การสั่งงานเมาส์ด้วยใบหน้า

CONTROL MOUSE BY FACE

ธฤตุ สุทธิธรรม
ปริวัตต์ กัปตัน
ปิติ บุษรากุล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมแมคคาทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

การสั่งงานเมาส์ด้วยใบหน้า

CONTROL MOUSE BY FACE

ธฤต สุทธิธรรม

ปริวัตต์ กัปตัน

ปิติ บุษรากุล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมแมคคาทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CONTROL MOUSE BY FACE

THARIT SUTTITHAM
PARIWATT CAPTAIN
PITI BUTSARAKUL

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF ENGINEEERING IN MECHATRONICS ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2013

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2556

สาขาวิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าคุณทหารลาดกระบัง

เรื่อง การสั่งงานเมาส์ด้วยใบหน้า

CONTROL MOUSE BY FACE

ผู้จัดทำ นายธฤต สุทธิธรรม 53010715

นายปริวัตต์ กัปตัน 53010948

นายปิติ บุษรากุล 53010984

_____ อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์สุมิตร พนาอุดมทรัพย์)

_____ อาจารย์ที่ปรึกษา

(รองศาสตราจารย์ ดร.เกียรติศักดิ์ คมวัชระ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสั่งงานเมาส์ด้วยใบหน้า

โดย

นายธฤต สุทธิธรรม 53010715
นายปริวัตต์ กัปตัน 53010948
นายปิติ บุษรากุล 53010984

อาจารย์ที่ปรึกษา
ผู้ช่วยศาสตราจารย์สุมิตร พนาอุดมทรัพย์
รองศาสตราจารย์ ดร.เกียรติศักดิ์ คมวัชระ

ปีการศึกษา 2556

บทคัดย่อ

งานวิจัยนี้เป็นการวิเคราะห์และพัฒนาโปรแกรมเกี่ยวกับการระบุตำแหน่งและติดตามใบหน้าเพื่อทำการสั่งงานเมาส์ โดยเมาส์จะเคลื่อนที่ตามทิศทางที่เราขยับใบหน้า จะคลิกซ้ายเมื่อเรากระพริบตาซ้าย และคลิกขวาเมื่อเรากระพริบตาขวา โดยอาศัยทฤษฎีการประมวลผลภาพเข้ามาช่วยในการวิเคราะห์ ซึ่งพัฒนาขึ้นด้วยซอฟแวร์ visual basic c# และ Emgu CV ไลบรารี่ บนระบบปฏิบัติการ window 7 โดยมีกล้องเว็บแคมราคาไม่แพงเป็นอุปกรณ์สำหรับรับภาพวิดีโอเพื่อมาประมวลผล ทั้งนี้งานวิจัยที่จัดทำขึ้นนี้จะเป็นประโยชน์ต่อบุคคลที่เป็นอัมพาตที่ไม่สามารถขยับแขนได้ ให้สามารถใช้งานคอมพิวเตอร์เบื้องต้นได้อย่างบุคคลปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CONTROL MOUSE BY FACE

By

Mr.Tharit Suttitham 53010715
Mr.Pariwatt Captain 53010948
Mr.Piti Budsarakul 53010984

Advisors
Asst.Prof. Sumit Panaudomsup
Assoc.Prof. Dr. Kiattisak Kumwachara

Academic Year 2013

ABSTRACT

This research is about analysis and development program to detecting and tracking face. Mouse cursor will move following your face and left click by blinking left eye or right click by blinking right eye. This research using image processing theory to analysis, using visual basic c# and Emgu CV library on window 7 to development program and using cheap web camera to receive video. Finally this research is useful for Paralytic who cannot move their hands to using mouse like a normally people.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ ประสบความสำเร็จลุล่วงได้ด้วยดี ด้วยคำแนะนำ คำปรึกษา และความเอาใจใส่เป็นอย่างดีจาก ผู้ช่วยศาสตราจารย์สุมิตร พนาอุดมทรัพย์ และรองศาสตราจารย์ ดร.เกียรติ ศักดิ์ ซึ่งเป็นอาจารย์ผู้ควบคุมปริญญานิพนธ์ คณะผู้จัดทำรู้สึกซาบซึ้ง และขอกราบขอบพระคุณท่านเป็นอย่างสูง

ขอกราบขอบพระคุณ ท่านคณาจารย์ทุกท่านที่ให้ความรู้ความเข้าใจในเนื้อหาวิชาตั้งแต่เริ่มเข้าศึกษา เพื่อนำความรู้ที่ได้จากคณาจารย์ทุกท่าน นำมาประกอบในการทำปริญญานิพนธ์ฉบับนี้

ขอกราบขอบพระคุณ บิดา มารดา ที่เป็นกำลังใจเสมอมา

ขอขอบคุณเพื่อนๆ พี่ๆ ที่คอยให้ความช่วยเหลือและคำแนะนำที่ดี

ผู้จัดทำ

นายธฤษ สุทธิธรรม

นายปริวัตต์ กัปตัน

นายปิติ บุษรากุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
บทคัดยอภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VI
สารบัญตาราง	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของโครงงาน	1
1.2 วัตถุประสงค์ในการทำโครงงาน	1
1.3 ขั้นตอนศึกษา และการจัดทำโครงงาน	2
1.4 ขอบเขตของโครงงาน	2
1.5 ผลที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง	3
2.1 การประมวลผลภาพดิจิตอล	3
2.2 รูปร่างของภาพ (Image Shape)	3
2.3 ความรู้พื้นฐานเกี่ยวกับภาพขาว-ดำ (Binary Image)	5
2.4 ความรู้พื้นฐานเกี่ยวกับภาพระดับสีเทา (Grayscale Image)	6
2.5 ความรู้พื้นฐานเกี่ยวกับภาพสีระบบอาร์จีบี (RGB image)	7
2.6 มาตรฐานของสี	8
2.6.1 ระบบสีอาร์จีบี	8
2.6.2 ระบบสีเอชเอสวี	9
2.6.3 ระบบสีเอชแอลเอส	11
2.6.4 การแปลงค่าจากระบบสีอาร์จีบีเป็นเอชเอสวี	12
2.6.5 การแปลงภาพสีระบบอาร์จีบีให้เป็นภาพระดับสีเทา	13
2.7 ลักษณะเด่นของฮาร์	14
2.8 CAMSHIFT algorithm	14
2.9 การหาขอบของภาพ (Edge Detection)	15
2.9.1 วิธีเกรเดียนต์	16
2.9.2 วิธีลาปลาเซียน	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.10 บริเวณที่สนใจ (Region of Interest: ROI)	16
2.11 ค่าขีดแบ่ง	16
2.12 Emgu CV	17
2.13 Visual basic C#	17
บทที่ 3 การวิเคราะห์และออกแบบระบบ	18
3.1 การออกแบบแผนผังโครงสร้างของระบบ	18
3.2 การออกแบบหน้าตาของโปรแกรม	19
3.2.1 Live cam	19
3.2.2 Face	20
3.2.3 Face Behavior	20
3.2.4 Eye Behavior	21
บทที่ 4 การพัฒนาและการทดสอบระบบ	22
4.1 การพัฒนาโปรแกรม	22
4.1.1 โปรแกรมและเครื่องที่ใช้ในการพัฒนาระบบ	22
4.1.2 พิจารณาในส่วนของการติดตามใบหน้า	23
4.2 การทดสอบโปรแกรม	27
4.2.1 วิธีการทดสอบการทำงานของโปรแกรม	27
4.2.2 ขั้นตอนการทดสอบการทำงานของโปรแกรม	28
บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ	34
5.1 ผลที่ได้รับ	34
5.2 ปัญหาและอุสรรค	34
5.3 การแก้ปัญหา	34
5.4 ข้อเสนอแนะ	35
เอกสารอ้างอิง	36
ภาคผนวก	37
ภาคผนวก ก Source code	38
ภาคผนวก ข อุปกรณ์ที่ใช้	52
ภาคผนวก ค ตารางการดำเนินงาน	53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	ภาพขาว-ดำและการเก็บข้อมูลลงในหน่วยความจำคอมพิวเตอร์	5
2.2	ภาพระดับสีเทาและการเก็บข้อมูลลงในหน่วยความจำคอมพิวเตอร์	6
2.3	ภาพสีอาร์จีบีและการเก็บข้อมูลลงในหน่วยความจำคอมพิวเตอร์	7
2.4	แสดงระบบสี RGB	8
2.5	แสดงระบบสี HSV	9
2.6	แสดงระบบสี HLS	11
2.7	การแปลงภาพสีอาร์จีบีเป็นภาพระดับสีเทา	13
2.8	รูปแบบของรูปเหลี่ยมที่ใช้ในการตรวจจับวัตถุด้วยลักษณะเด่นของฮาร์	14
2.9	ภาพใบหน้าจากวีดีโอและภาพ Gray Scale ของวัตถุที่ถูกเลือก	15
2.10	การติดตามใบหน้าตามทิศทางที่กำหนด	15
2.11	การติดตามใบหน้าตามแกน X, Y, Z และแกนหมุน	15
2.12	แสดงบริเวณที่สนใจ	16
3.1	แผนผังแสดงโครงสร้างการทำงานของโปรแกรมทั้งหมด	18
3.2	แสดงส่วนของ LiveCam	19
3.3	แสดงส่วนของข้อมูล Face	20
3.4	แสดงส่วนของ Face Behavior	20
3.5	แสดงส่วนของ Eye Behavior	21
3.6	แสดงหน้าตาการทำงานทั้ง 4 ส่วน	21
4.1	ภาพเฟรมแรกที่รับมาจากกล้อง	23
4.2	ภาพแสดงการตีกรอบบริเวณที่ใบหน้า	23
4.3	ภาพแสดงการตีกรอบบริเวณแถบดวงตา	24
4.4	ภาพแสดงการตีกรอบรอบดวงตาข้างซ้าย	24
4.5	ภาพแสดงการตีกรอบรอบดวงตาข้างขวา	25
4.6	ภาพแสดงการตีกรอบทุกบริเวณที่สนใจ	25
4.7	ภาพแสดงการจับดวงตาข้างซ้าย	26
4.8	ภาพแสดงการจับดวงตาข้างขวา	26
4.9	สถานะตาข้างขวา	27
4.10	สถานะตาข้างซ้าย	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.11	หน้าต่างของโปรแกรม	28
4.12	เปิดกล้องและเริ่มตรวจหาใบหน้า	28
4.13	โปรแกรมเริ่มการติดตาม	29
4.14	แสดงการติดตามเมื่อขยับหน้าขึ้น	29
4.15	แสดงการติดตามเมื่อขยับหน้าลง	30
4.16	แสดงการติดตามเมื่อขยับหน้าไปทางซ้าย	30
4.17	แสดงการติดตามเมื่อขยับหน้าไปทางขวา	31
4.18	แสดงการติดตามเมื่อขยับหน้าไปทางขวาบน	31
4.19	แสดงการจับสถานะเปิดปิดของตาซ้าย	32
4.20	แสดงการจับสถานะเปิดปิดของตาขวา	32
4.21	แสดงระยะที่ไกลที่สุดที่สามารถทำงานได้	32
4.22	แสดงระยะที่ใกล้ที่สุดที่สามารถทำงานได้	33
4.23	แสดงการจับใบหน้ากรณีที่มีใบหน้ามากกว่าหนึ่ง	33
4.24	แสดงการตรวจจับใบหน้าในบุคคลที่แตกต่าง	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
ค.1 แสดงขั้นตอนการทำงานในเทอม 1	53
ค.2 แสดงขั้นตอนการทำงานในเทอม 2	53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของโครงงาน

ในปัจจุบันเทคโนโลยีคอมพิวเตอร์เข้ามามีบทบาทในชีวิตประจำวันมากขึ้น ทั้งในเรื่องการติดต่อสื่อสารและสื่อบันเทิงต่าง ๆ แต่ในขณะเดียวกันก็ยังคงมีบุคคล ซึ่งป่วยเป็นอัมพาตไม่สามารถช่วยเหลือตัวเองได้ เช่น จากสมองพิการมีการบาดเจ็บที่สมองหรือโรคหลอดเลือดสมอง และถูกจำกัดความสามารถในการเคลื่อนไหวร่างกาย โดยบางคนสามารถขยับศีรษะได้ บางคนสามารถกระพริบตาหรือขยิบตา บางคนสามารถเคลื่อนไหวนัยน์ตาได้ โดยครอบครัวญาติพี่น้องและคนดูแลผู้ป่วย มักจะสังเกตการเคลื่อนไหวเหล่านี้เพื่อช่วยในการสื่อสาร และในปัจจุบันได้มีเทคโนโลยีซึ่งรู้จักกันในชื่อการประมวลผลภาพ ซึ่งการประมวลผลภาพได้มีการคิดค้นขึ้นในปี ค.ศ.1964 ที่ห้องเลป Jet Propulsion โดยงานชิ้นแรกคือการพิจารณาภาพถ่ายดาวเทียมของดวงจันทร์ หลังจากนั้นก็เกิดการจัดตั้งสาขาทางวิทยาศาสตร์ ชื่อว่า การประมวลผลภาพ หรือ Digital image processing ทำให้เกิดการสนใจในงานด้านการประมวลผลภาพเป็นอย่างมากและใช้กันอย่างกว้างขวางสำหรับงานในหลาย ๆ ด้าน เช่น การสื่อสารทางโทรทัศน์ ทางด้านการพิมพ์ ทางด้านกราฟฟิก การแพทย์ และการค้นคว้าทางวิทยาศาสตร์

Digital image processing จะเกี่ยวกับการแปลงข้อมูลภาพให้อยู่ในรูปแบบข้อมูลดิจิตอล (Digital format) ทำให้สามารถที่จะนำเอาข้อมูลภาพเหล่านี้ ไปผ่านกระบวนการต่าง ๆ ด้วยดิจิตอลคอมพิวเตอร์ได้ ในระบบของดิจิตอล อินพุตและเอาต์พุตของระบบจะอยู่ในรูปแบบดิจิตอลเท่านั้น

จากที่กล่าวมาข้างต้นจึงได้มีแนวคิดที่จะทำโครงงานเกี่ยวกับการสั่งงานคอมพิวเตอร์ด้วยใบหน้าเพื่ออำนวยความสะดวกของผู้ป่วยดังกล่าว โดยการประมวลผลภาพใบหน้าบนวีดีโอจากกล้องเว็บแคมเพื่อนำไปใช้ในการสั่งงานการทำงานของเคอเซอร์เมาส์ เพื่อใช้อำนวยความสะดวกในการเข้าถึงเทคโนโลยีคอมพิวเตอร์ให้แก่บุคคลเหล่านั้น

1.2. วัตถุประสงค์ในการทำโครงงาน

เพื่อช่วยผู้ป่วยอัมพาตให้สามารถใช้งานคอมพิวเตอร์ได้ โดยพัฒนาโปรแกรมที่สามารถตรวจจับการเคลื่อนที่ของใบหน้าเพื่อใช้ในการขยับเมาส์ในเครื่องคอมพิวเตอร์ และตรวจจับพฤติกรรมการเปิดปิดของตาเพื่อใช้ในการคลิกเมาส์ โดยใช้กล้องแว็บแคมเป็นสื่อกลางในการรับภาพและพัฒนาโปรแกรมให้เกิดความแม่นยำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขั้นตอนการศึกษา และการจัดทำโครงงาน

1. กำหนดหัวข้อโครงงาน และศึกษาความเป็นไปได้ของโครงงาน
2. กำหนดขอบเขตของโครงงาน เลือกอุปกรณ์และเทคโนโลยีที่จะใช้
3. ศึกษาโครงสร้างและวิธีการเขียนของภาษา C# และการใช้งานโปรแกรม Visual C#
4. ศึกษางานวิจัยที่ผ่านมาซึ่งเกี่ยวข้องกับประมวลผลภาพจากเว็บแคม
5. ศึกษาวิธีการตรวจจับวัตถุจากวีดีโอด้วยวิธีต่าง ๆ
6. วิเคราะห์และเลือกวิธีการตรวจสอบที่เหมาะสมที่สุดสำหรับโปรแกรมที่จะพัฒนา
7. วิเคราะห์และออกแบบขั้นตอนการทำงานของระบบ
8. พัฒนาโปรแกรมที่ใช้ในการปรับปรุงภาพ หาตำแหน่งภาพ และตรวจสอบ
9. วิเคราะห์ผลการทดลอง และแก้ไขส่วนที่ผิดพลาด
10. สรุปผลและข้อเสนอแนะ
11. จัดทำเอกสารประกอบโครงงาน

1.4 ขอบเขตของโครงงาน

1. ต้องไม่มีปัจจัยภายนอกมารบกวน เช่น แสงแดด
2. ใช้บนระบบปฏิบัติการ window 7 เท่านั้น
3. ผู้ใช้งานต้องไม่สวมใส่แว่นตาขณะใช้งาน

1.5 ผลที่คาดว่าจะได้รับ

1. ช่วยให้ผู้ป่วยอัมพาตได้ใช้งานคอมพิวเตอร์ได้ในระดับหนึ่ง
2. สามารถนำไปต่อยอดเพื่อพัฒนาโปรแกรมที่เกี่ยวข้องกับการตรวจจับใบหน้า เช่น โปรแกรมรักษาความปลอดภัย
3. ทำให้ผู้จัดทำมีความรู้เกี่ยวกับการเขียนภาษา C# และการประมวลผลภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและความรู้ที่เกี่ยวข้อง

2.1 การประมวลผลภาพดิจิตอล

การมองเห็นของมนุษย์เป็นสิ่งที่สำคัญ และเป็นกลไกการรับภาพที่ซับซ้อนอย่างหนึ่ง ซึ่งจะให้ข้อมูลที่มีความจำเป็นสำหรับใช้ในงานง่าย ๆ (ตัวอย่าง เช่น การจดจำวัตถุ) และสำหรับงานที่มีความซับซ้อน (ได้แก่ การวางแผนการตัดสินใจ การค้นคว้าทางวิทยาศาสตร์ การพัฒนาทางด้านความคิด) ดังคำ สุภาษิตของจีนกล่าวไว้ว่า "รูปภาพสามารถแทนคำ ได้เป็นพัน ๆ คำ " รูปภาพมีบทบาทมากสำหรับองค์กรต่าง ๆ เช่น หนังสือพิมพ์ โทรทัศน์ ภาพยนต์ซึ่งได้ใช้ภาพ (ภาพนิ่ง ภาพเคลื่อนที่) เป็นสื่อนำ เสนอข้อมูลข่าวสารต่าง ๆ สิ่งที่น่าสนใจของข้อมูลที่เกี่ยวกับการมองเห็นหรือข้อมูลภาพนั้นก็คือกระบวนการประมวลผลภาพ (Image Processing) โดยใช้ดิจิตอลคอมพิวเตอร์

Digital image processing จะเกี่ยวกับการแปลงข้อมูลภาพให้อยู่ในรูปแบบข้อมูลดิจิตอล (Digital format) ซึ่งสามารถที่จะนำเอาข้อมูลนี้จัดผ่านกระบวนการต่าง ๆ ด้วยดิจิตอลคอมพิวเตอร์ได้ ในระบบของดิจิตอล อินพุต และเอาต์พุตของระบบจะอยู่ในรูปแบบดิจิตอลเท่านั้น

Digital image analysis จะเกี่ยวกับวิธีการอธิบาย และการจดจำข้อมูลภาพดิจิตอล ซึ่งอินพุตของระบบจะเป็นข้อมูลภาพดิจิตอล และเอาต์พุตจะเป็นเครื่องหมายที่ใช้แทนข้อมูลภาพดิจิตอลเหล่านั้น ในการวิเคราะห์ภาพมีอยู่หลายวิธีด้วยกันที่ได้นำมาจากการทำงานของตามนุษย์ (human vision) นั่นก็คืองานทางด้าน Computer Vision เป็นลักษณะเดียวกับ Digital image analysis นั่นเอง การมองเห็นของมนุษย์นับว่าเป็นกระบวนการที่ซับซ้อนซึ่งลักษณะเทคนิค โดยทั่ว ๆ ไปในกระบวนการ Digital image analysis และ Computer Vision จะค่อนข้างซับซ้อนเช่นกัน

2.2 รูปร่างของภาพ (Image Shape)

วัตถุที่มีอยู่ตามธรรมชาติ และที่มนุษย์สร้างขึ้นมีรูปร่างที่แตกต่างกันไป ทั้งที่เป็นรูปทรงเรขาคณิต และไม่เป็นรูปทรงเรขาคณิต ในศาสตร์ของการประมวลผลภาพนั้น การกำหนดขอบเขตของภาพทุกภาพให้อยู่ในรูปสี่เหลี่ยม (Rectangular image model) เป็นวิธีที่นิยมใช้กันมากที่สุด เนื่องจากทำให้การอ่านภาพ การจัดเก็บข้อมูลภาพในหน่วยความจำ และการแสดงภาพออกทางอุปกรณ์ต่าง ๆ เป็นไปได้อย่างมีประสิทธิภาพ

การเก็บข้อมูลภาพลงหน่วยความจำของคอมพิวเตอร์สามารถทำได้ โดยการจองหน่วยความจำ ของเครื่องไว้ในรูปของตัวแปรอะเรย์ (array) โดยค่าในแต่ละช่องของอะเรย์แสดงถึงคุณสมบัติของจุดภาพ (pixel) และตำแหน่งของช่องอะเรย์เป็นตัวกำหนดตำแหน่งของจุดภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมมุติให้ Image เป็นตัวแปรแบบอะเรย์ขนาด M X N (M แถว และ N คอลัมน์) ที่ใช้เก็บภาพขนาด M x Nจุด (M จุดในแนวนอน และ N จุดในแนวตั้ง) ค่าสี (หรือความสว่าง ในกรณีที่เป็นภาพ grey level) ของจุดภาพในแถวที่ 5 คอลัมน์ที่ 4 จะตรงกับค่าของ Image (5,4) จะเห็นว่าเราใช้ตำแหน่งของจุดภาพทั้งสองแกนเป็นตัวชี้ค่าข้อมูลในอะเรย์

จากการใช้หน่วยความจำ เพื่อการเก็บภาพในลักษณะที่กล่าวมา เนื้อที่ในการเก็บภาพสามารถคำนวณได้จาก M x N x g เมื่อ g เป็นจำนวนเต็มที่แทนจำนวนบิตของข้อมูลในแต่ละจุดภาพ ตัวอย่างถ้า g มีค่าเท่ากับ 8 บิต เราจะสามารถเก็บความแตกต่างของระดับสีที่เป็นไปสูงสุด 256 ระดับ ค่า M และ N จะเป็นตัวบอกถึงความละเอียดของภาพสำหรับคอมพิวเตอร์ทั่วไปในระบบ VGA (Video Graphic Array) จะมีขนาด 640 x 480, 800 x 600 และ 1024 x 768 จุด เป็นต้น การกำหนดความละเอียดจะขึ้นอยู่กับงานที่จะใช้ ในงานบางอย่างใช้ความละเอียดแค่ 30 x 50 จุด ก็พอแล้วแต่ในงานบางชนิด ใช้ความละเอียดถึง 1000 x 1000 จุด ก็ยังไม่พอ

ปกติแล้วในการเก็บข้อมูลภาพ โดยเครื่องมือต่าง ๆ จะเก็บตามมาตรฐานของโทรทัศน์ซึ่งมีอัตราส่วน x ต่อ y เท่ากับ 4:3 สำหรับเครื่องมือเก็บข้อมูลภาพที่ไม่เป็นไปตามอัตราส่วน 4:3 เมื่อนำภาพนี้ไปแสดงในจอภาพมาตรฐานจะทำ ให้ภาพที่แสดงนั้นมีขนาดของจุดภาพไม่เป็นสี่เหลี่ยมจัตุรัส เช่นในบางระบบอาจจะใช้ความละเอียดในการแสดงเท่ากับ 640 x 512 ซึ่งจะทำให้ขนาดของจุดภาพที่ได้มีขนาดของด้านกว้างมีความยาวมากกว่าด้านสูง ซึ่งลักษณะดังกล่าวนี้เป็นหัวข้อที่ต้องสนใจสำหรับการเขียนโปรแกรมทางด้านกราฟฟิก และการจัดการข้อมูล

จำนวนสีสูงสุดที่เป็นไปได้ของแต่ละจุดภาพขึ้นอยู่กับจำนวนบิตที่ใช้ เมื่อมีการกำหนดให้ขนาดของบิตต่อจุดมากขึ้นจะทำให้จำนวนของสีมากขึ้นด้วย ตัวอย่างเช่น

1 บิต = 21 = 4 สี

2 บิต = 22 = 4 สี

4 บิต = 24 = 16 สี

8 บิต = 28 = 256 สี

16 บิต = 216 = 65536 สี เป็นต้น

สำหรับการแสดงข้อมูลภาพที่มีขนาด 1 บิต และ 8 บิตนั้นจะมีการทำงานที่จะใกล้เคียงกัน เนื่องจากหน่วยประมวลผลจะไม่สามารถจัดการกับข้อมูลที่เป็นบิตเดี่ยว ๆ ได้ ดังนั้นในการแสดงข้อมูลออกทางจอภาพตัวโปรเซสเซอร์จะทำการก็อปปี้ข้อมูลทั้ง 8 บิต (1 Byte) ส่งให้กับจอภาพซึ่งในกรณีที่ Pixel มีขนาด 1 บิต เมื่อโปรเซสเซอร์จะทำงานกับบิตแรกที่ต้องการแล้วก็จะทำการก็อปปี้ข้อมูลชุดใหม่ทันที โดยที่ไม่เกี่ยวกับข้อมูลอีก 7 บิตที่เหลือส่วนในกรณี Pixel ที่มีขนาด 8 บิต โปรเซสเซอร์จะทำการก็อปปี้ข้อมูลจุดใหม่ก็ต่อเมื่อโปรเซสเซอร์ทำงานกับทุกบิตแล้ว

ตัวอย่างสำหรับระบบที่มีความละเอียดเท่ากับ 800 x 600 และมีขนาด 16 บิตต่อ Pixel จะสามารถแสดงสีได้ทั้งหมด 65536 ระดับ และต้องใช้เนื้อที่ในการเก็บเท่ากับ 800 x 600 x 16 บิต

เอกสารนี้เป็นเอกสาร... ไม่ว่ากรณี... ...ไปใช้

2.3 ความรู้พื้นฐานเกี่ยวกับภาพขาว-ดำ (Binary Image)

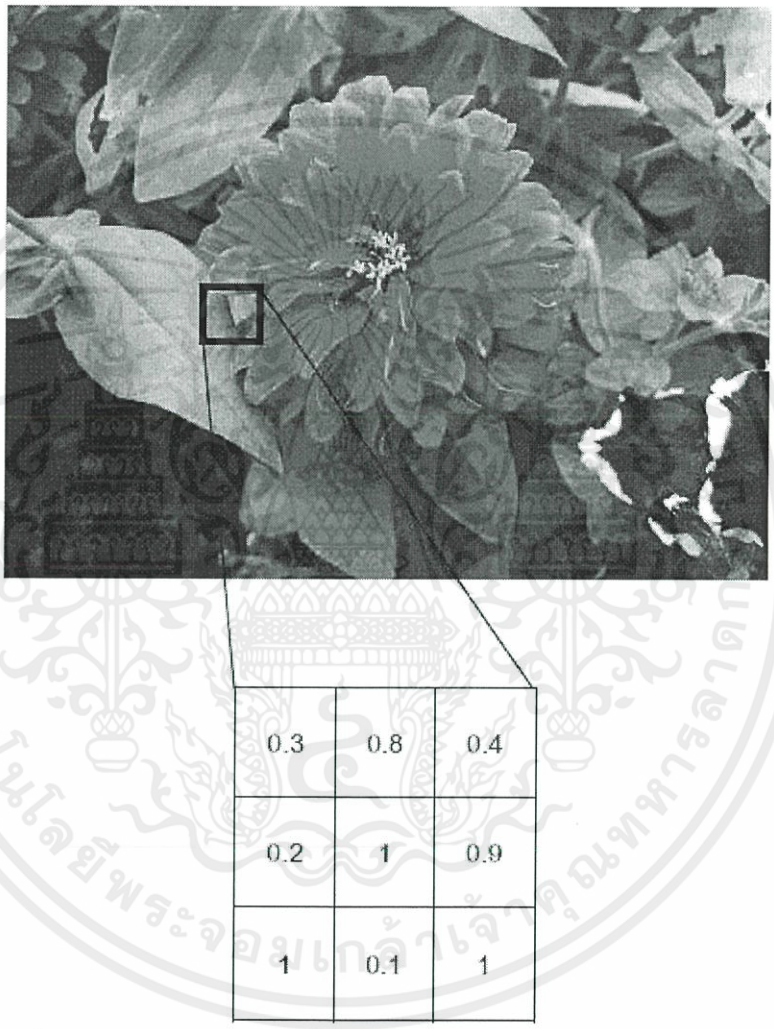
ภาพขาว-ดำนั้นเป็นภาพที่ประกอบเพียงสีขาว และสีดำเท่านั้น โดยการบันทึกภาพขาว-ดำลงบนหน่วยความจำคอมพิวเตอร์นั้นจะถูกบันทึกลงในรูปแบบของอาร์เรย์สองมิติ และค่าที่อยู่ในแต่ละช่องของอาร์เรย์จะแทนค่าของสีในแต่ละพิกเซล โดยสีดำจะถูกบันทึกด้วยค่า 0 และสีขาวจะถูกบันทึกด้วยค่า 1

รูปที่ 2.1 ภาพขาว-ดำ และการเก็บข้อมูลลงในหน่วยความจำคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ความรู้พื้นฐานเกี่ยวกับภาพระดับสีเทา (Grayscale Image)

ภาพระดับสีเทา คือ ภาพดิจิทัลที่ถูกเก็บอยู่ในรูปแบบของอาร์เรย์สองมิติ โดยค่าของแต่ละจุดภาพจะแทนค่าความเข้มของสี โดยค่าสีของภาพที่เป็นไปได้ทั้งหมดจะขึ้นอยู่กับจำนวนบิตที่ใช้เก็บภาพ เช่น ภาพระดับสีเทา 8 บิต จะมีระดับสีทั้งหมด 28 = 256 ระดับ โดยทั่วไปจะนิยมบันทึกในรูปช่วง 0-1 หรือ 0-255

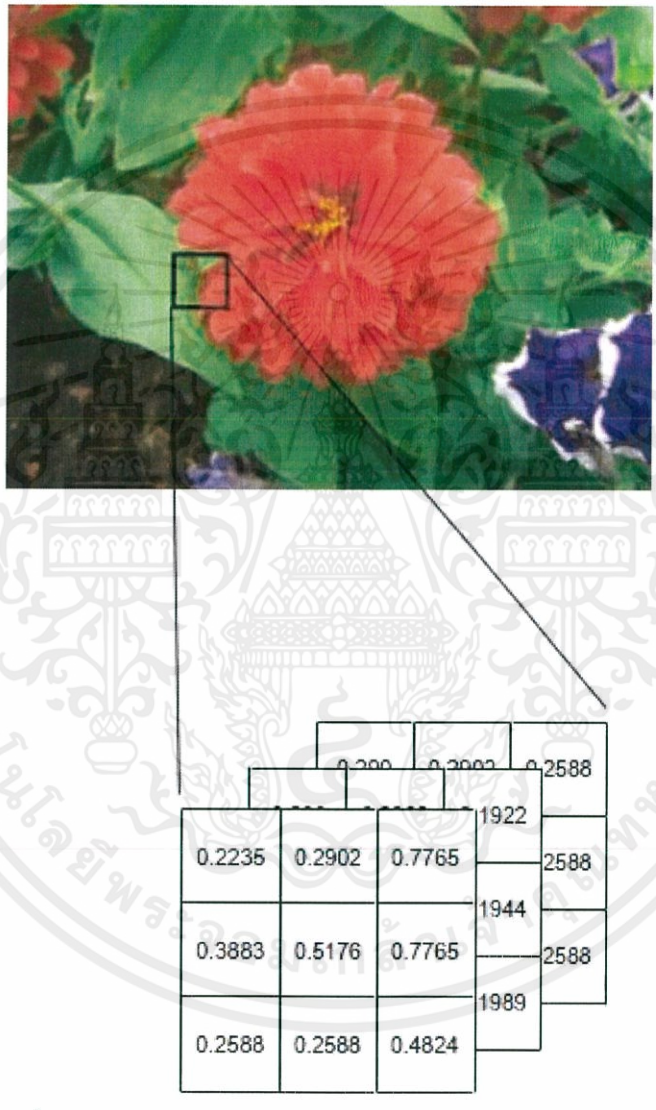


รูปที่ 2.2 ภาพระดับสีเทา และการเก็บข้อมูลลงในหน่วยความจำคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ความรู้พื้นฐานเกี่ยวกับภาพสีระบบอาร์จีบี (RGB Image)

ภาพสีระบบอาร์จีบีเป็นที่นิยมอย่างมากในการบันทึกภาพดิจิทัล โดยภาพจะแสดงสีในแต่ละพิกัดออกมาจากการผสมกันของ สีแดง (Red), สีเขียว (Green) และสีน้ำเงิน (Blue) ภาพสีระบบอาร์จีบีนั้นจะถูกเก็บอยู่ในรูปแบบของอาร์เรย์ 3 มิติที่มีขนาด m x n x 3 โดยที่ m และ n แทนขนาดความยาว และความกว้างของภาพ ส่วนมิติสุดท้ายนั้นจะทำการเก็บค่าของสีแดง, สีเขียว และสีน้ำเงิน



รูปที่ 2.3 ภาพสีอาร์จีบี และการเก็บข้อมูลลงในหน่วยความจำคอมพิวเตอร์

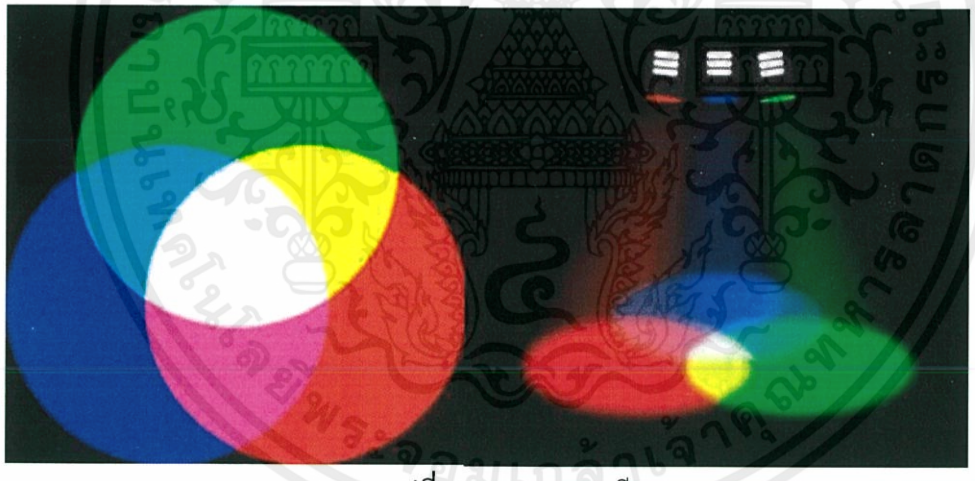
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 มาตรฐานของสี

มาตรฐานของสีที่ใช้อยู่ในปัจจุบันมีอยู่หลายระบบด้วยกัน ทั้งนี้จะขึ้นอยู่กับการนำ ไปใช้ แต่โดยทั่วไปแล้วทุกมาตรฐานจะมีแนวคิดเดียวกันคือ การแทนจุดสีด้วยจุดที่อยู่ภายในสเปส 3 มิติ โดยจะมีแกนอ้างอิงสำหรับจุดสีนั้นในสเปสซึ่งแต่ละแกนจะมีความเป็นอิสระต่อกัน ตัวอย่างเช่น ในระบบ RGB จะมีแกนสีคือ แกนสีแดง เขียว และน้ำเงิน ในระบบ HLS จะมีแกนเป็น ค่าสี (hue) ความสว่าง (lightness) และความบริสุทธิ์ของสี (saturation)

2.6.1 ระบบสี RGB

RGB ย่อมาจาก Red, Green และ Blue คือ กระบวนการผสมสีจากแม่สี 3 สี คือ สีแดง สีเขียว และสีน้ำเงิน การใช้สัดส่วนของสี 3 สีนี้ต่างกัน จะทำให้เกิดสีต่าง ๆ ได้อีกมาก แสดงให้เห็นในรูป ระบบสี RGB เป็นระบบสีที่เกิดจากการรวมกันของแสงสีแดง เขียว และน้ำเงิน ซึ่งโดยปกติจะนำไปใช้ในจอภาพแบบ CRT (Cathode Ray Tube) ในการใช้งานระบบสี RGB ยังมีการสร้างมาตรฐานที่แตกต่างกันออกไปที่นิยมใช้งานได้ แต่ RGBCIE และ RGBNTSC โดยแต่ละสีจะมีค่าความเข้ม (intensity) ตั้ง แต่ 0-255



รูปที่ 2.4 แสดงระบบสี RGB

- สีหลัก คือ สีแดง (Red) สีเขียว (Green) และสีน้ำเงิน (Blue)
- สีรอง เกิดจากสีหลัก 2 สีรวมกัน คือ สีม่วง (Cyan) เกิดจากสีเขียวผสมกับสีน้ำเงิน สีฟ้า (Magenta) เกิดจากสีแดงผสมกับสีน้ำเงิน และสีเหลือง (Yellow) เกิดจากสีแดงผสมกับสีเขียว
- สีขาว เกิดจากสีหลักผสมกัน
- สีดำ เกิดจากไม่มีสีใดเลย

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น รวมทั้งการนำไปดัดแปลงเนื้อหาและอ้างอิงในรูปแบบอื่นๆ ก่อนได้รับอนุญาตเป็นลายลักษณ์อักษรจากผู้เขียนก่อนนำไปใช้

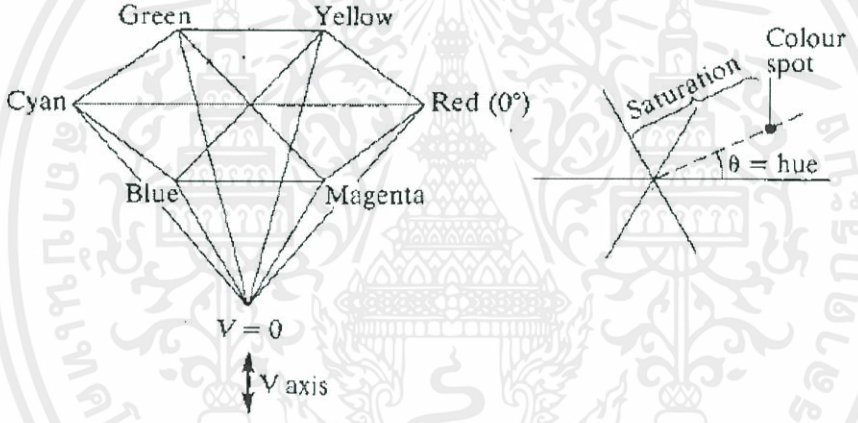
ความเข้มของสีหลัก จะเป็นตัวกำหนดความเหลือบของสีขาว หรือความสว่างของสี และหากความเข้มของแต่ละสีเท่ากัน จะเป็นลักษณะของการเหลือบของสีเทา

2.6.2 ระบบสี HSV

ระบบสี HSV (Hue Saturation Value) เป็นการพิจารณาสี โดยใช้ Hue Saturation และ Value ซึ่ง Hue คือค่าสีของสีหลัก (แดง เขียว และน้ำเงิน) ในทางปฏิบัติจะอยู่ระหว่าง 0 และ 255 ซึ่งถ้า Hue มีค่าเท่ากับ 0 จะแทนสีแดง และเมื่อ Hue มีค่าเพิ่มขึ้นเรื่อย ๆ สีก็จะเปลี่ยนแปลงไปตามสเปกตรัมของสีจนถึง 256 จึงจะกลับมาเป็นสีแดงอีกครั้ง ซึ่งสามารถแทนให้อยู่ในรูปขององศาได้ ดังนี้คือ สีแดง = 0 องศา สีเขียวเท่ากับ 120 องศา สีน้ำเงินเท่ากับ 240 องศา

Hue สามารถคำนวณได้จากระบบสี RGB ได้ดังนี้

$$red_h = red - \min(red, green, blue)$$
$$green_h = red - \min(red, green, blue)$$
$$blue_h = red - \min(red, green, blue)$$



รูปที่ 2.5 แสดงระบบสี HSV

จากลักษณะโมเดลของระบบ Hue พบว่าจะมีค่าอย่างน้อยหนึ่งค่าที่จะเท่ากับ 0 แต่ถ้ามีสองค่าเท่ากับ 0 แล้ว Hue จะเป็นมุมของสี (ค่าสี) มีค่าเป็นไปตามสีที่สาม และถ้าทั้งสามสีมีค่าเท่ากับ 0 แล้วจะทำให้ไม่มีค่าของ Hue หรือสีที่ได้จะมีค่าเท่ากับสีขาวนั่นเอง ตัวอย่างเช่น จอภาพขาว-ดำ ถ้าเกิดมีสีใดสีหนึ่งมีค่าเท่ากับ 0 จะทำให้ค่าสีที่ได้เป็นไปตามสีที่เหลือ การให้น้ำหนักในการพิจารณาเมื่อสีแดงมีค่าเท่ากับ 0

$$\frac{(240 \times blue_h) + (120 \times green_h)}{blue_h + green_h}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Saturation คือความบริสุทธิ์ของสีซึ่งถ้า Saturation มีค่าเท่ากับ 0 แล้วสีที่ได้จะไม่มี Hue ซึ่งจะเป็นสีขาวล้วนแต่ถ้า Saturation มีค่าเท่ากับ 255 แสดงว่าจะไม่มีแสงสีขาวผสมอยู่เลย
Saturation สามารถคำนวณได้ดังนี้

$$Saturation \frac{\max(red, green, blue) - \min(red, green, blue)}{\max(red, green, blue)}$$

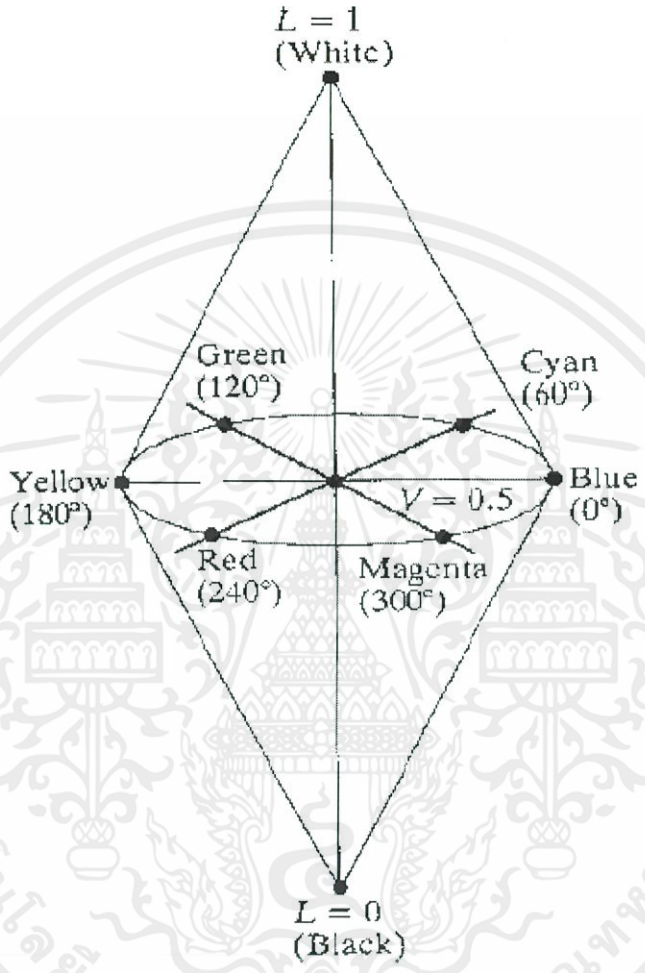
Value คือ ความสว่างของสี ซึ่งสามารถวัดได้ โดยค่าความเข้มของความสว่างของแต่ละสีที่ประกอบกัน สามารถคำนวณได้จาก

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.3 ระบบสีแบบ HLS

ระบบสีแบบ HLS (Hue lightness saturation) พัฒนาโดย Teletromix Incorporated จะมีลักษณะคล้ายกับ HSV ดังนี้คือสีของระบบจะขึ้นอยู่กับ Hue Lightness และ Saturation



รูปที่ 2.6 แสดงระบบสี HLS

Hue คือ ค่าของสีหลักซึ่งมีสีน้ำเงินอยู่ที่ 0 องศา สีเขียวอยู่ที่ 120 องศา และสีแดงอยู่ที่ 240 องศา Lightness คือค่าความสว่างซึ่งจะมีค่าเปลี่ยนแปลงตามแนวแกน L โดยที่ L = 0 จะเป็นสีดำ L = 1 จะเป็นสีขาว สามารถคำนวณได้ดังนี้

$$lightness = \frac{\max(red, green, blue) - \min(red, green, blue)}{2}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Saturation คือความบริสุทธิ์ของสีสามารถหาได้ดังนี้คือ

$$saturation = \begin{cases} \dfrac{\max(red, green, blue + \min(red, green, blue)}{\max(red, green, blue) - \min(red, green, blue)} if\ L \leq 0.5 \\ \dfrac{\max(red, green, blue + \min(red, green, blue)}{2 - \max(red, green, blue) - \min(red, green, blue)} if\ otherwise \end{cases}$$

2.6.4 การแปลงค่าจากระบบสี RGB เป็น HSV

$R' = R/255$

$G' = G/255$

$B' = B/255$

$Cmax = \max(R', G', B')$

$Cmin = \min(R', G', B')$

$\Delta = Cmax - Cmin$

Hue calculation:

$$H = \begin{cases} 60° \times \left(\dfrac{G' - B'}{\Delta} mod 6\right), Cmax = R' \\ 60° \times \left(\dfrac{B' - R'}{\Delta} + 2\right), Cmax = G' \\ 60° \times \left(\dfrac{R' - G'}{\Delta} + 4\right), Cmax = B' \end{cases}$$

Saturation calculation:

$$S = \begin{cases} 0 & , \Delta = 0 \\ \dfrac{\Delta}{Cmax} & , \Delta \neq 0 \end{cases}$$

Value calculation:

$V = Cmax$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.5 การแปลงภาพสีระบบอาร์จีบีให้เป็นภาพระดับสีเทา

การแปลงภาพสีระบบอาร์จีบีให้เป็นภาพระดับสีเทานั้นมีจุดมุ่งหมายในการทำเพื่อให้ระบบสามารถวิเคราะห์ภาพได้สะดวก และรวดเร็วยิ่งขึ้น โดยการแปลงนั้นจะทำการแปลงให้อยู่ในช่วง 0-255 โดยมีฟังก์ชันการแปลง ดังนี้

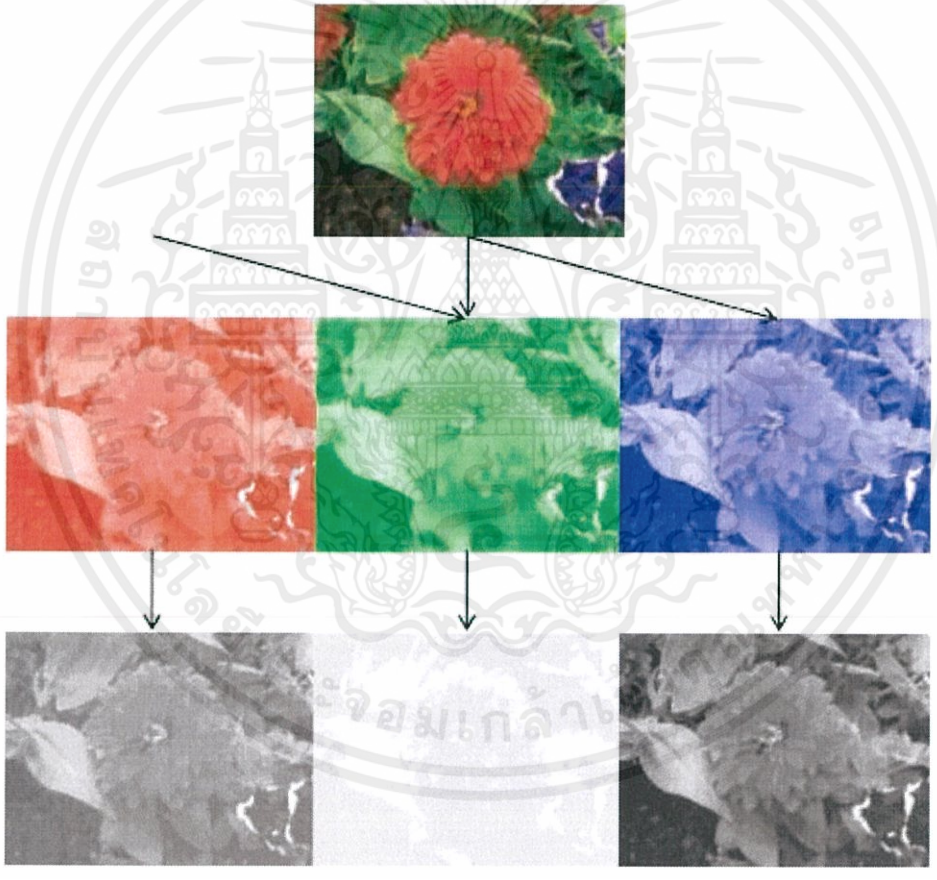
$$I = (0.2989 \times R) + (0.5870 \times G) + (0.1140 \times B$$

โดยที่ I = ความเข้มของสี

R = ค่าของสีแดง

G = ค่าของสีเขียว

B = ค่าของสีน้ำเงิน

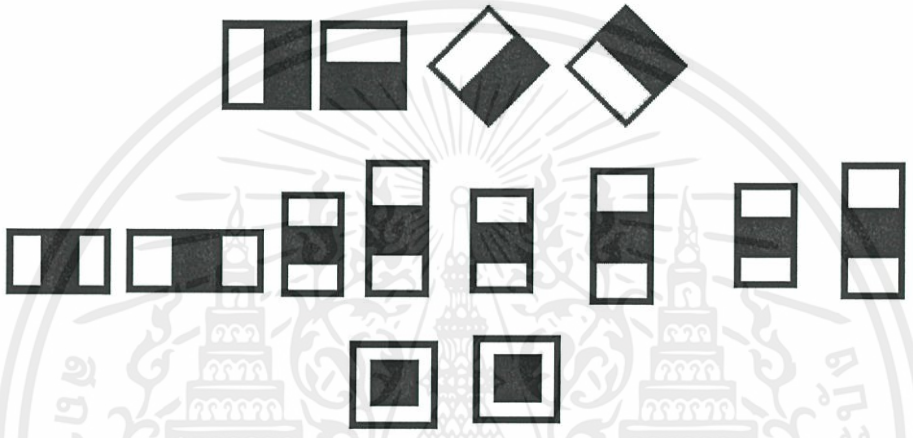


รูปที่ 2.7 การแปลงภาพสีอาร์จีบีเป็นภาพระดับสีเทา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 ลักษณะเด่นของฮาร์

ลักษณะเด่นของฮาร์เป็นคุณสมบัติที่ใช้รูปเหลี่ยมในการตรวจจับวัตถุในภาพดิจิทัล ในการวิเคราะห์วัตถุภายในภาพดิจิทัล ด้วยวิธีนี้จะใช้ความเข้มของสีในพิกเซลที่อยู่ติดกันมาพิจารณา เช่น ในการหาดวงตานั้นเป็นที่รู้กันดีว่าภายในบริเวณตาดำจะมีความเข้มมากกว่าบริเวณที่เป็นตาขาว และตาของมนุษย์ โดยทั่วไปนั้นจะประกอบไปด้วยสองข้างทั้งซ้าย และขวา ดังนั้นการหาดวงตาจึงหาจากบริเวณที่มีสีเข้มกว่าบริเวณโดยรอบซึ่งมีสองกลุ่มในแนวนอน เป็นต้น



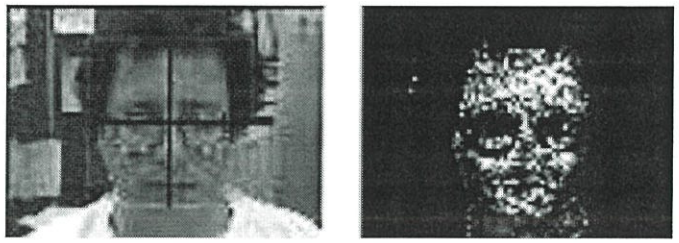
รูปที่ 2.8 รูปแบบของรูปเหลี่ยมที่ใช้ในการตรวจจับวัตถุด้วยลักษณะเด่นของฮาร์

สำหรับการสร้างรูปแบบของลักษณะเด่นของฮาร์ในการตรวจจับวัตถุที่ต้องการนั้น จะต้องใช้ภาพตัวอย่างจำนวนมากในการสร้างรูปแบบที่ต้องการ โดยใช้ทั้งภาพที่มี และไม่มีวัตถุที่ต้องการ จากนั้นจึงใช้วิธี AdaBoost ซึ่งวิธีนี้จะเป็นการเพิ่มประสิทธิภาพของกลุ่มตัวอย่างข้อมูล โดยจะทำการปรับน้ำหนักของตัวอย่างที่ใช้ในการแบ่งกลุ่มข้อมูลด้วยการลดน้ำหนักของตัวอย่างที่สามารถทำได้ดี และไปเพิ่มน้ำหนักสำหรับตัวอย่างที่มีความยากในการแบ่งกลุ่ม หลังจากนั้นจะได้รูปเหลี่ยมที่ใกล้เคียงกัน และไม่ใกล้เคียงกันเพื่อใช้ในการจัดประเภทของวัตถุที่เราต้องการ

2.8 CAMSHIFT algorithm

CAMSHIFT เป็นวิธีการหนึ่งที่ใช้ในการติดตามวัตถุที่เป็นภาพเคลื่อนไหว โดยจะรับอินพุตมาจากกล้องวีดีโอ แล้วนำมาเข้ากระบวนการ CAMSHIFT เพื่อสามารถทำการตรวจจับการเคลื่อนที่ของสิ่งที่เราสนใจได้ ซึ่ง CAMSHIFT เป็นวิธีการที่ง่าย และมีประสิทธิภาพในการคำนวณใบหน้า และการติดตามวัตถุที่มีสี ซึ่งในการติดตามจะทำการติดตามวัตถุที่ทำการเลือกหรือวัตถุที่สนใจ โดยใช้ค่าสีมาทำการติดตาม จะเห็นได้จากกราฟ Histogram ที่แสดงที่ความสว่างของภาพซึ่งอัลกอริธึมนี้จะติดตามวัตถุที่มีสีใกล้เคียงกับกราฟฮิสโตแกรมของวัตถุที่ถูกเลือกนั่นเอง

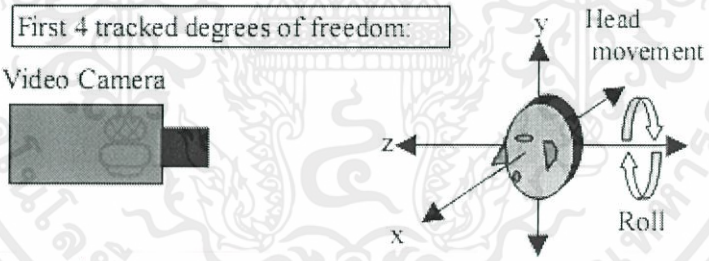
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 ภาพใบหน้าจากวีดีโอ และภาพ Gray Scale ของวัตถุที่ถูกเลือก

รูปที่ 2.10 การติดตามใบหน้าตามทิศทางที่กำหนด

ดังนั้น CAMSHIFT จะให้การคำนวณที่มีประสิทธิภาพ และมีเครื่องมือที่ง่าย ซึ่งจะมีการติดตามที่เป็นอิสระกันอยู่ 4 ทิศทาง ได้แก่ แกน X, แกน Y, แกน Z และแกนหมุน ดังรูปที่ 2.11



รูปที่ 2.11 การติดตามใบหน้าตามแกน X, Y, Z และแกนหมุน

2.9 การหาขอบของภาพ (Edge Detection)

การหาขอบของภาพ คือ การหาเส้นขอบของวัตถุ โดยผลลัพธ์ที่ได้อาจนำไปใช้ในการวิเคราะห์ และตีความเพื่อหาความหมายของวัตถุต่อไป การหาขอบของภาพนั้นกระทำโดยการหาความต่างของความเข้มแสงระหว่างจุดภาพกับจุดภาพที่ใกล้เคียงกัน หากค่าของความเข้มระหว่างจุดภาพมีค่าที่ต่างกันมาก ๆ ก็จะทำให้ได้ขอบที่ชัดเจน ในทางตรงกันข้ามหากค่าของความเข้มระหว่างจุดภาพไม่มากนักก็จะทำให้ได้ขอบที่ไม่ชัดเจน โดยการหาขอบของภาพนั้นสามารถแบ่งได้วิธีหลักได้สองวิธีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.1 วิธีเกรเดียนต์ (Gradient method)

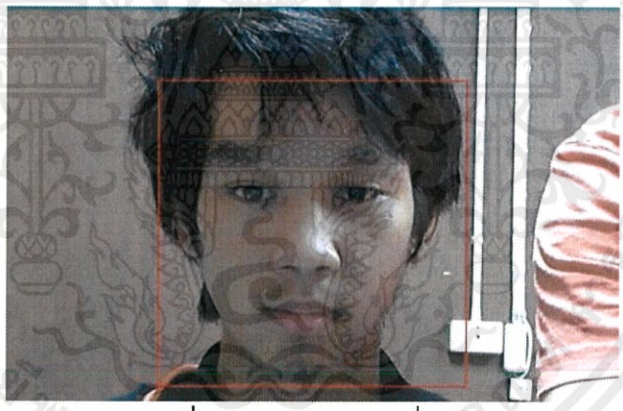
วิธีนี้เป็นการหาขอบภาพจากค่าอนุพันธ์อันดับหนึ่งของภาพ โดยดูจากค่าสูงสุด และต่ำสุด โดยจุดที่เป็นขอบจะมีอยู่เหนือค่าขีด

2.9.2 วิธีลาปลาเซียน (Laplacian method)

วิธีนี้เป็นการหาขอบของภาพจากค่าอนุพันธ์อันดับที่สองของภาพ โดยจะใช้จุดที่ค่าของ y มีค่าเป็นศูนย์หรือจุดตัดแกน x ซึ่งวิธีนี้จะใช้เวลาในการหาขอบของภาพมากกว่าวิธีเกรเดียนต์

2.10 บริเวณที่สนใจ (Region of Interest: ROI)

บริเวณที่สนใจ คือ บริเวณที่มีข้อมูลที่ระบบสนใจซึ่งอาจอยู่ในบริเวณใดก็ได้ในภาพ โดยบริเวณที่ระบบสนใจจะถูกตีกรอบล้อมรอบด้วยรูปเหลี่ยมใดๆ เพื่อที่จะนำภาพภายในบริเวณที่ถูกตีกรอบมาวิเคราะห์ และประมวลผลเพื่อหาข้อมูลที่ระบบสนใจ โดยการกระทำต่อบริเวณนั้นจะไม่มีผลกระทบกับส่วนอื่นๆ ที่อยู่ภายในภาพ



รูปที่ 2.12 แสดงบริเวณที่สนใจ

2.11 ค่าขีดแบ่ง

ใช้เพื่อเปลี่ยนทิศทางของภาพ โดยทำการแยกกลุ่มของภาพออกเป็นส่วน ๆ โดยการแปลงภาพระบบระดับสีเทา (Grayscale) ซึ่งมีค่าความเข้มของสี (intensity) อยู่ระหว่าง 0-255 ให้เป็นภาพที่มีค่าความเข้มสี (intensity) เพียงสองระดับ (Binary Image) โดยมีเงื่อนไขว่า ถ้าความเข้มแสงของจุดภาพใดมีค่าต่ำกว่าหรือเท่ากับค่า threshold ให้จุดภาพนั้นมีค่า เป็น 0 = สีดำ ถ้ามีค่าสูงกว่าค่าขีดแบ่ง ให้จุดภาพนั้นมีค่าเป็น 1 = สีขาว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.12 Emgu CV

Emgu CV คือไลบรารี่ของฟังก์ชันใช้สำหรับการพัฒนางานในด้านประมวลผลภาพ และคอมพิวเตอร์วิทัศน์แบบเรียลไทม์ ซึ่งมีให้เลือกกว่าห้าร้อยฟังก์ชัน ด้วยความที่ Emgu CV เป็นไลบรารี่ที่ไม่มีข้อจำกัดในเรื่องของอาร์ดแวร์จึงทำให้สามารถใช้งานได้หลายภาษาไม่ว่าจะเป็นภาษา C# , VB, VC++ และ IronPython เป็นต้น ใน Emgu CV นั้นจะประกอบไปด้วยชุดคำสั่งเกี่ยวกับระบบการจดจำใบหน้า การจำแนกวัตถุ และการติดตามการเคลื่อนไหวของวัตถุ เป็นต้น

2.13 Visual basic C#

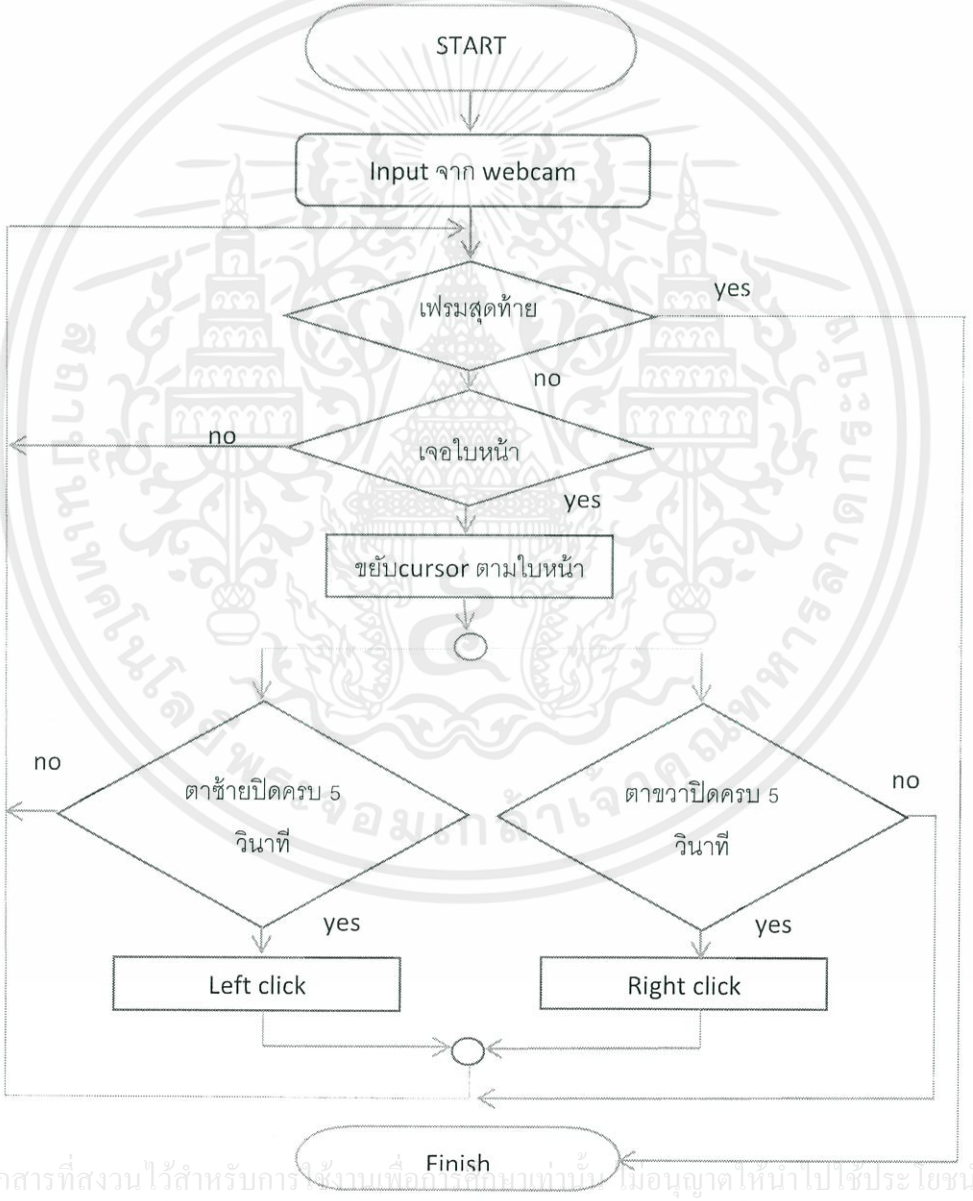
Visual basic C# คือ ภาษาคอมพิวเตอร์ประเภท object-oriented programming พัฒนาโดย Microsoft โดยมีจุดมุ่งหมายในการรวมความสามารถการคำนวณของ C++ ด้วยการโปรแกรมง่ายกว่าของ Visual Basic โดย C# มีพื้นฐานจาก C++ และเก็บส่วนการทำงานคล้ายกับ Java C# ได้รับการออกแบบให้ทำงานกับ .NET platform ของ Microsoft จุดมุ่งหมายคืออำนวยความสะดวกในการแลกเปลี่ยนสารสนเทศ และบริการผ่านเว็บ และทำให้ผู้พัฒนาสร้างโปรแกรมประยุกต์ในขนาดกระทัดรัด C# ทำให้โปรแกรมง่ายขึ้นผ่านการใช้ Extensible Markup Language (XML) และ Simple Object Access Protocol (SOAP) ซึ่งยอมให้เข้าถึง object ของโปรแกรมหรือ method โดยปราศจากความต้องการให้ผู้เขียนโปรแกรมเขียนคำสั่งเพิ่มในแต่ละขั้นตอน เนื่องจากผู้เขียนโปรแกรมสามารถสร้างบนคำสั่งที่มีอยู่ แทนที่การคัดลอกซ้ำ C# ภาษา C# ถูกพัฒนาขึ้นโดยเป็นส่วนหนึ่งในการพัฒนาโครงสร้างพื้นฐานของ .NET Framework เป็นการการนำข้อดีของภาษาต่างๆ (เช่น ภาษา Delphi , ภาษา C++) มาปรับปรุงเพื่อให้มีความเป็น OOP (โปรแกรมเชิงวัตถุ) มากขึ้น ขณะเดียวกันก็ลดความซับซ้อนในโครงสร้างของภาษาลง (เรียบง่ายกว่าภาษา C++) และมีสิ่งที่เกินความจำเป็นน้อยลง (เมื่อเทียบกับ Java).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การวิเคราห์และออกแบบระบบ

หลังจากได้เรียนรู้ทฤษฎีต่าง ๆ ที่เกี่ยวข้องในการพัฒนาโปรแกรมจากบทที่ 2 แล้วในบทนี้จะกล่าวถึงขั้นตอนการวิเคราะห์ และออกแบบโปรแกรม โดยนำทฤษฎีต่าง ๆ เหล่านั้นมาประยุกต์ใช้

3.1 การออกแบบแผนผังโครงสร้างของระบบ



รูปที่ 3.1 แผนผังแสดงโครงสร้างการทำงานของโปรแกรมทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบหน้าตาของโปรแกรม

ออกแบบหน้าตาโปรแกรมให้มี 4 ส่วนหลักได้แก่

3.2.1 Live cam

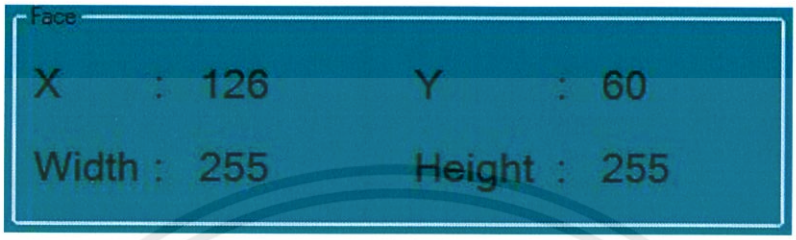
เป็นส่วนที่แสดงภาพวีดีโอแบบเรียลไทม์ มีปุ่ม Start และ Calibrate ตามรูปที่ 3.2

รูปที่ 3.2 แสดงส่วนของ LiveCam

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 Face

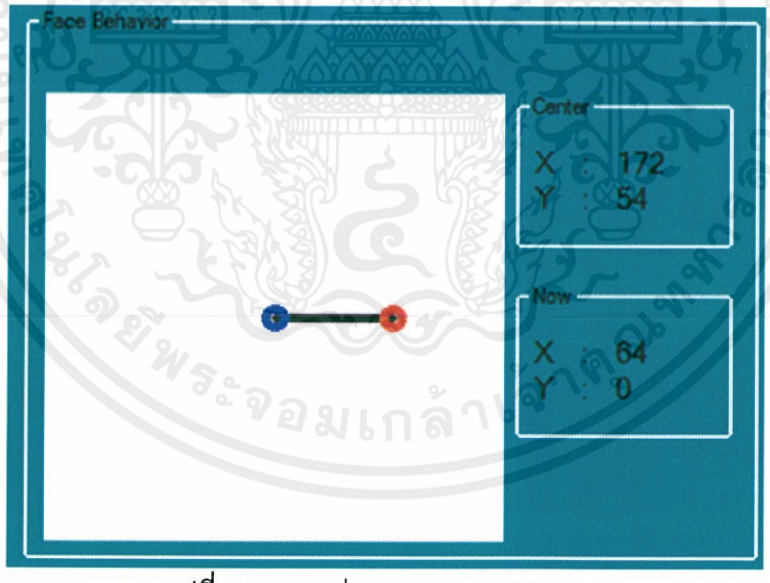
ในส่วนนี้จะให้แสดงค่า 4 ค่า โดย ค่า X และ Y แสดงพิกัดมุมซ้ายบนของกรอบหน้า โดยให้ตำแหน่งซ้ายบนของภาพเป็นจุด (0,0) ค่า Width แสดงค่าความกว้างของใบหน้า และค่า Height แสดงค่าความสูงของใบหน้า



รูปที่ 3.3 แสดงส่วนของข้อมูล Face

3.2.3 Face Behavior

สร้างจุดวงกลมสีน้ำเงินไว้เพื่อเป็นจุดอ้างอิง และให้แสดงค่าพิกัด X และ Y ในกรอบ Center ซึ่งเป็นตำแหน่งเริ่มต้นของใบหน้า และสร้างจุดวงกลมสีแดงเพื่อแสดงตำแหน่งปัจจุบันของใบหน้า โดยให้ X และ Y ในกรอบ Now คือค่าผลต่างของตำแหน่งปัจจุบันกับจุดเริ่มต้น

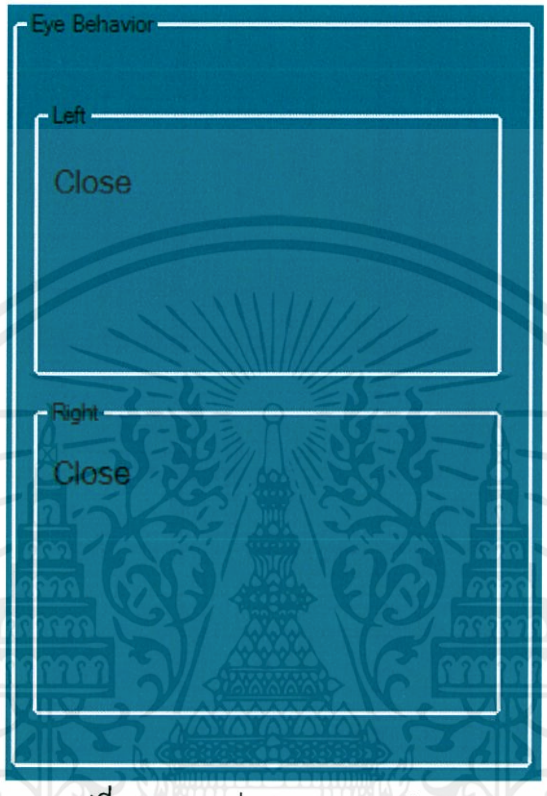


รูปที่ 3.4 แสดงส่วนของ Face Behavior

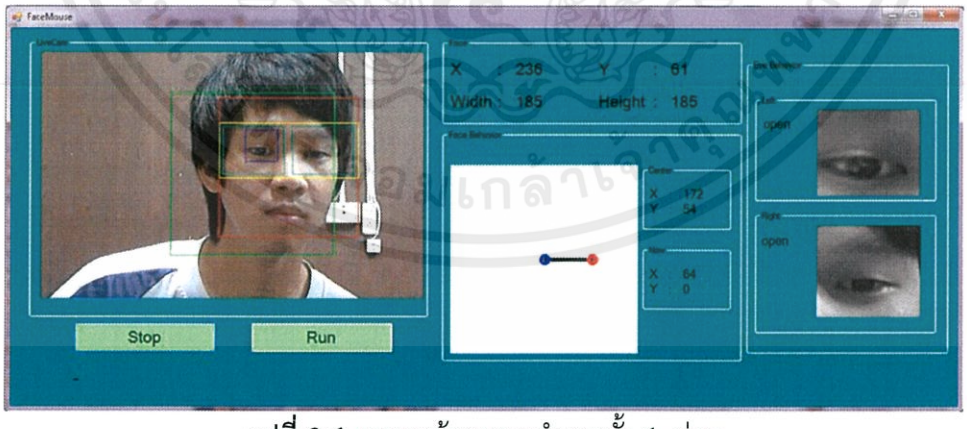
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 Eye Behavior

ส่วนนี้จะมีสองกรอบของตาซ้า ยและตาขวา โดยจะแสดงสถานะของตาว่าปิดหรือเปิดโดย หากเปิดจะแสดงว่า Open และปิดจะแสดงว่า Close



รูปที่ 3.5 แสดงส่วนของ Eye Behavior



รูปที่ 3.6 แสดงหน้าตาการทำงานทั้ง 4 ส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การพัฒนาและการทดสอบระบบ

ในบทนี้จะกล่าวถึงการพัฒนา และทดสอบระบบการติดตามใบหน้า และการสั่งงานเมาส์ โดยจะแบ่งเป็นสองส่วนได้แก่ การพัฒนาโปรแกรม และการทดสอบโปรแกรม

4.1 การพัฒนาโปรแกรม

4.1.1 โปรแกรมและเครื่องมือที่ใช้ในการพัฒนาระบบ

ซอฟต์แวร์:

- Microsoft Visual C# 2010
- EmguCV library
- ระบบปฏิบัติการ Windows 7

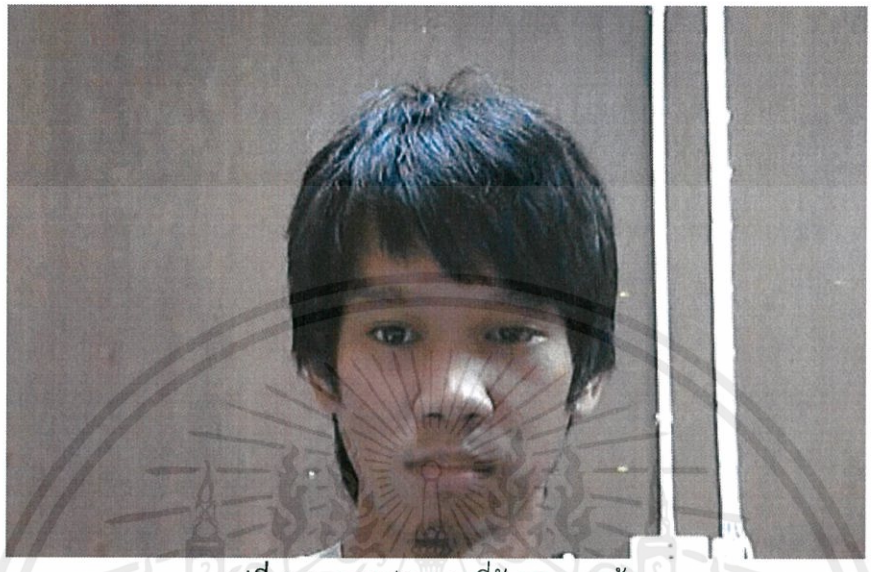
ฮาร์ดแวร์:

- Notebook
- Webcam
- LCD 14“

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 พิจารณาในส่วนของการติดตามใบหน้า

1. รับภาพเฟรมแรกมาจากกล้อง และนำมาพิจารณา



รูปที่ 4.1 ภาพเฟรมแรกที่รับมาจากกล้อง

2. เริ่มค้นหาใบหน้า โดยใช้ลักษณะเด่นของฮาร์มาช่วยเมื่อเจอใบหน้าแล้ว จะทำการตีกรอบสี่เหลี่ยมรอบใบหน้าด้วยสีแดง และจดจำตำแหน่งนี้ไว้เพื่อใช้เป็นกรอบอ้างอิง

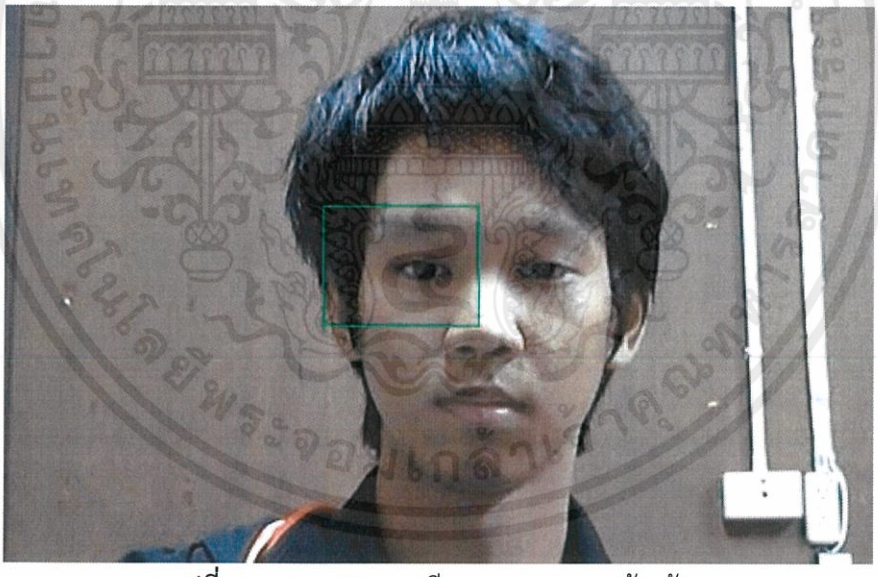
รูปที่ 4.2 ภาพแสดงการตีกรอบบริเวณที่ใบหน้า

3. เริ่มค้นหาแถบของดวงตาโดยใช้ลักษณะเด่นของฮาร์มาช่วย โดยเมื่อเจอแถบดวงตาแล้ว จะทำการตีกรอบสี่เหลี่ยมรอบแถบดวงตาด้วยสีเหลือง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.3 ภาพแสดงการตีกรอบบริเวณแถบดวงตา

4. เริ่มค้นหาดวงตาข้างซ้าย โดยใช้ลักษณะเด่นของฮาร์มาช่วยเมื่อเจอดวงตาข้างซ้ายแล้ว จะทำการตีกรอบสี่เหลี่ยบรอบดวงตาข้างซ้ายด้วยสีฟ้า



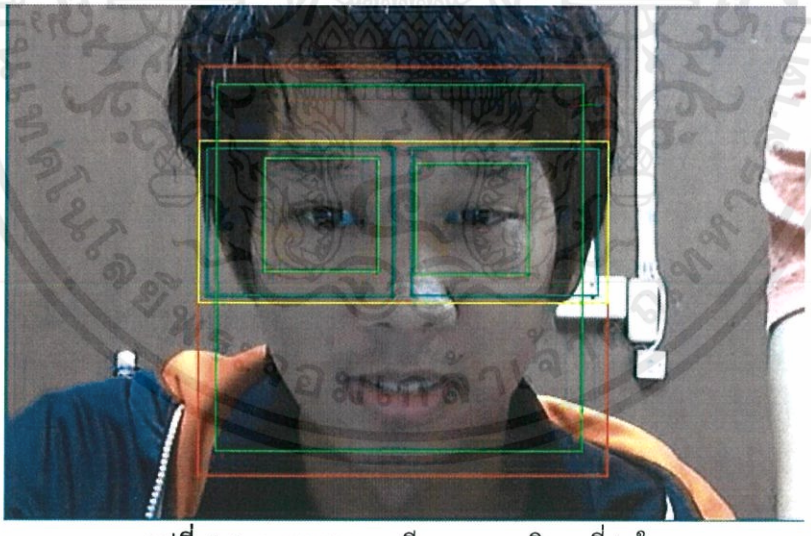
รูปที่ 4.4 ภาพแสดงการตีกรอบรอบดวงตาข้างซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. เริ่มค้นหาดวงตาข้างขวา โดยใช้ลักษณะเด่นของฮาร์มาช่วยเมื่อเจอดวงตาข้างขวาแล้ว จะทำการตีกรอบสี่เหลี่ยมรอบดวงตาข้างซ้ายด้วยสีฟ้า

รูปที่ 4.5 ภาพแสดงการตีกรอบรอบดวงตาข้างขวา

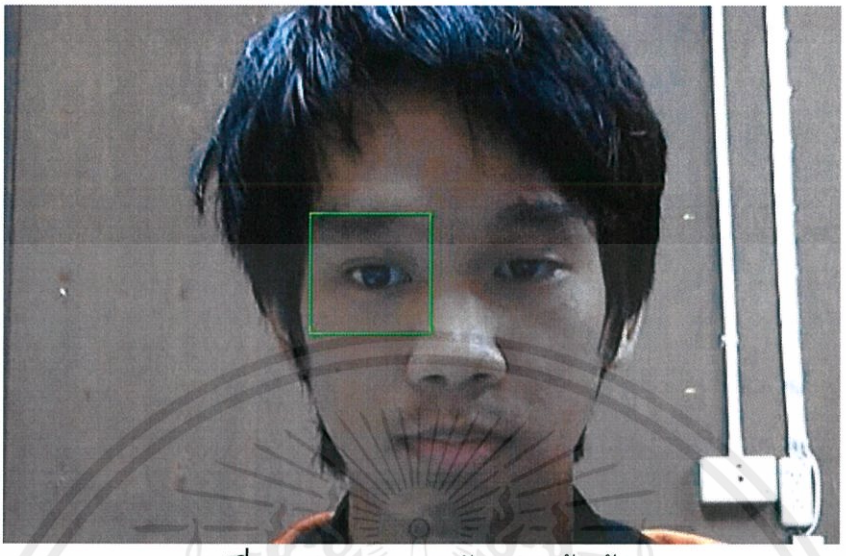
6. นำบริเวณที่สนใจทั้งหมดมารวมกันเพื่อเป็นจุดอ้างอิงในการขยับเมาส์



รูปที่ 4.6 ภาพแสดงการตีกรอบทุกบริเวณที่สนใจ

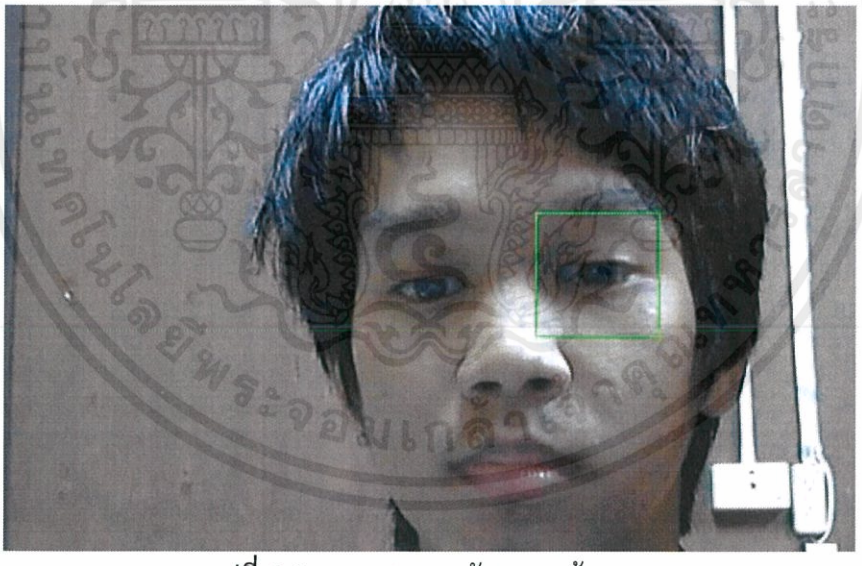
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ดึงภาพเฉพาะบริเวณตาข้างซ้ายออกมาเพื่อพิจารณาสถานะ



รูปที่ 4.7 ภาพแสดงการจับดวงตาข้างซ้าย

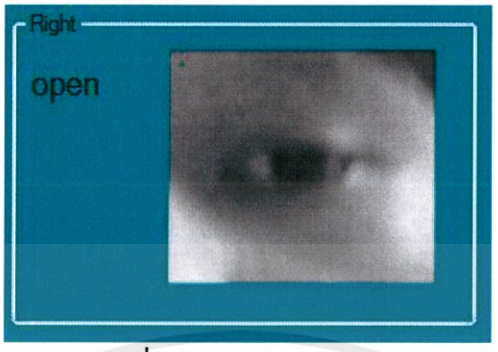
8. .ดึงภาพเฉพาะบริเวณตาข้างขวาออกมาเพื่อพิจารณาสถานะ



รูปที่ 4.8 ภาพแสดงการจับดวงตาข้างขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.เมื่อหาตาทั้งสองข้างเจอแล้วนำมาพิจารณาว่าตาข้างไหนมีสถานะเปิดตาหรือปิดตา



รูปที่ 4.9 สถานะตาข้างขวา

รูปที่ 4.10 สถานะตาข้างซ้าย

4.2 การทดสอบโปรแกรม

การทดสอบโปรแกรมเป็นส่วนสำคัญ เนื่องจากในส่วนนี้เป็นส่วนแก้ไขข้อผิดพลาดที่เกิดขึ้นจากการเขียนคำสั่งพัฒนาโปรแกรมนี้ขึ้นมา ทำให้เกิดการจำกัดขอบเขตของโปรแกรมให้แคบลงมาอีก ทั้งนี้ก็เพื่อให้ผู้ใช้สามารถใช้ โปรแกรมได้อย่างมีประสิทธิภาพ และเกิดความพึงพอใจสูงสุด

4.2.1 วิธีการทดสอบการทำงานของโปรแกรม

วิธีการทดสอบการทำงานของโปรแกรม อาศัยวิธีการดังนี้

- ทดสอบการทำงานของโปรแกรม ว่าสามารถตรวจจับ และทำการติดตามใบหน้าได้
- ทดสอบการทำงานของโปรแกรม ว่าสามารถบังคับเมาส์ได้ทุกทิศทาง
- ทดสอบการทำงานของโปรแกรม ว่าสามารถคลิกซ้าย และคลิกขวาได้
- ทดสอบการทำงานของโปรแกรมในบริเวณที่มีความสว่างของแสงแตกต่างกัน เพื่อพิสูจน์ว่าความสว่างแสงมีผลต่อการใช้งานโปรแกรมนี้อย่างไร
- ทดสอบการทำงานในการจับหาใบหน้า โดยหาระยะห่างที่ไกลจากกล้องเว็บแคมที่สุดที่ทำให้กล้องสามารถตรวจจับได้
- ทดสอบการทำงานในการจับหาใบหน้า โดยหาระยะห่างที่ใกล้จากกล้องเว็บแคมที่สุดที่

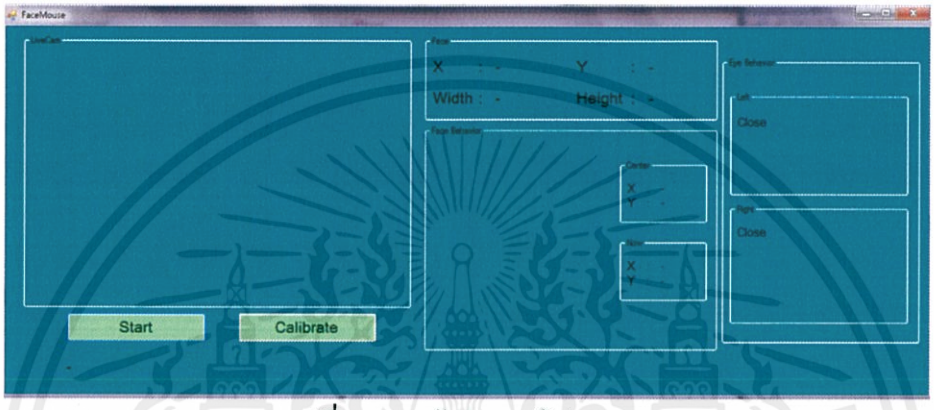
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำให้กล้องสามารถตรวจจับได้

- ทดสอบการทำงานของโปรแกรมว่าสามารถตรวจหาใบหน้าได้ทุกใบหน้า
- ทดสอบการทำงานของโปรแกรมกรณีที่มีใบหน้ามากกว่า 1

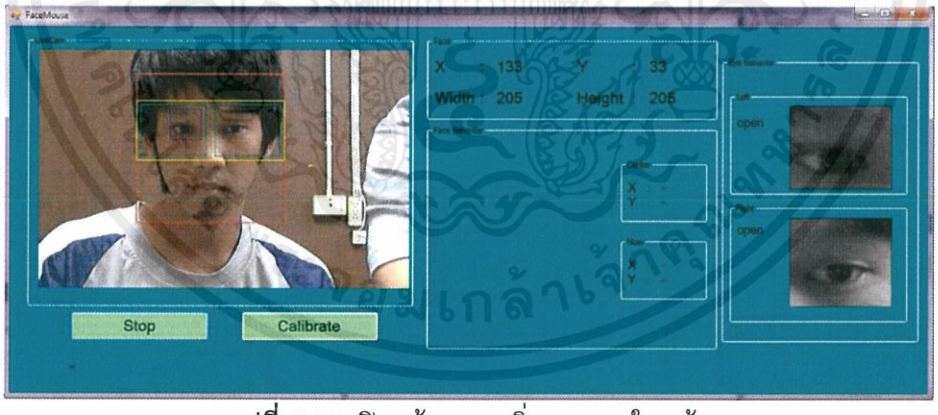
4.2.2 ขั้นตอนการทดสอบการทำงานของโปรแกรม

1. เปิดโปรแกรมขึ้นมา



รูปที่ 4.11 หน้าต่างของโปรแกรม

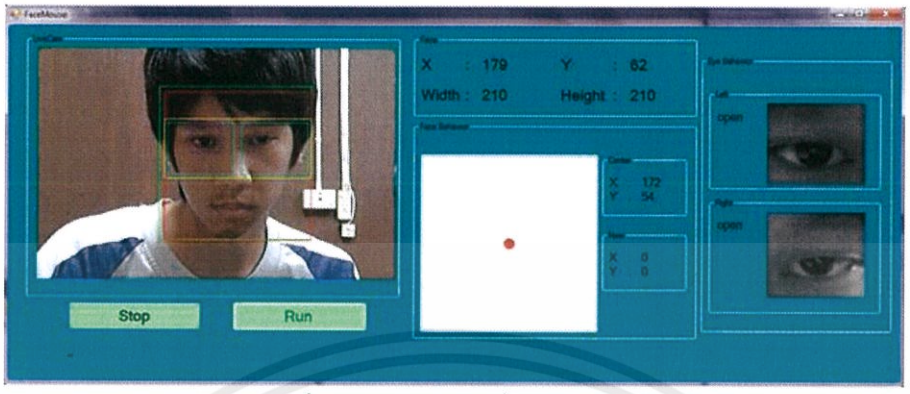
2. กดปุ่ม Start เพื่อเปิดกล้อง และเริ่มการทำงานของโปรแกรม



รูปที่ 4.12 เปิดกล้อง และเริ่มตรวจหาใบหน้า

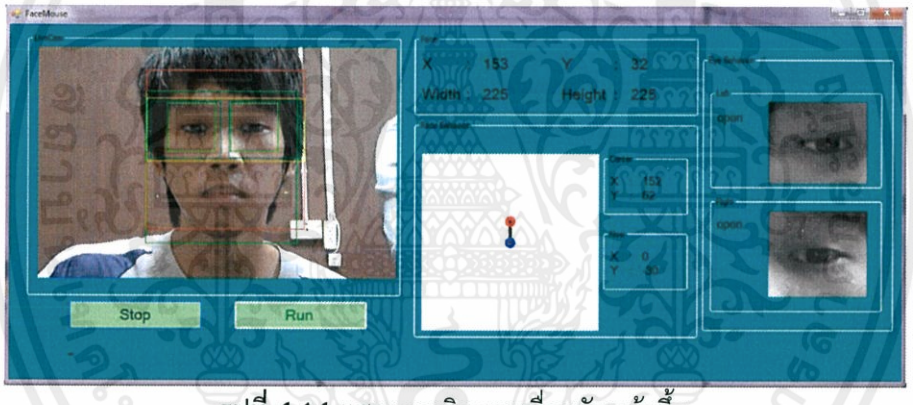
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. กดปุ่ม Calibrate เพื่อเริ่มใช้งานเมาส์



รูปที่ 4.13 โปรแกรมเริ่มการติดตาม

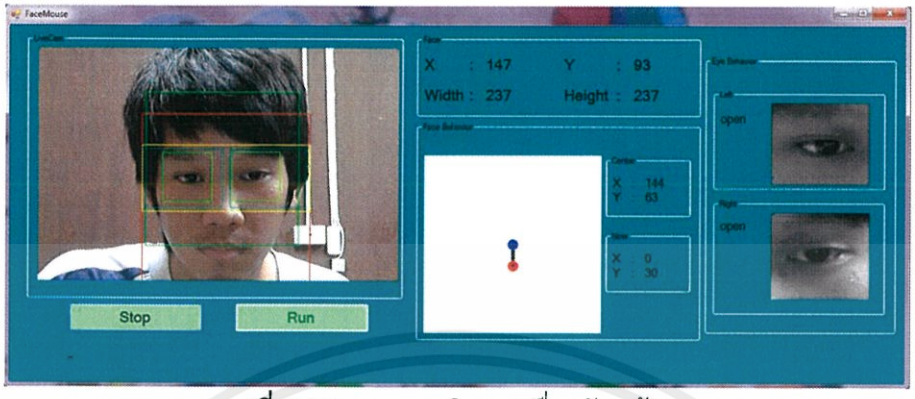
4. ขยับหน้าขึ้นเพื่อเลื่อนเมาส์ขึ้น



รูปที่ 4.14 แสดงการติดตามเมื่อขยับหน้าขึ้น

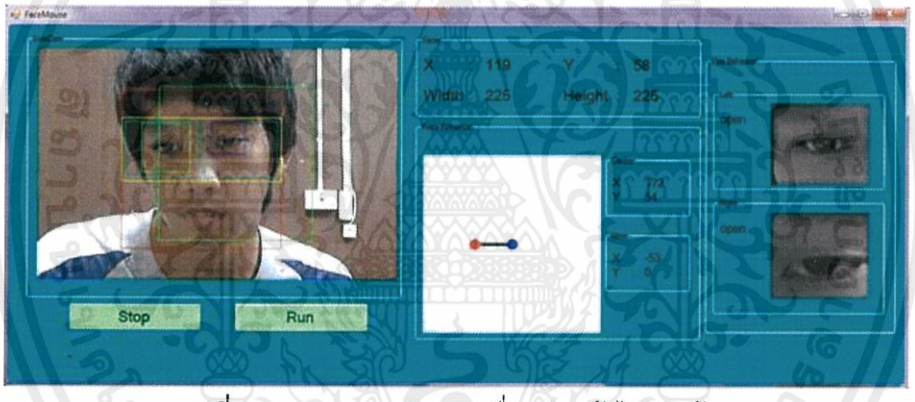
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ขยับหน้าลงเพื่อเลื่อนเมาส์ลง



รูปที่ 4.15 แสดงการติดตามเมื่อขยับหน้าลง

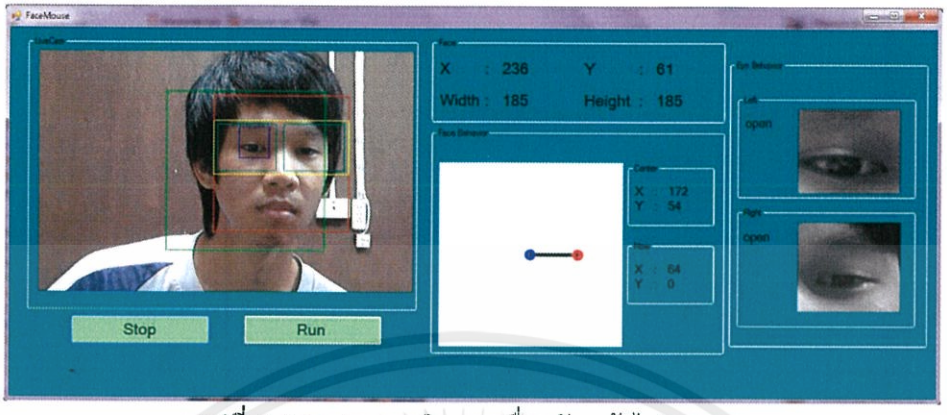
6. ขยับหน้าไปทางซ้ายเพื่อเลื่อนเมาส์ไปทางซ้าย



รูปที่ 4.16 แสดงการติดตามเมื่อขยับหน้าไปทางซ้าย

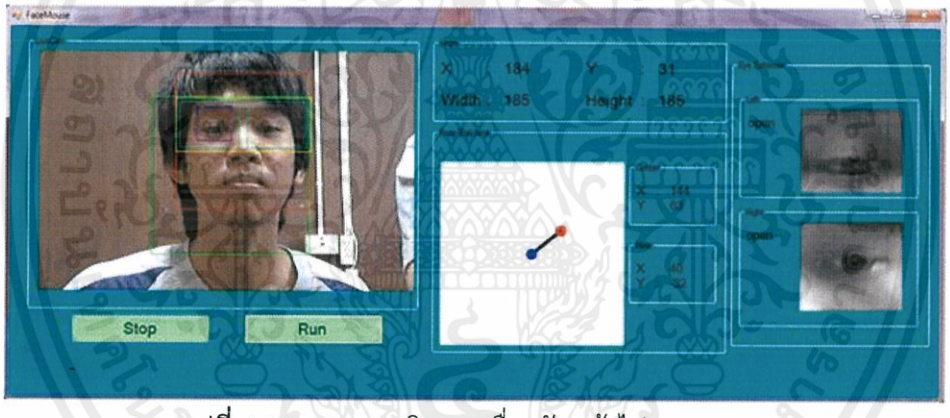
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ขยับหน้าไปทางขวาเพื่อเลื่อนเมาส์ไปทางขวา



รูปที่ 4.17 แสดงการติดตามเมื่อขยับหน้าไปทางขวา

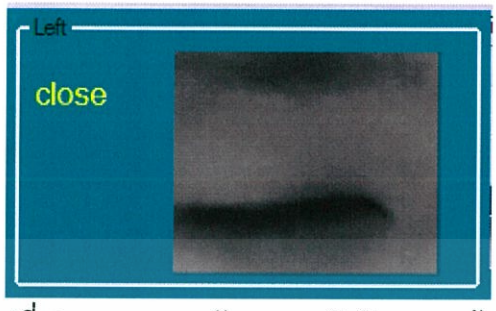
8. ขยับหน้าไปทางขวาบนเพื่อเลื่อนเมาส์ในแนวเฉียง



รูปที่ 4.18 แสดงการติดตามเมื่อขยับหน้าไปทางขวาบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. ปิดตาซ้ายเพื่อคลิกเมาส์ซ้าย

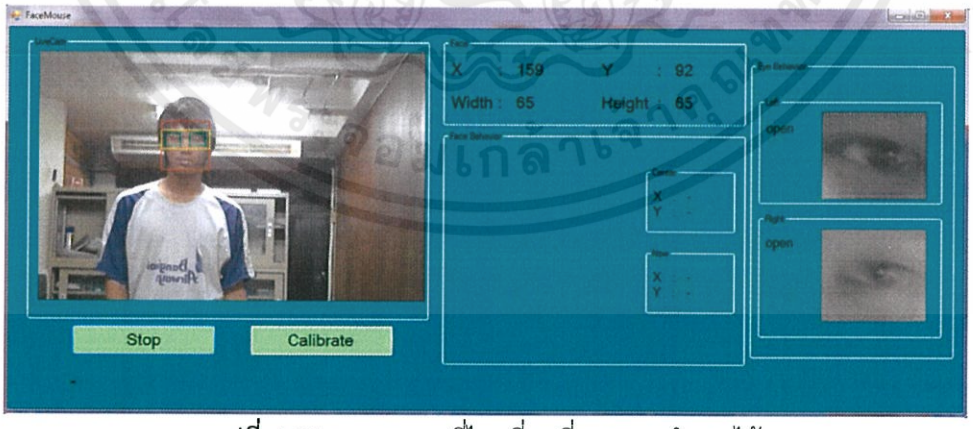


รูปที่ 4.19 แสดงการจับสถานะเปิดปิดของตาซ้าย

10. ปิดตาขวาเพื่อคลิกเมาส์ขวา

รูปที่ 4.20 แสดงการจับสถานะเปิดปิดของตาขวา

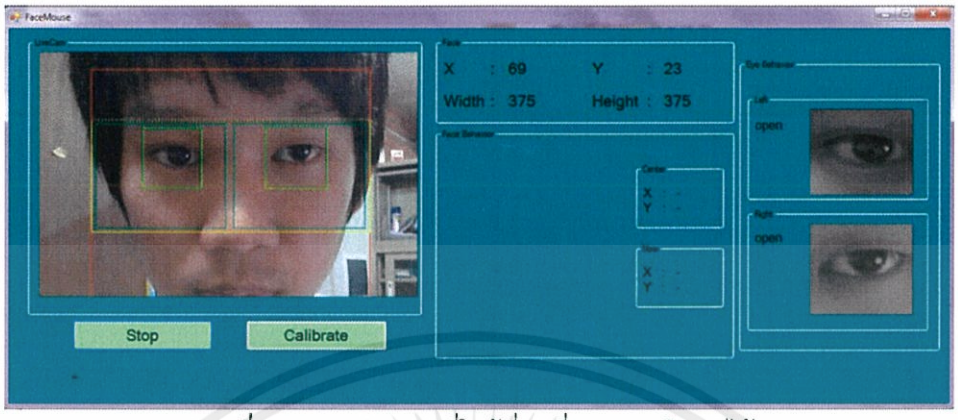
11. ทดสอบระยะที่ไกลที่สุดที่สามารถจับใบหน้าได้



รูปที่ 4.21 แสดงระยะที่ไกลที่สุดที่สามารถทำงานได้

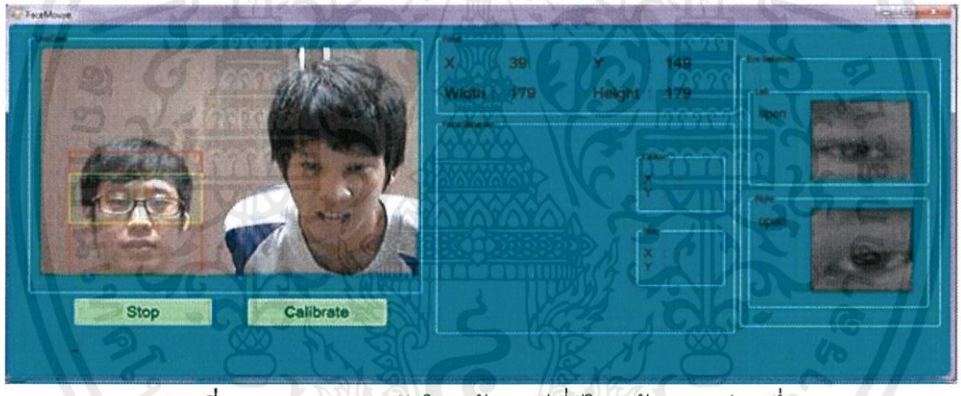
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. ทดสอบระยะที่ใกล้ที่สุดที่สามารถจับใบหน้าได้



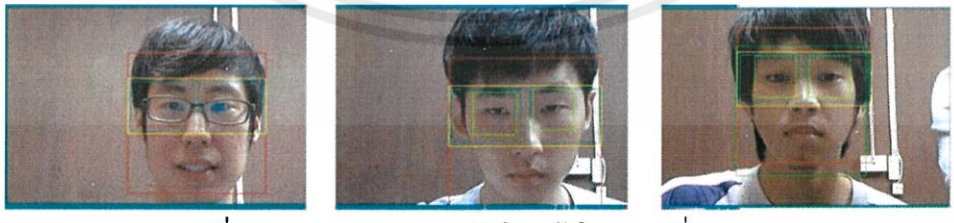
รูปที่ 4.22 แสดงระยะที่ใกล้ที่สุดที่สามารถทำงานได้

13. ทดสอบการจับใบหน้ากรณีที่มีใบหน้ามากกว่า 1



รูปที่ 4.23 แสดงการจับใบหน้ากรณีที่มีใบหน้ามากกว่าหนึ่ง

14. ทดสอบการทำงานของโปรแกรมว่าสามารถตรวจหาใบหน้าได้ทุกใบหน้า



รูปที่ 4.24 แสดงการตรวจจับใบหน้าในบุคคลที่แตกต่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการทดลองและข้อเสนอแนะ

5.1 ผลที่ได้รับ

จากการทดสอบโปรแกรมพบว่า โปรแกรมสามารถระบุบริเวณที่เป็นใบหน้า และดวงตาได้ และทำการติดตามบริเวณใบหน้าเพื่อใช้ในการเลื่อนเคอเซอร์ และโปรแกรมยังสามารถระบุได้ว่าเราปิดตาข้างไหนอยู่ ซึ่งการปิดตาสามารถใช้ในการคลิกเมาส์ทั้งซ้าย และขวาได้ ทำให้ผู้ใช้ที่ไม่สามารถขยับแขนหรือผู้ป่วยอัมพาตสามารถใช้งานคอมพิวเตอร์ได้อย่างคนปกติ อีกทั้งต้นทุนที่ใช้มีราคาไม่สูงมาก หาซื้อได้ตามร้านขายอุปกรณ์คอมพิวเตอร์ทั่วไป

5.2 ปัญหาและอุปสรรค

1. สมาชิกในกลุ่มไม่มีความรู้พื้นฐานของภาษา C# มาก่อน และไม่เคยใช้งานไลบรารีที่เกี่ยวข้องกับการประมวลผลภาพทำให้ใช้เวลาในการศึกษาเป็นระยะเวลานานจึงจะสามารถพัฒนาโครงงานนี้ได้

2. เนื้อหาเกี่ยวกับทฤษฎีการประมวลผลภาพเป็นเนื้อหาที่ยาก และใช้เวลาเรียนรู้นาน

3. กล้องเว็บแคมที่ติดมากับคอมพิวเตอร์โน๊ตบุ๊คแต่ละรุ่น มีความละเอียดภาพไม่เท่ากัน บางรุ่นความละเอียดต่ำทำให้มีปัญหาต่อการประมวลผล

4. ปัจจัยภายนอก เช่น ความสว่างของหลอดไฟ ความสว่างของแสงที่ส่องเข้ามา อาจมีผลต่อการประมวลผลภาพได้

5.3 การแก้ปัญหา

1. เนื่องจากระยะเวลาในการพัฒนาโปรแกรมมีอย่างจำกัด ทำให้ต้องศึกษาการใช้งานไปพร้อมกับการพัฒนาโปรแกรม ซึ่งสมาชิกในกลุ่มเลือกใช้ Microsoft Visual C# ซึ่งออกแบบให้ง่ายต่อการใช้งาน และมีเครื่องมือให้เลือกใช้อย่างสะดวก และใช้ EmguCV library ซึ่งมีข้อมูลที่สามารถค้นหาเพิ่มเติมได้จากทั้งทางหนังสือ และอินเตอร์เน็ตเป็นจำนวนมาก

2. เนื่องจากทฤษฎีการประมวลผลภาพมีความยาก และมีปริมาณมาก เราจึงแบ่งเนื้อหาให้สมาชิกแต่ละคนอย่างละเท่า ๆ กัน เพื่อให้เกิดความรวดเร็วต่อการศึกษา

3. ใช้กล้องเว็บแคมยี่ห้อ Logitech

4. จัดสถานที่ทดลอง ที่มีแสงสว่างเพียงพอ และไม่มีแสงแดดส่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 ข้อเสนอแนะ

1. ในการติดตามใบหน้าอาจใช้เทคนิคอื่นเพิ่มเติมเพื่อให้การติดตามลักษณะเด่นมีความแม่นยำมากขึ้น

2. ควรเพิ่มความเสถียรในการขยับเมาส์ให้มากขึ้นเพื่อให้ง่ายต่อการใช้งาน

3. ควรเพิ่มความสวยงามของหน้าตาโปรแกรมเพื่อให้ง่ายต่อการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] John C. Russ, J. Christian Russ.Introduction to image processing and analysis:Boca Raton : CRC Press, 2008

[2] สมเกียรติ อุดมหรรษากุล.การประมวลผลภาพดิจิตอลเบื้องต้น = Fundamentals of digital image processing:กรุงเทพฯ : ท้อป, 2554

[3] กิตินันท์ พลสวัสดิ์.คู่มือเรียนและใช้งาน Visual C# 2010 ฉบับสมบูรณ์:นนทบุรี : ไอดีซีฯ, 2554

[4] Andrew Duchowski.Eye tracking methodology:London : Springer, 2007

[5] จุฑามาส อุดมศิริรัตน์.การประมวลผลภาพดิจิตอล = Image processing,2531

[6] Petrou, Maria.Image processing : the fundamentals:Chichester : John Wiley, 2010

[7] Awcock, G.J.Applied image processing:New York : McGraw-Hill, 1996

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

Source code

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Emgu.CV;
using Emgu.CV.CvEnum;
using Emgu.CV.Structure;
using Emgu.CV.UI;
using System.Runtime.InteropServices;
namespace ImageProcessingProjectFinal
{
    public partial class Form1 : Form
    {
        Capture capture = new Capture(0);
        Timer timer = new Timer();
        bool isStart = false;
        HaarCascade haarFace = new
HaarCascade(@"haarcascades\haarcascade_frontalface_alt_tree.xml");
        HaarCascade haarEye= new HaarCascade(
@"haarcascades\haarcascade_eye.xml");
        HaarCascade haarLeftEye = new
HaarCascade(@"haarcascades\haarcascade_lefteye_2splits.xml");
        HaarCascade haarRightEye = new
HaarCascade(@"haarcascades\haarcascade_righteye_2splits.xml");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        HaarCascade haarEyePair = new
HaarCascade(@"haarcascades\haarcascade_mcs_eyepair_big.xml");
        bool isLeftClick = false;
        bool isRightClick = false;
        int LeftCount = 0;
        int RightCount = 0;
        int LeftOpenCount = 0;
        int RightOpenCount = 0;
        int ClickCount = 5;
        int stopValue = 10;
        int openFactor = 2;
        int openScale = 10;
        int closeScale = 15;
        int closeFactor = 0;
        double sensitiveX = 2;
        double sensitiveY = 1;
        Point centerPoint = new Point(125, 125);
        Point previousPoint = new Point(-1, -1);
        Point nowPoint = new Point(0, 0);
        Rectangle centerRect;
        bool isCalibrate = false;

        [DllImport("user32.dll", CharSet = CharSet.Auto, CallingConvention =
CallingConvention.StdCall)]
        public static extern void mouse_event(uint dwFlags, uint dx, uint dy, uint
cButtons, uint dwExtraInfo);

        private const int MOUSEEVENTF_LEFTDOWN = 0x02;
        private const int MOUSEEVENTF_LEFTUP = 0x04;
        private const int MOUSEEVENTF_RIGHTDOWN = 0x08;
        private const int MOUSEEVENTF_RIGHTUP = 0x10;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
public Form1()
{
    InitializeComponent();
    timer.Interval = 1;
    timer.Tick += timer_Tick;
}

void timer_Tick(object sender, EventArgs e)
{
    Run();
}
private void btnStart_Click(object sender, EventArgs e)
{

    if (!isStart)
    {
        timer.Start();
        btnStart.Text = "Stop";
        btnCalibrate.Enabled = true;
    }
    else
    {
        timer.Stop();
        btnStart.Text = "Start";
        btnCalibrate.Enabled = false;
    }
    isStart = !isStart;
}
private void btnCalibrate_Click(object sender, EventArgs e)
{
    if (isStart&& FaceEnable())
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MainControl.setCenter(int.Parse(lbFaceX.Text), int.Parse(lbFaceY.Text));
            isCalibrate = true;
            btnCalibrate.Text = "Run";
            btnCalibrate.Enabled = false;
            lbCenterX.Text = MainControl.centerX.ToString();
            lbCenterY.Text = MainControl.centerY.ToString();
        }
    }
    private bool FaceEnable()
    {
        int x = int.Parse(lbFaceX.Text);
        int y = int.Parse(lbFaceY.Text);
        int w = int.Parse(lbFaceWidth.Text);
        int h = int.Parse(lbFaceHeight.Text);
        return !(x == -1 && y == -1 && w == -1 && h == -1);
    }
    private void Run()
    {
        try
        {
            Image<Bgr, byte> img;
            img = capture.QueryFrame().Flip(FLIP.HORIZONTAL);

            MCvAvgComp face = new MCvAvgComp();
            Image<Gray, byte> grayImg = img.Convert<Gray, byte>();
            //Face
            var faces = grayImg.DetectHaarCascade(haarFace, 1.1, 1,
HAAR_DETECTION_TYPE.FIND_BIGGEST_OBJECT, new Size(img.Width / 10, img.Height /
10))[0];
            if (!isCalibrate)
            {
                #region calibrate
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if (img != null)
            {
                bool isfound = false;
                if (faces.Length > 0)
                {
                    face = faces[0];
                    int x = face.rect.X;
                    int y = face.rect.Y;
                    int width = face.rect.Width;
                    int height = face.rect.Height;
                    isfound = true;
                    img.Draw(face.rect, new Bgr(0, 0, 255), 1);
                    SetFaceLabel(x, y, width, height);
                    centerRect = face.rect;

                }
                if (!isfound) SetFaceLabel(-1, -1, -1, -1);
                imgBoxCapture.Image = img;
            }

            #endregion
        }
        else
        {
            #region Run
            if (img != null)
            {
                img.Draw(centerRect, new Bgr(0, 255, 0), 1);
                //var faces = grayImg.DetectHaarCascade(haar)[0];
                //MCvFont f = new MCvFont(FONT.CV_FONT_HERSHEY_COMPLEX,
1.0, 0.5);
                bool isfound = false;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if (faces.Length > 0)
            {

                face = faces[0];
                int x = face.rect.X;
                int y = face.rect.Y;
                int width = face.rect.Width;
                int height = face.rect.Height;
                isfound = true;
                img.Draw(face.rect, new Bgr(0, 0, 255), 1);
                SetFaceLabel(x, y, width, height);

            }
            //var eyes = grayimg.detecthaarcascade(haareye, 1.1, 4,
haar_detection_type.do_canny_pruning, new size(img.width / 8, img.height / 8))[0];
            //foreach (var eye in eyes)
            //{
            //    int x = eye.rect.x;
            //    int y = eye.rect.y;
            //    int width = eye.rect.width;
            //    int height = eye.rect.height;
            //    img.draw(eye.rect, new bgr(255,0, 0), 1); //วาดสี่เหลี่ยม
            //    //break;
            //}
            if (isfound)
            {
                MouseFoundFace();
            }
            else
            {
                SetFaceLabel(-1, -1, -1, -1);
                MouseNotFoundFace();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }
                imgBoxCapture.Image = img;
            }
            #endregion
        }
        #region Eye
        #region Eye ROI
        Point eyesPairPoint = new Point(face.rect.X, face.rect.Y + (face.rect.Height
* 2 / 11));
        Size eyesPairSize = new Size(face.rect.Width, (face.rect.Height * 2 / 5));
        Rectangle possibleROIeyePair = new Rectangle(eyesPairPoint,
eyesPairSize);
        Point eyesLeftPoint = new Point(face.rect.X + 5,eyesPairPoint.Y+ 5);
        Point eyesRightPoint = new Point(face.rect.X + (face.rect.Width / 2) + 5,
eyesPairPoint.Y + 5);
        Size eyesSize = new Size((face.rect.Width / 2) - 10, eyesPairSize.Height -
10);
        Rectangle possibleROILeftEye = new Rectangle(eyesLeftPoint, eyesSize);
        Rectangle possibleROIRightEye = new Rectangle(eyesRightPoint, eyesSize);

        img.Draw(possibleROIeyePair, new Bgr(0, 255, 255), 1);
        img.Draw(possibleROILeftEye, new Bgr(150, 200, 0), 1);
        img.Draw(possibleROIRightEye, new Bgr(150, 200, 0), 1);
        #endregion
        #region left eye
        //Left Eye
        grayImg.ROI = Rectangle.Empty;
        grayImg.ROI = possibleROILeftEye;
        var leftEyes = grayImg.DetectHaarCascade(haarEye, 1.1, openFactor,
HAAR_DETECTION_TYPE.DO_CANNY_PRUNING, new Size(img.Width / openScale,
img.Height / openScale))[0];
        if (leftEyes.Length > 0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            {
                MCvAvgComp leftEye = leftEyes[0];
                Rectangle leftRect = leftEye.rect;
                leftRect.Location = new Point(leftRect.X + eyesLeftPoint.X, leftRect.Y +
eyesLeftPoint.Y);
                img.Draw(leftRect, new Bgr(0, 255, 80), 1);
                grayImg.ROI = leftRect;
                Image<Gray, byte> imgCrop = new Image<Gray, byte>(leftRect.Width,
leftRect.Height);
                CvInvoke.cvCopy(grayImg, imgCrop, IntPtr.Zero);

                imgBoxLeftEye.Image = imgCrop;
                LeftOpenCount++;
                if (LeftOpenCount > ClickCount)
                {
                    lbLeftEye.Text = "open";
                    lbLeftEye.ForeColor = Color.Black;
                    isLeftClick = false;
                    LeftCount = 0;
                }
            }
            else
            {
                grayImg.ROI = possibleROILeftEye;
                var leftCloses = grayImg.DetectHaarCascade(haarLeftEye, 1.1,
closeFactor, HAAR_DETECTION_TYPE.DO_CANNY_PRUNING, new Size(img.Width /
closeScale, img.Height / closeScale))[0];
                if (leftCloses.Length > 0)
                {
                    MCvAvgComp leftClose = leftCloses[0];
                    Rectangle leftRect = leftClose.rect;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    leftRect.Location = new Point(leftRect.X + eyesLeftPoint.X,
leftRect.Y + eyesLeftPoint.Y);
                    img.Draw(leftRect, new Bgr(255, 0, 80), 1);
                    grayImg.ROI = leftRect;
                    Image<Gray, byte> imgCrop = new Image<Gray,
byte>(leftRect.Width, leftRect.Height);
                    CvInvoke.cvCopy(grayImg, imgCrop, IntPtr.Zero);
                    imgBoxLeftEye.Image = imgCrop;

                    LeftCount++;
                    if (!isLeftClick && LeftCount>ClickCount)
                    {
                        LeftClick();
                        LeftOpenCount = 0;
                        //LeftCount = 0;
                        isLeftClick = true;
                        lbLeftEye.Text = "close";
                        lbLeftEye.ForeColor = Color.Yellow;
                    }

                }
            }
            grayImg.ROI = Rectangle.Empty;
            #endregion
            #region righteye
            //Right Eye
            grayImg.ROI = possibleROIRightEye;
            var rightEyes = grayImg.DetectHaarCascade(haarEye, 1.1, openFactor,
HAAR_DETECTION_TYPE.DO_CANNY_PRUNING, new Size(img.Width / openScale,
img.Height / openScale))[0];
            if (rightEyes.Length > 0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            {
                MCvAvgComp rightEye = rightEyes[0];
                Rectangle rightRect = rightEye.rect;
                rightRect.Location = new Point(rightRect.X + eyesRightPoint.X,
rightRect.Y + eyesRightPoint.Y);
                img.Draw(rightRect, new Bgr(0, 255, 80), 1);
                grayImg.ROI = rightRect;
                Image<Gray, byte> imgCrop = new Image<Gray, byte>(rightRect.Width,
rightRect.Height);
                CvInvoke.cvCopy(grayImg, imgCrop, IntPtr.Zero);
                imgBoxRightEye.Image = imgCrop;

                RightOpenCount++;
                if (RightOpenCount > ClickCount)
                {
                    lbRightEye.Text = "open";
                    lbRightEye.ForeColor = Color.Black;
                    isRightClick = false;
                    RightCount = 0;
                }

            }
            else
            {
                grayImg.ROI = possibleROIRightEye;
                var rightCloses = grayImg.DetectHaarCascade(haarRightEye, 1.1,
closeFactor, HAAR_DETECTION_TYPE.DO_CANNY_PRUNING, new Size(img.Width /
closeScale, img.Height / closeScale))[0];
                if (rightCloses.Length > 0)
                {
                    MCvAvgComp rightClose = rightCloses[0];
                    Rectangle rightRect = rightClose.rect;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    rightRect.Location = new Point(rightRect.X + eyesRightPoint.X,
rightRect.Y + eyesRightPoint.Y);
                    img.Draw(rightRect, new Bgr(255, 0, 80), 1);
                    grayImg.ROI = rightRect;
                    Image<Gray, byte> imgCrop = new Image<Gray,
byte>(rightRect.Width, rightRect.Height);
                    CvInvoke.cvCopy(grayImg, imgCrop, IntPtr.Zero);
                    imgBoxRightEye.Image = imgCrop;
                    RightCount++;
                    if (!isRightClick && RightCount > ClickCount)
                    {
                        RightClick();
                        RightOpenCount = 0;
                        isRightClick = true;
                        lbRightEye.Text = "close";
                        lbRightEye.ForeColor = Color.Yellow;
                    }
                }


            }
            grayImg.ROI = Rectangle.Empty;
            #endregion
            #endregion
            lbError.Text = "-";
        }
        catch (Exception ex)
        {
            lbError.Text = "Error";
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
        private void SetFaceLabel(int x, int y, int width, int height)
        {
            lbFaceX.Text = x.ToString();
            lbFaceY.Text = y.ToString();
            lbFaceWidth.Text = width.ToString();
            lbFaceHeight.Text = height.ToString();
            //Application.DoEvents();
        }
        private void MouseFoundFace()
        {
            Image<Bgr, Byte> imgMouse = new Image<Bgr, byte>(250, 250, new Bgr(255,
255, 255));
            int x = int.Parse( lbFaceX.Text)- MainControl.centerX;
            int y = int.Parse( lbFaceY.Text)- MainControl.centerY;
            if (Math.Abs( x) < stopValue) x = 0;
            if (Math.Abs(y )< stopValue) y = 0;
            nowPoint.X = x+125;
            nowPoint.Y = y+125;
            lbNowX.Text = x.ToString();
            lbNowY.Text = y.ToString();
            previousPoint.X = int.Parse(lbFaceX.Text);
            previousPoint.Y = int.Parse(lbFaceY.Text);
            imgMouse.Draw(new LineSegment2D(nowPoint, centerPoint), new Bgr(0, 0, 0),
3);
            imgMouse.Draw(new CircleF(centerPoint, 5), new Bgr(255, 0, 0), 3);
            imgMouse.Draw(new CircleF(nowPoint, 5), new Bgr(0, 0, 255), 3);
            imgBoxMouse.Image = imgMouse;
            this.Cursor = new Cursor(Cursor.Current.Handle);
            int shiftX = (int)Math.Ceiling((double)x / sensitiveX);
            int shiftY = (int)Math.Ceiling((double)y / sensitiveY);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Cursor.Position = new Point(Cursor.Position.X +shiftX, Cursor.Position.Y
+shiftY);

        }
        private void MouseNotFoundFace()
        {
            Image<Bgr, Byte> imgMouse = new Image<Bgr, byte>(250, 250, new Bgr(255,
255, 255));
            int x = previousPoint.X - MainControl.centerX;
            int y = previousPoint.Y - MainControl.centerY;
            if (Math.Abs(x) < stopValue) x = 0;
            if (Math.Abs(y) < stopValue) y = 0;
            nowPoint.X = x + 125;
            nowPoint.Y = y + 125;
            lbNowX.Text = x.ToString();
            lbNowY.Text = y.ToString();
            imgMouse.Draw(new LineSegment2D(nowPoint, centerPoint), new Bgr(0, 0, 0),
3);
            imgMouse.Draw(new CircleF(centerPoint, 5), new Bgr(255, 0, 0), 3);
            imgMouse.Draw(new CircleF(nowPoint, 5), new Bgr(0, 0, 255), 3);
            imgBoxMouse.Image = imgMouse;
            this.Cursor = new Cursor(Cursor.Current.Handle);
            int shiftX = (int)Math.Ceiling((double)x / sensitiveX);
            int shiftY = (int)Math.Ceiling((double)y / sensitiveY);
            Cursor.Position = new Point(Cursor.Position.X + shiftX, Cursor.Position.Y +
shiftY);
        }
        public void LeftClick()
        {
            //Call the imported function with the cursor's current position
            int X = Cursor.Position.X;
            int Y = Cursor.Position.Y;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            mouse_event(MOUSEEVENTF_LEFTDOWN | MOUSEEVENTF_LEFTUP, (uint)X,
(uint)Y, 0, 0);
        }
        public void RightClick()
        {
            //Call the imported function with the cursor's current position
            int X = Cursor.Position.X;
            int Y = Cursor.Position.Y;
            mouse_event(MOUSEEVENTF_RIGHTDOWN | MOUSEEVENTF_RIGHTUP, (uint)X,
(uint)Y, 0, 0);
        }
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
อุปกรณ์ที่ใช้

รูปที่ ข.1 แสดงภาพกล้องเว็บแคม

Logitech HD Webcam C270H

Technical Specifications

HD video calling (1280 x 720 pixels) with recommended system
Video capture: Up to 1280 x 720 pixels
Photos: Up to 3.0 megapixels (software enhanced)
Built-in mic with Logitech RightSound™ technology
Hi-Speed USB 2.0 certified (recommended)
Universal clip fits laptops, LCD or CRT monitors

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

ตารางการดำเนินงาน

ขั้นตอนการดำเนินงาน	เทอมที่ 1 ปีการศึกษา 2556			
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.
1.วางแผนและคิดเรื่องที่จะทำ				
2.ศึกษาโครงสร้างและวิธีการเขียนของภาษา C#				
3.ศึกษาทฤษฎี Image processing				
4.ศึกษาการเขียน C# กับ Image processing				
5.ลงมือเขียนโปรแกรม				

ตารางที่ ค.1 แสดงขั้นตอนการทำงานในเทอม 1

ขั้นตอนการดำเนินงาน	เทอมที่ 2 ปีการศึกษา 2556			
	พ.ย.	ธ.ค.	ม.ค.	ก.พ.
1.ศึกษาการเขียนโปรแกรมเพิ่มเติม				
2.เขียนโปรแกรม				
3.ทดสอบการทำงานของโปรแกรม				
4.ปรับปรุงและแก้ไขโปรแกรม				
5.จัดทำรูปเล่ม				

ตารางที่ ค.2 แสดงขั้นตอนการทำงานในเทอม 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้