และสรุปผลการทดลอง

1.6 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ

1.6.1. ฮาร์ดแวร์

1.6.1.2 Notebook Dell Latitude 3440

- หน่วยประมวลผล 1.6 GHz intel Core i5-4200U
- RAM 4 GB DDR3
- ฮาร์ดดิสก์ 500 GB
- หน้าจอ 14 นิ้ว LCD display

1.6.2. ซอฟต์แวร์

1.6.2.1 Language and GUI

- Open CV
- C++

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

2.1 การประมวลผลภาพ

การประมวลผลภาพ คือการเอาภาพมาประมวลผล โดยคำนวณด้วยคอมพิวเตอร์ เพื่อให้คอมพิวเตอร์รู้ว่าภาพนั้นคือภาพอะไรใช่สิ่งที่เราต้องการหรือไม่

การมองเห็นของมนุษย์เป็นสิ่งที่สำคัญและเป็นกลไกการรับภาพที่ซับซ้อนอย่างหนึ่ง ซึ่งจะให้ข้อมูลที่มีความจำเป็นสำหรับใช้งานง่ายๆ (ตัวอย่างเช่น การจดจำวัตถุ) และสำหรับงานที่มีความซับซ้อน (ได้แก่ การวางแผน การตัดสินใจ การค้นคว้าทางวิทยาศาสตร์ การพัฒนาทางด้านความคิด) รูปภาพมีบทบาทมากสำหรับองค์กรต่างๆ เช่น หนังสือพิมพ์ โทรทัศน์ ภาพยนตร์ ซึ่งได้ใช้ภาพ (ภาพนิ่ง ภาพเคลื่อนไหว) เป็นสื่อนำเสนอข้อมูลข่าวสารต่างๆ สิ่งที่น่าสนใจของข้อมูลที่เกี่ยวกับการมองเห็นหรือข้อมูลภาพนั้นก็คือการประมวลภาพ (Image Processing) โดยใช้ดิจิตอลคอมพิวเตอร์

Digital image processing จะเกี่ยวกับการแปลงข้อมูลภาพให้อยู่ในรูปแบบข้อมูลดิจิตอล (Digital format) ซึ่งสามารถที่จะนำเอาข้อมูลนี้จัดผ่านกระบวนการต่างๆด้วยดิจิตอลคอมพิวเตอร์ได้ในระบบดิจิตอล อินพุตและเอาต์พุตของระบบจะอยู่ในรูปแบบดิจิตอลเท่านั้น

Digital image analysis จะเกี่ยวกับวิธีการอธิบายและการจดจำข้อมูลภาพดิจิตอล ซึ่งอินพุตของระบบจะเป็นข้อมูลดิจิตอลและเอาพุตจะเป็นเครื่องหมายที่ใช้แทนข้อมูลภาพดิจิตอลเหล่านั้น ในการวิเคราะห์ภาพมีอยู่หลายวิธีด้วยกันที่ได้นำมาจากการทำงานของตามนุษย์ (human vision) นั่นก็คืองานทางด้าน Computer Vision เป็นลักษณะเดียวกับ Digital image analysis นั่นเอง การมองเห็นของมนุษย์นับว่าเป็นกระบวนการที่ซับซ้อนซึ่งลักษณะเทคนิคโดยทั่วไปในกระบวนการ Digital image analysis และ Computer Vision จะค่อนข้างซับซ้อนเช่นกัน

การคำนวณนั้นมีหลายวิธี เช่นการนำเอาจัดสีแต่ละ Pixel มาคำนวณหรือการนำหลายๆจุดมารวมกันเป็นบริเวณกว้างขึ้น เช่น การดูPattern ของวัตถุ การวิเคราะห์หารูปร่างของวัตถุ เพื่อหาค่าอะไรบางอย่างที่ช่วยให้เข้าใจได้ว่า ภาพนั้นมีลักษณะอย่างไร

2.1.1 Pixel

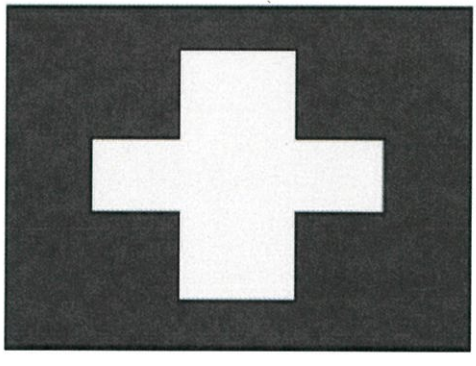
Pixel คือ พื้นที่เล็กๆจุดหนึ่งในภาพ โดยในแต่ละจุดนั้นจะมีค่าตัวเลขกำกับ ซึ่งตัวเลข เหล่านี้จะมาจากค่าของแม่สีสามสี R (สีแดง) G (สีเขียว) B (สีฟ้า) ใช้บอกระดับความเข้มของแต่ละ เฉดสี หากมี Pixel หลายๆจุดมาต่อกันจะกลายเป็นภาพซึ่งมีขนาด จำนวน Pixel ด้านกว้าง X จำนวน Pixel ด้านยาว ยกตัวอย่าง เช่น รูปภาพขนาด 800 x 600 pixels หมายความว่า รูปนี้มีความกว้าง 800 pixels และมีความยาว 600 pixels เป็นต้น

2.1.2 ระดับเทา (Gray level)

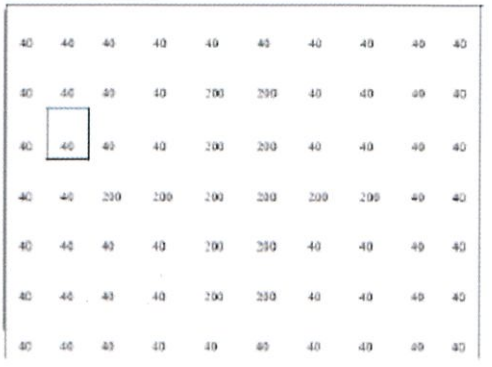
ระดับเทา (Gray Level) เป็นค่าซึ่งระบุความสว่างหรือความเข้ม ซึ่งมีค่าตั้งแต่ 0-255 (0 คือ ระดับความเข้ม 255 คือระดับความสว่าง) รวมทั้งพิกัดแนวนอนและแนวตั้ง ซึ่งใช้ระบุตำแหน่งในแถวลำดับ ภาพ (Image Array) เช่นจากรูปตัวอย่างที่ 2.1 และ 2.2 จุดภาพแถวนอนที่ 3 และแนวตั้งที่ 2 ในรูป 2.2 ซึ่งมีค่าระดับเทา 40 วิธีการหาค่าระดับเทา (Gray Level) สามารถทำได้ดังสมการ 2.1

$$Gray\ Level\ = \frac{R + G + B}{3}$$

(2.1)



รูปที่ 2.1 ภาพที่ใช้ในการทดสอบระดับเทา



รูปที่ 2.2 ระดับเทาของ pixel ในแถวแนวนอนที่ 3 และแถวแนวตั้งที่ 2

2.1.3 การแปลงภาพสีเป็นภาพขาว-ดำ

การแปลงภาพสีให้เป็นภาพขาว -ดำ (Thresholding) เป็นกระบวนการแปลงภาพสีให้มีการ แสดงผลได้แค่ 2 ระดับ คือ ขาว และดำ โดยจะแปลงข้อมูลภาพให้เป็นภาพ binary (Binary Image) มีกระบวนการแปลงภาพที่มีความเข้มหลายระดับ (Multilevel Image) ให้เป็นภาพที่มีความเข้มเพียง 2 ระดับ หรือ 1 บิต (bit) คือ 0 และ 1 โดย 0 แทนด้วยจุดที่มีภาพสีขาว และ 1 แทนด้วยจุดที่มีภาพสี ดำ Thresholding Technique คือการพิจารณาจุด pixel ในภาพว่าจุดใดควรจะเป็นจุดขาว หรือจุดใด ควรจะเป็นจุดที่มีค่าเท่ากับ 1 (จุดดำ) โดยจะทาการเปรียบเทียบค่าของแต่ละpixel (f(x,y)) กับค่าคงที่ ที่เรียกว่า Threshold (Threshold Value) เทคนิคนี้นิยมใช้กันมากในกรณีที่ความแตกต่างระหว่าง วัตถุ (Object) และพื้นหลัง (Background) ค่า pixel ในภาพที่มีค่าน้อยกว่าค่า Threshold จะถูก กำหนดเป็น 1 (จุดดำ) และถ้าค่าของ pixel ใด ๆ ในภาพมีค่ามากกว่าหรือเท่ากับค่า Threshold จะถูกกำหนดให้เป็น 0 (จุดขาว) ในการทำภาพ Binary โดยการทำ Thresholding ให้ได้ภาพดีและคมชัด ต้องเกิดจากการเลือกค่า Threshold ที่ถูกต้องและเหมาะสม ถ้าเลือกค่า Threshold ไม่เหมาะสม เช่น ค่า Threshold ที่มากหรือน้อยจนเกินไป ภาพที่ได้จะขาดความคมชัดหรืออาจทาให้รายละเอียดของ ภาพขาดหายไป หรือภาพที่ได้อาจจะมืดเกินไป หรือสว่างเกินไป หรืออาจจะเป็นภาพที่มีสิ่งรบกวน (Noise) เกิดขึ้น ทาให้ภาพผลลัพธ์ที่ได้ไม่ชัดเจน

2.1.4 ระบบสี RGB

ระบบสี RGB เป็นระบบสีที่เกิดจากการรวมกันของแสงสีแดง เขียวและน้ำเงินโดยมีการรวมกันแบบ Additive ซึ่งโดยปกติจะนำไปใช้ในจอภาพแบบ CRT (Cathode ray tube) ในการใช้งานระบบสี RGB ยังมีการสร้างมาตรฐานที่แตกต่างกันออกไปที่นิยมใช้งานได้แก่ RGBCIE และ RGBNTSC

2.1.4.1 ระบบสีแบบ RGB ของ CIE

เป็นระบบสีที่พัฒนาขึ้นโดย CIE (Commission International l 'Eclairage) ซึ่งอ้างอิงสีด้วยสีแดงที่ 700nm สีเขียวเท่ากับ 546.1 nm และสีน้ำเงิน 435.8 nm

2.1.4.2 ระบบสีแบบ RGB ของ NTSC

เป็นระบบที่พัฒนาโดย NTSC (National Television System Committee) เพื่อใช้สำหรับการแสดงภาพของจอภาพแบบ CRT เป็นมาตรฐานสาหรับผู้ผลิตแบบ CRT ให้มีลักษณะเดียวกัน

2.1.5 ระบบสี HSV

ระบบสี HSV (Hue Saturation Value) เป็นการพิจารณาสีโดยใช้ Hue Saturation และ Valueซึ่ง Hue คือค่าสีของสีหลัก (แดง เขียวและน้ำเงิน) ในทางปฏิบัติจะอยู่ระหว่าง 0 และ 255 ซึ่งถ้า Hue มีค่าเท่ากับ 0 จะแทนสีแดงและเมื่อ Hue มีค่าเพิ่มขึ้นเรื่อย ๆ สีก็จะเปลี่ยนแปลงไปตามสเปกตรัมของสีจนถึง256 จึงจะกลับมาเป็นสีแดงอีกครั้ง ซึ่งสามารถแทนให้อยู่ในรูปขององศาได้ ดังนี้คือ สีแดง = 0 องศา สีเขียวเท่ากับ 120 องศา สีน้ำเงินเท่ากับ 240 องศา Hue สามารถคำนวณได้จากระบบสี RGB ได้ดังนี้

$$redh = red - min(red, green, blue) \quad (2.2)$$

$$greenh = green - min(red, green, blue) \quad (2.3)$$

$$blueh = blue - min(red, green, blue) \quad (2.4)$$

จากลักษณะโมเดลของระบบ Hue พบว่าจะมีค่าอย่างน้อยหนึ่งค่าที่จะเท่ากับ 0 แต่ถ้ามีสองค่าเท่ากับ 0 แล้ว hue จะเป็นมุมของสี(ค่าสี)มีค่าเป็นไปตามสี่ที่สามและถ้าทั้งสามสีมีค่าเท่ากับ 0 แล้วจะทาให้ไม่มีค่าของ Hue หรือสีที่ได้จะมีค่าเท่ากับสีขาวนั่นเอง ตัวอย่างเช่น จอภาพขาว-ดา ถ้าเกิดมีสีใดสีหนึ่งมีค่าเท่ากับ 0 จะทาให้ค่าสีที่ได้เป็นไปตามสีที่เหลือ การให้น้ำหนักในการพิจารณาเมื่อสีแดงมีค่าเท่ากับ 0

$$240 \times blueh + (120 \times greenh)\ blueh + greenh \quad (2.5)$$

Saturation คือความบริสุทธิ์ของสีซึ่งถ้า Saturation มีค่าเท่ากับ 0 แล้วสีที่ได้จะไม่มี Hue ซึ่งจะเป็นสีขาวล้วน แต่ถ้า Saturation มีค่าเท่ากับ 255 แสดงว่าจะไม่มีแสงสีขาวผสมอยู่เลย

Saturation สามารถคำนวณได้ดังนี้

$$Saturation = \max red, green, blue - min(red, green, blue) max(red, green, blue) \quad (2.6)$$

Value คือความสว่างของสี ซึ่งสามารถวัดได้โดยค่าความเข้มของความสว่างของแต่ละสีที่ประกอบกันสามารถคำนวณได้จาก

$$value = max(red, green, blue) \quad (2.7)$$

2.1.6 การตรวจหาตำแหน่งดวงตา

การตรวจหาตำแหน่งบริเวณดวงตาจะแสดงดังรูป 2.3 ซึ่งได้แบ่งออกเป็นสองขั้นตอน ขั้นตอนแรกจะทาการหาบริเวณดวงตาจากค่าข้อมูลสี (Chrominance) และขั้นตอนที่สองเป็นการตรวจหาตำแหน่งบริเวณดวงตาจากค่าความสว่าง (Luminance) จากภาพสี YCbCr โดยตำแหน่งบริเวณดวงตาจากค่าข้อมูลสี (Chrominance) ได้จากการสังเกตค่าสูงสุดของค่าข้อมูลสี Cb และค่าต่ำสุด Cr ซึ่งหาได้จากสมการ ดังนี้

$$EMC = 13\{Cb_2 + Cr\sim_2 + (CbCr)\} \quad (2.8)$$

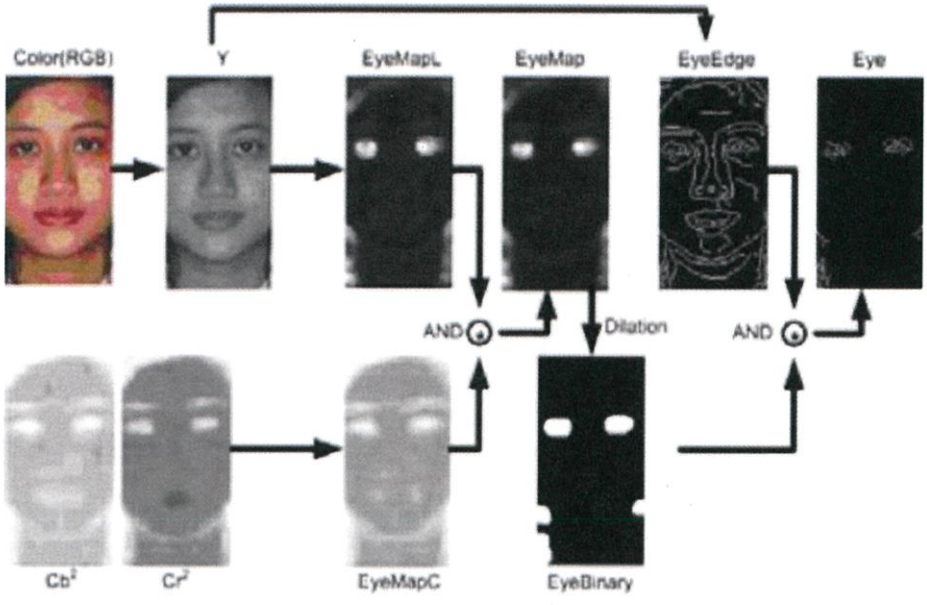
สมการ 2.8 คือการหา EyeMapC ดังรูป 2.3 โดย (C2b),(C~r 2) และ (Cb/Cr) โดยทาการปรับค่าทั้งหมดให้อยู่ ในช่วง (0-255)และ C~r คือ ค่าลบของ (255- Cr) เนื่องจาก ตำแหน่งบริเวณดวงตา จะมีลักษณะของจุดภาพที่สว่างและมืด อยู่ในค่าความสว่าง (Luminance) ซึ่งเป็นคุณสมบัติของตัว ดาเนินการมอร์โฟโลยีค่าระดับเทาโดยการไดเลชันและอีโรชัน ซึ่งในงานวิจัยจะใช้ การไดเลชัน และ อีโรชันค่าระดับเทาร่วมกับ 8 ส่วนประกอบโครงร่างแบบ hemispheric ซึ่งตำแหน่งบริเวณดวงตาที่ตรวจพบจากค่าความสว่าง หาได้จากสมการดังนี้

$$EML = Y\,x,y \oplus g\sigma(x,y)Y\,x,y\,\Theta g\sigma\,x,y\, + 1 \qquad (2.9)$$

สมการ 2.9 คือการหา EyeMapL ดังรูป 2.3 เมื่อตัวดาเนินการไดเลชันและอีโรชัน ค่าระดับเทาอยู่บน ฟังก์ชันพื้นฐาน f:F $\subset$ R2 -> Rโดยใช้ส่วนประกอบโครงร่างซึ่งสามารถเขียนเป็นฟังก์ชันได้ดังนี้ g:G $\subset$ R2 -> R เพื่อทาให้ภาพตำแหน่งบริเวณดวงตาที่ตรวจพบจากค่าข้อมูลสีมี คุณภาพดีขึ้น จะน้าภาพตำแหน่งบริเวณดวงตาที่ตรวจพบ จาก ค่าข้อมูลสีไปทาการประมวลผลภาพด้วยฮิสโตรแกรมอีควอไลเซชัน เพื่อให้ระดับความเข้มของแสงภาพมีการกระจายอย่างสม่ำเสมอ แล้วน้าภาพที่ได้จากการประมวลผลไป AND เข้ากับ ภาพตำแหน่งบริเวณดวงตาที่ตรวจพบจากค่าความสว่าง ซึ่งหาได้จากสมการดังนี้

$$EMAP = (EMC)AND(EML) \qquad (2.10)$$

สมการ 2.10 คือกระบวนการทำ EyeMap ดังรูป 2.3 ซึ่งภาพผลลัพธ์ที่ได้จากการตรวจหาตาแหน่งบริเวณดวงตา เรายังสามารถทาการปรับค่าไดเลชัน เพื่อเพิ่มความสว่างให้กับ ตำแหน่งบริเวณดวงตา รวมถึงการปรับปรุงภาพโดยใช้เทคนิค การกำหนดค่าเทรสโชล และค่าไบนารี ที่เหมาะสมให้กับภาพ จะทาให้ภาพผลลัพธ์มีประสิทธิภาพมากยิ่งขึ้นเมื่อทากระบวนการดังกล่าวแล้วจะได้ผลลัพธ์คือภาพ EyeBinary ดังรูป 2.3 จากนั้นนาภาพ EyeEdge และ EyeBinary มาทำกระบวนการ AND ซึ่งจะได้ผลลัพธ์คือขอบของดวงตาเท่านั้น ซึ่งกระบวนการตรวจหาตำแหน่งดวงตาจะแสดงดังรูป 2.3



รูปที่ 2.3 การตรวจหาดวงตา

2.2 OpenCV

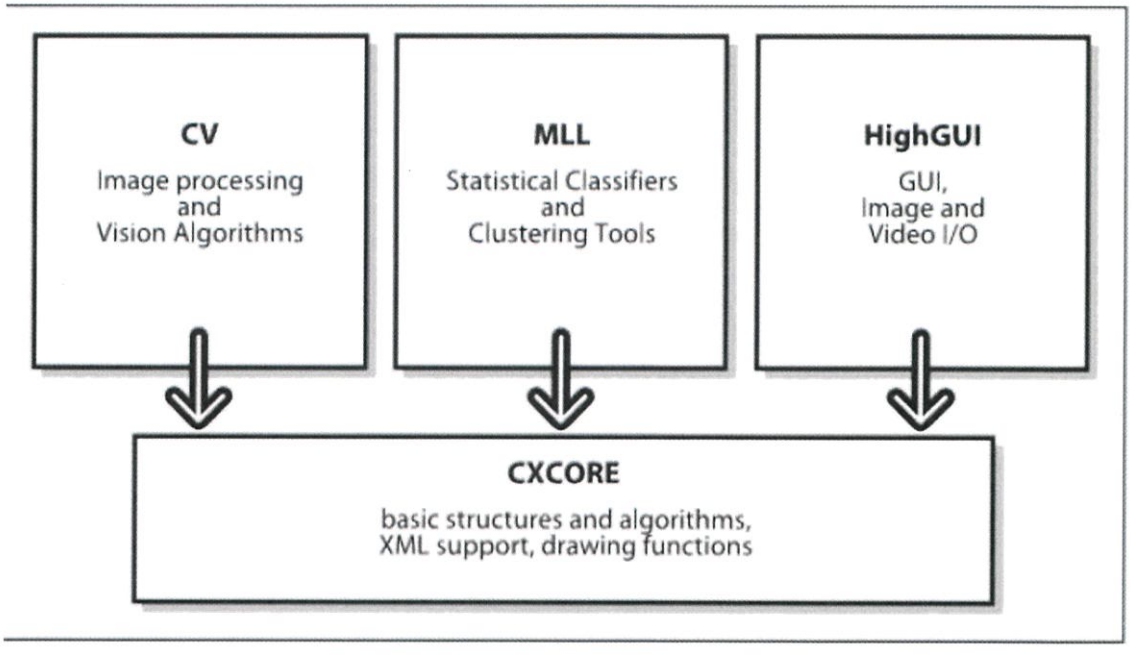
OpenCV ย่อมาจาก Open Source Computer Vision ซึ่ง OpenCV เป็นไลบรารีแบบ ครอสแพลตฟอร์ม (Cross-platform) โดยสามารถรันบนระบบปฏิบัติการลินุกซ์ (Linux), วินโดวส์ (Windows) และแมคโอเอสเอ็กซ์ (Mac OS X) OpenCV นั้นถูกออกแบบมา สาหรับการคำนวณที่มีประสิทธิภาพอย่างยิ่งสาหรับแอพพลิเคชั่นที่ทางานแบบเรียลไทม์ OpenCV นี้ถูกเขียนขึ้นโดยภาษาซีที่ถูกออฟติไมซ์แล้วและสามารถใช้ข้อได้เปรียบจากโปรเซสเซอร์แบบมัลติคอร์

จุดประสงค์หนึ่งของ OpenCV ก็คือการทำให้โครงสร้างพื้นฐานของคอมพิวเตอร์วิชัน นั้นง่ายต่อการใช้งาน ซึ่งช่วยให้ผู้ใช้สามารถสร้างวิชันแอพพลิเคชั่น (Vision application) ที่ค่อนข้างซับซ้อนได้อย่างรวดเร็ว OpenCV ไลบรารีนั้นประกอบไปด้วยฟังก์ชันมากกว่า 500 ฟังก์ชัน ซึ่งครอบคลุมหลายๆ หัวข้อของวิชั่น รวมถึงการตรวจสอบสินค้าจากโรงงาน การประมวลผลภาพในทางการแพทย์ ระบบรักษาความปลอดภัย ส่วนติดต่อกับผู้ใช้ การปรับเทียบกล้องสเตอริโอวิชัน และการใช้งานกับหุ่นยนต์ และเนื่องจากคอมพิวเตอร์วิชั่น และแมตชีนเลินนิง (Machine Learning) นั้นมักจะต้องทางานไปด้วยกันบ่อยครั้ง OpenCV จึงมีแมตชีนเลินนิงไลบรารี่รวมอยู่ด้วย ซึ่งไลบรารีย่อยนี้ เน้นไปที่การทำแพทเทิน

เรคคอกนิชัน (Pattern recognition) แบบสถิติ และการทำคลัสเตอริง (Clustering) ซึ่งแมตชีนเลินนิงไลบรารีนั้นสามารถใช้ได้เป็นอย่างดีกับงานที่เกี่ยวกับวิชัน ซึ่งสิ่งนี้คือหัวใจหลักของ OpenCV และโดยทั่วไปแล้วเพียงพอที่จะใช้ในปัญหาของแมตชีนเลินนิงต่างๆได้

2.2.1 โครงสร้างของ OpenCV และองค์ประกอบ

OpenCV นั้นมีโครงสร้างอยู่ห้าส่วนประกอบหลัก สี่ในห้าแสดงดังรูปต่อไปนี้โดยส่วนซีวี (CV) ประกอบไปด้วยการประมวลผลภาพพื้นฐาน และคอมพิวเตอร์วิชั่นอัลกอริทึมระดับสูงกว่า และแอ็มแอล (ML) คือแมตชีนเลินนิ่งไลบรารี่ซึ่งรวมถึงสแตทิสทิคอลคลาสสิไฟลเออร์ (Statistical classifiers) และคลัสเตอริ่งทูลในส่วนของไฮจียูไอ (HighGUI) จะประกอบไปด้วยการทำงานกับอินพุต เอาท์พุทและฟังก์ชันสาหรับการจัดเก็บและโหลดวีดีโอและภาพและ ซีเอ็กคอร์ (CXCore) ประกอบไปด้วยโครงสร้างข้อมูล (Data structures) และคอนเท็น (Content) พื้นฐาน รูป 2.4 แสดงโครงสร้างของ OpenCV



รูปที่ 2.4 โครงสร้างของ OpenCV

2.2.2 ระบบปฏิบัติการและสถาปัตยกรรมที่รองรับ

OpenCV นั้นถูกออกแบบมาให้สามารถใช้งานได้ง่าย เดิมทีนั้น OpenCV ถูกเขียนขึ้นและคอมไพล์ผ่าน Borland C++, Microsoft Visual C++ และอินเทลคอมไพล์เลอร์ซึ่งการใช้ภาษาC และ C++ ในการโค้ดเป็นมาตรฐานทำให้ OpenCV สามารถสนับสนุนการข้ามแพลตฟอร์มได้ง่ายขึ้น รูปต่อไปนี้แสดงถึงแฟลตฟอร์มที่ OpenCV สามารถทางานได้ เริ่มจากระบบ IA32 บนวินโดวส์ ตามด้วยลินุกซ์ บนสถาปัตยกรรมเดียวกัน บนแมคโอเอสเอ็กซ์ นั้นมีความสำคัญตั้งแต่แอปเปิล ได้เริ่มใช้โปรเซสเซอร์ของอินเทลตามด้วยสถาปัตยกรรมแบบ 64 บิต และบนซันฮาร์ดแวร์และบนระบบปฏิบัติการอื่นๆ โดยรูป 2.5 แสดงถึงระบบปฏิบัติการและสถาปัตยกรรมที่ OpenCV สนับสนุน

	IA32	EM64T	IA64	Other(PCC, Sparc)
Windows	✓ (w. MSVC6, .NET2005+OMP, ICC, GCC, BCC)	✓ (w. IPP; MSVC6+PSDK.NET2005+OMP, PSDK)	± (w. IPP; PSDK, some tests fail)	N/A
Linux	✓ (w. IPP; GCC, BCC)	✓ (w. IPP; GCC, BCC)	✓ (GCC, ICC)	✘
MacOSX	✓ (w. IPP; GCC, native APIs)	? (not tested)	N/A	✓ (iMac G5, GCC, native APIs)
Other (BSD, Solaris...)	✘	✘	✘	Reported to build on UltraSparc Solaris

รูปที่ 2.5 ระบบปฏิบัติการและสถาปัตยกรรมที่ OpenCV สนับสนุน

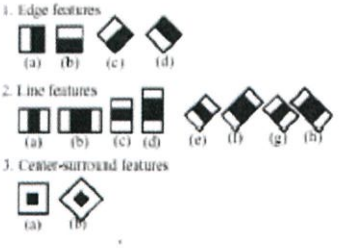
2.3 Haar like-feature

Haar like-Features เป็นวิธีการตรวจจับและตีความวัตถุภายในภาพ โดยใช้การสร้างรูปเหลี่ยม (Feature) โดยที่ภาพนี้แสดงถึงผลต่างระหว่างพื้นที่ส่วนสีขาว และส่วนที่เป็นสีดำ ซึ่งรูปเหลี่ยมที่สร้างขึ้นสามารถเปลี่ยนแปลงขนาด และตำแหน่งได้ ใช้สำหรับการตรวจจับลักษณะบนภาพแบบต่างๆ เช่น เส้นตรงวงกลม เป็นต้น รูปที่ 2.6 แสดงรูปแบบของรูปเหลี่ยมสำหรับการตรวจจับลักษณะแบบต่างๆ

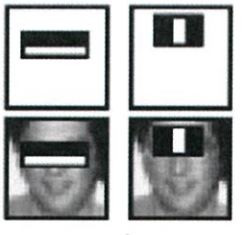
(1) ความสามารถของขอบ

(2) ความสามารถของเส้น

(3) ความสามารถของบริเวณที่ล้อมรอบจุดตรงกลาง



รูปที่ 2.6 รูปแบบของรูปเหลี่ยมสำหรับการตรวจจับลักษณะแบบต่างๆ

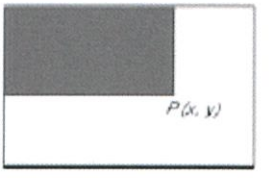


รูปที่ 2.7 ตัวอย่างการใช้รูปเหลี่ยมตรวจจับลักษณะต่างๆ

การคำนวณค่าของรูปเหลี่ยม (feature) นั้น ใช้หลักการคำนวณแบบ Integral image คือผลรวมของค่าในทุกๆ จุดภาพ ที่ตำแหน่ง (x, y) ใดๆ ดังสมการที่ 2.11 รูปที่ 2.8 แสดงการคำนวณแบบ Integral image

$$P\ x, y = x' \leq x, y' \leq y\ i(x', y')$$

(2.11)



รูปที่ 2.8 การคำนวณแบบ Integral image

ในการทำ Haar like-feature นั้น จำเป็นต้องมีภาพตัวอย่างจำนวนมาก ซึ่งใช้ในการคัดเลือกลักษณะของรูปต้องการตรวจจับและตีความหมาย ซึ่งมีสองลักษณะคือ Positive Image หรือรูปมีวัตถุนั้นๆประกอบอยู่ภายในภาพ และ Negative Image หรือภาพใดๆที่ไม่มีวัตถุที่เราต้องการอยู่ภายในภาพ จากนั้นใช้ขั้นตอนวิธีของ AdaBoost (Adaptive Boost) ซึ่งเป็นกระบวนการหารูปเหลี่ยมที่มีลักษณะใกล้เคียง และแตกต่างกับภาพนา เข้า สา หรับการจัดประเภทของภาพ โดยการถ่วงน้ำหนักให้ส่วนต่างๆ ภายในภาพ บนภาพ Positive และภาพ Negative เพื่อใช้หาลักษณะของวัตถุที่ “ใช่” และ “ไม่ใช่” ในลักษณะต่างๆ

2.4 Region of interest (ROI)

Region of interest (ROI) คือบริเวณที่เราสนใจซึ่งอาจจะเป็นบริเวณใดภายในภาพก็ได้ โดยการตีกรอบล้อมรอบบริเวณที่สนใจ ด้วยวงกลม กรอบสี่เหลี่ยม หรือกรอบรูปเหลี่ยมใดๆ เพื่อนำภาพเฉพาะส่วนดังกล่าวมาประมวลผลหรือเปลี่ยนแปลงภาพตามต้องการ โดยไม่มีผลกระทบกับส่วนอื่นๆซึ่งภายในหนึ่งภาพ สามารถกำหนดได้หลายๆ บริเวณที่สนใจ การตัดเฉพาะส่วนของภาพเพื่อนามาประมวลผลทาให้การประมวลผลภาพเร็วขึ้นเนื่องจาก Resolution ของภาพที่นามาประมวลผลมีขนาดเล็กลง ซึ่งในการประมวลผลภาพขนาดใหญ่จะทาให้เห็นความแตกต่างของประสิทธิภาพในการประมวลผลมาก รูปที่ 2.9 แสดงตัวอย่างการกำหนดบริเวณที่สนใจ

รูปที่ 2.9 ตัวอย่างการกำหนดบริเวณที่สนใจ

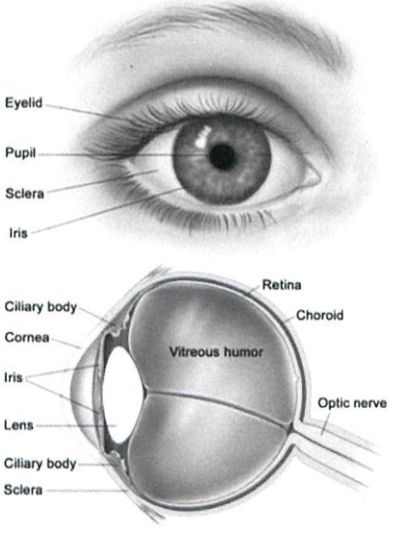
2.5 Circle Detection

Circle Detection คือการตรวจจับวงกลมในรูปภาพเพื่อใช้ในการตรวจหาตาดำ โดยจะใช้ภาพ Grey Scale เพื่อใช้ในการตรวจจับวงกลมโดยการตรวจจับวงกลมจะดูความลึกของสีซึ่งลักษณะของนัยน์ตาอาจมีสีดาหรือสีฟ้าซึ่งเมื่อแปลงเป็น Grey Scale แล้วจะทำให้มีระดับความลึกของสีอยู่ในระดับที่แตกต่างกันกับของตาขาวทำให้สามารถใช้การตรวจหาวงกลมเพื่อหานัยน์ตาได้ โดย EmguCV มี Library ที่สำเร็จรูปในการตรวจหาวงกลมซึ่งสามารถกำหนด Parameter 4 ค่าคือ ค่า Threshold ที่ใช้ในการทำ Canny Edge Detection ค่า Threshold ที่ใช้ในการทำ Accumulator ค่าอัตราส่วนของภาพที่จะทำการตรวจหาวงกลม ค่าระยะของรัศมีในการตรวจหามากสุด และน้อยสุดของวงกลม และค่าระยะห่างของวงกลมที่ตรวจสอบพบกับวงกลมถัดไป ซึ่งในการตรวจสอบนัยน์ตาจะต้องกำหนดค่าของรัศมีวงกลมให้พอเหมาะจึงจะสามารถตรวจหานัยน์ตาได้อย่างแม่นยำ

2.6 การติดตามลักษณะเด่นบนใบหน้า

การติดตามดวงตา (Eye Tracking)

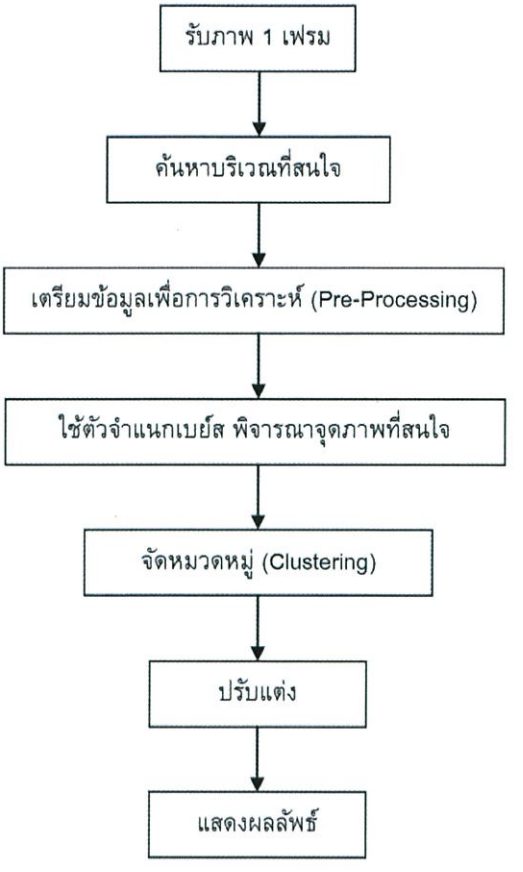
การมองเห็นอะไรได้ดีแถะชัดเจนนั้น ภาพที่มองจะต้องสามารถเดินทางผ่านเข้าไปในถูกตา โดยส่วนประกอบต่างๆ ของตา คือ กระจกตา (Cornea) และเลนส์แก้วตา (Lens) ไปตกที่จอประสาทตา (Retina) ซึ่งเป็นผนังชั้นในของลูกตา ที่ประกอบไปด้วยเซลล์ประสาทตาจำนวนมาก จากนั้นจะส่งสัญญาณภาพที่ได้ผ่านไปทางเส้นประสาทตา (Optic nerve) สู่สมอง เพื่อแปลงสัญญาณเป็นภาพที่มองเห็น ทำให้มนุษย์สามารถรับรู้ได้ว่าภาพนั้น เป็นภาพอะไร และสามารถทำกิจกรรมต่างๆ ส่วนม่านตา (Iris) เป็นที่ส่วนเป็นสีของนัยน์ตา ซึ่งอาจมีสีดำ สีน้ำตาล สีฟ้า หรือสีอื่นๆ ตามเชื้อชาติ โดยม่านตาทำหน้าที่ควบคุมปริมาณแสงให้พอเหมาะที่จะผ่านไปสู่เลนส์ตา ม่านตาสามารถเปิดกว้างมากน้อยตามความสว่างของแสง ถ้าแสงสว่างมากม่านตาจะเปิดน้อย แสงสว่างน้อยม่านตาจะเปิดกว้าง และรูม่านตา (Pupil) เป็นสีดำอยู่ตรงกลางนัยน์ตา ทำหน้าที่เป็นช่องให้แสงผ่านไปสู่เลนส์ตา



รูปที่ 2.10 ส่วนประกอบของดวงตา
ที่มา http://health-tune-ups.com/?p=195

เนื่องจากม่านตาคือส่วนที่เป็นสีของดวงตาที่เห็นอย่างชัดเจน ซึ่งทำให้ตรวจจับได้ง่าย จากงานวิจัยเรื่องนี้การติดตามดวงตาจริง ประกอบไปด้วย 4 ขั้นตอนหลัก คือ ขั้นตอนการเตรียมข้อมูล (Pre-processing) ขั้นตอนการจำแนก (Classification) ขั้นตอนการจัดหมวดหมู่ (Clustering) และขั้นตอนการปรับแต่ง (Post-processing) แสดงดังรูปที่ 2.1

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 2.11 ขั้นตอนวิธีการติดตามดวงตา

จากรูปที่ 2.11 เริ่มทำการรับภาพหนึ่งเข้าเฟรมมา และเริ่มค้นหาบริเวณที่สนใจ นั่นคือบริเวณดวงตา โดยจะหาตำแหน่งของดวงตาโดยประมาณ จากนั้นจะเริ่มเตรียมข้อมูลเพื่อการวิเคราะห์โดยพิจารณาจากสีของม่านตา ถัดมาจะใช้ตัวจำแนกเบย์ส (Bayesian Classifier) พิจารณาจุดภาพที่สนใจนั้นเป็นบริเวณของดวงตาหรือไม่ จากนั้นจะเริ่มจัดหมวดหมู่ (Clustering) ว่าบริเวณที่สนใจนั้นเป็นบริเวณของดวงตา หรือเป็นบริเวณอื่นๆ เพราะเป็นไปได้ว่าอาจมีบริเวณจุดภาพอื่นๆ ที่มีสีเช่นเดียวกับบริเวณสีของดวงตา เช่น เส้นผม เป็นต้น จากนั้นทำการปรับแต่ง โดยกำจัดบริเวณอื่นๆ ที่ไม่ใช่บริเวณของดวงตาออก และสุดท้ายแสดงผลลัพธ์โดยตีกรอบบริเวณดวงตาออกมา

149308

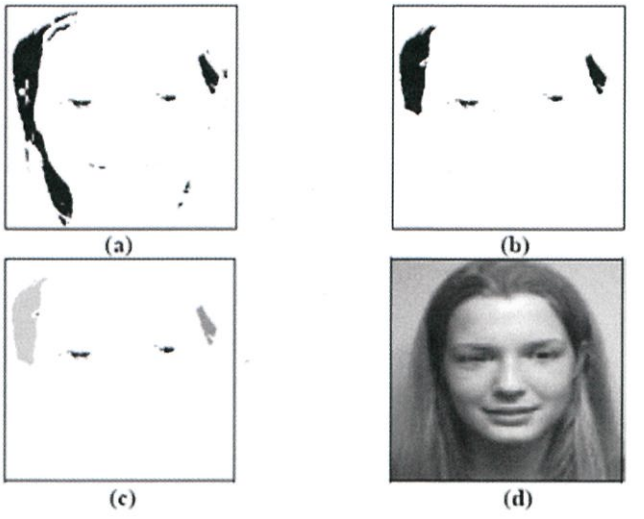


Figure 3: (a) After Pre-Processing. (b) After Bayesian Classifier. (c) Output of Clustering Algorithm. (d) Final Output.

รูปที่ 2.12 ตัวอย่างขั้นตอนวิธีการติดตามดวงตา

2.7 Gaussian Blur

Gaussian Blur คือการเบลอภาพโดยใช้ฟังก์ชันของเกาส์ ซึ่งใช้อย่างแพร่หลายในการทำงานทางคอมพิวเตอร์กราฟิก โดยทั่วไปนั้นจะใช้ทำการลดสิ่งรบกวนบนภาพและลดรายละเอียดของภาพลง ซึ่งจะทำให้ภาพดูนุ่มนวลขึ้น ดังรูปที่ 2.13 ซึ่งสมการฟังก์ชันของเกาส์แบบหนึ่งมิติมีรูปแบบดังนี้

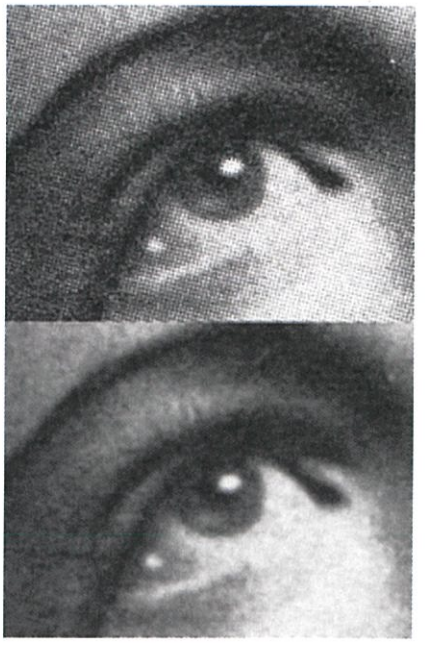
$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \qquad (2.12)$$

โดยค่า μ คือค่าเฉลี่ยของ x แต่เนื่องจากจุดศูนย์กลางคือจุดกำเนิดของภาพเมื่อมีการคำนวณหาเฉลี่ย ดังนั้น μ จึงมีค่าเท่ากับ 0 ดังสมการที่ 2.13

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2} \qquad (2.13)$$

และจากสมการที่ 2.13 ทำให้สามารถเขียนสมการในรูปแบบสองมิติได้ดังนี้

$$G(x,y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x^2+y^2)/2\sigma^2} \qquad (2.14)$$



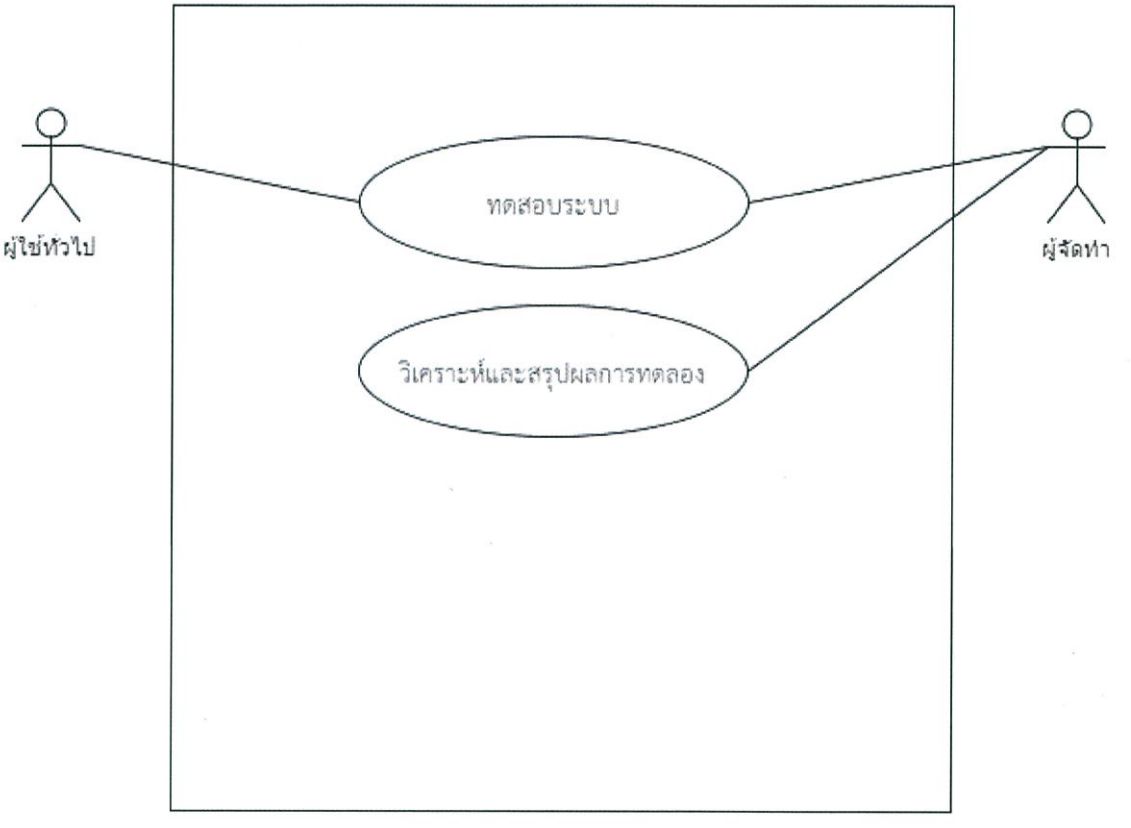
รูปที่ 2.13 ภาพที่ใช้ Gaussian blur

บทที่ 3

การวิเคราะห์ออกแบบและพัฒนาระบบ

3.1 การออกแบบระบบ

3.1.1 Use Case Diagram



รูปที่ 3.1 Use Case Diagram ของระบบ

ตารางที่ 1 แสดง Use Case Description ของการทดสอบระบบ

Use Case ID:	UC_01	
Use Case Name:	ทดสอบระบบ	
Actors:	ผู้จัดทำ, ผู้ใช้งานทั่วไป	
Description:		
Preconditions:	-	
Post conditions:	-	
Flow of Events:	Actor	System
	1.ผู้จัดทำเปิดใช้งานระบบขึ้นมา 3.ผู้ใช้มองไปที่ตำแหน่งที่ 1 4.ผู้จัดทำถ่ายภาพผู้ใช้มองที่ตำแหน่งที่ 1 5.ผู้ใช้มองไปที่ตำแหน่งที่ 2 6.ผู้จัดทำถ่ายภาพผู้ใช้มองที่ตำแหน่งที่ 2 7.ผู้ใช้มองไปที่ตำแหน่งที่ 3 8.ผู้จัดทำถ่ายภาพผู้ใช้มองที่ตำแหน่งที่ 3 9.ผู้ใช้มองไปที่ตำแหน่งที่ 4 10.ผู้จัดทำถ่ายภาพผู้ใช้มองที่ตำแหน่งที่ 4 11.ผู้ใช้มองไปที่ตำแหน่งที่ 5 12.ผู้จัดทำถ่ายภาพผู้ใช้มองที่ตำแหน่งที่ 5 13.ผู้ใช้มองไปที่ตำแหน่งที่ 6 14.ผู้จัดทำถ่ายภาพผู้ใช้มองที่ตำแหน่งที่ 6 15.ผู้ใช้มองไปที่ตำแหน่งที่ 7 16.ผู้จัดทำถ่ายภาพผู้ใช้มองที่ตำแหน่งที่ 7 17.ผู้ใช้มองไปที่ตำแหน่งที่ 8 18.ผู้จัดทำถ่ายภาพผู้ใช้มองที่ตำแหน่งที่ 8 19.ผู้ใช้มองไปที่ตำแหน่งที่ 9 20.ผู้จัดทำถ่ายภาพผู้ใช้มองที่ตำแหน่งที่ 9	2.จับภาพดวงตาของผู้ใช้งาน
Exception conditions:	21.ผู้ใช้งานเลิกใช้งานระบบ	

ตารางที่ 2 แสดง Use Case Description ของการวิเคราะห์และสรุปผลการทดลอง

Use Case ID:	UC_02	
Use Case Name:	วิเคราะห์และสรุปผลการทดลอง	
Actors:	ผู้จัดทำ	
Description:	-	
Preconditions:	ภาพของผู้ใช้งานที่มาทดสอบระบบ	
Post conditions:	-	
Flow of Events:	Actor	System
	1.ผู้จัดทำนำค่าที่ได้จาก output ของโปรแกรมที่ผู้ใช้ทำการทดสอบมาหาร้อยละของ output ที่แสดงตำแน่งการได้ถูกต้อง และตำแหน่งที่ผิดพลาด 2.สรุปผลการทดลอง	
Exception conditions:	-	

3.2 กระบวนการการทำงาน

ในการดำเนินงานของระบบเริ่มต้นระบบจะรับภาพจากกล้องวิดิโอเข้ามา และนำภาพในแต่ละเฟรมมาเข้าสู่กระบวนการทำงานในแต่ละขั้นตอน ซึ่งรายละเอียดการทำงานจะอธิบายดังต่อไปนี้

3.2.1 Face Localization

เป็นส่วนในการค้นหาใบหน้า โดยจะรับภาพมาจากกล้องหน้าของคอมพิวเตอร์หรือเว็บแคม แล้วนำภาพใบหน้าที่ได้จากกล้องมาเข้าสู่กระบวนการค้นหาใบหน้าตามหลักการ haarcascade ซึ่งใช้อัลกอริทึม Haar-like ในการค้นหาใบหน้า หลังจากได้ส่วนของใบหน้าแล้วจะทำการวาดกรอบสี่เหลี่ยมระบุส่วนที่เป็นใบหน้าแล้วทำการแสดงผลออกมา

3.2.2 Eye Detection

เป็นส่วนในการค้นหาดวงตา เริ่มต้นนั้นจะนำภาพใบหน้าที่ได้จากการค้นหาใบหน้ามาทำการเบลอภาพโดยใช้ฟังก์ชัน GaussianBlur ซึ่งใช้อัลกอริทึมของ Gaussian เพื่อทำการลดสิ่งรบกวนและรายละเอียดของภาพลง หลังจากนั้นจะทำการหาตำแหน่งของดวงตาข้างซ้ายและข้างขวา ซึ่งหลังจากได้ส่วนของดวงตาแล้วจะทำการวาดกรอบสี่เหลี่ยมระบุส่วนที่เป็นดวงตาทั้งซ้ายและขวา จากนั้นทำการหาจุดศูนย์กลางของดวงตาและวาดวงกลมระบุส่วนที่เป็นม่านตาแล้วทำการแสดงผลออกมาบนหน้าจอ

3.3 การดำเนินงานในส่วน Face Localization

เป็นส่วนของการค้นหาใบหน้า โดยจะรับภาพจากกล้องวิดีโอเข้ามาวิเคราะห์ระบุตำแหน่งใบหน้า แล้วนำภาพที่ได้มาคำนวณหาใบหน้า โดยใช้อัลกอริธึม Harr-like ในการค้นหา สำหรับฟังก์ชันที่ใช้ในการค้นหาใบหน้า คือ detectMultiScale ซึ่งอยู่ในไลบรารีของ OpenCV และได้มีรูปแบบไฟล์ xml ที่ใช้ในการระบุส่วนของใบหน้าอยู่ด้วยกัน 4 รูปแบบด้วยกัน ได้แก่

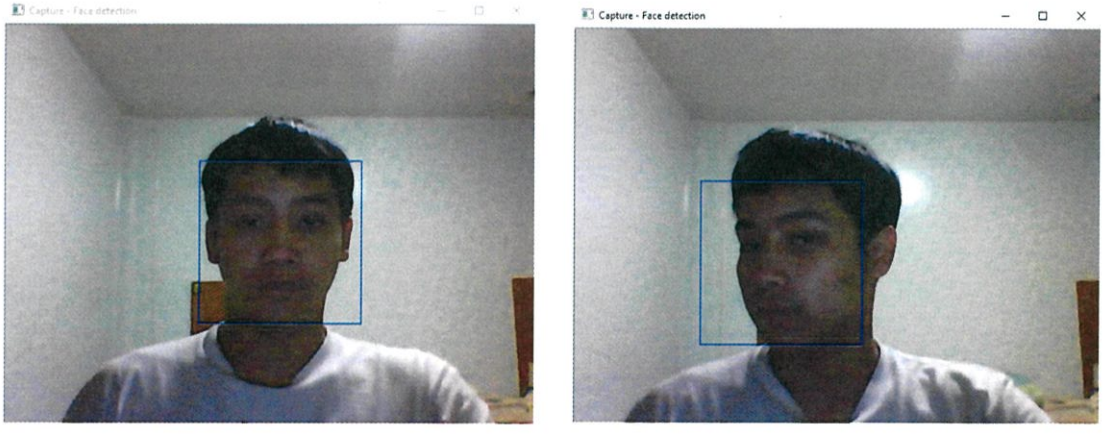
- Haarcascade_frontalface_atl.xml

- Haarcascade_frontalface_atl2.xml

- Haarcascade_frontalface_atl_tree.xml

- Haarcascade_frontalface_atl_defaul.xml

โดยการเขียนโปรแกรมส่วนนี้ จะเลือกรูปแบบไฟล์ xml มาเพียงรูปแบบเดียว นั่นคือแบบ Haarcascade_frontalface_atl2.xml เนื่องจากเป็นรูปแบบ xml ที่ดีที่สุดในการหาใบหน้าแนวตรง ซึ่งในแต่ละรูปนั้นสามารถนำมาระบุตำแหน่งใบหน้าได้เหมือนกัน แต่อาจจะค้นหาได้ในแต่ละองศาที่แตกต่างกันไปบ้าง เมื่อค้นหาใบหน้าเจอจะทำการเลือกบริเวณของใบหน้าโดยการวาดกรอบสี่เหลี่ยมในส่วนของใบหน้านั้น



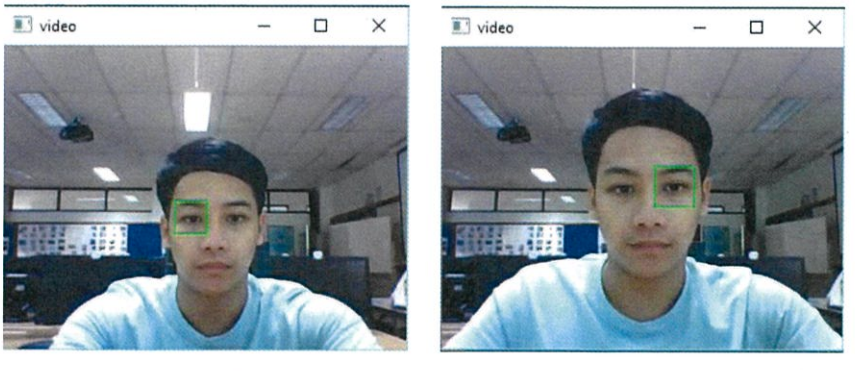
รูปที่ 3.2 ภาพที่ได้จากการค้นหาส่วนของใบหน้า

รูปที่ 3.3 ภาพที่ค้นหาใบหน้าไม่พบ

จากการรันโปรแกรม จะได้ว่า โปรแกรมจะทำการวาดกรอบส่วนของใบหน้า เมื่อมีการเจอใบหน้านั้นดังรูป 3.2 แต่จะค้นหาใบหน้าไม่พบหากมีการบังใบหน้าบางส่วนไว้ ดังรูปที่ 3.3

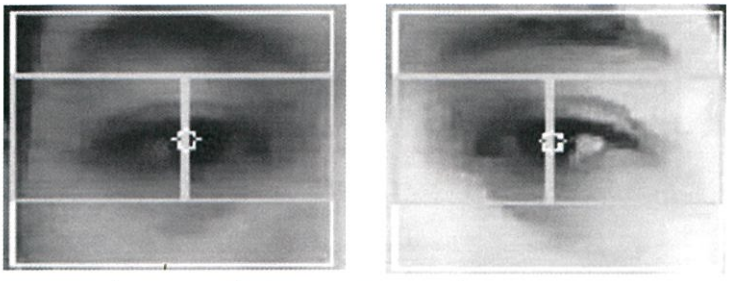
3.4 การดำเนินงานในส่วน Eye Detection

กระบวนการตรวจหาดวงตาโดยใช้ EmguCV Library โดยใช้ Haar Eye Detection ซึ่งใน EmguCV มี Function ที่สำเร็จรูปในการหาตำแหน่งของดวงตา โดยการหาตำแหน่งดวงตาเริ่มจากการนาภาพที่ถ่ายจากกล้อง Webcam มาทาการ Preprocess โดยการแปลงภาพเป็น Grey Scale และทำ Equalizehist เพื่อทำให้ระดับสีเทาของภาพมีความเข้มขึ้น จากนั้นจึงนำภาพที่ทำ Preprocess แล้วมาเป็นภาพ Input ของกระบวนการ Haar Eye Detection ซึ่งสามารถเลือกการตรวจหาภาพได้หลายแบบ เช่น Canny Pruning และ Biggest Object ซึ่งจะตรวจหาวัตถุที่ใหญ่ที่สุดผลลัพธ์ที่ได้จะเป็นวัตถุเพียงรูปเดียวเท่านั้น เป็นต้น เมื่อทำ Haar Eye Detection แล้วจะได้ตำแหน่งพิกัดมุมซ้ายบนของดวงตาที่ตรวจสอบพบและขนาดของสี่เหลี่ยมที่ตรวจสอบพบ รูป 3.4 แสดงการประมวลผลภาพเพื่อตรวจหาดวงตาโดยใช้ EmguCV



รูปที่ 3.4 ภาพที่ได้จากการค้นหาส่วนของดวงตา

หลังจากที่ทำการค้นหากรอบดวงตาจากใบหน้าระบบจะทำการระบุตำแหน่งแกน X,Y ของดวงตาทั้ง 2 ข้างจากนั้นจะทำการหาตำแหน่ง X,Y ที่เป็นจุดศูนย์กลางของดวงจาทั้ง 2 ข้าง เพื่อทำการวาดจุดวงกลมติดตามดวงตา โดยใช้ทฤษฎี The Gradient Algorithm ในการหาตำแหน่งของดวงตาทั้ง 2 ข้าง และแสดงผลออกทางหน้าจอตามรูปที่ 3.5



รูปที่ 3.5 ภาพที่ได้จากการค้นหาจุดกึ่งกลางดวงตา

บทที่ 4

การทดลองและผลการทดลอง

ในบทนี้จะกล่าวถึงการทดสอบระบบการควบคุมกล้องเว็บแคมด้วยการจับการเคลื่อนไหวของใบหน้าและดวงตา ซึ่งเราจะทำการทดลองด้วยการให้ผู้ใช้งานมองไปยังหน้าจอของโน้ตบุ๊ต โดยการให้ผู้ทดสอบจำลองหน้าจอโน้ตบุ๊คเป็นตารางขนาด 3x3 ทำการทดสอบทั้ง 9 จุด ผลที่ออกมาคือภาพทั้งหมด 9 ภาพต่อ 1 ผู้ทดสอบ เพื่อทดสอบการติดตามดวงตาของเคอร์เซอร์ในระบบ หลังจากนั้นทำการวัดประสิทธิภาพของระบบ ซึ่งสามารถอธิบายผลของการดำเนินงานในแต่ละขั้นตอนได้ดังนี้

4.1 การทดสอบ

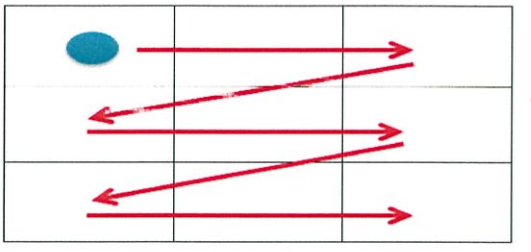
4.1.1 ผู้ทดสองจำลองตาราง

การทดสอบการตรวจจับการเคลื่อนไหวของดวงตาด้วยเคอร์เซอร์ติดตามดวงตา โดยให้ผู้ทดสอบจำลองหน้าจอโน้ตบุ๊คเป็นตาราง 3x3 ดังนี้

1	2	3
4	5	6
7	8	9

รูปที่ 4.1 ตารางจำลองหน้าจอโน้ตบุ๊ค

ขั้นตอนการเก็บข้อมูลเพื่อใช้สำหรับทดสอบระบบ ทีมพัฒนาให้ผู้ทดลองจำลองหน้าจอโน้ตบุ๊คเป็นตารางตามรูปที่ 4.1 จากนั้นให้ผู้ทดลองมองไปยังตำแหน่งที่เราทำการกำหนดไว้โดยเริ่มที่ตำแหนหมายเลข 1 ไปถึงหมายเลข 9 ทิศทางของการมอง โดยแสดงในรูปที่ 4.2

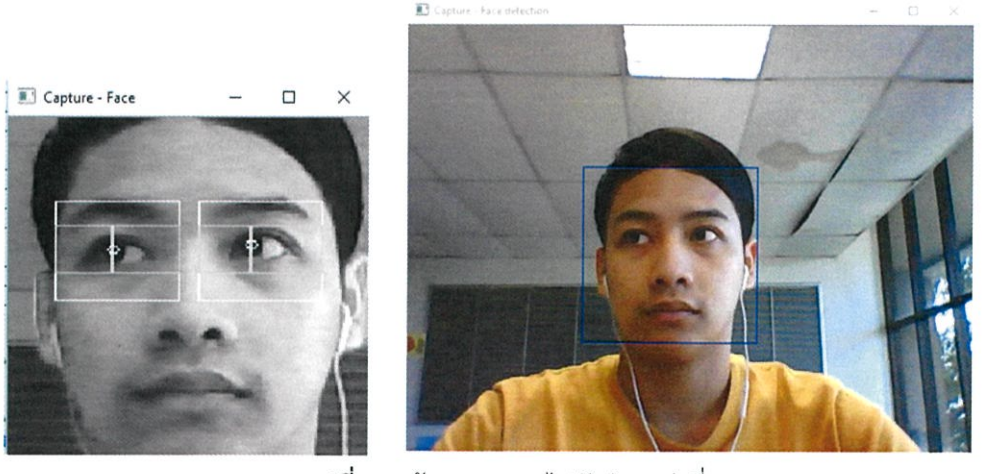


รูปที่ 4.2 ทิศทางในการมองบนหน้าจอโน้ตบุ๊ค

4.2 ผลการทดสอบ

4.2.1 ตำแหน่งที่ 1

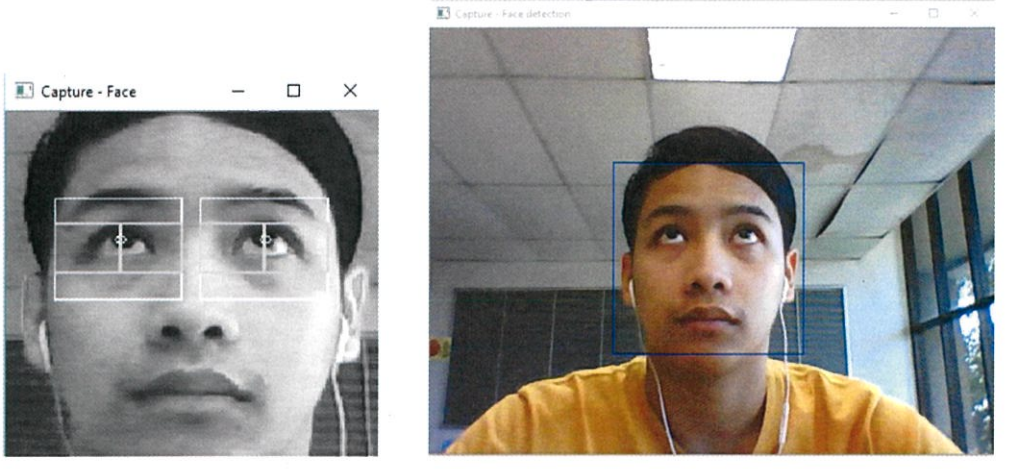
ให้ผู้ทดสอบมองไปยังตำแหน่งที่ 1 ตามรูปที่ 4.1 และตามทิศทางในการมองจากรูปที่ 4.2 ค้างไว้ ทำการถ่ายภาพเพื่อเก็บข้อมูลจะได้ข้อมูลตามรูปที่ 4.3



รูปที่ 4.3 ผู้ทดสอบมองไปยังตำแหน่งที่ 1

4.2.2 ตำแหน่งที่ 2

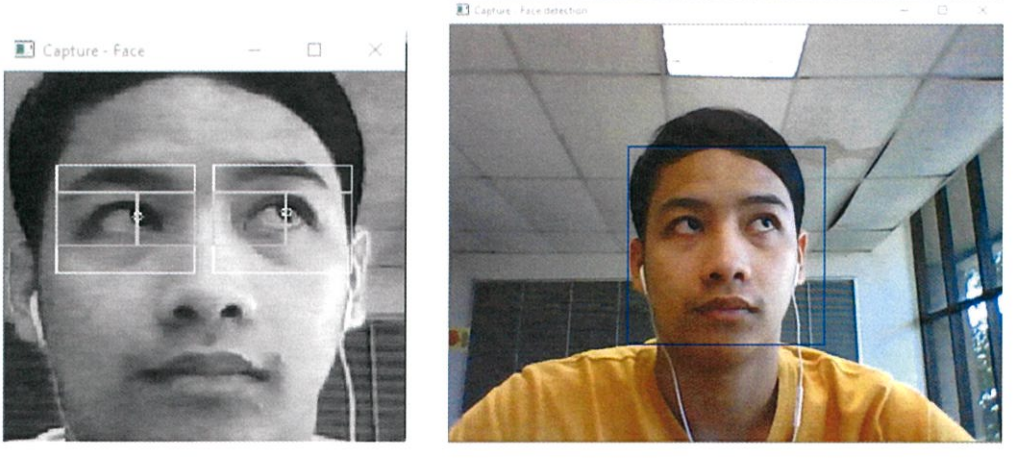
ให้ผู้ทดสอบมองไปยังตำแหน่งที่ 2 ตามรูปที่ 4.1 และตามทิศทางในการมองจากรูปที่ 4.2 ค้างไว้ ทำการถ่ายภาพเพื่อเก็บข้อมูลจะได้ข้อมูลตามรูปที่ 4.4



รูปที่ 4.4 ผู้ทดสอบมองไปยังตำแหน่งที่ 2

4.2.3 ตำแหน่งที่ 3

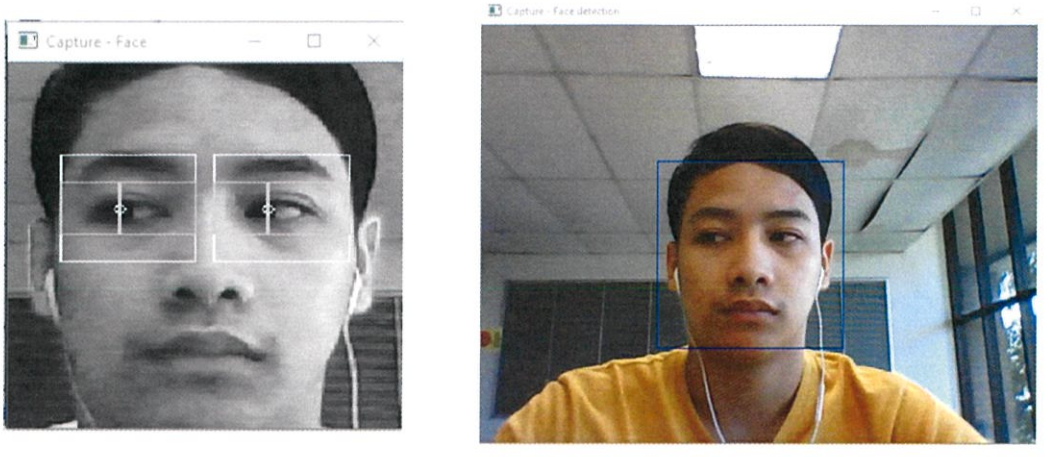
ให้ผู้ทดสอบมองไปยังตำแหน่งที่ 3 ตามรูปที่ 4.1 และตามทิศทางในการมองจากรูปที่ 4.2 ค้างไว้ ทำการถ่ายภาพเพื่อเก็บข้อมูลจะได้ข้อมูลตามรูปที่ 4.5



รูปที่ 4.5 ผู้ทดสอบมองไปยังตำแหน่งที่ 3

4.2.4 ตำแหน่งที่ 4

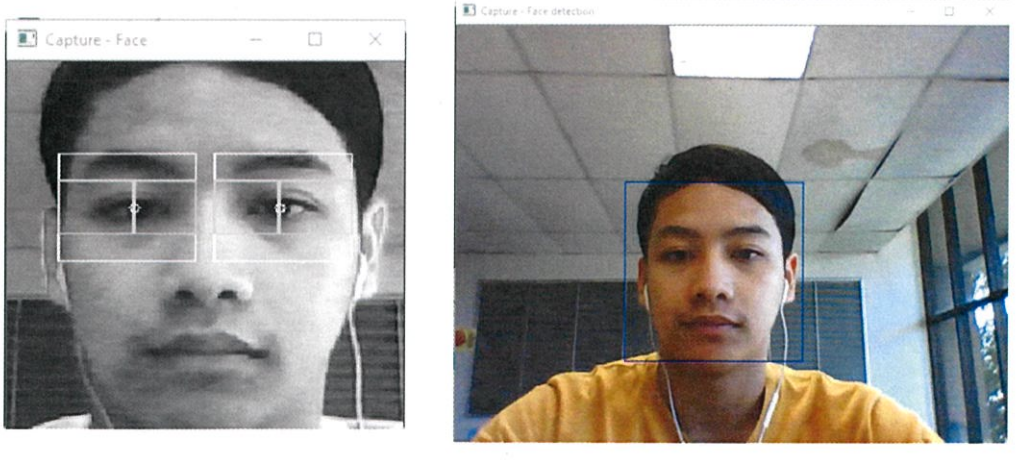
ให้ผู้ทดสอบมองไปยังตำแหน่งที่ 4 ตามรูปที่ 4.1 และตามทิศทางในการมองจากรูปที่ 4.2 ค้างไว้ ทำการถ่ายภาพเพื่อเก็บข้อมูลจะได้ข้อมูลตามรูปที่ 4.6



รูปที่ 4.6 ผู้ทดสอบมองไปยังตำแหน่งที่ 4

4.2.5 ตำแหน่งที่ 5

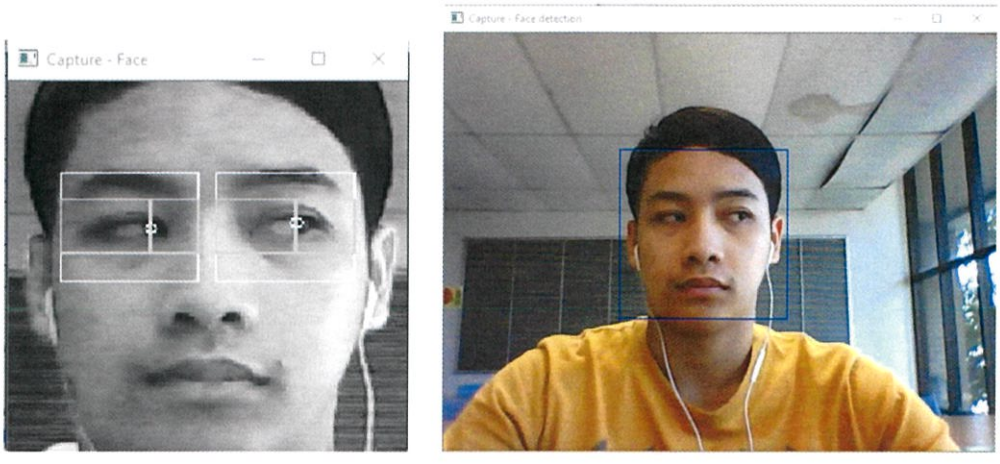
ให้ผู้ทดสอบมองไปยังตำแหน่งที่ 5 ตามรูปที่ 4.1 และตามทิศทางในการมองจากรูปที่ 4.2 ค้างไว้ ทำการถ่ายภาพเพื่อเก็บข้อมูลจะได้ข้อมูลตามรูปที่ 4.7



รูปที่ 4.7 ผู้ทดสอบมองไปยังตำแหน่งที่ 5

4.2.6 ตำแหน่งที่ 6

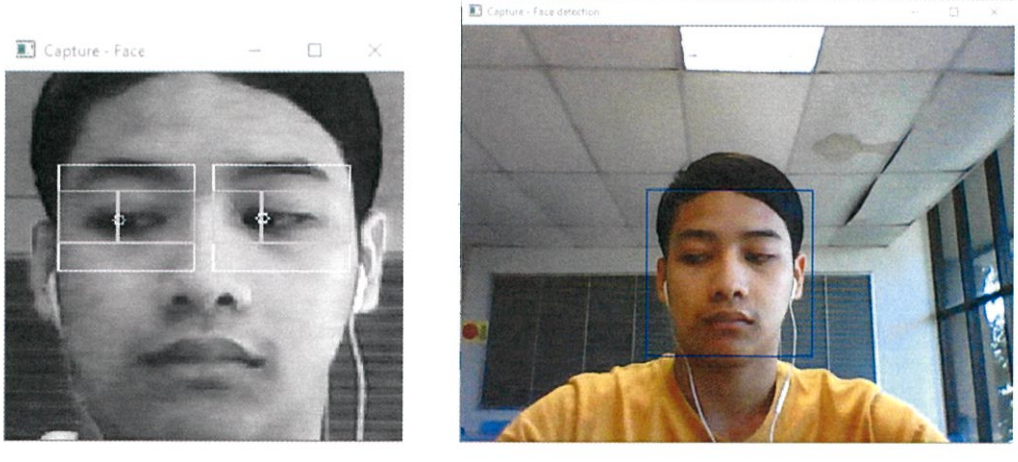
ให้ผู้ทดสอบมองไปยังตำแหน่งที่ 6 ตามรูปที่ 4.1 และตามทิศทางในการมองจากรูปที่ 4.2 ค้างไว้ ทำการถ่ายภาพเพื่อเก็บข้อมูลจะได้ข้อมูลตามรูปที่ 4.8



รูปที่ 4.8 ผู้ทดสอบมองไปยังตำแหน่งที่ 6

4.2.7 ตำแหน่งที่ 7

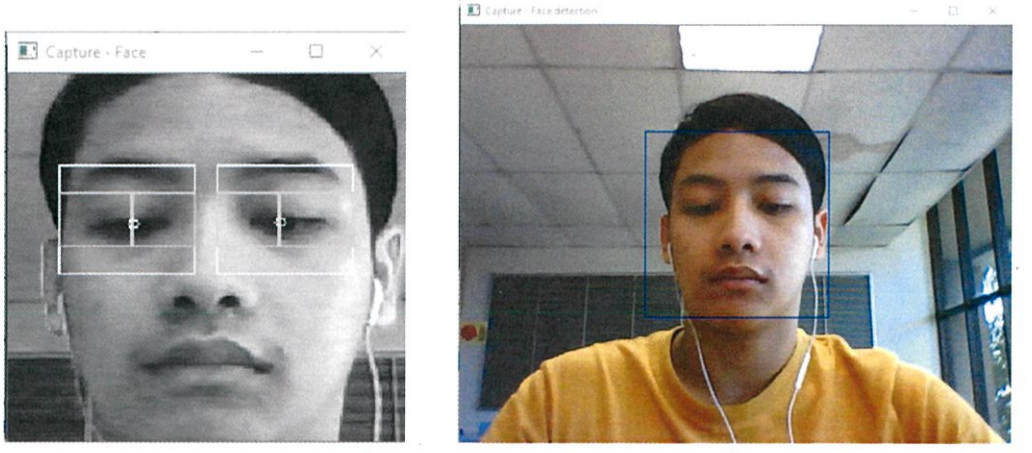
ให้ผู้ทดสอบมองไปยังตำแหน่งที่ 7 ตามรูปที่ 4.1 และตามทิศทางในการมองจากรูปที่ 4.2 ค้างไว้ ทำการถ่ายภาพเพื่อเก็บข้อมูลจะได้ข้อมูลตามรูปที่ 4.9



รูปที่ 4.9 ผู้ทดสอบมองไปยังตำแหน่งที่ 7

4.2.8 ตำแหน่งที่ 8

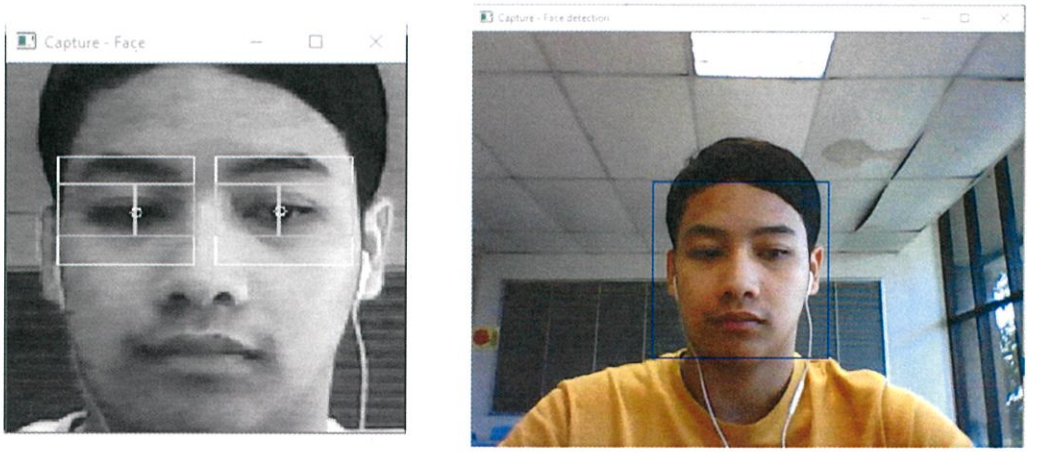
ให้ผู้ทดสอบมองไปยังตำแหน่งที่ 8 ตามรูปที่ 4.1 และตามทิศทางในการมองจากรูปที่ 4.2 ค้างไว้ ทำการถ่ายภาพเพื่อเก็บข้อมูลจะได้ข้อมูลตามรูปที่ 4.10



รูปที่ 4.10 ผู้ทดสอบมองไปยังตำแหน่งที่ 8

4.2.9 ตำแหน่งที่ 9

ให้ผู้ทดสอบมองไปยังตำแหน่งที่ 9 ตามรูปที่ 4.1 และตามทิศทางในการมองจากรูปที่ 4.2 ค้างไว้ ทำการถ่ายภาพเพื่อเก็บข้อมูลจะได้ข้อมูลตามรูปที่ 4.11



รูปที่ 4.11 ผู้ทดสอบมองไปยังตำแหน่งที่ 9

4.3 ผลการทดลอง

โดยการทดลองนั้นทางผู้จัดทำได้ทำการเก็บตำแหน่งจากคนที่มาทดลองในแนวแกน x และแกน y ทั้งหมด 9 ตำแหน่ง โดยจะกำหนดให้ผู้ทดลองมองตำแหน่งละ 10 ครั้ง โดยมีผู้ทดลองจำนวน 4 คนเพื่อนำมาทดสอบดูว่าในแต่ละตำแหน่งมีความแม่นยำในการจับภาพของดวงตามากเท่าไร จากนั้นจำทำผลลัพธ์ที่ได้มาคิดเป็นร้อยละของผลลัพธ์ที่คาดหวังและผลลัพธ์ที่ผิดพลาด

ตารางที่ 3 ตารางการทดสอบการใช้งานระบบตรวจจับดวงตา

ผู้ทดสอบ	ตำแหน่งของการมอง	จำนวนครั้งที่ทำการทดสอบ	จำนวนผลลัพธ์ที่คาดหวัง	จำนวนผลลัพธ์ที่ผิดพลาด
ผู้ทดสอบคนที่ 1	1	10	6	4
	2	10	7	3
	3	10	6	4
	4	10	9	1
	5	10	8	2
	6	10	9	1
	7	10	10	0
	8	10	10	0
	9	10	10	0
ผู้ทดสอบคนที่ 2	1	10	5	5
	2	10	6	4
	3	10	8	2
	4	10	7	3
	5	10	9	1
	6	10	10	0
	7	10	9	1
	8	10	10	0
	9	10	10	0
ผู้ทดสอบคนที่ 3	1	10	4	6
	2	10	3	7
	3	10	6	4
	4	10	8	2
	5	10	10	0
	6	10	7	3

ผู้ทดสอบ	ตำแหน่งของการมอง	จำนวนครั้งที่ทำการทดสอบ	จำนวนผลลัพธ์ที่คาดหวัง	จำนวนผลลัพธ์ที่ผิดพลาด
ผู้ทดสอบคนที่ 3	7	10	9	1
	8	10	10	0
	9	10	9	1
ผู้ทดสอบคนที่ 4	1	10	5	5
	2	10	4	6
	3	10	6	4
	4	10	8	2
	5	10	7	3
	6	10	6	4
	7	10	9	1
	8	10	10	0
	9	10	9	1
รวม		360	279	81

จากการทดลองพบว่า จากทำการทดสอบครบทั้งหมด 360 ครั้งจะเห็นได้ว่าเมื่อผู้ใช้แต่ละคนทำการมองไปยังแต่ละตำแหน่งที่ทำการทดสอบ output ของโปรแกรมจะแสดงตำแหน่งผลลัพธ์ออกมาทางหน้าจอ ซึ่งโปรแกรมสามารถคำนวณได้ว่า จากที่ผู้ใช้ทั้งหมดได้มองมานั้น มีจำนวน 279 ครั้งของการทดลองที่โปรแกรมสามารถจับตำแหน่งของตาได้ถูกต้อง คิดเป็นร้อยละ 77.5 และมีจำนวน 81 ครั้งของการทดลองที่โปรแกรมไม่สามารถจับตำแหน่งของดวงตาได้ถูกต้อง คิดเป็นร้อยละ 22.5 โดยตำแหน่งที่โปรแกรมจับตำแหน่งของดวงตาผิดบ่อย คือ ตำแหน่งที่ 1 ตำแหน่งที่ 2 และตำแหน่งที่ 3 เนื่องจากเป็นผลมาจากการใช้คอมพิวเตอร์โน๊ตบุ๊ค ที่ซึ่งมีกล้องหน้าติดอยู่ด้านบนของจอ จึงทำให้ส่วนในการจับดวงตาที่ตำแหน่งบนหน้าจอนั้นมีโอกาสผิดพลาดมากที่สุด และในส่วนที่มีโอกาสผิดพลาดน้อยที่สุดคือ ตำแหน่งที่ 7 ตำแหน่งที่ 8 และตำแหน่งที่ 9 เนื่องจากเป็นตำแหน่งที่อยู่ด้านล่างของหน้าจอ จึงทำให้กล้องสามารถจับตำแหน่งการมองของผู้ใช้ได้อย่างถูกต้อง และจะมีผิดพลาดเป็นบางครั้ง อาจเป็นผลมาจากการขยับในส่วนศีรษะหรือการมองไปยังตำแหน่งอื่นๆของผู้ทำการทดสอบ

บทที่ 5

สรุปผลการทดลอง

5.1 บทสรุป

ทางคณะผู้วิจัยสามารถพัฒนาขอบเขตที่วางไว้ทั้งหมด ดังนี้

แอพพลิเคชั่นประมวลผลภาพ

(1) สามารถตรวจหาและระบุตำแหน่งของดวงตาของผู้ใช้งานได้

(2) สามารถตรวจหาและระบุตำแหน่งม่านตาของผู้ใช้งานได้

(3) สามารถคำนวณทิศทางการมองของดวงตาได้

(4) สามารถตรวจหาและระบุตำแหน่งของใบหน้าของผู้ใช้งานได้

5.2 ปัญหาอุปสรรคและแนวทางการแก้ไข

(1) แอพพลิเคชั่นไม่สามารถระบุตำแหน่งของม่านตาได้เมื่อมองเห็นพื้นที่ของม่านตาไม่ถึง1/2ของพื้นที่ม่านตาทั้งหมด ซึ่งจะเกิดปัญหาเมื่อมองไปยังสุดของแกน X,Y เช่นการมองไปยังซ้ายสุดหรือขวาสุด ทำให้รูม่านตามีเพียงครึ่งเดียวทำให้ไม่สามารถตรวจจับวงกลมภายในภาพได้ ซึ่งแนวทางการแก้ไขทำโดยการใช้อัลกอริที่มในการตรวจจับครึ่งวงกลมเมื่อตรวจจับวงกลมในภาพไม่พบ

(2) แอพพลิเคชันจะมีการคำนวณค่าทิศทางการมองและเกิดความผิดพลาดได้เมื่อนำไปใช้กับคนที่มีดวงตาแตกต่างกันอาจเกิดความคลาดเคลื่อนเล็กน้อยในการคำนวณค่าทิศทางการมอง ซึ่งมีแนวทางการแก้ไขคือการทำ Machine Learning เพื่อสอนให้ระบบสามารถวิเคราะห์ความแม่นยำของแต่ละคนได้มากขึ้น

(3) อุปกรณ์กล้องโน้ตบุ๊คนั้นส่วนใหญ่ ความละเอียดของกล้องต่ำ และจะมีตำแหน่งอยู่ที่ด้านบนของหน้าจอ จึงทำให้จับภาพทิศทางของม่านตา ในการมองด้านบนมีความคลาดเคลื่อน ซึ่งแนวทางการแก้ไขทำโดยการใช้กล้องเว็บแคมที่มีความละเอียดสูง สามารถต่อพ่วงได้ และเลือกตำแหน่งของกล้องไว้ตำแหน่งจุดกึ่งกลางในระนาบเดียวกับใบหน้าผู้ใช้

(4) ตำแหน่งของผู้ใช้ การเคลื่อนไหวของศีรษะ ระยะห่างระหว่างกล้องกับผู้ใช้ เป็นปัจจัยที่ควบคุมได้ยาก จึงทำให้การวิเคราะห์การเคลื่อนไหวของดวงตาคลาดเคลื่อน ซึ่งมีแนวทางการแก้ไขคือการใช้กล้องที่นำมาติดกับแว่น และให้ผู้ใช้สวมใส่ จะทำให้รับข้อมูลได้แม่นยำขึ้น และนำมาวิเคราะห์จะได้ข้อมูลที่มีประสิทธิภาพมากขึ้น

5.3 แนวทางในการพัฒนาต่อ

(1) พัฒนาส่วนของหน้าแอพพลิเคชันให้ใช้งานง่ายยิ่งขึ้น

(2) นำการเคลื่อนไหวของดวงตาไปปรับใช้กับการควบคุมอย่างอื่นได้ เช่น การเคลื่อนที่ของหุ่นยนต์, การบังคับเป้าปืนด้วยสายตา เป็นต้น

(3) พัฒนาการแสดงผลภาพให้ออกทางอุปกรณ์เสริมแทนการแสดงทางแอพพลิเคชั่น เช่น การแสดงผลภาพบนแท็ปเล็ต เป็นต้น

(4) พัฒนาให้สามารถช่วยเพิ่มคุณภาพชีวิตของผู้พิการทางการเคลื่อนไหวได้ เช่น การวัดรูปด้วยการใช้ดวงตาในการวาด เป็นต้น

(5) พัฒนาให้แอพพลิเคชันสามารถแม่นยำในการตรวจจับดวงตามากยิ่งขึ้น

(6) พัฒนาให้แอพพลิเคชันมีหน้าตาที่น่าใช้งานมากยิ่งขึ้น

(7) ทำให้แอพพลิเคชันสามารถใช้งานร่วมกับระบบอื่นๆ ได้

บรรณานุกรม

ฐิตวัฒน์ เตชจรัสชีวิน และณพวีณา ฤกษ์ปรีดาพงศ์. 2551. "ระบบติดตามลักษณะเด่นบนใบหน้าโดยใช้กล้องเพียงตัวเดียว." วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิชาคณิตศาสตร์ ภาควิชาคณิตศาสตร์, จุฬาลงกรณ์มหาวิทยาลัย.

ธิมาภรณ์ เจริญรบ และอนาวิล ติสุวรรณ. 2550. "Face Recognition." วิทยานิพนธ์วิทยาศาสตร-บัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์, มหาวิทยาลัยสงขลานครินทร์.

รุสลี่ สุทธวีร์กูล และวิไลพร แซ่ลี้. 2554. "การตรวจจับใบหน้าด้วยวิธีการพื้นฐานของการจำลองรูปแบบ Haar-like" วารสารวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ. 6(2) : 34-43.

Kocejko, T., Bujnowski, A. and Wtorek, J. 2008. "Eye Mouse for Disabled." 199-201. in IEEE. **Proceedings Conference on Human System Interactions.** Krakow, Poland.

Lienhart, R. and Maydt, J. 2002. "An Extended Set of Haar-like Features for Rapid Object Detection." 900-903. in IEEE. **Proceedings International Conference on Image Processing.** California : Intel Corporation.

Peng, K., Chen, L., Ruan, S. and Kukharev, G. 2005. "A Robust Algorithm for Eye Detection on Gray Intensity Face without Spectacles." 127- 132. in **Journal of Computer Science and Technology.**

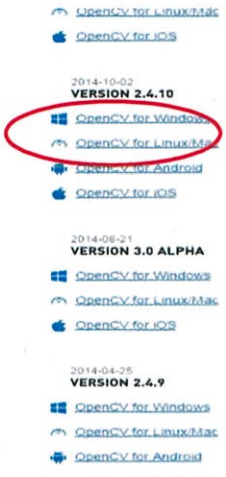
ภาคผนวก

ภาคผนวก ก

วิธีการติดตั้งไลบรารี OpenCV สำหรับใช้งานในภาษาซีพลัสพลัส

วิธีการติดตั้ง OpenCV 2.4.10 เพื่อพัฒนาโปรแกรมด้านคอมพิวเตอร์วิชันและการประมวลผลภาพดิจิทัลบนระบบปฏิบัติการ windows มีขั้นตอนดังนี้

1) ดาวน์โหลด OpenCV จาก www.opencv.org/downloads โดยให้คลิกที่ OpenCV for Windows

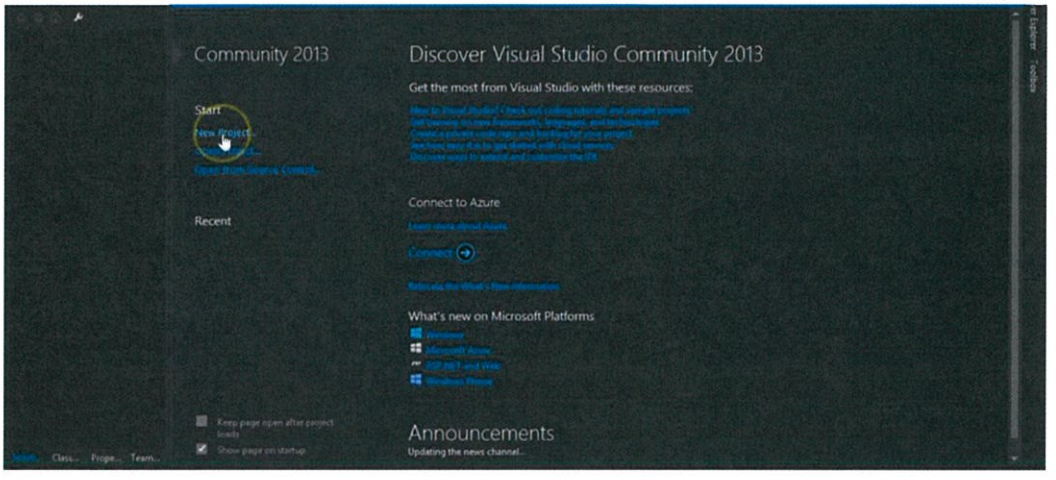


รูปที่ ก.1 หน้าต่างของเว็บไซต์ www.opencv.org/downloads

2) บันทึกไฟล์โดยสร้างโฟลเดอร์ OpenCV2410 ในไดรฟ์ C

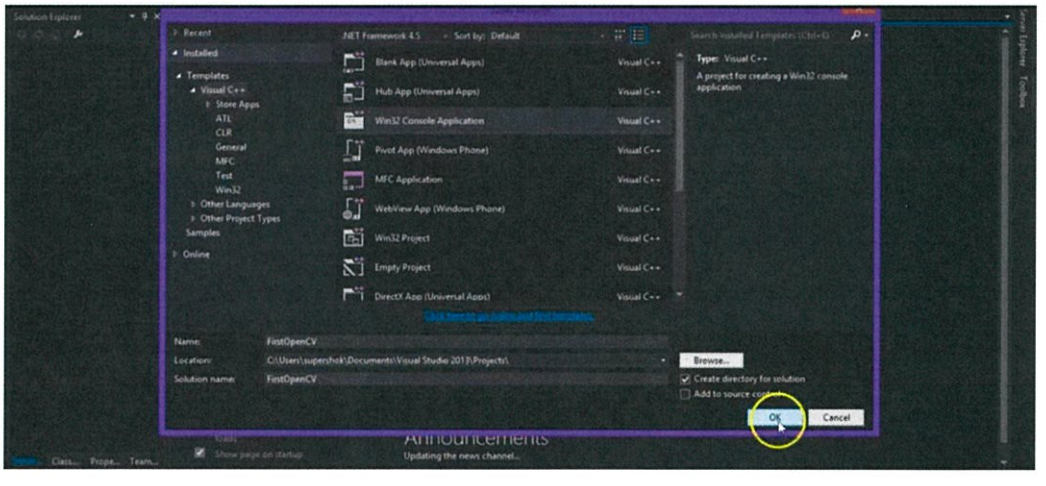
3) เมื่อทำการติดตั้งเสร็จเรียบร้อยแล้วโปรแกรม Visual Studio จำเป็นที่จะต้องสร้างโปรเจค และตั้งค่าดังนี้

3.1) New -> Project



รูปที่ ก.2 แสดงการสร้างโปรเจค

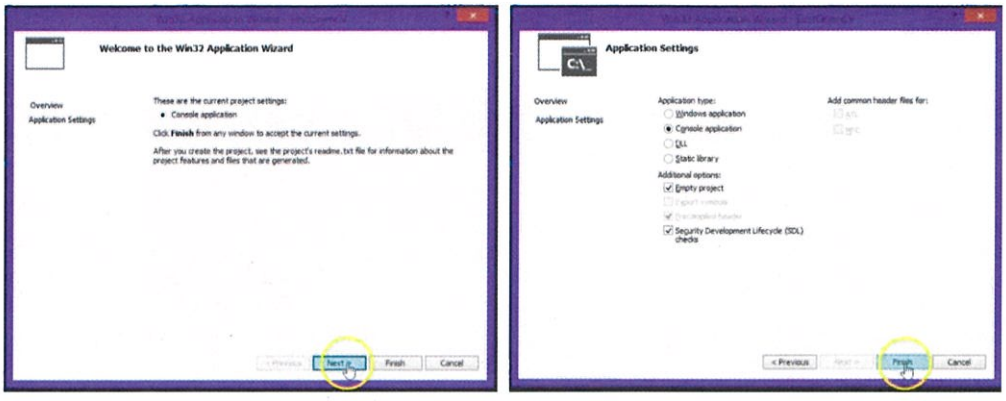
3.2) เลือก Win32 Console Application ตั้งชื่อโปรเจค และกดปุ่ม OK



รูปที่ ก.3 แสดงหน้าต่างการตั้งค่า

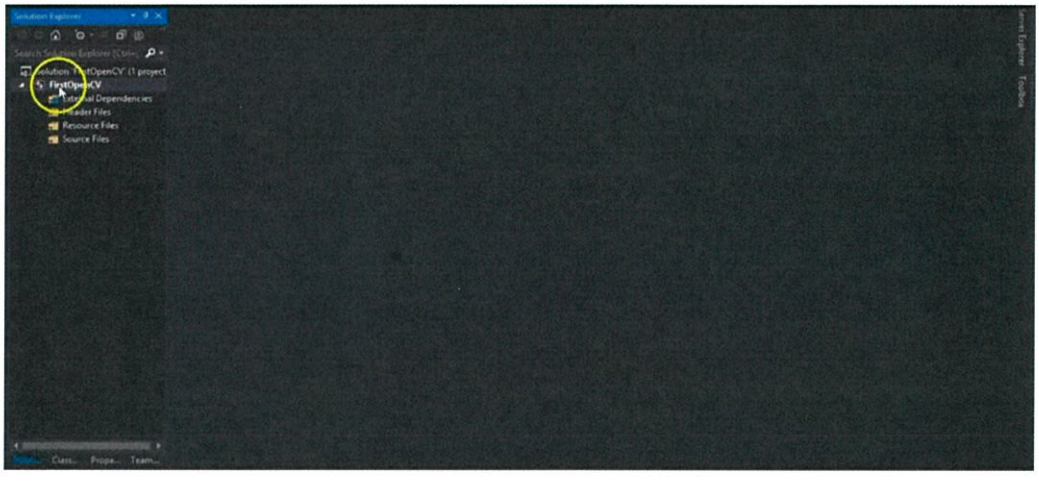
3.3) Application Settings

Application type เลือก console Application และ Additional options เลือก Empty Project

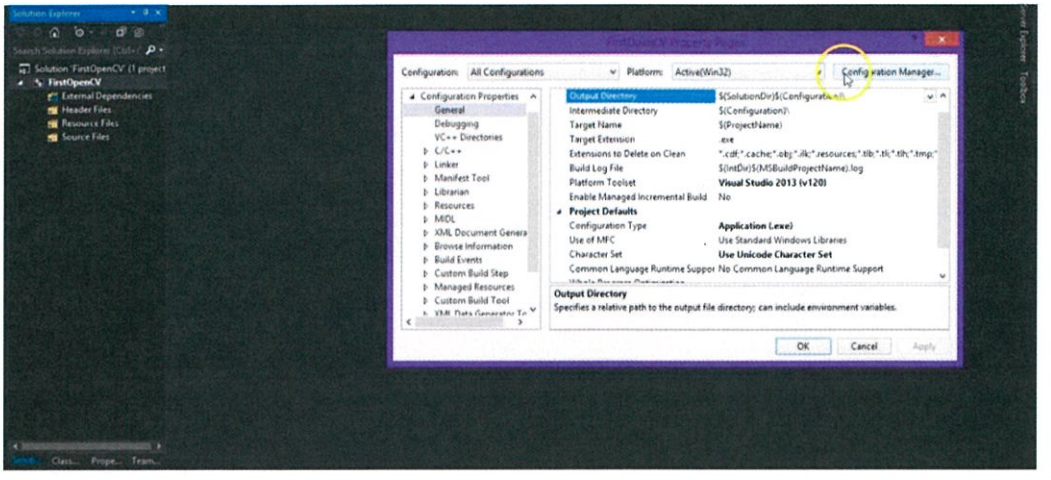


รูปที่ ก.4 แสดงหน้าต่างการตั้งค่า

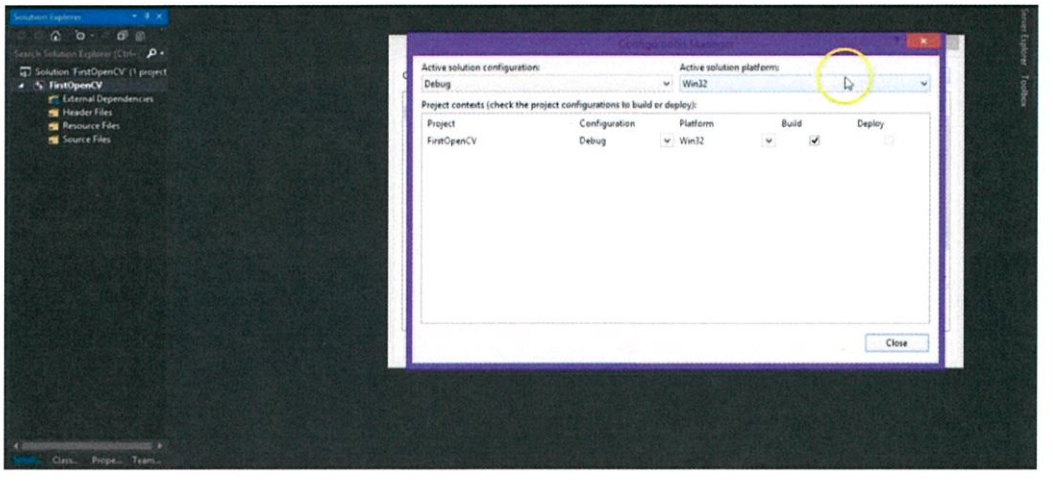
3.4) คลิกที่ชื่อโปรเจคเลือก property และ Configuration Manager จะปรากฏหน้าต่าง ดังรูปที่ ก.7



รูปที่ ก.5 แสดงหน้าต่างการคลิกเลือก

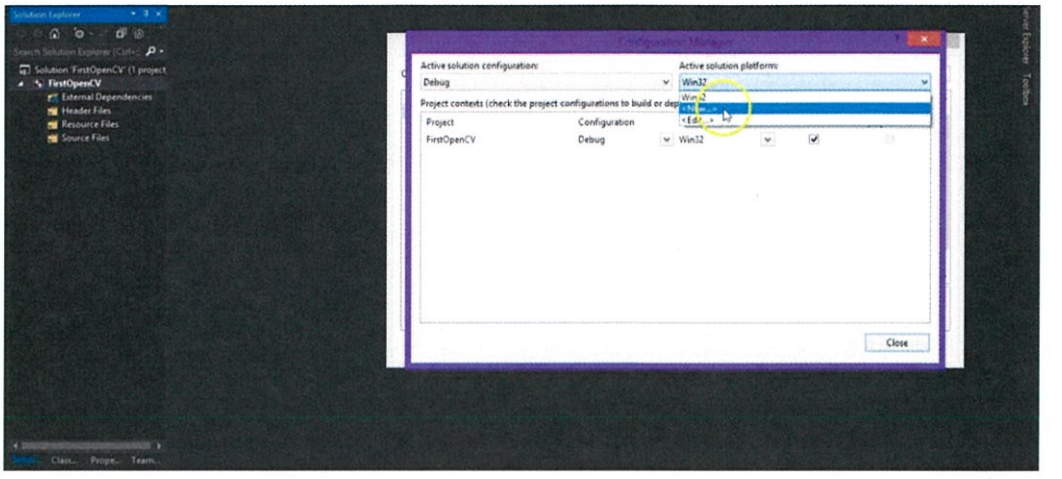


รูปที่ ก.6 แสดงหน้าต่างคลิกเลือก Configuration Manager

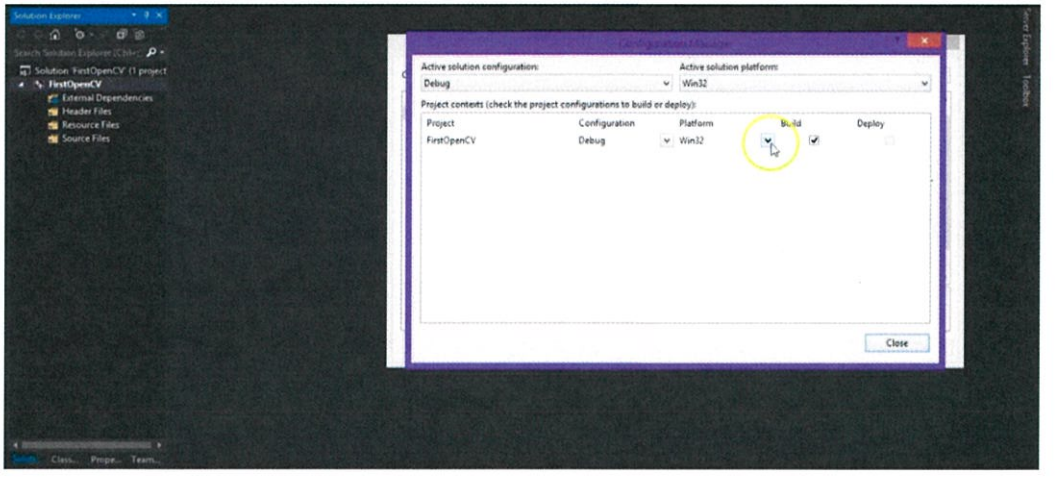


รูปที่ ก.7 แสดงหน้าต่าง Configuration Manager

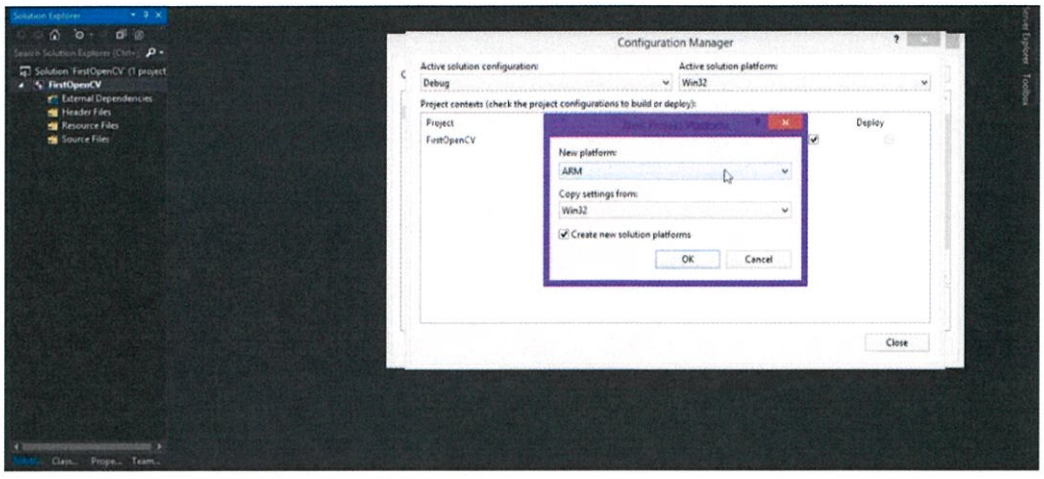
3.5) เลือก Win32 คลิกที่ new และเลือก Win32 ตรง platform คลิก new จะได้หน้าต่างแสดงดังรูปที่ ก.8 และก.9



รูปที่ ก.8 แสดงการตั้งค่าใน Configuration Manager

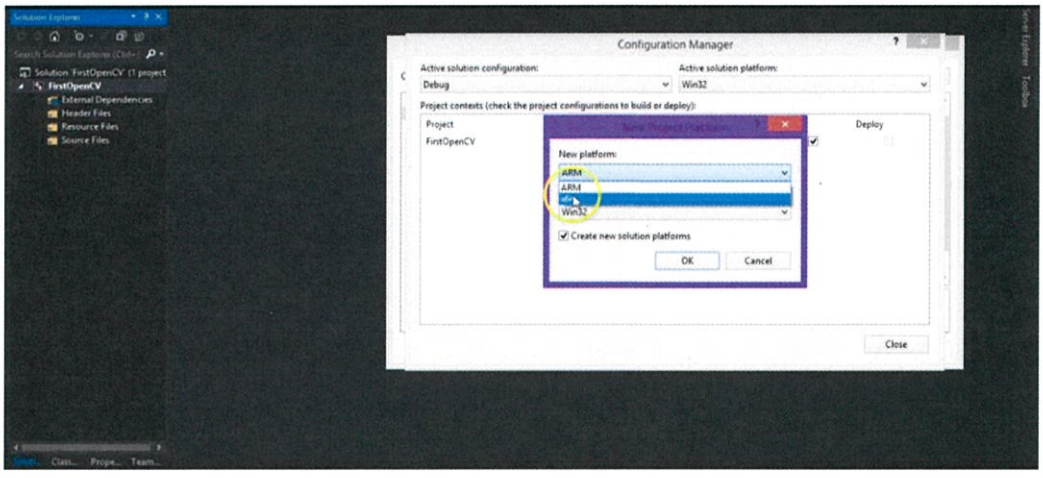


รูปที่ ก.9 แสดงการตั้งค่าใน Configuration Manager



รูปที่ ก.10 แสดงหน้าต่างเมื่อคลิกปุ่ม New ที่ Win32

3.6) จากนั้นคลิกที่ ARM เลือก x64 และกด OK



รูปที่ ก.11 แสดงการตั้งค่าใน Configuration Manager

3.7) เลือก property คลิกที่ C/C++ -> General -> Additional Include Directories กด Edit ใส่

C:\OpenCV2410\opencv\build\include

C:\OpenCV2410\opencv\build\include\opencv

C:\OpenCV2410\opencv\build\include\opencv2

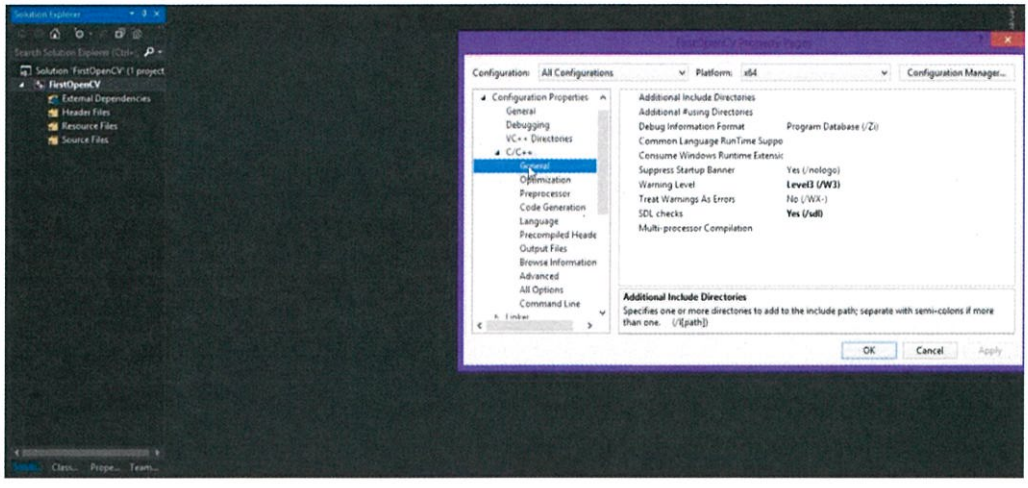
จากนั้นเลือก Linker -> General -> Additional Library Directories

กด Edit ใส่ C:\OpenCV2410\opencv\build\x64\vc12\lib

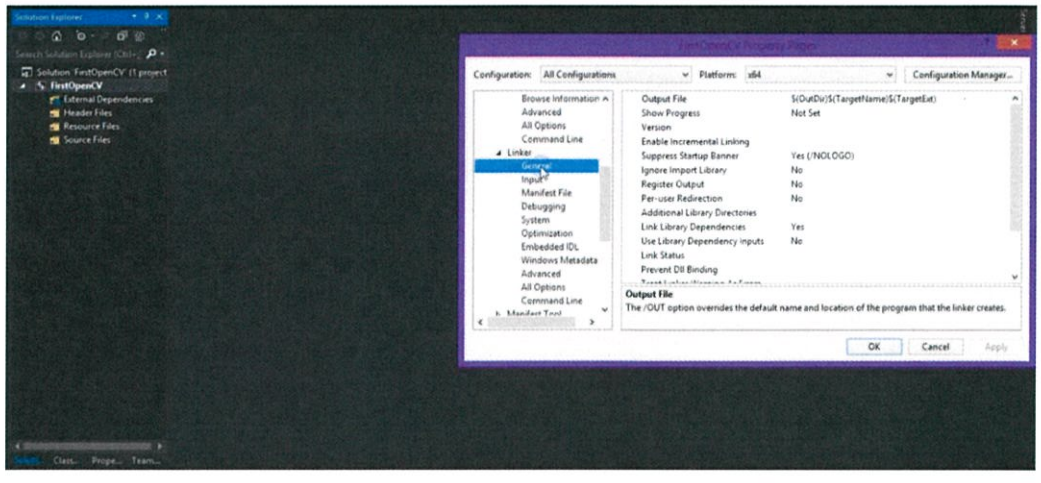
และเลือก Linker -> Input -> Additional Dependencies

กด Edit ใส่
opencv_calib3d2410d.lib
opencv_contrib2410d.lib
opencv_core2410d.lib
opencv_features2d2410d.lib
opencv_flann2410d.lib
opencv_gpu2410d.lib
opencv_highgui2410d.lib
opencv_imgproc2410d.lib
opencv_legacy2410d.lib
opencv_ml2410d.lib
opencv_nonfree2410d.lib
opencv_objdetect2410d.lib
opencv_ocl2410d.lib
opencv_photo2410d.lib
opencv_stitching2410d.lib
opencv_superres2410d.lib
opencv_ts2410d.lib
opencv_video2410d.lib
opencv_videostab2410d.lib

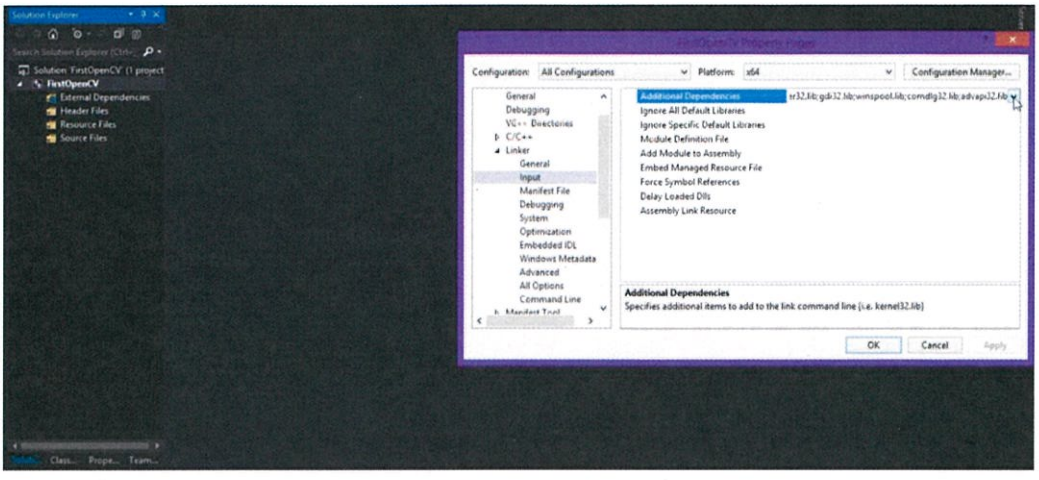
แสดงขั้นตอนที่กล่าวมาดังรูปที่ ก.12, ก.13, ก.14 และก.15



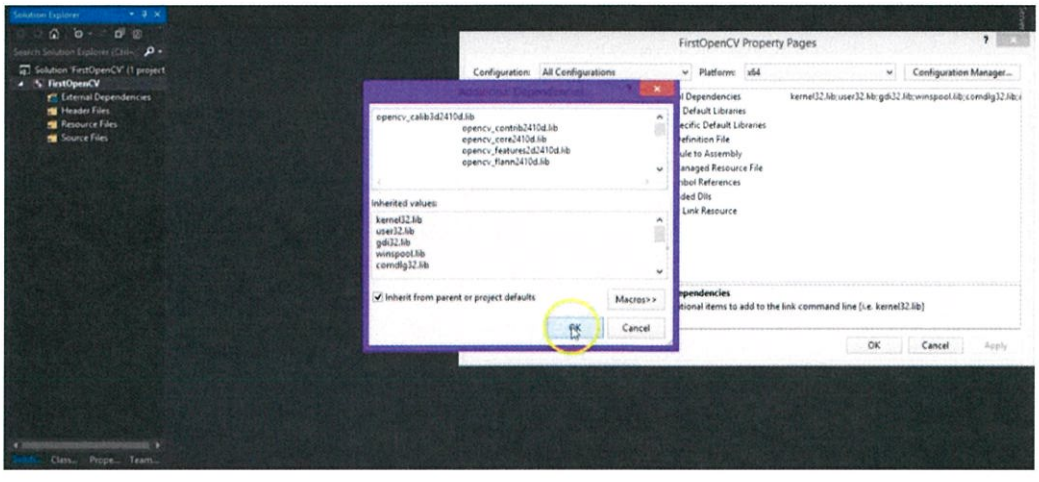
รูปที่ ก.12 แสดงขั้นตอนการตั้งค่าใน Property



รูปที่ ก.13 แสดงขั้นตอนการตั้งค่าใน Property



รูปที่ ก.14 แสดงขั้นตอนการตั้งค่าใน Property



รูปที่ ก.15 แสดงขั้นตอนสุดท้ายของการตั้งค่าใน Property

4) เพิ่ม opencv path ลงในระบบ

วิธีการ set path เข้าไปที่

Control Panel -> System -> Advanced System Setting -> Environment Variables จะปรากฏหน้าต่างนี้ขึ้นตรง System variables เลือก path -> edit

Variable value ให้ add C:\OpenCV2410\opencv\build\x64\vc12\bin

จากขั้นตอนที่กล่าวมาข้างต้นทำให้สามารถใช้งานฟังก์ชัน ตัวแปร และประเภทข้อมูลต่างๆ ของไลบรารี OpenCV ได้

ทดสอบจากการใช้งาน ดังนี้
นำโค้ดตัวอย่างที่ใช้งานไลบรารี OpenCV มารันบนระบบ
โค้ดแสดงรูปภาพ

```
#include "opencv2/highgui/highgui.hpp"
#include <iostream>
using namespace cv;
using namespace std;
int main( int argc, const char** argv )
{
Mat img = imread("MyPic.JPG", CV_LOAD_IMAGE_UNCHANGED);
            if (img.empty())
            {
                system("pause");
                return -1;
            }
namedWindow("MyWindow", CV_WINDOW_AUTOSIZE);
imshow("MyWindow",img);
waitKey(0);
destroyWindow("MyWindow");
return 0;
}
```

รูปที่ ก.16 ตัวอย่างการใช้งานไลบรารี OpenCV เพื่อแสดงรูปภาพ