

ระบบแจ้งเตือนการหลับใหล
Doze-off Warning System



ปริญญาโท วิทยานิพนธ์เป็นส่วนหนึ่งของงานที่ศึกษาตามหลักสูตรปริญญาโท สาขาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

ระบบแจ้งเตือนการหลับใน
Doze-off Warning System



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2556

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบแจ้งเตือนการหลับใน

Doze-off Warning System

ผู้จัดทำ

1. นายเบญจพงศ์ เทียนภาสกร

รหัสนักศึกษา 53010884

2. นายวัชรพล ตอบุญธง

รหัสนักศึกษา 53011451



..... อาจารย์ที่ปรึกษา
(รองศาสตราจารย์ ดร. บุญวัฒน์ อัดชู)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบแจ้งเตือนการหลับใน

นายเบญจพงศ์	เทียนภาสกร	53010884
นายวัชรพล	ตอบุญธง	53011451
รศ.ดร. บุญวัฒน์	อัฐชู	อาจารย์ที่ปรึกษา
ปีการศึกษา 2556		

บทคัดย่อ

ในปัจจุบันการเกิดอุบัติเหตุตามท้องถนนเกิดขึ้นเป็นประจำทุกวัน การหลับใน เป็นหนึ่งในสาเหตุหลักของการเกิดอุบัติเหตุ มักเกิดจากการพักผ่อนไม่เพียงพอของผู้ขับขี่ยานพาหนะ โปรแกรมแจ้งเตือนการหลับในจะเป็นตัวช่วยแจ้งเตือนให้กับผู้ขับขี่เมื่อมีอาการหลับในให้มีสติขึ้นมา โครงการนี้นำเสนอการพัฒนาโปรแกรมแจ้งเตือนการหลับใน ซึ่งจะช่วยลดอุบัติเหตุที่เกิดจากการหลับใน โปรแกรมจะทำงานโดยรับภาพจากกล้องวิดีโอหรือไฟล์วิดีโอ และนำมาประมวลผล โปรแกรมแจ้งเตือนการหลับในจะตรวจสอบการหลับใน โดยดูจากคันทาลูกตาตาของผู้ขับขี่ หากไม่พบลูกตาตา จะทำการแจ้งเตือนผู้ใช้ ตรวจสอบริมฝีปากของผู้ใช้ เพื่อนำไปตรวจจับอาการหาวของผู้ใช้ และการตรวจหาอาการเหนื่อยล้าจากการเคลื่อนไหวของศีรษะของผู้ใช้ โดยเก็บตำแหน่งใบหน้าของผู้ใช้ ซึ่งโปรแกรมสามารถตรวจจับและแจ้งเตือนอาการหลับในรวมถึงอาการเหนื่อยล้าของผู้ใช้ได้

Doze-off Warning System

Mr. Benjapong	Tienpassakorn	53010884
Mr. Watcharaphon	Torboontong	53011451
Assoc. Prof. Dr. Boonwat	Attachoo	Advisor

Academic Year 2013

ABSTRACT

Nowadays, there are many fatal car accidents happen every day which could be causes of death. One of these primary causes are somnolent as well as sleepiness which can cause a driver dozed off while driving. To reduce the risk of drowsy driving, Doze-off Warning System will be a good choice. It will let the driver know if he is in danger of falling sleeping at the wheel by alerting driver to be conscious again. This project presents a development of a Doze-off Warning System that reduce accident caused by dozed off. The application works by capturing a sequence of images via webcam or video file. Then the application will process these images using digital image processing. These are three main categories of detection. First, the application will detect pupil from user's eyes. If pupil were not detected application will alert user. Second, the application will detect user's lip for yawning detection. Third, the application will detect user's face position and process for fatigue detection. In conclusion, the application can detect doze off and fatigue and warn to driver.

กิตติกรรมประกาศ

รายงานฉบับนี้สำเร็จลุล่วงได้เป็นอย่างดีด้วยคำแนะนำและคำปรึกษาเกี่ยวกับการดำเนินงาน การศึกษาและวิจัย โดยได้รับความช่วยเหลือจากบุคคลากรต่อไปนี้

ขอขอบพระคุณรองศาสตราจารย์ ดร. บุญวัฒน์ อัดชู ซึ่งเป็นอาจารย์ที่ปรึกษาโครงการ ที่ได้ให้ความรู้ คำแนะนำ อีกทั้งความช่วยเหลือในแนวทางการแก้ปัญหาต่างๆที่เกิดขึ้นระหว่างทำโครงการ จนทำให้รายงานเล่มนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณห้องวิจัย ISAG และห้องวิจัย 803 ที่ได้ให้การสนับสนุนสถานที่ในการจัดทำโครงการ ตลอดจนเครื่องมือ หนังสือต่างๆ ตลอดจนเพื่อนๆ พี่ๆ น้องๆ ที่ได้กำลังใจและคำแนะนำ เสมอมา

สุดท้ายนี้ข้าพเจ้าขอขอบพระคุณบิดา มารดา และครอบครัว ที่ได้ให้การสนับสนุนและคอย เป็นกำลังให้เสมอมา

นาย เบญจพงศ์

เทียนภาสกร

นาย วิชระพล

ต่อบุญธง

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VIII
สารบัญรูป.....	IX
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญและที่มาของโครงการ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบเขตของโครงการ.....	1
1.4 วิธีการดำเนินการ.....	1
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.6 ส่วนประกอบของรายงาน.....	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	3
2.1 อาการหลับใหล.....	3
2.1.1 สาเหตุของการหลับใหล.....	3
2.1.2 สิ่งแวดล้อมที่ส่งเสริมให้เกิดการง่วงนอนและหลับใหล.....	3
2.1.3 การหลีกเลี่ยง หรือ การป้องกัน “การหลับใหล”.....	3
2.2 การประมวลผลภาพดิจิทัล.....	4
2.2.1 Introduction.....	4
2.2.2 รูปร่างของภาพ (Image Shape).....	5
2.2.3 มาตรฐานของสี.....	6
2.3 การปรับปรุงภาพ.....	10

สารบัญ (ต่อ)

	หน้า
2.3.1 การแปลงรูปแบบสีจาก RGB เป็น YCbCr.....	10
2.3.3 การแปลงภาพระดับสีเทาให้เป็นภาพไบนารี.....	12
2.3.4 การปรับเปลี่ยนหรือการเปลี่ยนแปลงค่าความเข้มแสงของภาพ (intensity manipulation).....	13
2.3.5 การปรับความเข้มแสงโดยใช้ฟังก์ชันลอการิทึม(Logarithm Function).....	14
2.3.6 การขยายภาพ (Dilation).....	14
2.3.7 การย่อภาพ (Erosion).....	15
2.3.8 โอเพอเรชัน Closing และ Opening.....	16
2.3.9 การกระจายตัวของสัญญาณรบกวนแบบเกาส์เซียน (Gaussian Noise).....	16
2.3.10 การปรับภาพเชิงพื้นที่.....	17
2.3.11 การกรองภาพแบบคอนโวลูชัน (convolution filtering).....	18
2.3.12 เครื่องกรองผ่านความถี่ต่ำ (low-pass filter).....	19
2.3.13 การจับขอบ (edge detection).....	20
2.3.14 การเน้นขอบ (edge enhancement).....	21
2.4 แมทแลป (Matlab).....	22
2.4.1 ความหมายและที่มาของแมทแลป.....	22
2.4.2 โครงสร้างและองค์ประกอบของแมทแลป.....	23
2.4.3 ข้อดีของแมทแลป.....	24
2.5 คอมพิวเตอร์วิชัน (Computer Vision).....	25
2.6 การตรวจจับใบหน้าโดยวิธีของวิโอล่าและโจนส์ (Face Detection by Viola-Jones method).....	26
2.6.1 ขั้นตอน Integral Image.....	27
2.6.2 ขั้นตอนการเรียนรู้ด้วยวิธี Adaboost.....	30
2.6.3 ขั้นตอนการรวมตัวจำแนกกลุ่มแบบต่อเรียง.....	31
2.7 การตรวจหาริมฝีปาก.....	32

สารบัญ (ต่อ)

	หน้า
รูปที่ 2.21 ตัวอย่างรูปภาพที่ผ่านการตรวจหาริมฝีปาก.....	33
2.8 Hough Circle Transform	33
บทที่ 3 การออกแบบ และพัฒนา.....	34
3.1 ภาพรวมของโปรแกรม.....	34
3.2 การตรวจจับใบหน้า.....	35
3.2.1 ภาพรวมโฟลว์ชาร์ตแสดงขั้นตอนการตรวจจับใบหน้า.....	35
3.3 การตรวจหาอาการหลับจากดวงตา.....	37
3.3.1 ภาพรวมการตรวจหาอาการหลับจากดวงตา.....	37
3.3.2 ภาพรวมของการหาลูกตาต่ำ.....	39
3.3.3 สถานะในการแจ้งเตือนการหลับ.....	41
3.4 การตรวจหาอาการเหนื่อยล้าจากการเคลื่อนไหวของศีรษะ.....	43
3.4.1 ภาพรวมการตรวจหาอาการเหนื่อยล้าจากการเคลื่อนไหวของศีรษะ.....	43
3.4.2 ตารางแสดงช่วงค่าส่วนเบี่ยงเบนมาตรฐานกับความอ่อนล้า.....	44
3.5 ตรวจหาอาการหาว.....	45
3.5.1 การหาริมฝีปาก.....	45
3.5.2 การนำริมฝีปากมาประมวลผลเพื่อหาการหาว.....	47
3.5.3 การตรวจหาการหาวจากหลายๆเฟรม.....	49
บทที่ 4 การทดลองและผลการทดลอง.....	50
4.1 ทดลองเปิดใช้งานกล้องเว็บแคมบนแมทแลป.....	50
4.2 ทดลองตรวจจับภาพใบหน้าโดยใช้กล้องเว็บแคมผ่านโปรแกรมแมทแลป.....	53
4.3 ทดลองตรวจจับใบหน้าและดวงตาพร้อมกันโดยกล้องเว็บแคมผ่านโปรแกรมแมทแลป.....	54
4.3 การทดลองหาลูกตาต่ำและอาการหลับ.....	55

สารบัญ (ต่อ)

	หน้า
4.3.1 การทดลองตรวจจับดวงตาโดยใช้ Haar Like Feature เมื่อผู้ใช้ลืมตาอยู่.....	55
4.3.2 การทดลองตรวจจับดวงตาโดยใช้ Haar Like Feature เมื่อผู้ใช้หลับตา.....	58
4.3.3 การทดลองตรวจหาอาการหลับ.....	59
4.3.4 การตรวจจับดวงตาและหาลูกตาดำเมื่อผู้ใช้สวมแว่นตา.....	63
4.4 การทดลองตรวจหาอาการเหนื่อยล้าจากเคลื่อนที่ของศีรษะ.....	66
4.5 การทดลองตรวจหาอาการหาว.....	67
4.5.1 การทดลองตรวจหาริมฝีปาก.....	67
4.5.2 การทดลองหาอาการหาวจากภาพริมฝีปากในกรณีริมฝีปากปิดปกติ.....	68
4.5.3 การทดลองหาอาการหาวจากริมฝีปากในกรณีปากอ้ากว้าง.....	70
4.5.2 การทดลองตรวจสอบการหาวในไฟล์วีดีโอทดลอง.....	72
4.5.3 การตรวจหาการหาวกรณีผู้ใช้มีการพูดคุย.....	75
บทที่ 5 บทสรุปและข้อเสนอแนะ.....	79
5.1 บทสรุปโครงงาน.....	79
5.2 ปัญหาอุปสรรคและแนวทางการแก้ไข.....	79
5.3 แนวทางการพัฒนาต่อ.....	80
บรรณานุกรม.....	81

สารบัญตาราง

ตารางที่

หน้า

ตารางที่ 3.1 ตารางแสดงช่วงค่าส่วนเบี่ยงเบนมาตรฐานกับความอ่อนล้า..... 44



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
รูปที่ 2.1 แสดงองค์ประกอบของข้อมูลภาพในระบบสี RGB	7
รูปที่ 2.3 แสดงองค์ประกอบของข้อมูลภาพในระบบสีระดับสีเทา.....	9
รูปที่ 2.4 แสดงองค์ประกอบของข้อมูลภาพในระบบสี binary	9
รูปที่ 2.5 แสดงภาพก่อนและหลังการแปลงภาพสีจาก RGB เป็น YCbCr	11
รูปที่ 2.6 แสดงภาพก่อนและหลังการแปลงภาพสีจาก RGB เป็นระดับสีเทา	12
รูปที่ 2.7 แสดงภาพก่อนและหลังการแปลงภาพสีจากระดับสีเทาเป็นไบนารี	13
รูปที่ 2.8 รูปแสดงกระบวนการขยายภาพ	15
รูปที่ 2.9 รูปภาพแสดงกระบวนการย่อภาพ	15
รูปที่ 2.10 ความถี่เชิงพื้นที่ของจุดภาพ ในภาพพื้นที่หนึ่ง.....	17
ภาพที่ 2.11 การเคลื่อนที่ของหน้าต่างเคลื่อนที่.....	18
(a) หน้าต่างเคลื่อนที่ ขนาด 3×3 จุดภาพ เคลื่อนที่ไปบนภาพที่จะทำการประมวล	18
(b) เลื่อนไปตามแถวจากจุดภาพหนึ่งไปยังอีกจุดภาพหนึ่งจนหมดแถว	18
(c) เลื่อนหน้าต่างเคลื่อนที่จากแถวหนึ่งไปยังอีกแถวหนึ่ง.....	18
รูปที่ 2.12 การปรับภาพโดยใช้การกรองด้วยหน้าต่างเคลื่อนที่แบบต่างๆ (ภาพบริเวณอ่าวปัตตานี) 20	
รูปที่ 2.13 โครงสร้างและองค์ประกอบของแมทแลป.....	23
รูปที่ 2.14 สิ่งที่คอมพิวเตอร์มองเห็นคือกริดของตัวเลขจำนวนมาก	26
รูปที่ 2.15 ตัวอย่างการจำลองรูปแบบ Haar-like.....	27
รูปที่ 2.16 ตัวอย่างการคำนวณความเข้มในพื้นที่สี่เหลี่ยม D โดยวิธีการทั่วไป	28
รูปที่ 2.17 ตัวอย่างภาพ Integral image.....	28
รูปที่ 2.18 ตัวอย่างการคำนวณผลรวมความเข้มในพื้นที่สี่เหลี่ยม D โดยวิธี Integral image	29
รูปที่ 2.19 แสดงตัวอย่างของตัวจำแนกแบบแข็งแกร่งที่ได้จากกระบวนการเรียนรู้ด้วย Adaboost... 31	
รูปที่ 2.20 แสดงขั้นตอนการรวมตัวจำแนกกลุ่มแบบต่อเรียง.....	32
รูปที่ 2.21 ตัวอย่างรูปภาพที่ผ่านการตรวจหาริมฝีปาก.....	33
รูปที่ 3.1 ภาพรวมของโปรแกรม	34
รูปที่ 3.2 โพลวชาร์ตแสดงการตรวจจับใบหน้า.....	35
รูปที่ 3.3 โพลวชาร์ตภาพรวมการตรวจหาการหลับจากดวงตา.....	37
รูปที่ 3.4 โพลวชาร์ตแสดงภาพรวมของการหาลูกตาดำ.....	39

สารบัญรูป (ต่อ)

รูปที่	หน้า
รูปที่ 3.5 แผนผังสถานะแสดงการแจ้งเตือนการกลับ.....	41
รูปที่ 3.6 โพลวชาร์ตแสดงการตรวจหาอาการเหนื่อยล้าจากการเคลื่อนไหวของศีรษะ	43
รูปที่ 3.7 โพลวชาร์ตแสดงการหาริมฝีปาก	45
รูปที่ 3.8 โพลวชาร์ตแสดงขั้นตอนการนำริมฝีปากมาประมวลผลเพื่อการหา.....	47
รูปที่ 4.1 แสดงผลลัพธ์เมื่อใช้คำสั่ง imaqwhinfo('winvideo').....	50
รูปที่ 4.2 แสดงผลลัพธ์เมื่อใช้คำสั่ง imaqhwinfo('winvideo',1)	51
รูปที่ 4.3 แสดงผลลัพธ์เมื่อใช้คำสั่ง dev.SupportedFormats.....	51
รูปที่ 4.4 แสดงผลลัพธ์เมื่อให้แมทแอสแสดงภาพจากกล้องเว็บแคม	52
รูปที่ 4.5 แสดงผลการทดลองจับภาพใบหน้าบนความละเอียด 355 x288 ในระบบสี YCbCr.....	53
รูปที่ 4.6 แสดงผลการทดลองจับภาพใบหน้าบนความละเอียด 355 x288 พิกเซล ในระบบสี RGB	54
รูปที่ 4.7 แสดงผลการทดลองจับภาพใบหน้าและดวงตาบนความละเอียด 320x 240 พิกเซล.....	55
รูปที่ 4.8 ภาพใบหน้าที่ได้จากการตรวจจับใบหน้า.....	55
รูปที่ 4.9 ภาพดวงตาทั้งสองข้างที่ได้จากการตรวจจับดวงตา.....	56
รูปที่ 4.10 ภาพดวงตาทั้งสองข้างเมื่อผ่าน Gaussian Filter	56
รูปที่ 4.11 ภาพดวงตาทั้งสองข้างในระบบสี Grayscale.....	56
รูปที่ 4.12 ภาพดวงตาทั้งสองข้างเมื่อผ่านการฟังก์ชันลอการิทึม.....	56
รูปที่ 4.13 ภาพดวงตาทั้งสองข้างในระบบสี Binary	56
รูปที่ 4.14 ภาพลูกตาดำทั้งสองข้างหลังจากผ่านการทำ Closing Operation และหาวงกลมบริเวณที่เป็นลูกตาดำ.....	57
รูปที่ 4.15 ภาพผู้ใช้เมื่อผ่านการหาลูกตาดำ	57
รูปที่ 4.16 ภาพใบหน้าที่ได้จากการตรวจจับใบหน้า	58
รูปที่ 4.17 ภาพดวงตาทั้งสองข้างที่ได้จากการตรวจจับดวงตา.....	58
รูปที่ 4.18 ภาพดวงตาทั้งสองข้างเมื่อผ่าน Gaussian Filter	58
รูปที่ 4.19 ภาพดวงตาทั้งสองข้างในระบบสี Grayscale.....	58
รูปที่ 4.20 ภาพดวงตาทั้งสองข้างเมื่อผ่านการฟังก์ชันลอการิทึม	59
รูปที่ 4.21 ภาพดวงตาทั้งสองข้างในระบบสี Binary	59
รูปที่ 4.22 รูปภาพเมื่ออยู่ในสถานะปกติ	60

สารบัญรูป (ต่อ)

รูปที่	หน้า
รูปที่ 4.23 รูปแสดงเมื่อเข้าสู่สถานะ A	60
รูปที่ 4.24 รูปแสดงเมื่อเข้าสู่สถานะ B	61
รูปที่ 4.25 รูปแสดงเมื่อเข้าสู่สถานะแจ้งเตือน	61
รูปที่ 4.26 รูปแสดงเมื่อเข้าสู่สถานะ C	62
รูปที่ 4.27 รูปแสดงสถานะ D	62
รูปที่ 4.28 รูปแสดงเมื่อโปรแกรมกลับสู่สถานะปกติ	63
รูปที่ 4.29 รูปแสดงการทดลองเมื่อผู้ใช้สวมใส่แว่นตา	64
รูปที่ 4.30 รูปแสดงการทดลองเมื่อผู้ใช้สวมใส่แว่นตา	64
รูปที่ 4.31 รูปแสดงการทดลองเมื่อผู้ใช้สวมใส่แว่นตา	65
รูปที่ 4.32 รูปแสดงการทดลองเมื่อผู้ใช้สวมใส่แว่นตา	65
รูปที่ 4.33 รูปแสดงการทดลองเมื่อผู้ใช้สวมใส่แว่นตา	66
รูปที่ 4.34 แสดงภาพที่ได้จากโปรแกรม	66
รูปที่ 4.35 แสดงตำแหน่งจุดศูนย์กลางของภาพในแนวแถวและหลัก (Row , Column)	67
รูปที่ 4.36 แสดงส่วนแสดงผลเกี่ยวกับค่าส่วนเบี่ยงเบนมาตรฐาน	67
รูปที่ 4.37 แสดงผลลัพธ์ของการหาส่วนเบี่ยงเบนมาตรฐานจากข้อมูลสิบเฟรมแรก	67
รูปที่ 4.38 ภาพใบหน้าครึ่งล่าง	67
รูปที่ 4.39 ภาพที่ได้จากสมการ Lip Map	68
รูปที่ 4.40 รูป Binary ของริมฝีปาก	68
รูปที่ 4.41 รูปริมฝีปากที่ผ่าน Opening Operation	69
รูปที่ 4.42 พื้นที่ของวัตถุภายในภาพ	69
รูปที่ 4.43 รูปแสดงจุดศูนย์กลางวัตถุทั้งสองชิ้น	69
รูปที่ 4.44 แสดงร้อยละระหว่างระยะห่างของริมฝีปากกลางและบนกับความสูงของรูปครึ่งล่างของ ใบหน้า	70
รูปที่ 4.45 รูปภาพใบหน้าส่วนล่าง	70
รูปที่ 4.46 รูปภาพใบหน้าส่วนล่างที่ผ่านสมการ Lip map	70
รูปที่ 4.47 รูปภาพครึ่งล่างใบหน้าผ่านสมการ Lip map และแปลงเป็นระบบสี binary	71
รูปที่ 4.48 รูปภาพใบหน้าส่วนล่างที่ผ่าน Opening Operation	71

สารบัญรูป (ต่อ)

รูปที่	หน้า
รูปที่ 4.49 แสดงผลลัพธ์ของพื้นที่ของวัตถุทั้งหมดในรูปภาพ	71
รูปที่ 4.50 แสดงรูปภาพริมฝีปากบนและล่าง	72
รูปที่ 4.51 แสดงผลลัพธ์ของจุดศูนย์กลางของริมฝีปากบนและล่าง	72
รูปที่ 4.52 แสดงร้อยละระหว่างระยะห่างของริมฝีปากกับความสูงของภาพครึ่งล่างของใบหน้า	72
รูปที่ 4.53 รูปแสดงการตรวจหาการหา	73
รูปที่ 4.54 รูปแสดงการตรวจหาการหา	73
รูปที่ 4.55 รูปแสดงการตรวจหาการหา	74
รูปที่ 4.57 รูปแสดงการตรวจหาการหา	75
รูปที่ 4.58 รูปแสดงอาการพูดคุยทั่วไป	76
รูปที่ 4.59 รูปแสดงอาการพูดคุยทั่วไป	76
รูปที่ 4.60 รูปแสดงอาการพูดคุยทั่วไป	77
รูปที่ 4.61 รูปแสดงอาการพูดคุยทั่วไป	77
รูปที่ 4.61 รูปแสดงอาการพูดคุยทั่วไป	78

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของโครงการ

ในปัจจุบันการอุบัติเหตุตามท้องถนนเกิดขึ้นเป็นประจำทุกวัน การหลับใน เป็นหนึ่งในสาเหตุหลักของการเกิดอุบัติเหตุ มักเกิดจากการพักผ่อนไม่เพียงพอของผู้ขับขี่ยานพาหนะ และมาตรการการป้องกันในปัจจุบันยังมีไม่เพียงพอ ทำให้เพียงแต่รณรงค์ขังวงไม่ขับเท่านั้น โปรแกรมแจ้งเตือนการหลับในจะเป็นตัวช่วยแจ้งเตือนให้กับผู้ขับขี่เมื่อมีอาการหลับใน ให้มีสติขึ้นมา ซึ่งจะช่วยลดอุบัติเหตุที่เกิดจากการหลับในได้

1.2 วัตถุประสงค์ของโครงการ

- 1) ศึกษาการใช้งานการประมวลผลภาพผ่านโปรแกรมแมทแลป
- 2) สร้างและพัฒนาโปรแกรมแจ้งเตือนการหลับใน
- 3) สามารถแจ้งเตือนผู้ขับขี่ที่หลับในได้อย่างทันท่วงที

1.3 ขอบเขตของโครงการ

โปรแกรมแจ้งเตือนการหลับในจะตรวจสอบการหลับใน โดยตรวจสอบจากสามสิ่งคือ

- 1) ดวงตาของผู้ใช้งาน โดยได้จากการค้นหาลูกตาดำของผู้ใช้งานแล้วนำไปประมวลผลว่ามีอาการหลับในหรือไม่ หากมีการหลับในจะแจ้งเตือนผู้ใช้เพื่อให้ผู้ใช้ตื่นจากอาการหลับใน
- 2) การเคลื่อนไหวของศีรษะของผู้ใช้ โดยตรวจสอบตำแหน่งใบหน้าของผู้ใช้ในแนวตั้งเพื่อนำไปประมวลผลหาอาการอ่อนล้าของผู้ใช้
- 3) ตรวจหาริมฝีปากของผู้ใช้เพื่อนำไปวิเคราะห์อาการหาวของผู้ใช้

1.4 วิธีการดำเนินการ

- 1) ศึกษาเกี่ยวกับการใช้งานการประมวลผลภาพผ่านพื้นฐานโดยใช้โปรแกรมแมทแลป
- 2) ศึกษาพฤติกรรมกรการหลับในของมนุษย์ที่สามารถสังเกตได้
- 3) ศึกษากระบวนการการค้นหาองค์ประกอบบนใบหน้าภายในภาพ
- 4) พัฒนาโปรแกรมเพื่อตรวจหาการลูกตาดำ ตำแหน่งของใบหน้า และตรวจหาริมฝีปากของผู้ใช้

- 5) วิเคราะห์ผลของโปรแกรมแจ้งเตือนการหลับใน และแก้ไขข้อผิดพลาดเพื่อให้โปรแกรมทำงานได้ดีที่สุด

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ระบบแจ้งเตือนการหลับในเป็นส่วนสำคัญที่ช่วยลดโอกาสในเกิดอุบัติเหตุทางท้องถนน
- 2) ได้รับความรู้ในการเขียนโปรแกรมเพื่อประมวลผลภาพ
- 3) ได้รับความรู้ในการใช้งานโปรแกรมแมทแลป และไลบรารีที่ใช้งานในการประมวลผลภาพในแมทแลป
- 4) นำโปรแกรมไปประยุกต์ใช้งานจริง และสามารถพัฒนาต่อยอดเพื่อให้ความสมบูรณ์ขึ้นในอนาคต

1.6 ส่วนประกอบของรายงาน

ปริญญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาโดยทั่วไปออกเป็น 5 บทด้วยกันคือ
 บทที่ 1 บทนำ กล่าวถึงความสำคัญและที่มาของโครงการ วัตถุประสงค์ของโครงการ ขอบเขตของโครงการ วิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปริญญานิพนธ์

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในโครงการ ประกอบด้วยอาการหลับใน การประมวลผลภาพดิจิทัล การปรับปรุงภาพ โปรแกรมแมทแลป คอมพิวเตอร์วิชัน การตรวจจับใบหน้าด้วยวิธีการของวีโอล่าและโจนส์ และการตรวจหากริมฝีปาก

บทที่ 3 การออกแบบและพัฒนา กล่าวถึงรายละเอียดของโครงการ ภาพรวมของโปรแกรม, การตรวจจับใบหน้า การตรวจหาอาการหลับจากดวงตา การตรวจหาอาการเหนื่อยล้าจากการเคลื่อนไหวของศีรษะ และการตรวจหาอาการหาว

บทที่ 4 การทดลองและผลการทดลอง กล่าวถึงข้อมูลที่ใช้ในการทดลอง การทดลองเปิดใช้งานกล้องเว็บแคมบนแมทแลป การทดลองตรวจจับใบหน้าโดยใช้กล้องเว็บแคมผ่านโปรแกรมแมทแลป การทดลองตรวจจับใบหน้าและดวงตาพร้อมกันโดยกล้องเว็บแคมผ่านโปรแกรมแมทแลป การทดลองหาจุดตาดำและอาการหลับ การทดลองตรวจหาอาการเหนื่อยล้าจากเคลื่อนไหวของศีรษะ และการทดลองตรวจหาอาการหาว

บทที่ 5 บทสรุป กล่าวถึงบทสรุปและข้อเสนอแนะ กล่าวถึงบทสรุปโครงการ ปัญหาอุปสรรค และแนวทางการแก้ไข และแนวทางการพัฒนาต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 อาการหลับใหล

2.1.1 สาเหตุของการหลับใหล

จากผลการวิจัยและข้อมูลต่างๆ[1]พอสรุปสาเหตุของการหลับใหลได้ดังนี้

- 1) ทำงานหนัก และพักผ่อนไม่เพียงพอ
- 2) นอนหลับพักผ่อนไม่เพียงพอ (ติดหนึ่การนอน) การพักผ่อนที่เพียงพอต้องนอนหลับสนิทติดต่อกันนาน 7-8 ชั่วโมง ต่อวันคนนอนกรน มักนอนหลับไม่สนิท จึงติดหนึ่การนอน มักหลับใหลง่าย
- 3) เป็นคนหลับง่าย คนที่มีมวลกายสูง (อ้วน) เป็นคนหลับง่าย
- 4) ขับรถในช่วงพักผ่อนของร่างกาย คือช่วงหลังเที่ยงวันถึงบ่าย 3 โมง และหลังเที่ยงคืน-หกโมงเช้า ซึ่งเป็นช่วงเวลาที่นาฬิกาชีวภาพประจำตัว หยุดปลุก ทำให้ง่วงนอน โดยเฉพาะอย่างยิ่ง หากหากขับรถหลังรับประทานอาหารหรือสุราหรือของมีน เมาแล้วขับรถ
- 5) การกินยาบางชนิดที่มีฤทธิ์ กดประสาทและทำให้มีอาการง่วงนอนและหลับใหลเช่น ยาแก้แพ้ ยาลดน้ำมูก ยาแก้หวัด ยาแก้ไอ ยานอนหลับ ยาคลายเครียด ยาแก้ปวด เป็นต้น ตลอดจนการดื่มเครื่องดื่มแอลกอฮอล์

2.1.2 สิ่งแวดล้อมที่ส่งเสริมให้เกิดการง่วงนอนและหลับใหล

- 1) อุณหภูมิเย็นๆ (เปิดเครื่องปรับอากาศแรง)
- 2) ความเร็วรถสม่ำเสมอ
- 3) ฟังเพลง
- 4) ขับรถคนเดียว หรือไม่มีการพูดคุยในเรื่องที่น่าตื่นเต้น

2.1.3 การหลีกเลี่ยง หรือ การป้องกัน “การหลับใหล”

การหลับใหลขณะขับรถสามารถป้องกันได้ อาการการง่วง ..เป็นอาการที่บอกให้เราทราบว่าร่างกายต้องการพักผ่อน ดังนั้นการง่วงจึงเป็นสิ่งบอกเหตุและสามารถใช้เป็นสัญญาณแจ้งให้รับทำการป้องกันการหลับใหล “หาว” โดยสัญญาณเตือนที่สังเกตได้ว่าจะเกิดอาการหลับใหลได้แก่อาการหาวบ่อยและหาวต่อเนื่อง ใจลอยไม่มีสมาธิ รู้สึกเหนื่อยล้า หงุดหงิดและกระวนกระวาย เกิดอาการมึน สมองไม่ชัดเจนมองเห็นภาพไม่ชัดและหนักศีรษะ จากการวิจัยพบว่าเมื่อร่างกายง่วง จะสามารถลดทอนต่อการหลับใหล

ได้ประมาณ 20 นาที และคนที่มีอายุน้อยมักทนต่อการหลับในได้น้อย (25-35 ปี) กว่าคนที่มีอายุมาก ๆ ดังนั้นวิธีหลีกเลี่ยง หรือป้องกันการหลับใน เมื่อเกิดการง่วง ควรปฏิบัติดังนี้ได้อย่างดี คนเราเมื่อง่วงจะแสดงออกโดยการ

- 1) เมื่อขับรถแล้วง่วง ให้หาที่จอดรถที่ปลอดภัย แล้วจับหลับสัก 15-20 นาที แล้วทำการล้างหน้า ทำร่างกายให้สดชื่น แล้วจึงขับรถต่อไป
- 2) เมื่อเริ่มง่วง และกำลังมองหาที่จอดรถที่ปลอดภัย ให้บรรเทาการง่วงโดย
 - บีบหัวนมแรงๆ หรือหยิกหัวนมเบา จะทำให้ร่างกายตื่นตัว หายง่วงไปสักระยะหนึ่ง
 - เปิดกระจก ให้ลมพัดเข้าหน้าแรงๆ และจะช่วยระบายอากาศเสียออกจากห้องโดยสารของรถ เพื่อลดปริมาณคาร์บอนมอนนอกไซด์ในรถ และเพิ่มออกซิเจน
 - ตะโกนดังๆ หรือร้องเพลงดังๆ (วิธีนี้ไม่ค่อยแนะนำเพราะอาจขาดสมาธิในการขับรถ)
- 3) ก่อนขับรถ เพื่อป้องกันการหลับใน ให้ปฏิบัติดังนี้
 - นอนหลับพักผ่อนให้เพียงพอ นอนหลับประมาณ 7-9 ชั่วโมงต่อวัน
 - หลีกเลี่ยงการขับรถในช่วงเวลาที่ร่างกายต้องการพักผ่อน (หลังเที่ยงวันถึงบ่าย 3 โมง และหลังเที่ยงคืน ถึง 6 โมงเช้า)
 - หลีกเลี่ยงการขับรถหลังรับประทานอาหารอิ่มใหม่ๆ
 - ดื่มเครื่องดื่มบำรุงกำลัง หรือดื่มกาแฟ เพื่อช่วยให้ร่างกายสดชื่น
 - หลีกเลี่ยงการกินยาบางชนิดที่มีฤทธิ์ กดประสาทและทำให้มีอาการง่วงนอน เช่น ยาแก้แพ้ ยาลดน้ำมูก ยาแก้หวัด ยาแก้ไอ ยานอนหลับ ยาคลายเครียด ยาแก้คัน เป็นต้น
 - ตลอดจนงดเว้นการดื่มเครื่องดื่มแอลกอฮอล์อย่างเด็ดขาด
 - วางแผนการเดินทาง เตรียมสภาพร่างกายให้พร้อมก่อนออกเดินทาง หากต้องเดินทางไกลควรมีเพื่อนร่วมเดินทางไปด้วย เพื่อพูดคุยหรือผลัดเปลี่ยนกันขับรถ
 - คนนอนกรนและอ้วนมีความเสี่ยงสูงที่จะหลับใน ในขณะที่ขับรถ จึงควรหลีกเลี่ยงไม่ให้คนอ้วนขับรถทางไกล

2.2 การประมวลผลภาพดิจิทัล

2.2.1 Introduction

การมองเห็นของมนุษย์เป็นสิ่งที่สำคัญและเป็นกลไกการรับภาพที่ซับซ้อนอย่างหนึ่งซึ่งจะให้ข้อมูลที่มีความจำเป็นสำหรับใช้ในางานง่ายๆ (ตัวอย่างเช่น การจดจำวัตถุ) และสำหรับงานที่มีความซับซ้อน สิ่งที่น่าสนใจของข้อมูลที่เกี่ยวข้องกับการมองเห็นหรือข้อมูลภาพนั้นก็คือกระบวนการประมวลผลภาพ (Image Processing) [2] โดยใช้คอมพิวเตอร์ ความพยายามทางด้านกระบวนการประมวลผลภาพได้เริ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขึ้นในปี 1964 ได้นำกระบวนการการประมวลผลภาพมาใช้ในการพิจารณาภาพถ่ายดาวเทียมของดวงจันทร์ต่อมาได้มีการตั้งสาขาทางวิทยาศาสตร์สาขาใหม่มีชื่อว่า Digital image processing หลังจากนั้นงานทางด้าน การประมวลผลภาพก็พัฒนาขึ้นเรื่อย ๆ และใช้กันอย่างกว้างขวางสำหรับงานในหลาย ๆ ด้านตัวอย่าง เช่น ทางด้านสื่อสารโทรคมนาคม การสื่อสารทางโทรทัศน์ ทางด้านการพิมพ์ทางด้านกราฟฟิก การแพทย์และการค้นคว้าทางวิทยาศาสตร์

2.2.2 รูปร่างของภาพ (Image Shape)

วัตถุที่มีอยู่ตามธรรมชาติและที่มนุษย์สร้างขึ้นมีรูปร่างที่แตกต่างกันไป ทั้งที่เป็นรูปทรงเรขาคณิตและไม่เป็นรูปทรงเรขาคณิต ในศาสตร์ของการประมวลผลภาพนั้น การกำหนดขอบเขตของภาพทุกภาพให้อยู่ในรูปสี่เหลี่ยม (Rectangular image model) เป็นวิธีที่นิยมใช้กันมากที่สุด เนื่องจากทำให้การอ่านภาพ การจัดเก็บข้อมูล ภาพในหน่วยความจำ และการแสดงภาพออกทางอุปกรณ์ต่าง ๆ เป็นไปได้อย่างมีประสิทธิภาพ

การเก็บข้อมูลภาพลงหน่วยความจำของคอมพิวเตอร์สามารถทำได้โดยการจองหน่วยความจำของเครื่องไว้ในรูปของตัวแปรอะเรย์ (array) โดยค่าในแต่ละช่องของอะเรย์แสดงถึงคุณสมบัติของจุดภาพ (pixel) และตำแหน่ง ของช่องอะเรย์เป็นตัวกำหนดตำแหน่งของจุดภาพ

สมมติให้ image เป็นตัวแปรแบบอะเรย์ขนาด $M \times N$ (M แถว และ N คอลัมน์) ที่ใช้เก็บภาพขนาด $M \times N$ จุด (M จุดในแนวนอน และ N จุดในแนวตั้ง) ค่าสี (หรือความสว่าง ในกรณีที่เป็นภาพ grey level) ของจุดภาพใน แถวที่ 5 คอลัมน์ที่ 4 จะตรงกับค่าของ image(5,4) จะเห็นว่าเราใช้ตำแหน่งของจุดภาพทั้งสองแกนเป็นตัวชี้ค่าข้อมูลในอะเรย์จากการใช้หน่วยความจำเพื่อการเก็บภาพ ในลักษณะที่กล่าวมา เนื้อที่ในการเก็บภาพสามารถคำนวณได้ จาก $M \times N \times g$ เมื่อ g เป็นจำนวนเต็มที่แทนจำนวนบิตของข้อมูลในแต่ละจุดภาพตัวอย่างถ้า g มีค่าเท่ากับ 8 บิต เราจะสามารถเก็บความแตกต่างของระดับที่เป็นไปสูงสุด 256 ระดับ ค่า M และ N จะเป็นตัวบอกถึงความละเอียด ของภาพ สำหรับคอมพิวเตอร์ทั่วไปในระบบ VGA (Video Graphic Array) จะมีขนาด 640x480, 800x600 และ 1024x768 จุดเป็นต้น การกำหนดความละเอียดจะขึ้นอยู่กับงานที่จะใช้ ในงานบางอย่างใช้ความละเอียดแค่ 30x50 จุด ก็พอแล้วแต่ในงานบางชนิด ใช้ความละเอียดถึง 1000 x 1000 จุด ก็ยังไม่พอ

ปกติแล้วในการเก็บข้อมูลภาพโดยเครื่องมือต่าง ๆ จะเก็บตามมาตรฐานของโทรทัศน์ซึ่งมีอัตราส่วน x ต่อ y เท่ากับ 4:3 สำหรับเครื่องมือเก็บข้อมูลภาพที่ไม่เป็นไปตามอัตราส่วน 4:3 เมื่อนำภาพนี้ไปแสดงในจอภาพมาตรฐานจะทำให้ภาพที่แสดงนั้นมีขนาดของจุดภาพไม่เป็นสี่เหลี่ยมจัตุรัส เช่น ในบางระบบอาจจะใช้ความละเอียดในการแสดงเท่ากับ 640 x 512 ซึ่งจะทำให้ขนาดของจุดภาพที่ได้มีขนาดของด้านกว้างมีความยาวมากกว่าด้านสูง

จำนวนสีสูงสุดที่เป็นไปได้ของแต่ละจุดภาพขึ้นอยู่กับจำนวนบิตที่ใช้เมื่อมีการกำหนดให้ขนาดของบิตต่อ จุด มากขึ้นจะทำให้จำนวนของสีมากขึ้นด้วย ตัวอย่างเช่น

$$1 \text{ บิต} = 2^1 = 2 \text{ สี}$$

$$2 \text{ บิต} = 2^2 = 4 \text{ สี}$$

$$4 \text{ บิต} = 2^4 = 16 \text{ สี}$$

$$8 \text{ บิต} = 2^8 = 256 \text{ สี}$$

$$16 \text{ บิต} = 2^{16} = 65536 \text{ สี เป็นต้น}$$

สำหรับการแสดงข้อมูลภาพที่มีขนาด 1 บิตและ 8 บิตนั้นจะมีการทำงานที่จะใกล้เคียงกัน เนื่องจากหน่วยประมวลผลจะไม่สามารถจัดการกับข้อมูลที่เป็นบิตเดี่ยว ๆ ได้ดังนั้นในการ

แสดงข้อมูลออกจากจอภาพตัวโปรเซสเซอร์จะทำการก๊อปปี้ข้อมูลทั้ง 8 บิต (1 Byte) ส่งให้กับจอภาพซึ่งในกรณีที่มี Pixel มีขนาด 1 บิต เมื่อโปรเซสเซอร์จะ ทำงานกับบิตแรกที่ต้องการแล้วก็ จะทำการก๊อปปี้ข้อมูลชุดใหม่ที่โดยที่ไม่เกี่ยวกับข้อมูลอีก 7 บิตที่เหลือส่วนใน กรณี Pixel ที่มีขนาด 8 บิต โปรเซสเซอร์จะทำการก๊อปปี้ข้อมูลชุดใหม่ก็ต่อเมื่อโปรเซสเซอร์ทำงานกับทุกบิตแล้ว

2.2.3 มาตรฐานของสี

มาตรฐานของสีที่ใช้ในปัจจุบันมีอยู่หลายระบบด้วยกัน ทั้งนี้ขึ้นอยู่กับนำไปใช้แต่ โดยทั่วไปแล้วทุกมาตรฐานจะมีแนวคิดเดียวกันคือ การแทนจุดสีด้วยจุดที่อยู่ภายในสเปซ 3 มิติ โดยจะมีแกนอ้างอิงสำหรับจุดสี นั้นในสเปซซึ่งแต่ละแกนจะมีความเป็นอิสระต่อกัน ตัวอย่าง เช่นในระบบ RGB จะมีแกนสีคือ แขนงสีแดง เขียว และ น้ำเงินในระบบ HLS จะมีแกนเป็น ค่าสี(hue) ความสว่าง (lightness) และความบริสุทธิ์ของสี (saturation)

ตัวอย่างระบบสีที่ใช้ในโครงการนี้ได้แก่ระบบ RGB YCbCr ระดับสีเทา และโมนารี

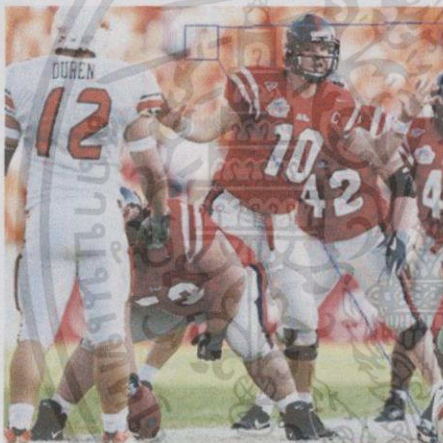
- 1) ระบบสี RGB คือ True color Image เป็นรูปที่เก็บโดยใช้อาร์เรย์ 3 มิติ ขนาด $m \times n \times 3$ โดยที่ m คือความยาว และ n คือความกว้างของภาพในหน่วยจุดภาพ ส่วนมิติสุดท้ายนั้น ในแต่ละมิติจะเก็บค่าสีแยกกัน คือสีแดง (Red) สีเขียว (Green) และสีน้ำเงิน (Blue) RGB ย่อมาจาก red, green และ blue คือระบบสีของแสง เกิดจากการหักเหของแสงกลายเป็นสีรุ้งด้วยกัน 7 สี ซึ่งเป็นช่วงแสงที่ตาของพวกเราสามารถมองเห็นได้ แสงสีม่วงจะมีความถี่สูงสุด เรียกว่า อัลตราไวโอเล็ต และแสงสีแดงจะมีความถี่ต่ำสุด เรียกว่าอินฟราเรด คลื่นแสงที่มีความถี่สูงกว่าสีม่วง และต่ำ กว่าสีแดงนั้น สายตาของมนุษย์ไม่สามารถรับได้ แสงสีทั้งหมดเกิดจาก แสงสี 3 สี คือ สีแดง (Red) สีน้ำเงิน (Blue)และสีเขียว (Green)ทั้งสามสีถือเป็นแม่สีของแสง

แม่สีของแสงมีด้วยกัน 3 สี คือ สีแดง(R) ,สีเขียว(G),สีน้ำเงิน(B) และแต่ละแม่สีเมื่อรวมกันก็จะได้สีดังนี้ สีแดง+สีเขียว ได้ สีเหลือง Yellow

สีเขียว+น้ำเงิน ได้ สีฟ้า Cyan

สีแดง+สีน้ำเงิน ได้ สีแดงอมชมพู Magenta

เมื่อนำแม่สีของแสงทั้ง 3 มาผสมกัน ในปริมาณแสงสว่างเท่ากันก็จะได้เป็นแสงที่สีขาว แต่ถ้าผสมกันระหว่างแสงระดับความสว่างต่างกัน ก็จะได้ผลที่เป็นแสงสีๆ มากมายเป็นล้านสีทีเดียว ส่วนใหญ่การใช้สีลักษณะนี้จะใช้ในอุปกรณ์ที่เกี่ยวกับแสง เช่น จอภาพ กล้องดิจิทัล สแกนเนอร์ เป็นต้น ระบบสี RGB จะการแสดงผลออกมา เป็นรูปแบบการรับแสงแสดงผลด้วยแสงที่เป็นแม่สีได้แก่ สีแดง สีเขียว สีน้ำเงิน ซึ่งอุปกรณ์อิเล็กทรอนิกส์ต่างๆไม่ว่าจะเป็น จอภาพ,สแกนเนอร์,กล้องดิจิทัลหรือดวงตาคนเราล้วนแต่รับและแปลผลเป็นสีต่างๆด้วยแสงเหล่านี้

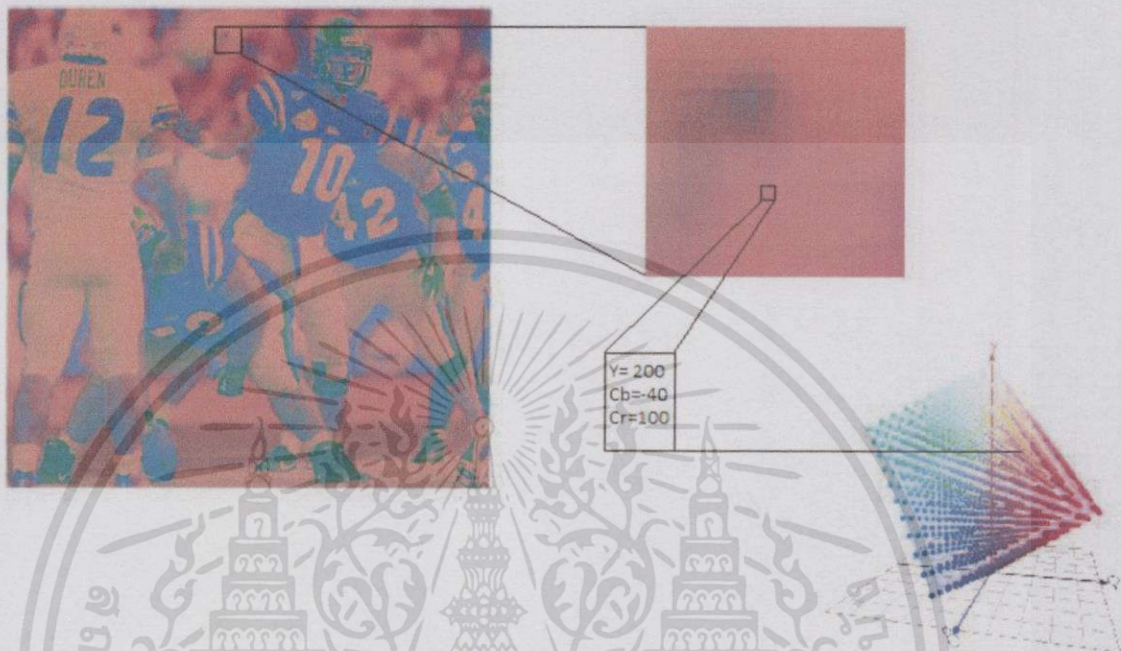


					215	192	176	59	73	128	30	20
					170	197	187	109	104	171	49	28
	164	253	130	67	196	215	192	176				24
	191	229	225	97	108	170	197	187				44
45	5	17	24	24	18	21	14	163				104
0	16	34	7	25	22	17	31	89				171
13	16	18	23	26	14	15	49	33				192
17	20	15	17	27	30	32	45	25				175
15	22	17	16	22	28	30	29	23				169
24	17	18	18	16	16	18	22	11				175
21	12	16	18	13	13	20	28	13				93
25	17	13	12	13	17	23	22	28				4
25	21	13	10	11	14	17	20	56				5
12	17	14	14	15	16	19	34	189				25
10	23	22	17	18	24	19	24	255				21
2	7	20	28	28	34	28	25	250				2
32	20	35	34	20	23	27	28	171				24
31	4	16	24	25	35	39	33	13				
7	20	24	19	17	22	17	11	36				
11	5	5	0	9	12	23	40					
6	0	30	68	97	125	148	168					

รูปที่ 2.1 แสดงองค์ประกอบของข้อมูลภาพในระบบสี RGB

- ระบบโครงสร้างสีวายซีบีซีอาร์ (YCbCr Color Model) ในรูปแบบของโครงสร้างนี้ ค่าปริมาณของแสงในการส่องสว่างจะเก็บข้อมูลนี้ไว้ในส่วนของ (Y) และในส่วนความแตกต่างของสีนั้นจะแบ่งเป็น 2 สี คือ Cb และ Cr โดย Cb จะแสดงให้เห็นถึงความแตกต่างของสีฟ้าและอ้างถึงค่าในหมวดสีฟ้านั้น ส่วน Cr จะแสดงให้เห็นถึงความแตกต่างของสีแดง และอ้าง

ถึงค่าในหมวดสีแดง โดยโครงสร้างสีวายซีบีซีอาร์มีความเที่ยงตรงและแม่นยำมาก โดยค่า Y จะมีค่าในช่วงตั้งแต่ 0 ถึง 255 และค่า Cb,Cr อยู่ในช่วงตั้งแต่ -128 ถึง 127



รูปที่ 2.2 แสดงความสัมพันธ์ของแกนสี YCbCr กับพิกเซลในภาพ

- 3) **Grayscale Image (ภาพระดับสีเทา)** ภาพระดับสีเทาเป็นภาพที่แสดงค่าความเข้มแสง ณ แต่ละตำแหน่งของจุดภาพภายในภาพ ซึ่งภาพจะมีลักษณะเป็นโทนสีเทา โดยค่าความเข้มแสงที่สามารถแสดงได้ ระดับสีเทา จะขึ้นอยู่กับจำนวนบิตที่ใช้แทนภาพในแต่ละภาพได้จาก $L=2^k$ เมื่อ L คือระดับสีเทา และ k คือค่าจำนวนบิตที่ใช้แทนภาพเช่น ภาพระดับสีเทาขนาด 8 บิตสามารถแสดงค่าระดับสีเทาได้ทั้งหมด 256 ระดับ และค่าที่เป็นไปได้คือ 0-255

ภาพระดับสีเทาหมายถึงภาพที่มีค่าความสว่างของแต่ละจุดภาพอยู่ในช่วงสีดำ สีเทา เรื่อยไปจนถึงสีขาวซึ่งเป็นค่าความสว่างที่มากที่สุด สำหรับค่าของระดับสีเทานั้น ปกติแล้วจะเป็นกำลังของ 2 ซึ่งโดยทั่วไปที่ใช้จะเป็น 8 บิต (256ระดับ) หรืออาจใช้จำนวนบิตมากหรือน้อยกว่านี้ขึ้นอยู่กับความละเอียดของภาพที่ความต้องการว่ามากกว่าหรือน้อยเพียงใด

2.3 การปรับปรุงภาพ

การปรับปรุงภาพ (Image enhancement) [3] เป็นกระบวนการในการแปลงข้อมูลภาพตัวเลขเพื่อที่จะสร้างภาพที่เน้นรายละเอียดที่ต้องการ หรือปรับพิสัยของโทนแสงที่ต้องการของภาพ เมื่อเปรียบเทียบกับข้อมูลหรือรายละเอียดอื่นๆ ของภาพ เมื่อทำการปรับภาพจะมีการเน้นสารสนเทศใน ข้อมูลบางส่วน และอาจจะไปกดสารสนเทศในข้อมูลอีกส่วนหนึ่ง ผู้ใช้อาจจะต้องตัดสินใจใช้วิธีการปรับปรุงภาพหลายๆ แบบในการทำงานครั้งหนึ่งๆ ซึ่งการปรับปรุงภาพแต่ละวิธี จะช่วยในการแปลภาพในแง่มุมที่แตกต่างกันในโครงการหนึ่งๆ

ผู้ใช้จำเป็นต้องรู้ลักษณะการสะท้อนแสงของพื้นที่ที่ถ่ายภาพ พอๆ กับที่ต้องรู้จักผลที่ได้จากการใช้วิธีต่างๆ ในการปรับปรุงภาพ เพราะมิฉะนั้นจะทำให้การแปลความหมายผิดไป หรือไปกดสารสนเทศที่สำคัญ การปรับปรุงภาพโดยการเน้นบางส่วน และลดบางส่วน จะช่วยให้ผู้ใช้สามารถสกัดสารสนเทศที่สำคัญได้อย่างประหยัด ถูกต้องและแม่นยำ แต่เนื่องจากการมอง เป็นผลทั้งจาก สรีระและจิตวิทยา ดังนั้นผู้แปลแต่ละคนจะมีความชอบแตกต่างกันไป การเลือกประเภทการแปลจะ ขึ้นอยู่กับผู้แปลเท่านั้น ซึ่งมีวิธีการมากมาย อีกประการหนึ่ง คือ ทุกครั้งที่มีการเปลี่ยนวิธีในการปรับ ภาพ ผู้ใช้จะต้องเรียนรู้ความหมายของสีที่แสดงออกมา ทางที่ดี คือ พยายามใช้วิธีการที่มีอยู่เท่าที่ จำเป็น อย่าใช้หลายอย่างปนกัน มิฉะนั้นข้อมูลบางส่วนอาจจะหายไป ที่สำคัญอีกประการ คือ ต้องมี จอสีที่มี คุณภาพสูง เร็ว และราคาถูกในการแสดงผลภาพ

2.3.1 การแปลงรูปแบบสีจาก RGB เป็น YCbCr

$$Y = (0.299 \times R) + (0.587 \times G) + (0.114 \times B) \quad (2.1)$$

$$Cb = (-0.168736 \times R) - (0.331264 \times G) + (0.5 \times B) \quad (2.2)$$

$$Cr = (0.5 \times R) - (0.418688 \times G) - (0.081312 \times B) \quad (2.3)$$

โดยกำหนดให้ค่าของ R,G และ B เป็นค่าของสีแดง เขียว และสีน้ำเงินตามลำดับ เมื่อทำการแยกทำที่สีพิกลเซลล์ที่ของรูปภาพแล้วจะได้ภาพที่เป็นเฉดสี YCbCr



รูปที่ 2.5 แสดงภาพก่อนและหลังการแปลงภาพสีจาก RGB เป็น YCbCr

2.3.2 การแปลงภาพธรรมดาให้เป็นภาพระดับสีเทา

การแปลงภาพธรรมดาให้เป็นภาพระดับสีเทาคือการแปลงภาพอินพุทภาพจริงที่เป็นภาพสีหรือภาพ RGB ให้เป็นภาพระดับสีเทาที่มีความเข้มแสงตั้งแต่ขาวแล้วเข้มขึ้นเป็นสีเทาจนเข้มมาที่สุดเป็นสีดำ โดยจะกำจัดเฉดสีให้ภาพกลายเป็นภาพที่เป็นภาพสีเทาที่มีระดับความเข้มของสีเทาตั้งแต่ 0-255 หรือ 8 บิต โดยฟังก์ชันจะใช้สูตรทางคณิตศาสตร์คือ

$$\text{Gray} = (0.299 \times R) + (0.587 \times G) + (0.114 \times B) \quad (2.4)$$

Gray = ค่าความเข้มของสีเทาโดยจะมีค่าระหว่าง 0-255

R = ค่าความเข้มของสีแดงโดยจะมีค่าระหว่าง 0-255

G = ค่าความเข้มของสีเขียวโดยจะมีค่าระหว่าง 0-255

B = ค่าความเข้มของสีน้ำเงินโดยจะมีค่าระหว่าง 0-255

โดยขั้นตอนการคำนวณจะทำที่ละพิกเซลไปเรื่อยๆจนครบทุกพิกเซลเมื่อทำตามสูตรแล้วก็จะได้ภาพใหม่ซึ่งจะเป็นภาพที่มีโทนสีขาวไปจนถึงดำ

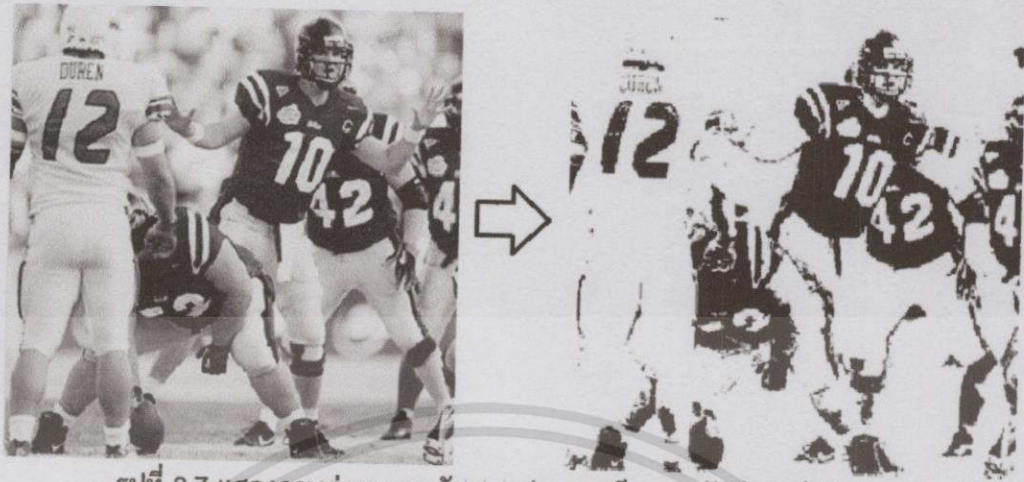


รูปที่ 2.6 แสดงภาพก่อนและหลังการแปลงภาพสีจาก RGB เป็นระดับสีเทา

2.3.3 การแปลงภาพระดับสีเทาให้เป็นภาพไบนารี

การแปลงภาพระดับสีเทาให้เป็นภาพไบนารีจะเป็นการแปลงภาพต่างๆ ให้การเป็นภาพที่มีเฉดสีเพียง 2 เฉดคือขาวและดำ โดยแทนค่าด้วยเลขไบนารี 0 คือสีดำ 1 คือสีขาว โดยจะให้ผู้ใช้งานกำหนดเองว่าจะให้เฉดสีตั้งแต่เท่าไรเป็นค่าแยกความแตกต่างระหว่างสีขาวและสีดำ โดยการแปลงค่าจากภาพระดับสีเทาเป็นไบนารีนั้นจะขึ้นอยู่กับค่า เทรสโธลด์ ค่าเทรสโธลด์คือค่าในการแบ่งว่า พิกเซลนั้นจะเป็นสีขาวหรือสีดำ โดยเมื่อนำค่าในภาพเกรสเกลที่มีค่าตั้งแต่ 0-255 นั้นมาหารด้วย 255 จะได้ค่าหนึ่งที่มีค่า 0-1 แล้วจะใช้ค่าเทรสโธลด์เป็นตัวแบ่งว่าจะป็นสีขาวหรือสีดำ เมื่อได้ค่าที่ได้จากการหารด้วย 255 เมื่อค่าในพิกเซลนั้นมีค่าน้อยกว่า 0.4 พิกเซลนั้นจะมีค่าเป็น 0 (สีดำ) และถ้าหากมีค่าตั้งแต่ 0.4-1 ค่านั้นจะมีค่าเป็น 1 (สีขาว) ถ้าหากว่าค่าเทรสโธลด์มีค่าน้อย (เข้าใกล้ 0) ภาพไบนารีก็จะมีพื้นที่สีขาวมาก แต่ถ้าค่าเทรสโธลด์มีค่ามาก (เข้าใกล้ 1) ภาพไบนารีก็จะมีพื้นที่สีดำมาก โดยอัลกอริธึมนี้จะช่วยในการแยกแยะวัตถุต่างๆ ออกจากกันอย่างชัดเจน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แสดงภาพก่อนและหลังการแปลงภาพสีจากระดับสีเทาเป็นไบนารี

2.3.4 การปรับเปลี่ยนหรือการเปลี่ยนแปลงค่าความเข้มแสงของภาพ (intensity manipulation)

การปรับเปลี่ยนหรือการเปลี่ยนแปลงค่าความเข้มแสงของภาพ [4] เป็นเทคนิคที่อาศัยฟังก์ชันทางคณิตศาสตร์หรือค่าทางสถิติของภาพ เพื่อทำการแปลงค่าความเข้มแสงของภาพต้นฉบับให้ได้คุณภาพของภาพหลังการแปลงค่าความเข้มแสงมีคุณภาพดีขึ้นเหมาะกับการนำไปใช้งาน ค่าพารามิเตอร์ของการแปลงสามารถกำหนดได้โดยตรงจากผู้ใช้งาน (Transform Technique) หรือสามารถปรับเปลี่ยนค่าการเปลี่ยนแปลงอัตโนมัติตามคุณสมบัติทางสถิติของภาพโดยตรง (Histogram-based technique)

เทคนิคในการแปลงความเข้มแสงของภาพ (Intensity Transform Technique)

เทคนิคในการแปลงความเข้มแสงของภาพเป็นเทคนิคที่ง่ายที่สุดสำหรับการปรับปรุงคุณภาพของภาพ โดยจะใช้ฟังก์ชันการแปลงความเข้มแสง (Intensity Transform Function) ฟังก์ชันเหล่านี้จะใช้เพื่อทำการแปลงความเข้มแสงขาเข้า ($f(x, y)$) ให้เปลี่ยนเป็นความเข้มแสงผลลัพธ์ ($g(x, y)$) ดังสมการ

$$(g(x, y)) = T(f(x, y)) \quad (2.5)$$

โดยที่

T	คือ ฟังก์ชันการเปลี่ยนแปลงความเข้มแสง
$(f(x, y))$	คือ ค่าความเข้มแสงขาเข้า
$(g(x, y))$	คือ ค่าความเข้มแสงผลลัพธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในทางปฏิบัติเนื่องจากเรารู้ช่วงของค่า $(f(x, y))$ ที่แน่นอนดังนั้นค่าของ $(g(x, y))$ สามารถคำนวณเป็นตารางการแปลงความเข้มแสงล่วงหน้าได้ เมื่อต้องการแปลงความเข้มแสงเพียงอ่านค่าความเข้มแสงขาเข้า $(f(x, y))$ เพื่อนำมาเปิดค่าการแปลงความเข้มแสงที่เก็บไว้ในตารางอ่านค่าเพื่อนำไปเป็นความเข้มแสงผลลัพธ์ การทำเช่นนี้ทำให้ลดเวลาในการคำนวณลงได้มาก เนื่องจากเวลาที่ใช้ในการคำนวณตารางการแปลงความเข้มแสงจะคงที่ขึ้นอยู่กับจำนวนระดับค่าความเข้มแสงของภาพ แต่ไม่ขึ้นกับขนาดของภาพ ดังนั้นไม่ว่าภาพจะใหญ่หรือเล็กเวลาในการแปลงความเข้มแสงจะไม่ต่างกันมากนัก

2.3.5 การปรับความเข้มแสงโดยใช้ฟังก์ชันลอการิทึม(Logarithm Function)

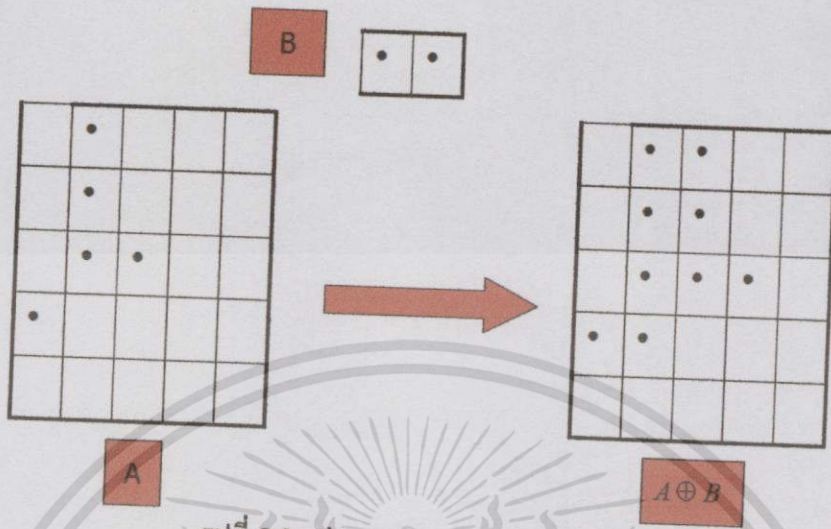
เราสามารถปรับปรุงคุณภาพของภาพได้โดยใช้ฟังก์ชันลอการิทึมได้ดังสมการ

$$g(x, y) = \log(1 + f(x, y)) \quad (2.6)$$

ฟังก์ชันลอการิทึมนี้มีประโยชน์มากในการขยายรายละเอียดพื้นที่มืดในภาพ โดยแลกเปลี่ยนกับการสูญเสียรายละเอียดในพื้นที่สว่างไปบางส่วน การขยายโดยใช้ฟังก์ชันลอการิทึมนี้ให้ผลของการปรับในช่วงความเข้มแสงในพื้นที่ที่ต้องการ และมีผลกระทบกับส่วนอื่นของภาพเพียงเล็กน้อยเมื่อเปรียบเทียบกับการใช้ฟังก์ชันเส้นตรง

2.3.6 การขยายภาพ (Dilation)

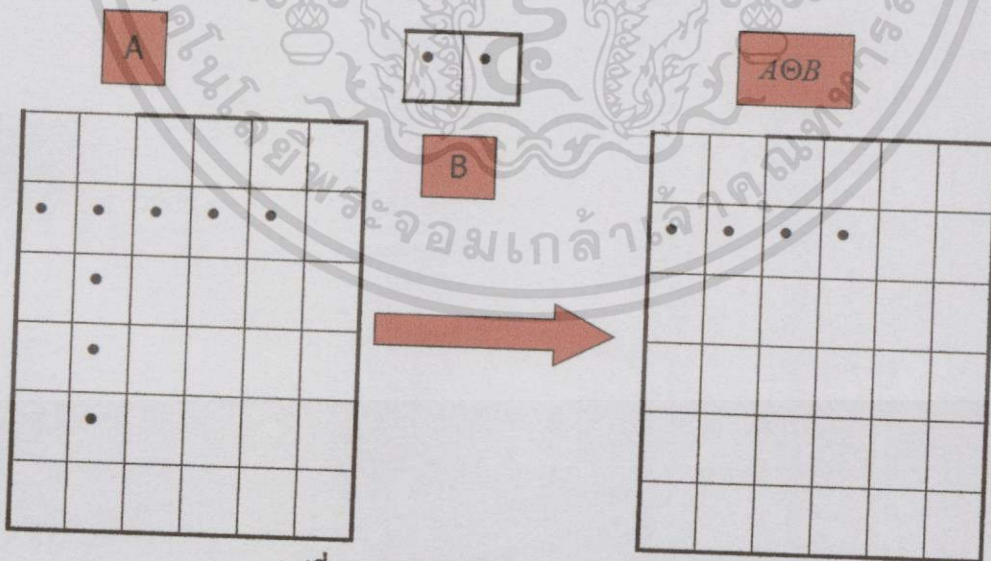
การขยายภาพในที่นี้จะพิจารณาสำหรับข้อมูลภาพที่เป็นแบบไบนารีโดยการใช้เทคนิคการฮิต และ มิส การขยายภาพจะทำได้โดยกำหนดเทมเพลต ซึ่งสามารถสร้างได้จาก 0 และ 1 โดยมีจุดเริ่มต้น (Origin) ที่กำหนด และนำเทมเพลต นี้สแกนไปบนข้อมูลภาพตามลำดับตลอดทั้งภาพ ซึ่งในขณะที่จุดเริ่มต้น (Origin) ของเทมเพลตตรงกับตำแหน่งข้อมูลภาพที่พิกเซลมีค่าเท่ากับ 1 นั่นก็จะปรับค่าพิกเซลของภาพที่เทมเพลตนั้นหาพบอยู่ให้เป็น 1 ทั้งหมด



รูปที่ 2.8 รูปแสดงกระบวนการขยายภาพ

2.3.7 การย่อภาพ (Erosion)

การย่อภาพเป็นลักษณะของการลบข้อมูลภาพบริเวณขอบของภาพ การย่อภาพสามารถทำได้มีลักษณะคล้ายกับการขยายภาพโดยการสร้างเทมเพลต ขึ้นแล้วนำเทมเพลตไปสแกนตามข้อมูลภาพสำหรับทุกตำแหน่งที่เลื่อนเทมเพลต ไปบนภาพก็จะมีการเปรียบเทียบกับข้อมูลภาพ ถ้าข้อมูลภาพมีค่าเหมือนกับเทมเพลตจะทำ การกำหนดค่าข้อมูลภาพในตำแหน่งที่ตรงกับจุดเริ่มต้น ของเทมเพลตถูกกำหนดให้มีค่าเท่ากับ 1



รูปที่ 2.9 รูปภาพแสดงกระบวนการย่อภาพ

2.3.8 โอเพอเรชัน Closing และ Opening

โอเพอเรชันการ Open

กำหนดให้ OPEN (I, T) เป็นการกระทำ Opening ของภาพ I โดยใช้เทมเพลต T ซึ่งมีลักษณะดังสมการต่อไปนี้

$$\text{OPEN (I, T)} = D(E(I)) \quad (2.7)$$

จากสมการจะได้อาการกระทำโอเพอเรชัน OPEN คือการนำขอมูลสภาพ I ผ่านการทำให้ขยายภาพ (Erosion) แล้วตามด้วยการย่อภาพ (Dilation) โดยใช้เทมเพลตชุดเดียวกันคือ T

โอเพอเรชันการ Close

กำหนดให้ CLOSE (I, T) เป็นการกระทำแบบ Closing ของภาพ I โดยใช้เทมเพลต T ซึ่งมีลักษณะดังสมการต่อไปนี้

$$\text{CLOSE (I, T)} = E(D(I)) \quad (2.8)$$

จากสมการจะได้อาการกระทำโอเพอเรชัน CLOSE คือการนำขอมูลสภาพ I ผ่านการทำให้การย่อภาพ (Dilation) แล้วตามด้วยการขยายภาพ (Erosion) โดยใช้เทมเพลต ชุดเดียวกันคือ T

2.3.9 การกระจายตัวของสัญญาณรบกวนแบบเกาส์เซียน (Gaussian Noise)

สัญญาณรบกวนแบบเกาส์เซียนเป็นรูปแบบของสัญญาณรบกวนที่ถูกนำไปใช้ในทางปฏิบัติจริงบ่อยครั้ง เนื่องจากทางในคณิตศาสตร์ค่าฟังก์ชันการกระจายของความน่าจะเป็นอยู่ในรูปของการกระจายตัวแบบปกติ (Normal Distribution) ซึ่งมีรูปแบบเดียวกันในโดเมนตำแหน่งและโดเมนความถี่ด้วยข้อเด่นนี้แบบจำลองเกาส์เซียนมักถูกอ้างอิงและนำไปประยุกต์ใช้ในงานต่างๆที่มีแนวโน้มว่าจะใช้แบบจำลองเกาส์เซียนนี้ได้ดีที่สุด

ฟังก์ชันการกระจายตัวของความน่าจะเป็นของตัวแปรสุ่มเกาส์เซียน (Gaussian Random Variable, z) แสดงดังต่อไปนี้

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \quad (2.9)$$

โดยค่า z แทนระดับความเข้มแสงสีเทา μ เป็นค่าเฉลี่ยของค่า z และ σ เป็นค่าเบี่ยงเบนมาตรฐาน ซึ่งค่ากำลังสองของค่าเบี่ยงเบนมาตรฐานเรียกว่า “ค่าความแปรปรวนของค่า z (Variance of z)”

2.3.10 การปรับภาพเชิงพื้นที่

ในขณะที่การปรับภาพเชิงรังสีเป็นการแปลงค่าจุดภาพเดี่ยว แต่การปรับภาพเชิงพื้นที่เป็นการเน้นภาพที่ใช้ ค่าจากจุดภาพที่อยู่รอบๆ จุดภาพนั้นมาคำนวณด้วย การปรับภาพเชิงพื้นที่จะเกี่ยวข้องกับ “ความถี่เชิงพื้นที่” ซึ่งหมายถึง ความแตกต่างระหว่างค่าสูงสุดและต่ำสุดของกลุ่มจุดภาพที่อยู่ติดกันหรืออยู่ใกล้เคียงกัน หรือจำนวนการเปลี่ยนแปลงค่าความสว่างต่อหน่วยระยะทาง ในส่วนหนึ่งของภาพ ภาพที่ 2.3 แสดงความถี่เชิงพื้นที่แบบง่ายๆ เช่น ความถี่เชิงพื้นที่เป็นศูนย์ เป็นภาพเรียบๆ ทุกจุดภาพมีค่าเท่ากัน ความถี่เชิงพื้นที่ต่ำ เป็นภาพที่ค่อยๆ มีการเปลี่ยนแปลงของค่าหรือโทนสี ความถี่เชิงพื้นที่สูง เป็นภาพที่มีการติดกันของโทนสีขาวดำอย่างชัดเจน หรือมีการเปลี่ยนโทนสีอย่างฉับพลัน



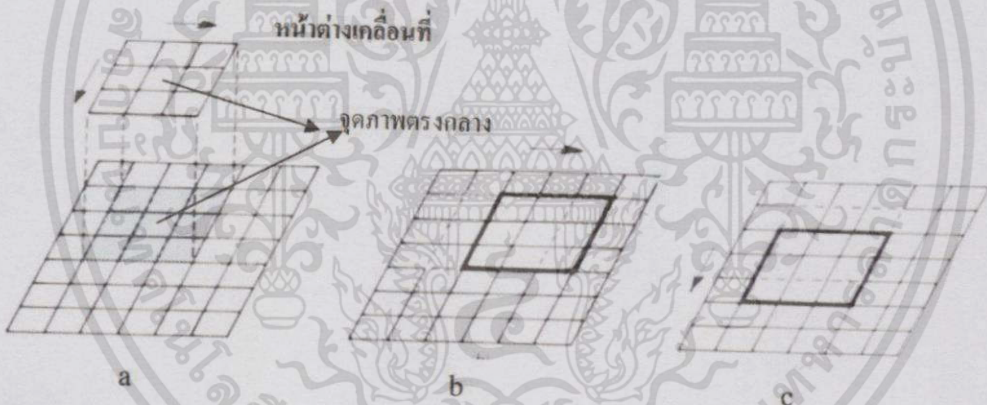
รูปที่ 2.10 ความถี่เชิงพื้นที่ของจุดภาพ ในภาพพื้นที่หนึ่ง

การปรับภาพเชิงพื้นที่มักจะใช้เทคนิคการกรองภาพ (filtering) ซึ่งหมายถึงการแปลงข้อมูลเพื่อลดสัญญาณรบกวน หรือปรับลักษณะบางอย่างของภาพโดยการเน้น หรือลดความถี่ข้อมูลเชิงพื้นที่ บางครั้งเป็นการปรับภาพเพื่อการสกัดลักษณะเด่น (feature extraction) ให้เห็นชัดขึ้น ผลจากการใช้เทคนิคนี้ช่วยให้การแปลภาพด้วยสายตาดูดีขึ้น และมักจะเปลี่ยนค่าข้อมูลดั้งเดิมอย่างถาวร ดังนั้นการใช้เทคนิคการกรองภาพจะใช้ในกรณีที่ ข้อมูลดั้งเดิมไม่ใช่สิ่งสำคัญในการเอาไปใช้ต่อในการคำนวณ

2.3.11 การกรองภาพแบบคอนโวลูชัน (convolution filtering)

การกรองภาพจะใช้วิธีการทางคณิตศาสตร์ที่เรียกว่า การกรองภาพแบบคอนโวลูชัน ซึ่งเป็นกระบวนการ การเฉลี่ยค่าของจุดภาพภายในเขตเล็กหนึ่งๆ และกระทำตลอดทั่วทั้งภาพ

- 1) หน้าต่างเคลื่อนที่ (moving window) จะทำหน้าที่กลายเป็นแมทริกซ์ (matrix) ซึ่งได้จากค่าสัมประสิทธิ์ (coefficients) หรือปัจจัยน้ำหนัก (weighting factors) ที่ใช้เป็นตัวถ่วงน้ำหนัก ทำหน้าที่เหมือนตะแกรงที่มีช่องขนาดต่างๆ หน้าต่างเคลื่อนที่นี้ตามภาษาคณิตศาสตร์เรียกว่า เคอร์เนล(kernel) หน้าต่างเคลื่อนที่ที่จะเป็นตารางสี่เหลี่ยมของจุดภาพจำนวนคือ เช่น 3×3 หรือ 5×5 หรือ 7×7 จุดภาพ
- 2) หน้าต่างเคลื่อนที่ที่จะเคลื่อนที่ผ่านที่ละจุดภาพตลอดทั่วทั้งภาพดั้งเดิม และค่าตัวเลขที่อยู่จุดภาพตรงกลางของหน้าต่างเคลื่อนที่ จะถูกคำนวณเพื่อเอาไปใส่ในภาพใหม่ตรงตำแหน่งเดิมโดยวิธีการ คูณสัมประสิทธิ์แต่ละตัวในหน้าต่างเคลื่อนที่ ด้วยค่าตัวเลขของภาพดั้งเดิมที่ทับโดยหน้าต่างเคลื่อนที่ แล้วเอาผลคูณทั้งหมดมาบวกกัน การคำนวณนี้จะทำที่ละจุดภาพในภาพดั้งเดิม



ภาพที่ 2.11 การเคลื่อนที่ของหน้าต่างเคลื่อนที่

- (a) หน้าต่างเคลื่อนที่ ขนาด 3×3 จุดภาพ เคลื่อนที่ไปบนภาพที่จะทำการประมวล
- (b) เลื่อนไปตามแถวจากจุดภาพหนึ่งไปยังอีกจุดภาพหนึ่งจนหมดแถว
- (c) เลื่อนหน้าต่างเคลื่อนที่จากแถวหนึ่งไปยังอีกแถวหนึ่ง

2.3.12 เครื่องกรองผ่านความถี่ต่ำ (low-pass filter)

เป็นเทคนิคการกรองภาพที่ บริเวณที่มี ความถี่เชิงพื้นที่ต่ำสามารถผ่านไปได้ ส่วนบริเวณที่มีความถี่สูงกว่าที่กำหนดโดยหน้าต่างเคลื่อนที่จะถูกกรองให้ลดความถี่ลง การกรองเพื่อลดความถี่เชิงพื้นที่นี้ ทำให้ภาพที่มีความถี่เชิงพื้นที่สูงมีความนุ่มนวลขึ้น มักใช้ในการลบหรือลดสัญญาณรบกวนที่เกิดขึ้นในขณะที่ตรวจจับภาพ หรือในขณะที่ บันทึกภาพ โดยทั่วไปการกรองด้วยหน้าต่างเคลื่อนที่ในลักษณะนี้ จะใช้กับข้อมูลที่ได้จากภาพที่ถ่าย ด้วยเรดาร์ ตัวอย่างการกรองแบบนี้ ได้แก่ การเฉลี่ยค่าจุดภาพในหน้าต่างเคลื่อนที่ 3×3 หรือ 5×5 แล้วแทนที่จุดภาพตรงกลาง หรือ การใช้ค่าฐานนิยมหรือค่ามัธยฐาน แทนที่จุดภาพตรงกลางของ หน้าต่างเคลื่อนที่ การกรองโดยใช้ค่าฐานนิยมมักจะใช้กับภาพที่มีการจำแนกแล้ว (post classification) เพื่อจะกำจัดจุดภาพเล็ก ๆ ที่เกิดจากการจำแนกผิด ส่วนการกรองด้วยค่ามัธยฐาน ใช้กับข้อมูลที่ ตัวเลขมีความหมาย หรือมีลำดับ การกรองด้วยค่าเฉลี่ยใช้กับข้อมูลแบบต่อเนื่องเช่นเดียวกับ ข้อมูลภาพดาวเทียม เพื่อปรับภาพให้ดูนุ่มนวลขึ้น ตัวอย่างภาพที่ได้จากการกรองภาพด้วยวิธีนี้ แสดง ในภาพที่ 2.5 ซึ่งภาพผลลัพธ์จะเบลอกว่าภาพเดิม ขนาดของหน้าต่างเคลื่อนที่ มีความสำคัญต่อการกรองแบบนี้ ถ้าหน้าต่างเคลื่อนที่มีขนาดใหญ่ขึ้น ภาพจะยิ่งเบลอมากขึ้น เพราะจุดภาพตรงกลางจะถูกคาดคะเนมาจากจุดภาพจำนวนมากขึ้นตามขนาดของหน้าต่างเคลื่อนที่ ตัวอย่างการกรองแบบเครื่องกรองผ่านความถี่ต่ำ ผลรวมของตัวเลขในหน้าต่างเคลื่อนที่จะมีค่าเป็น 1

$1/9$	$1/9$	$1/9$		0	$1/5$	0
$1/9$	$1/9$	$1/9$	หรือ	$1/5$	$1/5$	$1/5$
$1/9$	$1/9$	$1/9$		0	$1/5$	0



รูปที่ 2.12 การปรับภาพโดยใช้การกรองด้วยหน้าต่างเคลื่อนที่แบบต่างๆ (ภาพบริเวณอ่าวปัตตานี)

2.3.13 การจับขอบ (edge detection)

เทคนิคนี้มีวัตถุประสงค์เพื่อแสดงถึงรอยต่อเชิงพื้นที่ของวัตถุ ไม่ว่าจะเป็น จุด เส้น หรือโชน การกรองนี้ใช้กับปรากฏการณ์ที่การกระจายเชิงพื้นที่ไม่ติดต่อกัน วิธีการคำนวณที่ใช้เรียกว่า การเปลี่ยนแปลงเชิงตำแหน่ง (spatial gradient) การคำนวณการเปลี่ยนแปลงเชิงตำแหน่ง สำหรับแต่ละจุดภาพ ใช้การกรองแบบเครื่องกรองผ่านความถี่ สูง ลักษณะเฉพาะของเครื่องกรองนี้ คือ ผลบวกของสัมประสิทธิ์ในหน้าต่างเคลื่อนที่มีค่าเป็นศูนย์ (zero-sum) ดังนั้นผลรวมของค่าในหน้าต่างเคลื่อนที่ไม่ต้องเอาไปหารค่าที่คำนวณได้ เพราะการหารด้วยศูนย์หาค่าไม่ได้ โดยทั่วไปผลลัพธ์ที่ได้จากการใช้หน้าต่างเคลื่อนที่แบบนี้ คือ พื้นที่ที่ข้อมูลนำเข้ามีค่าเท่ากัน ผลที่ได้จะมีค่าเป็นศูนย์ พื้นที่ที่มีความถี่ต่ำ ผลที่ได้จะมีค่าต่ำ พื้นที่ที่มีความถี่สูง ค่าผลที่ได้จะมีค่าสูงมาก ดังนั้น หน้าต่างเคลื่อนที่ zero-sum จึงทำให้บริเวณที่มีความถี่เชิงพื้นที่ต่ำ มีค่าเป็นศูนย์หรือมีค่าต่ำ ในขณะที่บริเวณที่มีความถี่สูง มีค่าแตกต่างกันอย่างชัดเจน เช่น บริเวณที่เป็นรอยต่อของพื้นที่ภาพผลลัพธ์ที่ได้ คือ บริเวณที่เป็นขอบจะเห็นชัดเจน ในขณะที่บริเวณกลุ่มจุดภาพที่มีความกลมกลืนกัน (homogenous) จะมีค่าเป็นศูนย์ หรือปรากฏเป็นสีดำ

ขอบซึ่งจะเห็นเฉพาะขอบ แต่ส่วนอื่นจะหายไป เทคนิคนี้มีประโยชน์ในการทำภาพพิมพ์ หรือต้องการให้ภาพบนหน้าจอชัดขึ้น และไม่จำเป็นต้องใช้กับข้อมูลที่เป็นแผนที่

ตัวอย่างการใช้เครื่องกรองแบบนี้จะเน้นที่จุดกลางของหน้าต่างเคลื่อนที่

-1	-1	-1
-1	16	-1
-1	-1	-1

ถ้าเราใช้หน้าต่างเคลื่อนที่นี้กับกลุ่มของจุดภาพที่มีค่าสูงล้อมรอบจุดภาพตรงกลางที่มีค่าต่ำ จุดภาพที่มีค่าต่ำ (100) ก็จะมีค่าต่ำลง (0)

2.4 แมทแลป (Matlab)

2.4.1 ความหมายและที่มาของแมทแลป

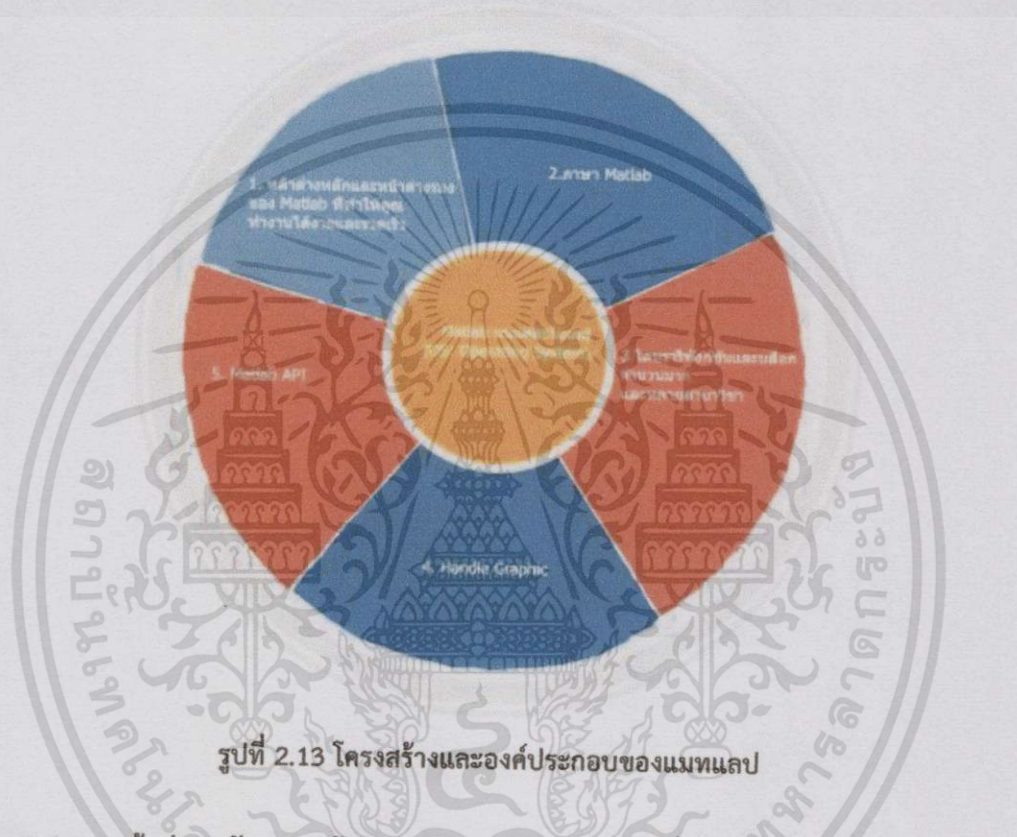
แมทแลป[5] เป็น ภาษาคอมพิวเตอร์ระดับสูงที่มาพร้อมด้วยสภาพแวดล้อมการทำงานเชิงโต้ตอบ(คล้ายเครื่องคิดเลข) ซึ่งสามารถคำนวณคณิตศาสตร์ที่ซับซ้อนได้อย่างรวดเร็วมากกว่าภาษาคอมพิวเตอร์สมัยก่อน เช่น ภาษา ซี(C), ซีพลัสพลัส(C++) และ ฟอรัทราน (Fortran)

แมทแลป เป็นภาษาคอมพิวเตอร์ระดับสูงที่ใช้สำหรับคำนวณเชิงตัวเลข (Numerical Computing) แสดงผลกราฟฟิก และเขียนแอปพลิเคชัน สามารถคำนวณผลลัพธ์ พัฒนาอัลกอริทึม สร้างแบบจำลอง และแอปพลิเคชันได้ง่ายและรวดเร็วมาก ภายในตัว แมทแลป ประกอบด้วย ภาษาคอมพิวเตอร์ ทูลบ็อกซ์(Toolbox: กลุ่มฟังก์ชันสำเร็จรูปในแต่ละสาขาวิชา) และฟังก์ชันพื้นฐานจำนวนมาก ทำให้การวิเคราะห์ทำได้หลากหลายวิธี พร้อมกับคำตอบที่รวดเร็วกว่าโปรแกรมตารางคำนวณ(Spreadsheet) หรือภาษาคอมพิวเตอร์สมัยก่อน เช่น ภาษาซี, ซีพลัสพลัส, ฟอรัทราน, จาวา และอื่นๆ

แมทแลป สามารถนำไปประยุกต์ใช้งานได้หลายสาขา ทั้ง การประมวลผลสัญญาณ(Signal Processing) การสื่อสาร(Communication) การประมวลผลภาพและวิดีโอ(Image and Video Processing) ระบบควบคุม(ControlSystem) การวัดและควบคุม(Instruments and Control) การคำนวณทางเศรษฐศาสตร์(Economic) การคำนวณทางชีววิทยา(Biology) และอื่นๆ มีนักวิทยาศาสตร์และวิศวกรหลายล้านคนทั้งในแวดวงการศึกษาและอุตสาหกรรมที่ใช้แมทแลปในการคำนวณเชิงตัวเลข

2.4.2 โครงสร้างและองค์ประกอบของแมทแลป

แมทแลป[5] ได้ออกแบบมาเพื่อสนับสนุนการทำงานของผู้ใช้อยู่ 5 ส่วน เพื่อช่วยในการวิเคราะห์ข้อมูล แสดงผลข้อมูล เชื่อมต่อกับสิ่งต่างๆ ภายนอกได้อย่างรวดเร็วและมีประสิทธิภาพสูงสุดดังรูป



รูปที่ 2.13 โครงสร้างและองค์ประกอบของแมทแลป

2.4.2.1 หน้าต่างหลักและหน้าต่างรองของ แมทแลปหรือที่เรียกกันว่า Matlab Desktop Environment ช่วยให้ผู้ใช้ทำงานได้ง่ายและรวดเร็วมากยิ่งขึ้น ในส่วนนี้ประกอบด้วยชุดเครื่องมือที่ช่วยให้เราสามารถใส่ฟังก์ชันและไฟล์ต่างๆ ด้วยเครื่องมือแบบรูปภาพ (GUI) ประกอบด้วยหน้าต่างย่อย Current Folder, Command Windows, Command History และ Workspace

2.4.2.2 ภาษา แมทแลป ใช้สำหรับเขียนแอปพลิเคชันหรือฟังก์ชันของตนเอง ไว้ใช้งาน โดยเฉพาะ แมทแลป โดยใช้ Matlab Editor เขียนในรูปฟังก์ชัน m-file พร้อมเครื่องมือตรวจสอบบัคในโปรแกรม

2.4.2.3 ไลบรารีฟังก์ชันและบล็อกโคอะแกรมจำนวนมากและครอบคลุมหลายสาขาวิชา ส่วนนี้จะรวบรวมฟังก์ชัน m-file หรือ mdl ของ Simulink เป็นไฟล์ย่อยๆ ไว้ โดยแต่ละไฟล์จะเป็นไฟล์ที่สร้างขึ้นมาเพื่อใช้กำหนดลักษณะในการคำนวณ เราเรียกว่า อัลกอริทึม (Algorithms) แบบต่างๆ เริ่มจากฟังก์ชันง่ายๆ เช่นการบวก ฟังก์ชันตรีโกณมิติพื้นฐาน เช่น sine, cos, tan ไปจนถึงฟังก์ชันที่มีความซับซ้อนมีขั้นตอนในการคำนวณมากๆ เช่น การหาอินเวิร์สของเมตริกซ์ การหาค่าสมการเชิงซ้อน หรือการหาทรูโกลค์ส เป็นต้น ส่วนนี้ประกอบด้วยฟังก์ชันพื้นฐาน เช่น บวก ลบ คูณ ทหาร sine, cost, log และฟังก์ชันเฉพาะสาขาวิชา ซึ่งเราเรียกว่าทูลบ็อกซ์ (Toolbox) เช่น Control System, Bioinformatics , Signal Processing, Fuzzy Logic , Aerospace, Image Processing, Econometrics และอื่นๆ

2.4.2.4 Handle Graphics เป็นไลบรารีฟังก์ชันในแมทแลปสำหรับแสดงผลข้อมูล เป็นกราฟฟิก รูปภาพ เสียง วิดีโอ พร้อมด้วย Library Guide เป็นเครื่องมือสำหรับสร้างแอปพลิเคชันแบบ GUI (Graphic User Interface) โดยเลียนแบบการเขียนโปรแกรมจาก Visual Basic ดังนั้นทำให้คุณสามารถนำอัลกอริทึมไปพัฒนาแอปพลิเคชันบน แมทแลปเพื่อใช้งานเองหรือแจกจ่ายได้อย่างรวดเร็ว

2.4.2.5 Matlab API (API: Application Program Interface) เป็นส่วนติดต่อกับภาษาคอมพิวเตอร์อื่นๆ โปรแกรมภายนอก นามสกุลไฟล์ต่างๆ และสามารถพัฒนาให้เชื่อมต่อกับฮาร์ดแวร์ได้ง่าย

2.4.3 ข้อดีของแมทแลป

- โปรแกรมใช้งานง่ายทำงานได้หลากหลายรูปแบบด้วยสภาพแวดล้อมเชิงโต้ตอบคล้ายเครื่องคิดเลขสามารถตรวจสอบค่าต่างๆ เข้าได้อย่างรวดเร็ว ประยุกต์ใช้ในการออกแบบและแก้ปัญหาทางคณิตศาสตร์ได้ง่ายและมีประสิทธิภาพสูงมาก
- มีทูลบ็อกซ์หรือไลบรารีฟังก์ชันทางคณิตศาสตร์มากมาย สำหรับ พีชคณิต สถิติ การวิเคราะห์ฟูเรียร์ ฟังก์ชันลอจิก การประมวลผลภาพและวิดีโอ การคำนวณทางเศรษฐศาสตร์ และชีววิทยา เป็นต้น
- มี Simulink ที่เป็นซอฟต์แวร์สนับสนุนการสร้างแบบจำลองทางคณิตศาสตร์บน แมทแลป พร้อมด้วยบล็อกเชื่อมหลายสาขาวิชาเพื่อใช้ในการวิเคราะห์ระบบต่างๆการสร้างแอปพลิเคชันหรือสร้างอุปกรณ์สำเร็จรูป
- ประกอบด้วยฟังก์ชันสำหรับแสดงผลกราฟฟิกขั้นสูง ในการแสดงผลข้อมูลที่หลากหลายได้อย่างสวยงาม นอกจากนี้ยังสามารถปรับแต่งการพล็อตได้ง่าย

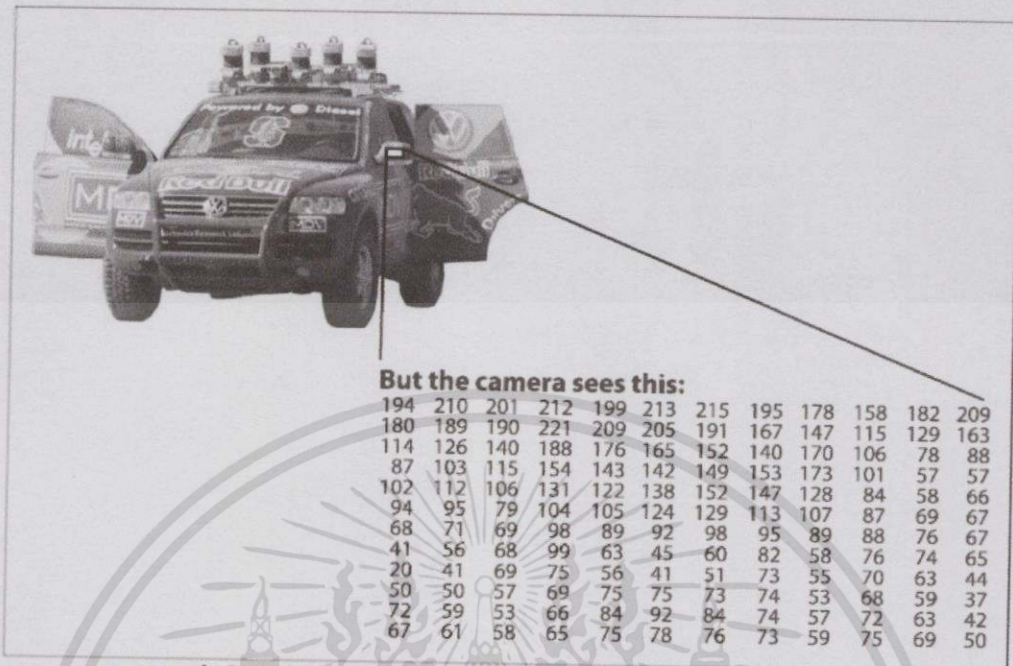
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- แมทแลป มีเครื่องมือช่วยปรับปรุงโค้ดต่างๆ ที่ต้องการจะสร้างเป็นแอปพลิเคชันหรืออุปกรณ์สำเร็จรูป
- จัดหาเครื่องมือสำหรับสร้างแอปพลิเคชันบน แมทแลป ด้วย GUI (Graphic User Interface) นั่นคือสามารถสร้างแอปพลิเคชันบนแมทแลปด้วยเครื่องมือคล้ายๆ กับ Visual Basic
- แมทแลปสามารถเชื่อมการทำงานร่วมกับ ภาษาซี จาวา คอทเน็ต เอ็กเซล หรือฮาร์ดแวร์ภายนอก

2.5 คอมพิวเตอร์วิชัน (Computer Vision)

คอมพิวเตอร์วิชัน[6] ก็คือการแปลงสัญญาณข้อมูลจากภาพนิ่งหรือจากกล้องวิดีโอ ไปยังอีกรูปแบบหนึ่งและเนื่องจากมนุษย์เป็นสิ่งมีชีวิตที่สามารถมองเห็นได้ จึงอาจจะเข้าใจผิดว่างานในรูปแบบของคอมพิวเตอร์วิชันนั้นเป็นเรื่องง่าย แต่จริงๆ แล้วนั้น การทำงานในแบบของคอมพิวเตอร์วิชันไม่ใช่เรื่องง่ายเลย เนื่องจากคอมพิวเตอร์นั้นไม่ใช่สิ่งมีชีวิต และไม่สามารถมองเห็นได้แบบที่มนุษย์มองเห็น สิ่งที่คอมพิวเตอร์มองเห็นนั้นย่อมนับเป็นข้อมูลดิจิทัล ซึ่งคอมพิวเตอร์จะรับกริด (Grid) ของตัวเลขหลายๆตัวจากกล้อง

ในรูปต่อไปนี้จะแสดงรูปของรถยนต์ ซึ่งในรูป สามารถมองเห็นกระจกสองหลังที่อยู่ทางฝั่งเดียวกับที่นั่งของคนขับของรถ แต่สิ่งที่คอมพิวเตอร์เห็นนั้นเป็นกริดของตัวเลขจำนวนมาก ตัวเลขที่อยู่ภายในกริดนั้นมีสัญญาณรบกวนอยู่เป็นจำนวนมาก และนี่ก็คือทั้งหมดที่คอมพิวเตอร์เห็น งานที่ต้องทำคือทำให้กริดที่มีสัญญาณรบกวนของตัวเลขเหล่านี้กลายเป็นกระจกมองหลังที่ไม่มีสัญญาณรบกวน



รูปที่ 2.14 สิ่งที่คอมพิวเตอร์มองเห็นคือกริดของตัวเลขจำนวนมาก

2.6 การตรวจจับใบหน้าโดยวิธีของวิโอล่าและโจนส์ (Face Detection by Viola-Jones method)

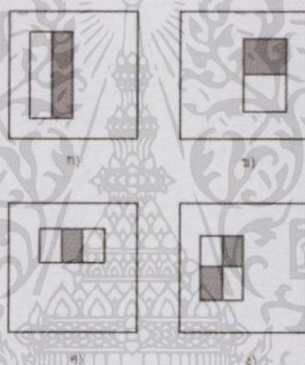
Paul viola และ Michael J. Jones ได้นำเสนอเทคนิคการตรวจจับใบหน้าที่มีความเร็วและมีความถูกต้องในการตรวจจับสูงในปี 2001 เทคนิคการตรวจจับใบหน้าของ Viola-Jones[7] เป็นเทคนิคที่ได้รับการยอมรับและรู้จักในงานวิจัยเรื่องการตรวจจับใบหน้าเป็นอย่างมาก โดยเทคนิคการตรวจจับใบหน้า Viola-Jones นี้สามารถแบ่งออกเป็น 3 ขั้นตอนคือการคำนวณรูปแบบการจำลองด้วย Integral Image การค้นหาแบบจำลองด้วย Adaboost และการรวมตัวจำแนกกลุ่มแบบต่อเรียง (Cascaded Classifier) โดยแต่ละขั้นตอนมีรายละเอียดดังต่อไปนี้

หลักการพื้นฐานของเทคนิคการตรวจจับใบหน้าของ Viola-Jones คือ การนำภาพที่ต้องการตรวจหาใบหน้ามาแบ่งเป็นภาพย่อย (Sub-window) จากนั้นภาพย่อยดังกล่าวจะถูกพิจารณาเป็นภาพอินพุตของกระบวนการตรวจหาใบหน้า เทคนิคทั่วไปในการตรวจหาใบหน้าจะทำการปรับขนาดของภาพอินพุตแตกต่างกันหลายขนาด และใช้ตัวตรวจจับ (Detector) ที่มีขนาดคงที่ค้นหาวัตถุข้อเสียของการตรวจจับใบหน้าแบบนี้คือ ระยะเวลาในการคำนวณไม่คงที่ ดังนั้น Viola-Jones จึงเสนอเทคนิคการตรวจจับใบหน้าใหม่ โดยใช้การจำลองรูปแบบ Haar-like เป็นตัวตรวจจับ ทำการปรับขนาดของตัวตรวจจับแทนการปรับขนาดภาพอินพุต และใช้ตัวตรวจจับทำการตรวจจับใบหน้าหลายๆรอบ โดยแต่ละรอบใช้ขนาดของตัวตรวจจับแตกต่างกัน ซึ่งเมื่อทำการเปรียบเทียบกับวิธีเดิม

พบว่า เวลาที่ใช้ในการคำนวณไม่ได้แตกต่างกันมากแต่ใช้เวลาในการคำนวณการตรวจจับภาพใบหน้า แต่ละรอบมีค่าคงที่ แม้ขนาดของตัวตรวจจับจะแตกต่างกันก็ตาม เทคนิคที่ Viola-Jones นำเสนอนี้ ทำการคำนวณการจำลองรูปแบบ Haar-like ด้วย Integral Image

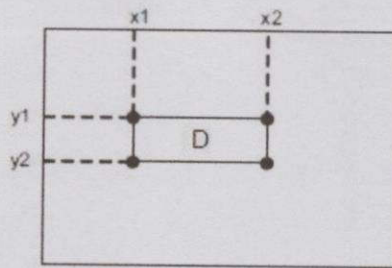
2.6.1 ขั้นตอน Integral Image

ตัวอย่างของการจำลองรูปแบบ Haar-like ที่ใช้ในเทคนิคการตรวจจับใบหน้าของ Viola-Jones แสดงในรูปที่ 2.15 โดยแต่ละลักษณะเด่นจะประกอบด้วยพื้นที่สี่เหลี่ยม 2 ประเภทคือ ส่วนที่ แรเงา และส่วนที่ไม่ได้แรเงา การหาค่าการจำลองรูปแบบ Haar-like คือการหาผลต่างระหว่างความเข้มในส่วนที่แรเงากับส่วนที่ไม่ได้แรเงา จากนั้นนำผลลัพธ์ที่ได้ไปเปรียบเทียบกับค่าขีดแบ่ง (Threshold) กับขั้ว (Polarity) ที่ใช้ในการตัดสินใจว่าภาพที่รับเข้ามาควรถูกจัดให้เป็นบวก(ภาพ ใบหน้า) หรือเป็นลบ (ไม่ใช่ภาพใบหน้า) ผลลัพธ์ที่ได้จะถูกนำไปพิจารณาในขั้นตอนต่อไป



รูปที่ 2.15 ตัวอย่างการจำลองรูปแบบ Haar-like

ตัวอย่างการจำลองรูปแบบ Haar-like ที่แสดงในรูปที่ 2.15 มี 4 แบบคือ การจำลองรูปแบบ Haar-like ที่ประกอบด้วยพื้นที่สี่เหลี่ยมสองภาพหรือ Two-rectangle feature (รูปที่ 2.8ก และ 2.8ข) การจำลองรูปแบบ Haar-like ที่ประกอบด้วยพื้นที่สี่เหลี่ยมสามภาพหรือ Three-rectangle feature (รูปที่ 2.8ค) และการจำลองรูปแบบ Haar-like ที่ประกอบด้วยพื้นที่สี่เหลี่ยมสี่ภาพหรือ Four-rectangle feature (รูปที่ 2.8ง)



รูปที่ 2.16 ตัวอย่างการคำนวณความเข้มในพื้นที่สี่เหลี่ยม D โดยวิธีการทั่วไป

ในรูปที่ 2.16 เป็นวิธีการทั่วไปในการหาค่าผลรวมความเข้มของพื้นที่สี่เหลี่ยมใด ๆ ในรูปภาพ สามารถเขียนแทนได้ด้วยสมการต่อไปนี้

$$i_s(x_1, x_2, y_1, y_2) = \sum_{x=x_1}^{x_2} \sum_{y=y_1}^{y_2} i(x, y) \quad (2.10)$$

โดยที่

$i(x, y)$ แทนค่าความเข้มที่จุด x และ y ในรูปภาพ
 $i_s(x_1, x_2, y_1, y_2)$ แทนผลรวมความเข้มภายในพื้นที่สี่เหลี่ยม
 x_1, x_2, y_1 และ y_2 เป็นพิกัดมุมทั้งสี่ของรูปสี่เหลี่ยม

จากสมการที่ 2.10 จะเห็นว่าเวลาที่ใช้ในการคำนวณนั้นจะขึ้นอยู่กับขนาดของรูปสี่เหลี่ยม ในการเพิ่มประสิทธิภาพ Viola และ Jones ได้นำเสนอให้ใช้เทคนิคที่เรียกว่า Integral Image ในการคำนวณการจำลองรูปแบบ Haar-like เทคนิค Integral image คือ การรวมความเข้ม (Intensity) ของแต่ละพิกเซลเข้าด้วยกัน



รูปที่ 2.17 ตัวอย่างภาพ Integral image

รูปที่ 2.17 ก) แสดง Integral image ของรูป 2.17 ข) การคำนวณหา Integral image ที่จุด (x, y) สามารถเขียนแทนได้ด้วยสมการต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

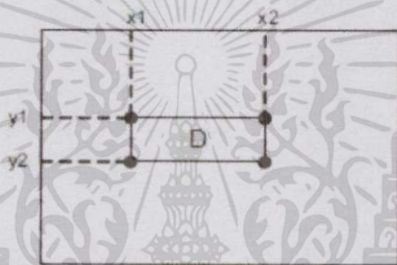
$$(ii(x, y)) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (2.11)$$

เมื่อ

$ii(x, y)$ คือ ค่าของ Integral image ที่ตำแหน่งจุดที่ตำแหน่งจุด (x, y)

$i(x', y')$ คือ ค่าความเข้มในแต่ละพิกเซลของภาพต้นฉบับ

ตัวอย่างการคำนวณค่าผลรวมความเข้มของพื้นที่สี่เหลี่ยม D โดยใช้ค่า Integral image จำนวน 4 จุด ผลลัพธ์ที่ได้คือ $4+1-(2+3)$ ดังแสดงในรูปที่ 4 ซึ่งจะเห็นได้ว่าไม่ว่าสี่เหลี่ยมจะมีขนาดเท่าใดก็จะใช้เวลาในการคำนวณค่าผลรวมความเข้มคงที่



รูปที่ 2.18 ตัวอย่างการคำนวณผลรวมความเข้มในพื้นที่สี่เหลี่ยม D โดยวิธี Integral image

กำหนดให้การจำลองรูปแบบ Haar-like ที่ได้หลังจากทำการเปรียบเทียบกับค่าขีดแบ่ง เรียกว่า ตัวจำแนกแบบอ่อนแอ (Weak Classifier) เนื่องจากการจำลองรูปแบบ Haar-like ที่ได้รับจะให้คำตอบที่ถูกต้องมากกว่าการเดาสุ่มเพียงเล็กน้อย ซึ่งการหาตัวจำแนกแบบอ่อนแอสามารถเขียนแทนด้วยสมการต่อไปนี้

$$h(x_i, f, p, \theta) = \begin{cases} 1 & \text{if } pf(x_i) \leq p\theta \\ 0 & \text{otherwise} \end{cases} \quad (2.12)$$

เมื่อ

x_i คือภาพย่อย

$h(x_i, f, p, \theta)$ คือตัวจำแนกแบบอ่อนแอที่พิจารณาจากการจำลองรูปแบบของ Haar-like

f เทียบกับค่าขีดแบ่ง

p, θ คือขีดซึ่งเป็นตัวกำหนดทิศทางของสมการ

2.6.2 ขั้นตอนการเรียนรู้ด้วยวิธี Adaboost

เทคนิคการตรวจจับใบหน้าของ Viola-Jones ในขั้นตอนนี้จะนำการจำลองรูปแบบ Haar-like ที่ได้จากขั้นตอนแรกมาเข้ากระบวนการเรียนรู้ของเครื่อง (Machine Learning) ที่เรียกว่า “Adaptive Boost” หรือ “Adaboost” ซึ่งกระบวนการนี้ใช้การเร่งหาตัวจำแนกแบบอ่อนแอ และ กำหนดค่าถ่วงน้ำหนักที่ทำให้ค่าผิดพลาดน้อยที่สุดในแต่ละรอบของกระบวนการ เพื่อสร้างตัวจำแนกแบบแข็งแรง (Strong Classifier)

ขั้นตอนการเรียนรู้ด้วย Adaboost มีดังต่อไปนี้

- 1) พิจารณาเลือกกลุ่มภาพตัวอย่างจำนวน N ภาพเพื่อใช้ในเรียนรู้ กลุ่มตัวอย่างภาพสามารถเขียนได้ดังสมการข้างล่าง

$$s = \{(x_i, z_i), i = 1, 2, \dots, N\} \quad (2.13)$$

เมื่อ

x_i คือ ภาพตัวอย่างที่ i

z_i คือค่าที่ระบุว่าภาพตัวอย่างเป็นภาพที่เป็นใบหน้าคน ($z_i = 1$) หรือไม่ใช่ใบหน้าคน ($z_i = 0$)

- 2) กำหนดจำนวนตัวจำแนกแบบอ่อนแอที่มีค่าเท่ากับ $t = 1, 2, \dots, T$ รอบ กำหนดค่าถ่วงน้ำหนักเริ่มต้น ($w_{t,i}$) ตัวจำแนกแบบอ่อนแอตัวแรก (เมื่อ $t=1$) มีค่าเท่ากับ $(w_{1,i}) = \frac{1}{2m} \cdot \frac{1}{2l}$ สำหรับ $z_i = 0.1$ เมื่อ m และ l คือจำนวนภาพที่เป็นใบหน้าและไม่ใช่ใบหน้าตามลำดับ

- 3) ขั้นตอนการพิจารณาตัวจำแนกแบบอ่อนแอทีละค่ามีดังต่อไปนี้

a. ทำการนอร์มัลไลซ์ (Normalize) ค่าถ่วงน้ำหนัก $w_{t,i} \approx \frac{w_{t,i}}{\sum_{j=1}^n w_{t,i}}$

b. ค้นหาตัวจำแนกแบบอ่อนแอที่ทำให้ค่าความผิดพลาด ϵ_t ต่ำที่สุดตามสมการ

ต่อไปนี้

$$\epsilon_t = \min_{f,p,\theta} \sum_i w_i |h(x_i, f, p, \theta) - z_i| \quad (2.14)$$

c. กำหนดให้ $h_t(x) = h(x, f_t, p_t, \theta_t)$

เมื่อ f_t, p_t และ θ_t คือค่าที่ทำให้ได้ค่าต่ำที่สุดของ ϵ_t

d. จากนั้นทำการปรับปรุงค่าถ่วงน้ำหนักของแต่ละตัวอย่างมีค่าเท่ากับ

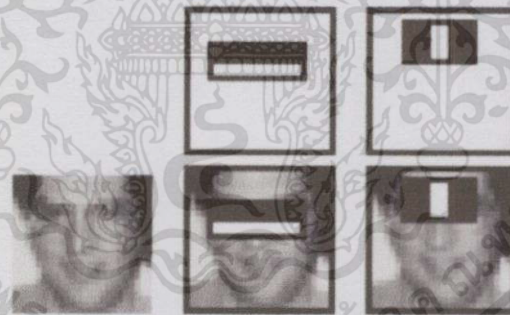
$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_t} \quad (2.15)$$

เมื่อ $e_t = 0$ ถ้าตัวอย่าง x_i ถูกเลือกอย่างถูกต้อง และ $e_t = 1$ ในกรณีอื่นๆ และค่า $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$ ผลลัพธ์สุดท้ายที่ได้คือ ตัวจำแนกแบบแข็งแกร่งค่าเท่ากับ

$$C(x) = \begin{cases} 1 & \sum_{t=1}^T a_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{otherwise} \end{cases} \quad (2.16)$$

$$\text{เมื่อ } \alpha_t = \log \frac{1}{\beta_t}$$

รูปที่ 2.19 แสดงตัวอย่างการจำลองรูปแบบ Haar-like สองลำดับแรกที่ได้จากกระบวนการเรียนรู้ ด้วยวิธี Adaboost จากรูปที่ 2.28 สังเกตเห็นว่าการจำลองรูปแบบ Haar-like แรกเป็นการจำลองรูปแบบ Haar-like ที่ได้จากความแตกต่างบริเวณดวงตาซึ่งจะมีค่าน้อยกว่าบริเวณโหนกแก้ม สำหรับการจำลองรูปแบบ Haar-like ตัวที่สองได้จากความแตกต่างบริเวณดวงตาทั้งสองข้างซึ่งจะมีค่าน้อยกว่าบริเวณสันจมูก

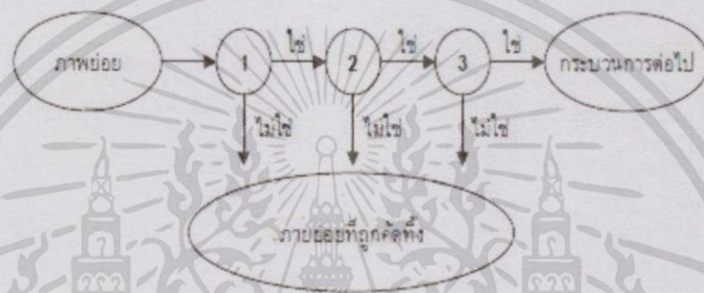


รูปที่ 2.19 แสดงตัวอย่างของตัวจำแนกแบบแข็งแกร่งที่ได้จากกระบวนการเรียนรู้ด้วย Adaboost

2.6.3 ขั้นตอนการรวมตัวจำแนกกลุ่มแบบต่อเรียง

ขั้นตอนที่ 3 ที่ Viola-Jones นำเสนอเป็นเทคนิคที่ใช้ในการเพิ่มประสิทธิภาพในการตรวจจับให้มีความถูกต้อง และยังใช้เวลาในการคำนวณลดลงด้วย โดยแนวความคิดของขั้นตอนนี้ คือ การสร้างการรวมตัวจำแนกกลุ่มแบบต่อเรียง (Cascaded Classifier) โดยเทคนิคนี้จะนำตัวจำแนก (Classifier) หลาย ๆ ตัวต่อกันเป็นลำดับ ดังแสดงในรูปที่ 2.13 โดยที่ตัวจำแนกในลำดับต้น ๆ มักจะ

มีความซับซ้อนที่น้อยกว่า อาจสร้างมาจากตัวจำแนกแบบอ่อนแอม่ก็ตัวเพื่อลดปริมาณการคำนวณ และลดระดับของค่าแบ่งขีด (Threshold) ลงเพื่อเพิ่มลดการความผิดพลาดในการตรวจจับไม่พบ ใบหน้า ซึ่งภาพย่อยที่ถูกจัดประเภทเป็นไม่ใช่ใบหน้า (Non-face) จะถูกทิ้งไปเป็นจำนวนมากในลำดับ ต้น ๆ ในทางตรงข้าม ถ้าภาพย่อยนั้นถูกจำแนกว่ามีโอกาสเป็นภาพใบหน้า จะถูกส่งต่อไปยังตัว จำแนกตัวถัดไปซึ่งมีความซับซ้อนสูงขึ้นตามลำดับ และกล่าวได้ว่ายังมีจำนวนชั้นของตัวจำแนกมาก เท่าใด โอกาสที่ภาพย่อยจะเป็นใบหน้าจะยิ่งมีมากขึ้น



รูปที่ 2.20 แสดงขั้นตอนการรวมตัวจำแนกกลุ่มแบบต่อเรียง

2.7 การตรวจหาริมฝีปาก

การตรวจหาริมฝีปาก[8]จะใช้สมการ Lip map โดยยึดตามหลักแนวคิดคือ บริเวณริมฝีปาก จะมีองค์ประกอบสีแดงมากที่สุด ในขณะที่เดียวกันก็จะมีองค์ประกอบสีน้ำเงินน้อยที่สุด เพราะฉะนั้น จึงสรุปได้ว่า องค์ประกอบสี C_r มีค่ามากกว่า C_b ในบริเวณริมฝีปาก โดยสมการต่อไปนี้จะสามารถ นำมาใช้ทำให้ริมฝีปากภายในบริเวณใบหน้าเด่นชัดขึ้นมาได้

$$Lip_{map} = (C_r)^2 \times \left((C_r)^2 - \frac{n \times C_r}{C_b} \right)^2 \quad (2.17)$$

$$n = 0.95 \frac{\frac{1}{n} \sum_{(x,y)} C_r(x,y)^2}{\frac{1}{n} \sum_{(x,y)} \left(\frac{C_r(x,y)}{C_b(x,y)} \right)} \quad (2.18)$$

Lip map = ค่าพิกเซลของรูปภาพใหม่

$C_r(x,y)$ = ค่าพิกเซล C_r ในระบบสี YCrCb

$C_b(x,y)$ = ค่าพิกเซล C_b ในระบบสี YCrCb

ซึ่งผลลัพธ์ที่ได้จะเป็นภาพระดับสีเทา ซึ่งจะมีพิกเซลที่สว่างแสดงบริเวณริมฝีปาก



รูปที่ 2.21 ตัวอย่างรูปภาพที่ผ่านการตรวจหาริมฝีปาก

2.8 Hough Circle Transform

Hough transform เป็นเทคนิคที่สามารถนำมาใช้เพื่อแยกคุณลักษณะของรูปร่างเฉพาะภายในภาพ เพราะต้องการคุณสมบัติในรูปแบบพารามตริก Hough transform ถูกใช้มากที่สุดใน การตรวจหาของเส้นโค้งปกติ เช่น เส้นตรง, วงกลม, วงรีและอื่นๆ โดยสมการในการหาเส้นตรง (Straight Line Hough Transform) คือ

$$x \cos \theta + y \sin \theta = r \quad (2.19)$$

และสมการที่ใช้ในการหาเส้นโค้ง (Circular Hough Transform) คือ

$$(x - c)^2 + (y - c)^2 = r^2 \quad (2.20)$$

เมื่อ r คือระยะจากจุดศูนย์กลางถึงเส้นตรง และ θ คือมุมระหว่าง r กับแนวแกน x เมื่อ กำหนดจุดของค่า r และ θ สำหรับทุกๆ ค่า x และ y ที่เป็นไปได้ แล้วจะได้เป็นภาพในรูปของ ฮัฟสเปซ (Hough Space) โดยในฮัฟสเปซจะสามารถระบุค่าที่เป็นค่าสูงสุด (Local Maxima) ที่มีอยู่ ได้ ซึ่งจุดนั้นจะเป็นจุดที่เส้นตรงอยู่นั่นเอง

บทที่ 3

การออกแบบ และพัฒนา

3.1 ภาพรวมของโปรแกรม



รูปที่ 3.1 ภาพรวมของโปรแกรม

โปรแกรมแจ้งเตือนการหลับในมีการทำงานแบ่งออกเป็นสามส่วนหลักคือ

ตรวจจับใบหน้า การตรวจจับใบหน้า ใช้ Haar-Like Feature ในการตรวจจับและตีกรอบบริเวณใบหน้าให้เป็นจุดสนใจของโปรแกรม เพื่อนำมาใช้ประมวลผลต่อไป

ตรวจหาอาการหลับในจากดวงตา การตรวจจับดวงตา ใช้ตัวตรวจจับ Haar-Like เช่นเดียวกับการตรวจจับใบหน้า แต่เปลี่ยนเป็นตรวจจับดวงตาแทน โดยบริเวณดวงตาจะเป็นจุดที่เราสนใจ และนำไปประมวลผลต่อไป

ตรวจหาอาการเหนี่ยยล้าจากการเคลื่อนไหวของคิริษะ การตรวจหาการเหนี่ยยล้าจากการเคลื่อนไหวของคิริษะ ใช้วิธีการตรวจหาตำแหน่งจุดศูนย์กลางของใบหน้า โดยเลือกใช้ตำแหน่งในแนวตั้ง นำมาประมวลผลและพิจารณาว่าผู้ขับขี่มีอาการเหนี่ยยล้าหรือไม่

ตรวจหาอาการหวา การตรวจหาอาการหวา เริ่มต้นโดยนำบริเวณส่วนล่างของใบหน้านำมาประมวลผล เพื่อหาริมฝีปาก และนำภาพริมฝีปากที่ได้ ไปประมวลผลต่อ ว่าผู้ขับขี่มีอาการหวาหรือไม่

3.2 การตรวจจับใบหน้า

3.2.1 ภาพรวมโฟลว์ชาร์ตแสดงขั้นตอนการตรวจจับใบหน้า



รูปที่ 3.2 โฟลว์ชาร์ตแสดงการตรวจจับใบหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1.1 คำอธิบายแต่ละแคว่ทิวติของโพล์วชาร์ตแสดงขั้นตอนการตรวจจับใบหน้า

เริ่มต้น : เริ่มต้นการจับใบหน้า

- รับภาพ 1 เฟรม : นำภาพจากกล้องวิดีโอหรือไฟล์วิดีโอเข้าสู่โปรแกรม 1 เฟรม โดยภาพที่ได้ จะมีขนาด 640 x 480 พิกเซล และใช้ระบบสี RGB
- ค้นหาใบหน้าโดยใช้ Harr-Like Feature : นำตัวตรวจจับ Harr – Like มาตรวจจับใบหน้าบนเฟรมภาพ เพื่อหาจุดสนใจของภาพ
- พบใบหน้าหรือไม่ : เมื่อตรวจจับใบหน้าจากรูปภาพเสร็จ หากพบใบหน้า จะทำการตัดรูปภาพส่วนที่เป็นใบหน้าก่อน แต่หากไม่ จะจบการทำงานของโปรแกรมในส่วนของการตรวจจับใบหน้าทันที
- ตัดส่วนรูปภาพที่เป็นใบหน้า : ตัดรูปภาพทั้งหมดให้เหลือเพียงใบหน้า ซึ่งเป็นจุดสนใจที่เราต้องการ
- ค้นหาดวงตาซ้ายและขวาโดยใช้ Haar – Like Feature : ใช้ตัวตรวจจับสำหรับตรวจหาดวงตา ตรวจหาดวงตาทั้งสองข้าง
- ตัดแบ่งภาพครึ่งล่างของครึ่งหน้า : นำภาพใบหน้ามาแบ่งครึ่งแล้วตัดแบ่งให้เหลือเพียงครึ่งหน้า
- จบการทำงาน : จบการทำงานของการทำงานตรวจหาใบหน้า

3.2.1.2 คำอธิบายโดยรวมของโพล์วชาร์ตแสดงขั้นตอนการตรวจจับใบหน้า

เมื่อเริ่มใช้งาน โปรแกรมจะรับภาพจากกล้องหรือไฟล์วิดีโอมา 1 เฟรม จากนั้นจะทำเฟรมภาพที่ได้ มาตรวจจับหาใบหน้า หากตรวจไม่พบจะจบการทำงานทันที แต่เมื่อตรวจพบใบหน้า จะนำใบหน้ามาประมวลผลเพื่อเตรียมรูปภาพมาประมวลผลต่อ โดยจะตรวจจับดวงตาซ้ายและขวา และตัดภาพครึ่งล่างของใบหน้า เพื่อนำภาพทั้งสองไปประมวลผลต่อไป

3.3 การตรวจหาอาการหลักจากดวงตา

3.3.1 ภาพรวมการตรวจหาอาการหลักจากดวงตา



รูปที่ 3.3 โฟลวชาร์ตภาพรวมการตรวจหาอาการหลักจากดวงตา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

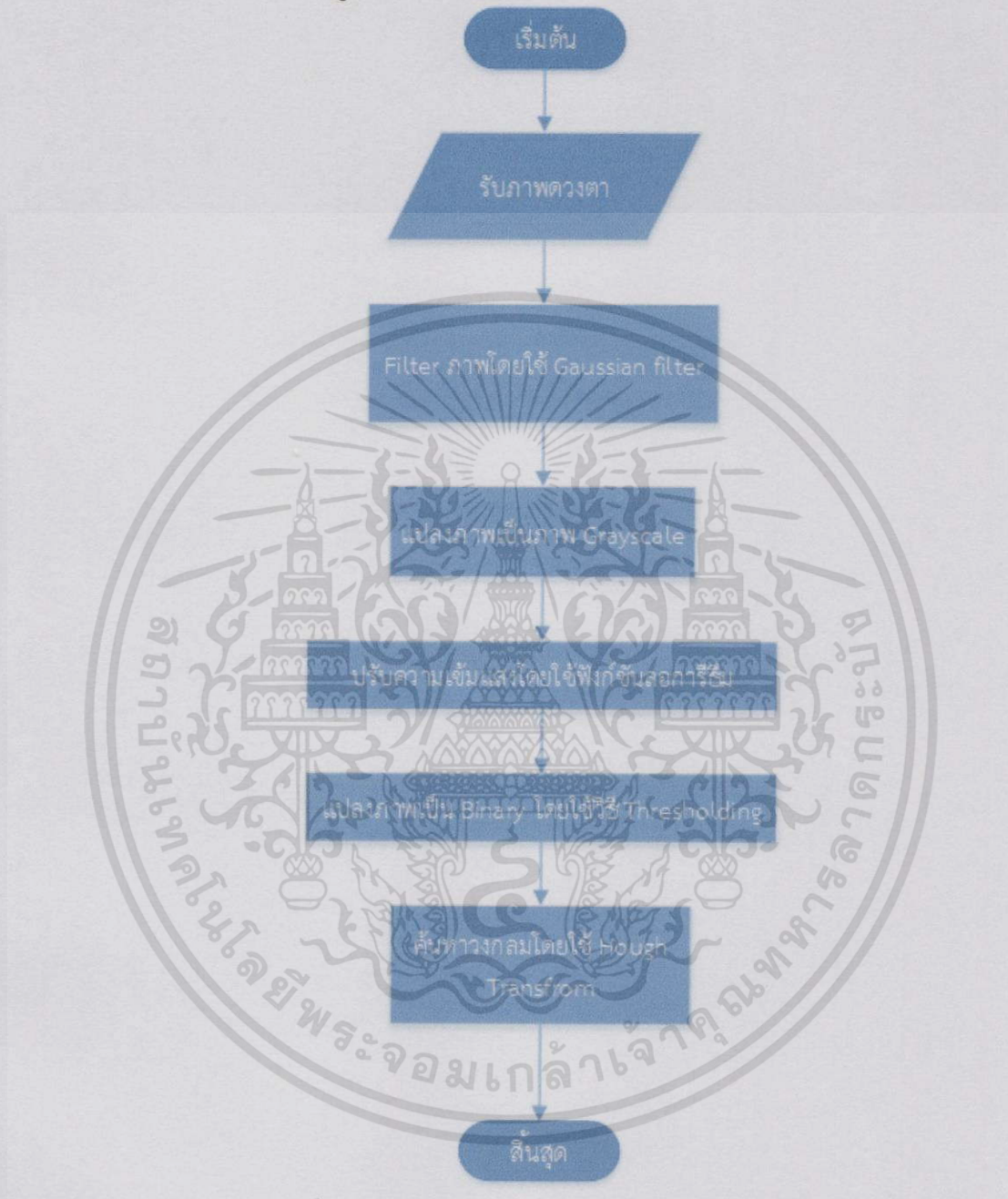
คำอธิบายแต่ละแอคทิวิตีของโฟลว์ชาร์ตแสดงภาพรวมการหาอาการหลักจากดวงตา

- เริ่มต้น : เริ่มต้นการทำงานเมื่อได้รับภาพดวงตาจากการตรวจจับดวงตา
- รับภาพดวงตา : รับภาพดวงตาที่ได้รับจากการตรวจจับดวงตา
- หาลูกตาดำ : นำภาพลูกตาดำที่ได้มาหาลูกตาดำทั้งสองข้าง
- พบลูกตาดำทั้งสองข้างหรือไม่ ? : นำลูกตาดำที่หาได้ทั้งสองข้างซึ่งแต่ละข้างเมื่อหาพบจะได้สถานะเป็น 1 หากไม่พบให้สถานะเป็น 0
- สถานะลูกตาดำ = 0 , สถานะลูกตาดำ = 1 : เมื่อหาลูกตาดำสำเร็จ ให้มีสถานะตามกันพบลูกตาดำทั้งสองข้างคือ ไม่พบเป็น 0 พบเป็น 1
- แสดงสถานะการแจ้งเตือน : นำเอาสถานะลูกตาดำมาประมวลเทียบกับสถานะที่ได้กับเฟรมก่อนหน้าว่าจะแจ้งเตือนหรือไม่
- สิ้นสุด : สิ้นสุดการทำงานในส่วนนี้

คำอธิบายภาพรวมของการตรวจหาอาการหลักจากดวงตา

เมื่อรับภาพดวงตาจากการค้นหาใบหน้า ระบบจะค้นหาลูกตาดำแต่ละข้างจากภาพดวงตาที่ได้รับ หากพบดวงตาทั้งสองข้าง จะมีสถานะเป็น 1 หากไม่พบข้างใดข้างหนึ่งหรือทั้งสองข้างจะให้สถานะเป็น 0

3.3.2 ภาพรวมของการหาลูกตาดำ



รูปที่ 3.4 โฟลวชาร์ตแสดงภาพรวมของการหาลูกตาดำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำอธิบายแต่ละแอคทิวิตีของโฟลว์ชาร์ตแสดงการหาลูกตาดำ

- เริ่มต้น : เริ่มต้นการทำงาน
- รับภาพดวงตา : รับภาพดวงตาที่ได้รับมาจากการตรวจหาดวงตาโดยใช้ Haar Like Feature
- Filter ภาพโดยใช้ Gaussian filter: นำภาพที่ได้มา Filter โดยใช้ตัวกรอง Gaussian เพื่อให้ภาพเบลอลงซึ่งจะลบรายละเอียดเล็กน้อยที่เราไม่ต้องการออกไป เช่นขอบตา เป็นต้น
- แปลงภาพเป็น Grayscale : ภาพที่ได้จากการตรวจหาดวงตานั้นเป็นภาพสีแบบ RGB และจะนำมาแปลงภาพเป็น Grayscale เพื่อนำไปประมวลผลต่อ
- ปรับความเข้มแสงโดยใช้ฟังก์ชันลอการิทึม : นำภาพที่ผ่านการกรอง มาปรับความเข้มแสงโดยใช้ฟังก์ชันลอการิทึม
- ซึ่งจะช่วยขยายรายละเอียดในที่มืด เมื่อภาพที่ได้มีแสงค่อนข้างน้อย และลบรายละเอียดในที่สว่าง เช่นส่วนของผิวหนังซึ่งเราไม่ต้องการออกไป
- แปลงภาพเป็น Binary โดยใช้วิธี Thresholding : นำภาพที่ได้ มาแปลงเป็นภาพ Binary โดยใช้วิธีหาค่าขีดแบ่ง (Thresholding) เพื่อนำภาพไปประมวลผลต่อ
- ค้นหาวงกลมโดยใช้ Hough Transform : นำภาพ Binary ที่ได้จากกระบวนการข้างต้น มาหาวงกลมซึ่งจะเป็นลูกตาดำที่ต้องการไปประมวลผลต่อไป
- สิ้นสุด : จบการทำงานในส่วนของการหาลูกตาดำ

คำอธิบายโดยรวมของโฟลว์ชาร์ตแสดงการหาลูกตาดำ

เมื่อได้รับภาพดวงตา ก็จะทำการแปลงภาพดวงตาที่ได้ให้เป็นภาพ Grayscale และทำให้ภาพเบลอลงเพื่อลบรายละเอียดบางส่วนของภาพที่เราไม่ต้องการออกไป โดยใช้ Gaussian filter และใช้ฟังก์ชันลอการิทึม เพื่อลบรายละเอียดในที่สว่างและขยายรายละเอียดในที่มืด ต่อจากนั้นจึงแปลงภาพให้เป็นภาพ Binary เพื่อนำมาหาวงกลมในภาพโดยใช้ Hough Transform

3.3.3 สถานะในการแจ้งเตือนการหลัก

แผนผังสถานะแสดงการแจ้งเตือนการหลัก

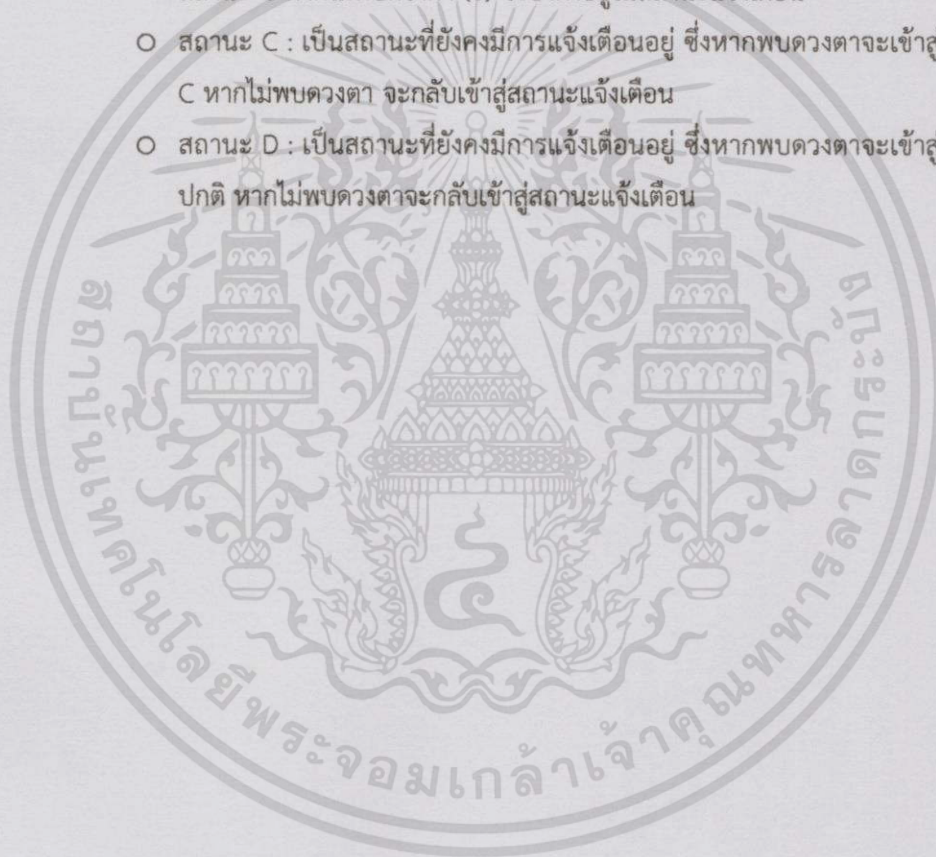


รูปที่ 3.5 แผนผังสถานะแสดงการแจ้งเตือนการหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

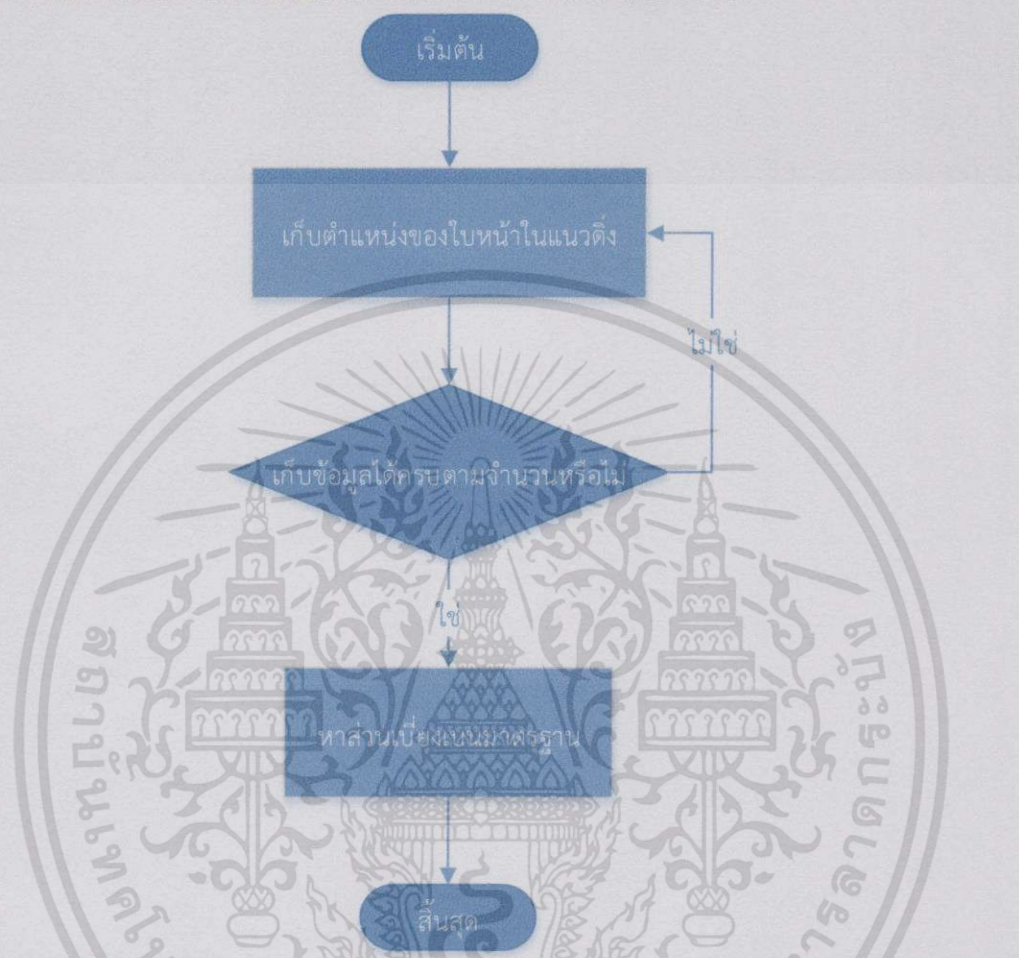
คำอธิบายแผนผังสถานะแจ้งเตือนการหลับ

- สถานะปกติ : หากพบดวงตา (1) ก็ยังคงอยู่ในสถานะปกติ แต่หากไม่พบดวงตา (0) จะเข้าสู่สถานะ A
- สถานะ A : หากพบดวงตา (1) ก็จะกลับเข้าสู่สถานะปกติ แต่หากไม่พบดวงตา (0) จะเข้าสู่สถานะ B
- สถานะ B : หากพบดวงตา (1) ก็จะกลับเข้าสู่สถานะ A แต่หากไม่พบดวงตา (0) จะเข้าสู่สถานะแจ้งเตือน
- สถานะแจ้งเตือน : เป็นสถานะที่จะมีการแจ้งเตือนผู้ใช้ หากพบดวงตา (1) จะเข้าสู่สถานะ C หากไม่พบดวงตา (0) จะยังคงอยู่ในสถานะแจ้งเตือน
- สถานะ C : เป็นสถานะที่ยังคงมีการแจ้งเตือนอยู่ ซึ่งหากพบดวงตาจะเข้าสู่สถานะ C หากไม่พบดวงตา จะกลับเข้าสู่สถานะแจ้งเตือน
- สถานะ D : เป็นสถานะที่ยังคงมีการแจ้งเตือนอยู่ ซึ่งหากพบดวงตาจะเข้าสู่สถานะปกติ หากไม่พบดวงตาจะกลับเข้าสู่สถานะแจ้งเตือน



3.4 การตรวจหาอาการเหนื่อยล้าจากการเคลื่อนไหวของศีรษะ

3.4.1 ภาพรวมการตรวจหาอาการเหนื่อยล้าจากการเคลื่อนไหวของศีรษะ



รูปที่ 3.6 โฟลวชาร์ตแสดงการตรวจหาอาการเหนื่อยล้าจากการเคลื่อนไหวของศีรษะ

คำอธิบายแต่ละแอกทิวิตีของโฟลว์ชาร์ตแสดงขั้นตอนหาอาการเหนื่อยล้าจากการเคลื่อนไหวของศีรษะ

- เริ่มต้น : เริ่มต้นการตรวจหาความเหนื่อยล้า
- เก็บตำแหน่งของใบหน้าในแนวตั้ง : เก็บตำแหน่งตรงกลางของใบหน้าในแนวแกนตั้งนำมาเก็บไว้ในอาร์เรย์ขนาดตามที่กำหนดไว้
- เก็บข้อมูลได้ครบตามที่กำหนดหรือไม่ : ตรวจสอบข้อมูลในอาร์เรย์ว่ามีตามที่กำหนดหรือไม่ หากไม่ครบให้เก็บตำแหน่งจากเฟรมถัดไปต่อ แต่หากครบแล้วให้คำนวณหาส่วนเบี่ยงเบนมาตรฐาน

- ทาสวนเบียงเบนมาตรฐาน : นำค่าตำแหน่งที่เก็บได้ทุกค่ามาทาสวนเบียงเบนมาตรฐานเพื่อนำไปประมวลผลต่อไป

คำอธิบายโดยรวมของโพล์ชาร์ตแสดงขั้นตอนหาอาการเหน้อยล้าจากการเคลื่อนไหวของศีรษะ

เมื่อโปรแกรมได้ตำแหน่งกึ่งกลางใบหน้าของผู้ใช้ จะนำตำแหน่งมาเก็บไว้ในอาร์เรย์หนึ่งเพื่อรอการประมวลผลเมื่อเก็บข้อมูลได้จำนวนตามที่ต้องการ เช่น เก็บเป็นจำนวน 1 นาที ถ้าเก็บภาพวินาทีละ 30 เฟรม จะได้ข้อมูลทั้งหมด 1800 หน่วย และนำข้อมูลทั้งหมดนั้น มาหาค่าส่วนเบียงเบนมาตรฐาน

3.4.2 ตารางแสดงช่วงค่าส่วนเบียงเบนมาตรฐานกับความอ่อนล้า

ตารางที่ 3.1 ตารางแสดงช่วงค่าส่วนเบียงเบนมาตรฐานกับความอ่อนล้า

ส่วนเบียงเบนมาตรฐาน	ความอ่อนล้า
0 - 2	ปกติ
2 - 3.5	เล็กน้อย
4-4.8	ปานกลาง
4.8 เป็นต้นไป	มาก

เมื่อโปรแกรมทาสวนเบียงเบนมาตรฐานเรียบร้อยแล้ว จะแสดงสถานะความอ่อนล้าให้ผู้ใช้ได้ทราบ เพื่อให้ผู้ใช้ตัดสินใจว่าควรจะพักผ่อนหรือไม่

3.5 ตรวจสอบอาการหา

การตรวจสอบอาการหาแบ่งออกเป็น 2 ส่วนคือ การหาริมฝีปากของผู้ใช้ และนำริมฝีปากของผู้ใช้มาประมวลผลเพื่อหาอาการหา

3.5.1 การหาริมฝีปาก



รูปที่ 3.7 โฟลวชาร์ตแสดงการหาริมฝีปาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

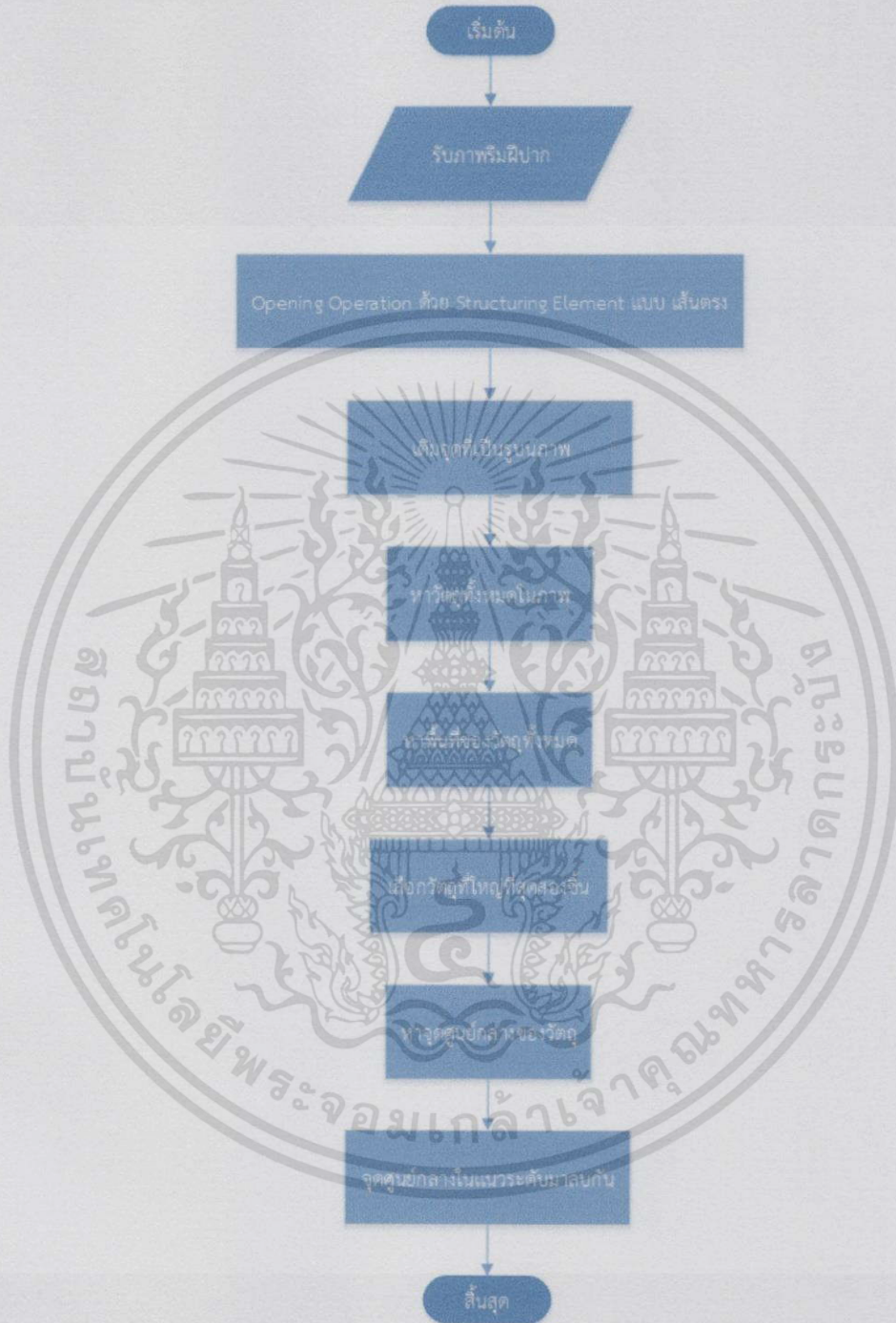
คำอธิบายแต่ละแคว่ทวิติของโพล์ชาร์ตแสดงชั้นตอนหาริมฝีปาก

- เริ่มต้น : เริ่มต้นหาริมฝีปาก
- รับภาพครึ่งล่างของใบหน้า : นำภาพใบหน้าที่มีนการแบ่งครึ่งส่วนล่างมาแล้วมาใช้งาน
- แปลงภาพจาก RGB เป็น YCbCr : นำภาพใบหน้าซึ่งเป็นภาพแบบ RGB มาแปลงระบบสีเป็น YCbCr
- นำค่า Pixel แต่ละ Pixel มาคำนวณในสมการ Lip_map : นำค่าพิกเซลของรูปภาพครึ่งล่างใบหน้าทั้งหมดมาใช้สมการ Lip_map โดนผลลัพธ์ที่ได้บริเวณที่เป็นริมฝีปากจะมีสีขาว และบริเวณอื่นๆจะเป็นสีดำ
- แปลงภาพเป็น Binary โดยวิธีการ Thresholding : นำภาพริมฝีปากที่ได้มาแปลงเป็นภาพ Binary เพื่อนำไปประมวลผลต่อ
- สิ้นสุด : สิ้นสุดการหาริมฝีปาก

คำอธิบายโดยรวมของโพล์ชาร์ตแสดงชั้นตอนหาริมฝีปาก

เมื่อเริ่มทำงาน จะนำภาพครึ่งล่างของใบหน้าแปลงเป็นระบบสี YCrCb ในแต่ละ pixel มาเข้าสู่สมการ Lip map เพื่อบริเวณที่เป็นริมฝีปาก เมื่อคำนวณเสร็จ ก็จะนำภาพที่ได้จากสมการ Lip map มาแปลงเป็นภาพ Binary เพื่อประมวลผลต่อไป

3.5.2 การนำริมฝีปากมาประมวลผลเพื่อหาการหา



รูปที่ 3.8 โฟลวชาร์ตแสดงขั้นตอนการนำริมฝีปากมาประมวลผลเพื่อหาการหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำอธิบายแต่ละแคว่ทวิติของการนำริมฝีปากมาประมวลผลเพื่อหาการหา

- เริ่มต้น : เริ่มต้นการประมวลผล
- รับภาพริมฝีปาก : รับภาพริมฝีปากที่ผ่านการประมวลผลและแปลงเป็นภาพแบบ Binary เรียบร้อยแล้ว
- Opening Operation ด้วย Structuring Element แบบ Line : นำภาพริมฝีปากที่ได้มากเข้ากระบวนการ Opening เพื่อกำจัดสิ่งรบกวนที่เกิดขึ้นในภาพ
- เติมจุดที่เป็นรูบนรูปภาพ : นำภาพที่ผ่านการ Opening มาแล้ว มาเติมบริเวณที่เป็นหลุมบนรูปภาพ เพื่อให้รูปภาพที่ได้เป็นวัตถุแบบเต็มโดยไม่มีจุดมารบกวนภายในวัตถุ
- หาวัตถุทั้งหมดในภาพ : ตรวจหาวัตถุทั้งหมดในรูปภาพ
- หาพื้นที่ของวัตถุทั้งหมด : นำวัตถุในรูปภาพที่ได้ มาหาพื้นที่ของวัตถุ
- เลือกวัตถุที่ใหญ่ที่สุดสองชิ้น : เลือกพื้นที่ของวัตถุที่ใหญ่ที่สุดอันดับสอง จากนั้นลบวัตถุที่มีขนาดเล็กกว่าพื้นที่นี้ออกจากรูปภาพ
- หาจุดศูนย์กลางของวัตถุ : นำวัตถุที่ยังเหลืออยู่ในรูปภาพทั้ง 2 ชิ้นมาหาจุดศูนย์กลางของวัตถุนั้น โดยจะได้พิกัดในแนวระนาบและแนวตั้ง
- จุดศูนย์กลางในแนวระดับมาลบกััน : นำค่าในแนวระดับของจุดศูนย์กลางของวัตถุมาลบกัันเพื่อผลต่าง จากนั้นคำนวณผลต่างเทียบกับความยาวของภาพริมฝีปากทั้งภาพเทียบเป็นร้อยละ
- สิ้นสุด : สิ้นสุดการหาการหา

คำอธิบายโดยรวมของการนำริมฝีปากมาประมวลผลการหา

ภาพริมฝีปากที่ถูกแปลงเป็น Binary เรียบร้อยแล้วจะถูกนำมาประมวลผลต่อโดย ทำ Open Operation จากนั้นจึงเติมหลุมในรูปภาพให้เต็ม จะได้รูปภาพวัตถุหลายๆวัตถุในรูปภาพ จากนั้นหาวัตถุที่ใหญ่เป็นอันดับสองเพื่อลบวัตถุที่เล็กกว่าพื้นที่อันดับสองออกจากภาพ จะได้ภาพริมฝีปากบนและล่าง ต่อมานำจุดศูนย์กลางของภาพทั้งสองในแนวระดับมาลบกัันเพื่อหาผลต่างซึ่งจะนำไปเทียบกับขนาดความสูงของภาพริมฝีปากโดยเทียบเป็นร้อยละเพื่อนำไปหาการหา

3.5.3 การตรวจหาการทาวจากหลายๆเฟรม

เมื่อได้ภาพริมฝีปากและรอยละความแตกต่างของริมฝีปากบนและล่างเทียบกับความสูงของรูป ค่าที่ได้จะถูกนำมาประมวลผลโดยเริ่มจากภาพที่มี ร้อยละมากกว่า 20 จะถือว่ามียปากที่อ้ากว้างกว่าปกติทั่วไป และนำมาตรวจสอบหลายๆเฟรมโดยตรวจสอบจากเฟรมที่มีร้อยละมากกว่า 20 ติดต่อกันเป็นจำนวนหนึ่ง จะถือว่าผู้ใช้มีอาการทาว และเก็บเป็นข้อมูลในการแจ้งเตือนอาการเหนื่อยล้าต่อไป



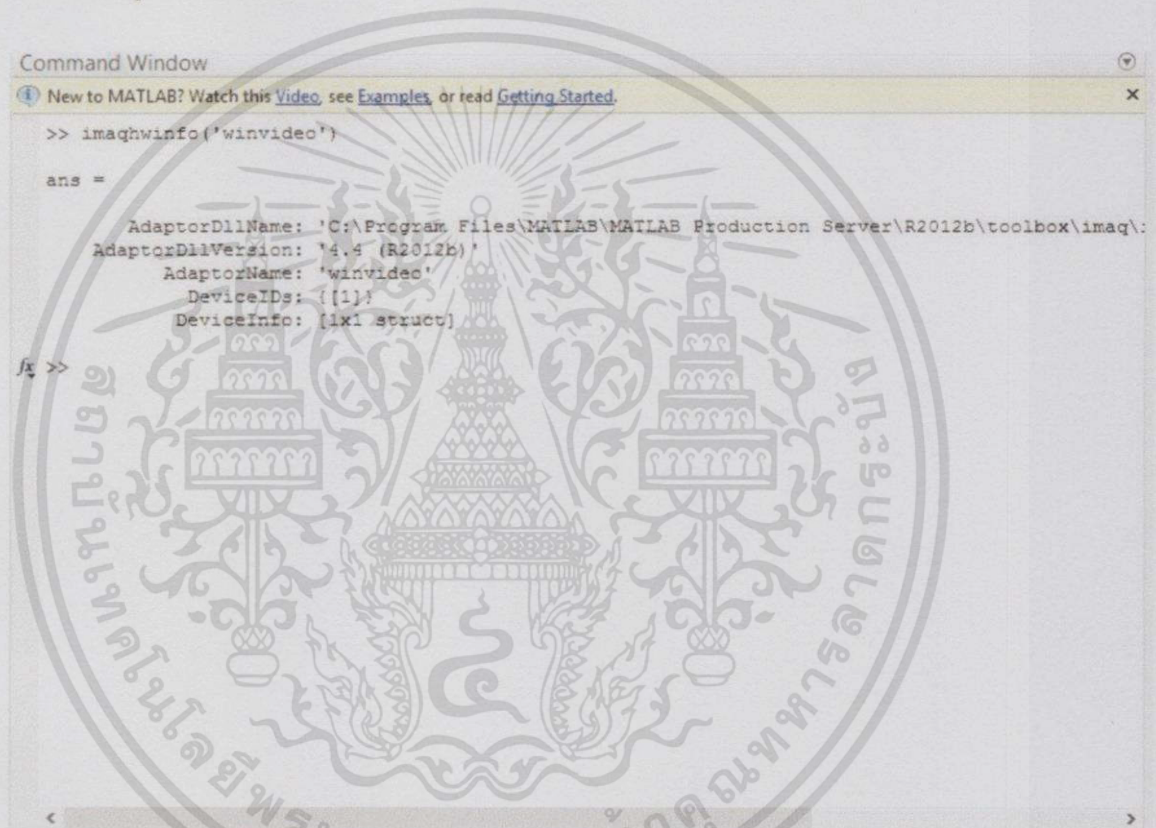
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 ทดลองเปิดใช้งานกล้องเว็บแคมบนแมทแลป

- 1) เข้าสู่โปรแกรมแมทแลป หลังจากนั้นใช้คำสั่ง `imagwhinfo('winvideo')`; เพื่อตรวจสอบข้อมูลเกี่ยวกับวิดีโอ



```
Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.
>> imagwhinfo('winvideo')
ans =
    AdaptorDllName: 'C:\Program Files\MATLAB\MATLAB Production Server\R2012b\toolbox\imaq\
    AdaptorDllVersion: '4.4 (R2012b)'
    AdaptorName: 'winvideo'
    DeviceIDs: {[1]}
    DeviceInfo: [1x1 struct]
```

รูปที่ 4.1 แสดงผลลัพธ์เมื่อใช้คำสั่ง `imagwhinfo('winvideo')`

จากรูปจะเห็นข้อมูลเกี่ยวกับวิดีโอของเครื่องที่ใช้อยู่ โดยมี DeviceIDs: {[1]} หมายถึง ณ ขณะปัจจุบันมีกล้องเว็บแคมเชื่อมต่ออยู่กับคอมพิวเตอร์จำนวน 1 ตัว

- 2) หลังจากนั้น จะทำการตรวจสอบรายละเอียดของกล้องเว็บแคมโดยใช้คำสั่ง

```
>> dev = imaqhwinfo('winvideo',1);
>> dev

dev =

    DefaultFormat: 'YUY2_160x120'
    DeviceFileSupported: 0
    DeviceName: 'ASUS USB2.0 WebCam'
    DeviceID: 1
    VideoInputConstructor: 'videoinput('winvideo', 1)'
    VideoDeviceConstructor: 'imaq.VideoDevice('winvideo', 1)'
    SupportedFormats: {1x5 cell}

fx >> |
```

รูปที่ 4.2 แสดงผลลัพธ์เมื่อใช้คำสั่ง imaqhwinfo('winvideo',1)

จากรูปเราใช้ตัวแปร dev ในการเก็บค่าข้อมูลรายละเอียดของกล้องเว็บแคมที่มี ID = 1 และแสดงค่าออกมาโดยมีรายละเอียดของกล้องที่น่าสนใจคือ โดยปกติกล้องใช้ระบบสี YUV ที่ความละเอียด 160 x 120 มี ID = 1

- 3) ต่อมา จะทำการตรวจสอบว่ากล้องเว็บแคมรองรับระบบสีใดที่ความละเอียดเท่าใดบ้าง โดยใช้คำสั่ง dev.SupportedFormats

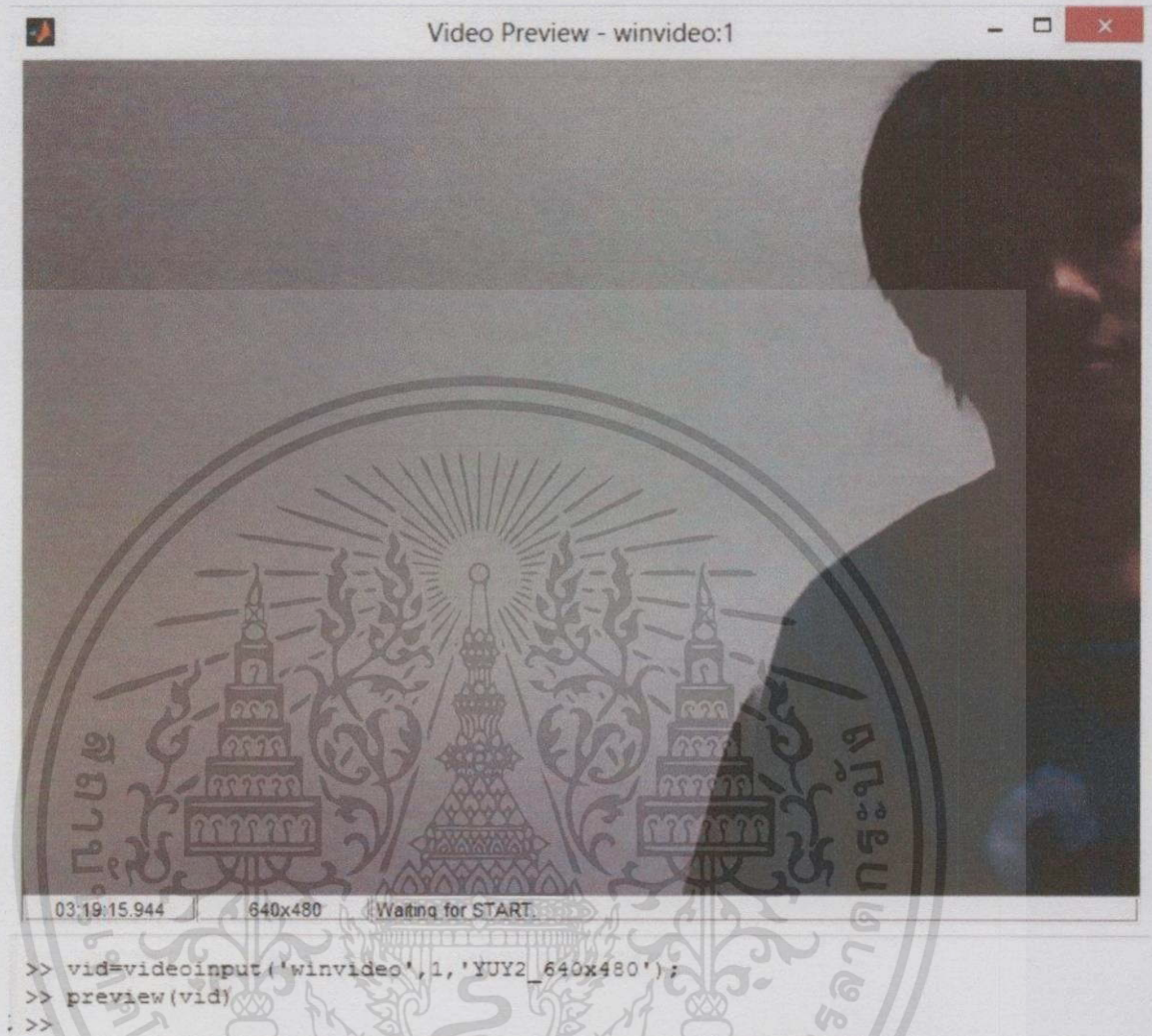
```
>> dev.SupportedFormats
ans =

    'YUY2_160x120'    'YUY2_176x144'    'YUY2_320x240'    'YUY2_352x288'    'YUY2_640x480'
```

รูปที่ 4.3 แสดงผลลัพธ์เมื่อใช้คำสั่ง dev.SupportedFormats

จากรูปจะเห็นว่า กล้องเว็บแคมตัวนี้รองรับการใช้งานบนระบบสี YUV ที่ความละเอียดที่แตกต่างกันเป็นจำนวน 5 แบบ คือ 160x120 , 176x144 , 320x240 , 352x288 และ 640x480

- 4) หลังจากทราบข้อมูลเบื้องต้นเกี่ยวกับกล้องแล้ว เราจะสร้างตัวแปรที่เก็บข้อมูลรายละเอียดของกล้องเพื่อนำไปใช้งานต่อไป โดยใช้คำสั่ง vid = videoinput('winvideo', 1, 'YUY2_640x480'); และหลังจากนั้นจะทดลองให้แมทแลปแสดงภาพจากกล้องเว็บแคมโดยใช้คำสั่ง preview(vid)

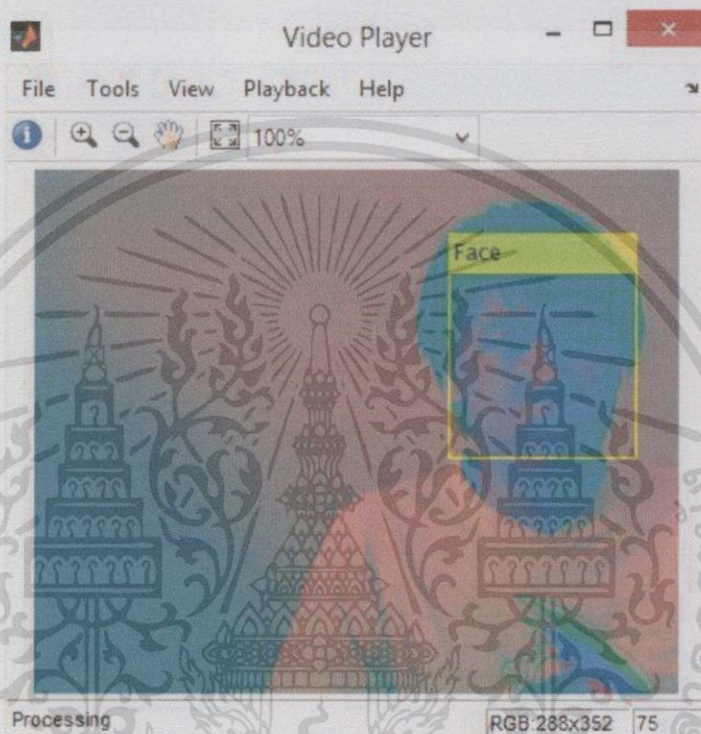


รูปที่ 4.4 แสดงผลลัพธ์เมื่อให้แมทแลปแสดงภาพจากกล้องเว็บแคม

จากภาพจะเห็นว่า แมทแลปสามารถแสดงผลภาพจากกล้องเว็บแคมได้ตามที่เราต้องการ ซึ่งเมื่อเราต้องการใช้งานในคราวต่อไป ก็สามารถนำคำสั่งสร้างตัวแปรที่เก็บรายละเอียดของกล้องเว็บแคมไปใช้งานต่อได้ เพื่อให้โปรแกรมของเราใช้งานกล้องเว็บแคมด้วยบนความละเอียดและระบบสีที่ต้องการ

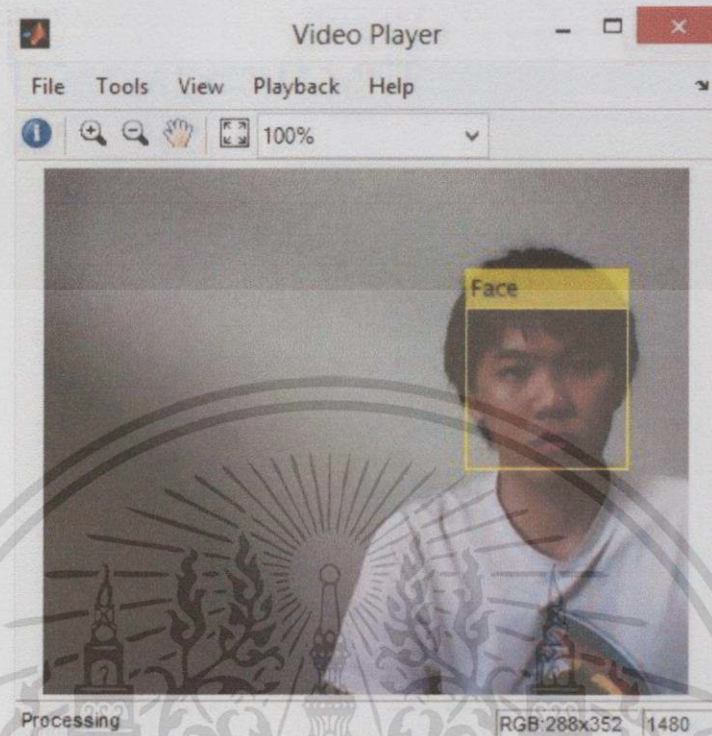
4.2 ทดลองตรวจจับภาพใบหน้าโดยใช้กล้องเว็บแคมผ่านโปรแกรมแมทแลป

ตรวจจับภาพใบหน้าใช้วิธีการของวิโอล่า-โจนส์ (Violar-Jones detection algorithm) ซึ่งได้ทดลองตรวจจับภาพใบหน้า (Face Detection) และติดตามใบหน้า (Face Tracking) โดยทำที่ความละเอียด 352x288 pixel บนระบบสี YCbCr



รูปที่ 4.5 แสดงผลการทดลองจับภาพใบหน้าบนความละเอียด 355 x288 ในระบบสี YCbCr

เนื่องจากระบบสีทั่วไปของ Video Player รวมถึงฟังก์ชันต่างๆในแมทแลปมักจะรองรับระบบสี RGB จึงต้องแปลงเฟรมของภาพวิดีโอที่รับมาจากเว็บแคมให้เป็น RGB จะได้ผลลัพธ์ดังภาพ

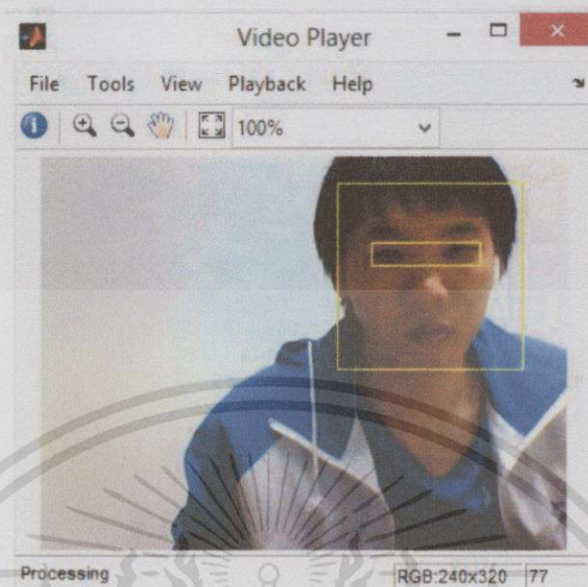


รูปที่ 4.6 แสดงผลการทดลองจับภาพใบหน้าบนความละเอียด 355 x288 พิกเซล ในระบบสี RGB

ทดลองตรวจจับภาพใบหน้าโดยใช้วิธีของวิโอล่า-โจนส์เป็นเวลา 1 นาที ที่ความละเอียดต่างๆ ที่กล้องเว็บแคมรองรับ เพื่อเป็นข้อมูลประกอบการตัดสินใจในการเลือกความละเอียดที่เหมาะสม

4.3 ทดลองตรวจจับใบหน้าและดวงตาพร้อมกันโดยกล้องเว็บแคมผ่านโปรแกรมแมทแลป

เมื่อสามารถตรวจจับใบหน้าได้ ดวงตาคือสิ่งต่อมาที่โปรแกรมสนใจ ซึ่งการตรวจจับดวงตา ใช้วิธีการเดียวกันกับการตรวจจับใบหน้าคือวิธีการของวิโอล่า-โจนส์ (Violar-Jones detection algorithm)



รูปที่ 4.7 แสดงผลการทดลองจับภาพใบหน้าและดวงตาบนความละเอียด 320x 240 พิกเซล

ในการทดลอง ได้ทำการวัดเฟรมเรตของของวิดีโอที่แสดงผลต่อเนื่อง โดยใช้การวัดเป็นระยะเวลา 60 วินาที และคำนวณเป็นเฟรมเรตออกมา พบว่า บนความละเอียด 320 x 240 พิกเซล พบว่าสามารถทำเฟรมเรตได้ที่ 9.23 เฟรมต่อวินาที

4.3 การทดลองหาลูกตาดำและอาการหลับ

4.3.1 การทดลองตรวจจับดวงตาโดยใช้ Haar Like Feature เมื่อผู้ใช้ลืมตาอยู่

เมื่อสามารถตรวจจับใบหน้าได้ ขั้นตอนต่อมาของโปรแกรมคือตรวจจับดวงตาของผู้ใช้งาน เพื่อนำไปประมวลผลต่อ จากขั้นตอนแรก เราจะรับภาพเฉพาะส่วนของใบหน้าของผู้ใช้ก่อน ดังรูป



รูปที่ 4.8 ภาพใบหน้าที่ได้จากการตรวจจับใบหน้า

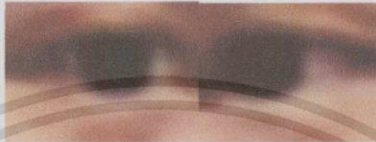
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อมา เมื่อได้รับภาพใบหน้า จะทำการหาดวงตา และตัดแบ่งภาพของดวงตา ดังรูป



รูปที่ 4.9 ภาพดวงตาทั้งสองข้างที่ได้จากการตรวจจับดวงตา

ต่อมา นำภาพดวงตาที่ได้ผ่าน Gaussian Filter เพื่อทำให้ภาพเบลอ ดังรูป



รูปที่ 4.10 ภาพดวงตาทั้งสองข้างเมื่อผ่าน Gaussian Filter

ขั้นตอนถัดไป นำภาพดวงตาที่ได้จากการกรอง มาแปลงเป็นระบบสี Grayscale จะได้ดังรูป



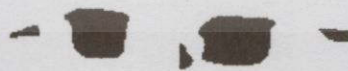
รูปที่ 4.11 ภาพดวงตาทั้งสองข้างในระบบสี Grayscale

จากนั้น นำภาพGrayscale มาปรับเพิ่มรายละเอียดในที่มืด เพื่อแยกรายละเอียดของลูกตาดำออกมา โดยใช้ฟังก์ชันลอการิทึม จะได้ผลลัพธ์ดังภาพ



รูปที่ 4.12 ภาพดวงตาทั้งสองเมื่อผ่านการฟังก์ชันลอการิทึม

ถัดมานำภาพที่ผ่านการเพิ่มรายละเอียดแล้ว นำมาแปลงผลลัพธ์เป็นภาพ Binary ดังรูป

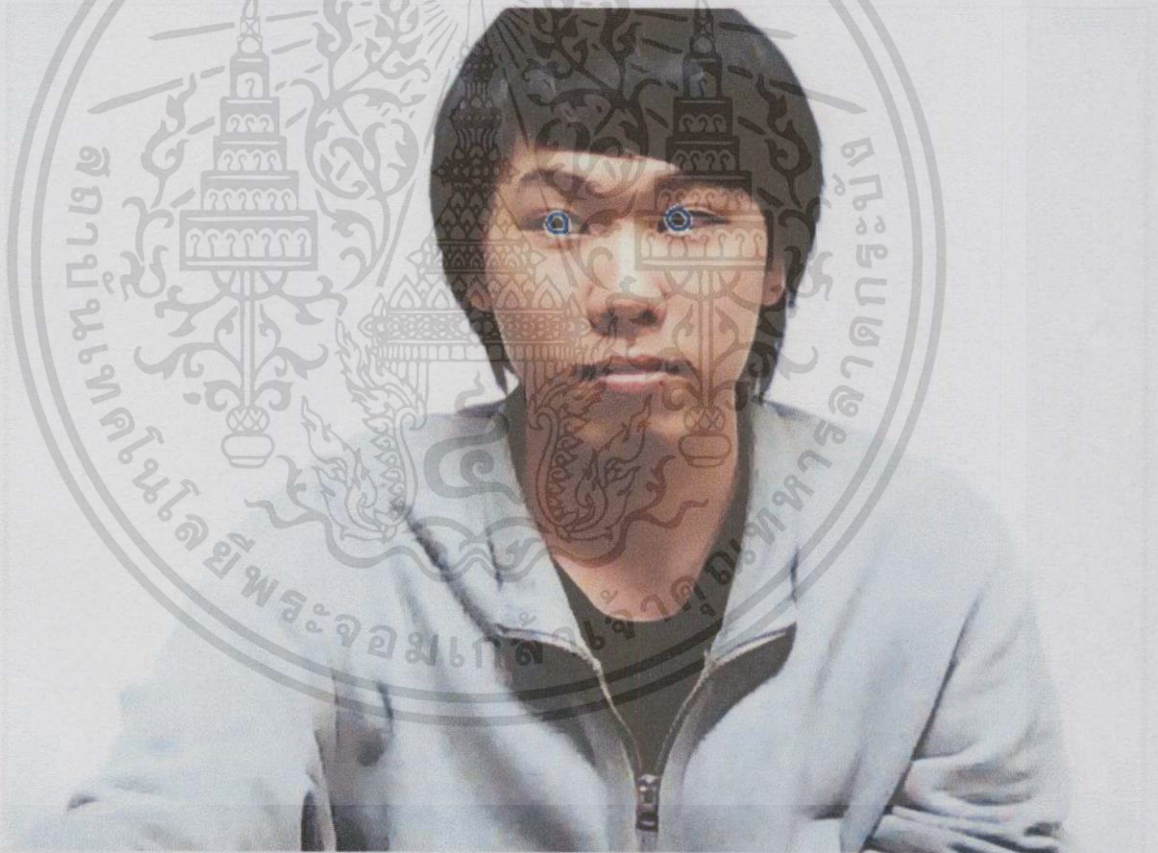


รูปที่ 4.13 ภาพดวงตาทั้งสองในระบบสี Binary

จากนั้นใช้วิธีการ Closing Operation กับภาพ เพื่อให้ได้ภาพวงกลมของลูกตาดำและใช้ Hough Transform เพื่อหาวงกลม จะได้ดังรูป

รูปที่ 4.14 ภาพลูกตาดำทั้งสองข้างหลังจากผ่านการทำ Closing Operation และหาวงกลม บริเวณที่เป็นลูกตาดำ

จากนั้นนำตำแหน่งและรัศมีของลูกตาดำไปแทรกในรูปภาพทั้งหมด จะได้ดังภาพ

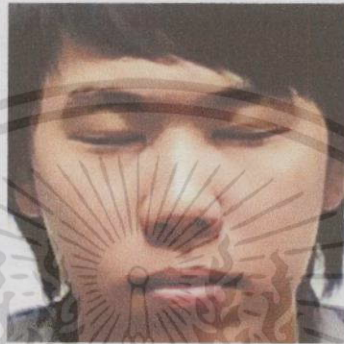


รูปที่ 4.15 ภาพผู้ใช้เมื่อผ่านการหาลูกตาดำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 การทดลองตรวจจับดวงตาโดยใช้ Haar Like Feature เมื่อผู้ใช้หลับตา

เมื่อสามารถตรวจจับใบหน้าได้ ขั้นตอนต่อมาของโปรแกรมคือตรวจจับดวงตาของผู้ใช้งาน เพื่อนำไปประมวลผลต่อ จากขั้นตอนแรก เราจะรับภาพเฉพาะส่วนของใบหน้าของผู้ใช้ก่อน ดังรูป



รูปที่ 4.16 ภาพใบหน้าที่ได้จากการตรวจจับใบหน้า

ต่อมา เมื่อได้รับภาพใบหน้า จะทำการหาดวงตา และตัดแบ่งภาพของดวงตา ดังรูป



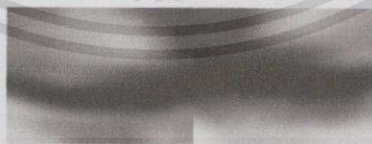
รูปที่ 4.17 ภาพดวงตาทั้งสองข้างที่ได้จากการตรวจจับดวงตา

ต่อมา นำภาพดวงตาที่ได้ผ่าน Gaussian Filter เพื่อทำให้ภาพเบลอ ดังรูป



รูปที่ 4.18 ภาพดวงตาทั้งสองข้างเมื่อผ่าน Gaussian Filter

ขั้นตอนถัดไป นำภาพดวงตาที่ได้จากการกรอง มาแปลงเป็นระบบสี Grayscale จะได้ดังรูป



รูปที่ 4.19 ภาพดวงตาทั้งสองข้างในระบบสี Grayscale

จากนั้น นำภาพ Grayscale มาปรับเพิ่มรายละเอียดในที่มืด เพื่อแยกรายละเอียดของลูกตา
ดำออกมา โดยใช้ฟังก์ชันลอการิทึม จะได้ผลลัพธ์ดังภาพ

รูปที่ 4.20 ภาพดวงตาทั้งของเมื่อผ่านการฟังก์ชันลอการิทึม

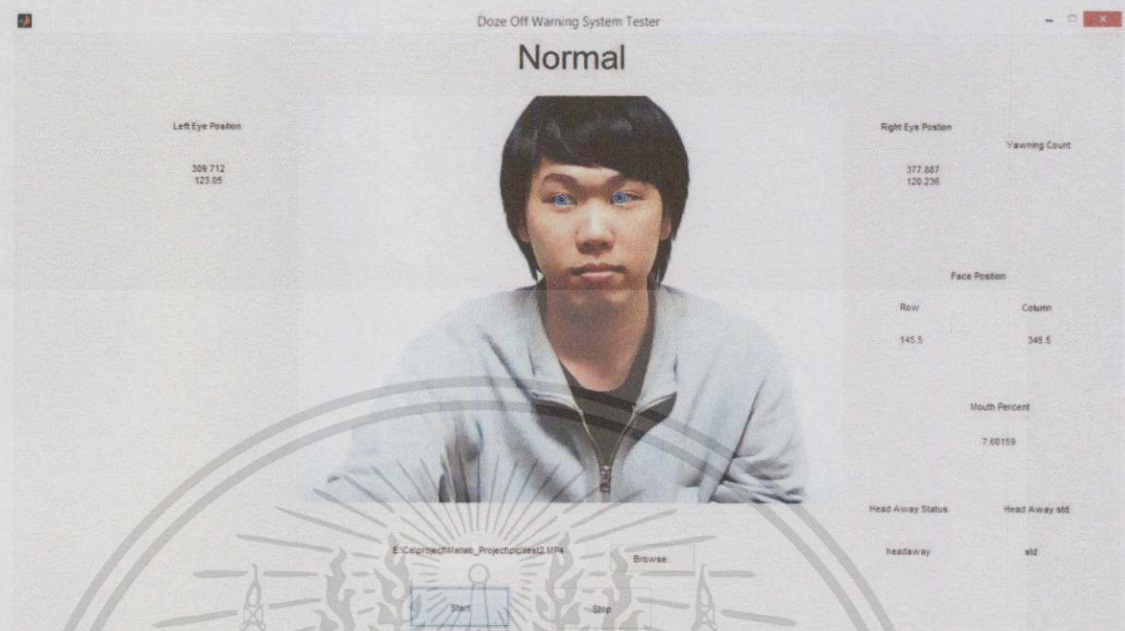
ถัดมานำภาพที่ผ่านการเพิ่มรายละเอียดแล้ว นำมาแปลงผลลัพธ์เป็นภาพ Binary ดังรูป

รูปที่ 4.21 ภาพดวงตาทั้งของในระบบสี Binary

จากนั้นใช้วิธีการ Closing Operation กับภาพ เพื่อให้ได้ภาพวงกลมของลูกตาดำและใช้
Hough Transform เพื่อนหาวงกลม จะได้ผลลัพธ์เป็นภาพขาว เพราะการ Closing ได้ลบภาพที่ไม่ได้
มีลักษณะคล้ายวงกลมออกไปจนหมด

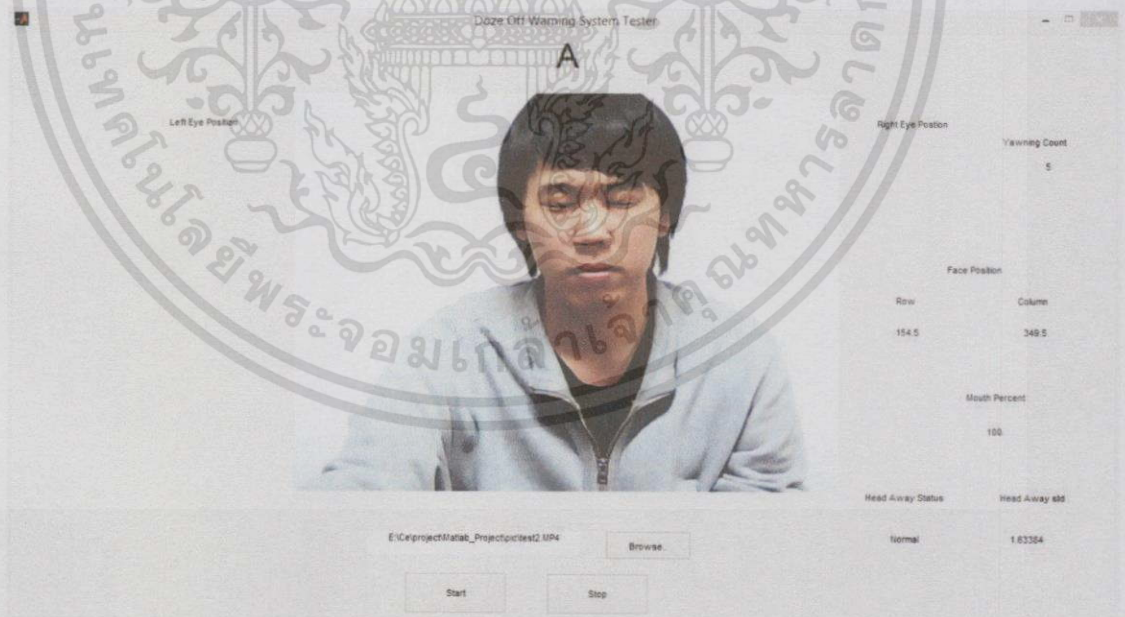
4.3.3 การทดลองตรวจหาอาการหลับ

ในการทดลอง จะใช้ไฟล์วิดีโอจำลองสถานการณ์เมื่อดวงตาอยู่ในสถานะปกติ จะได้ผลลัพธ์
ดังรูป



รูปที่ 4.22 รูปภาพเมื่ออยู่ในสถานะปกติ

โดยบริเวณค่านซ้ายจะบอกตำแหน่งของลูกตาดำซ้าย ด้านขวาจะบอกตำแหน่งของลูกตาดำด้านขวา และมีสถานะบอกอยู่ด้านบน ในรูปคือ Normal คือสถานะปกติต่อมา โปรแกรมตรวจหาลูกตาดำไม่พบ จึงเข้าสู่สถานะ A



รูปที่ 4.23 รูปแสดงเมื่อเข้าสู่สถานะ A

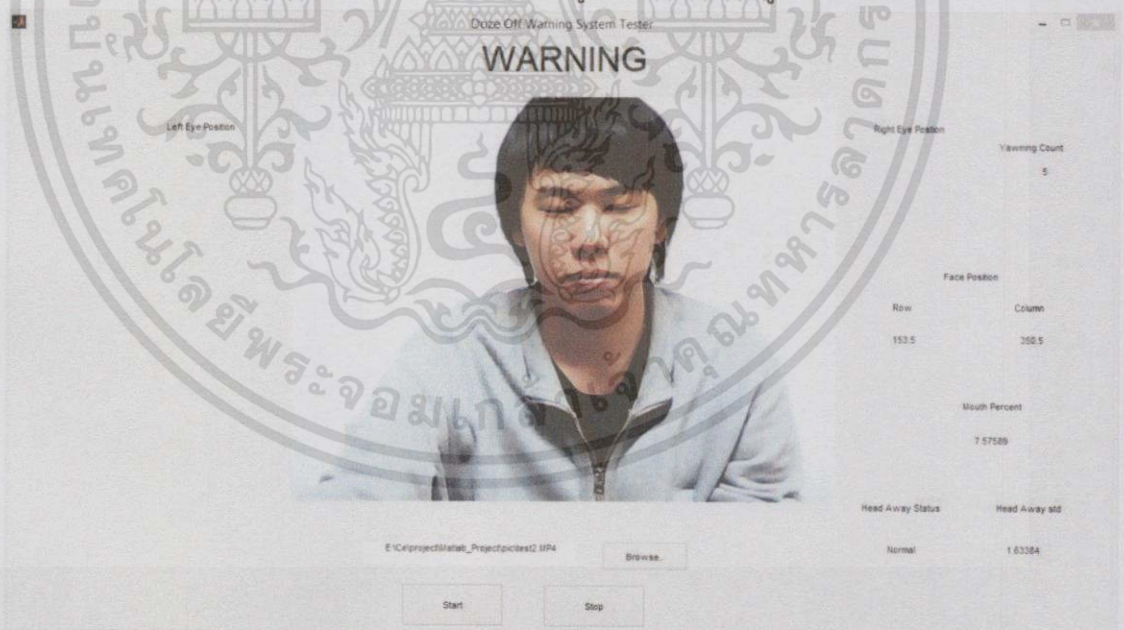
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในเฟรมถัดมา โปรแกรมก็ยังคงตรวจหาลูกตาดำไม่พบ จึงเข้าสู่สถานะ B



รูปที่ 4.24 รูปแสดงเมื่อเข้าสู่สถานะ B

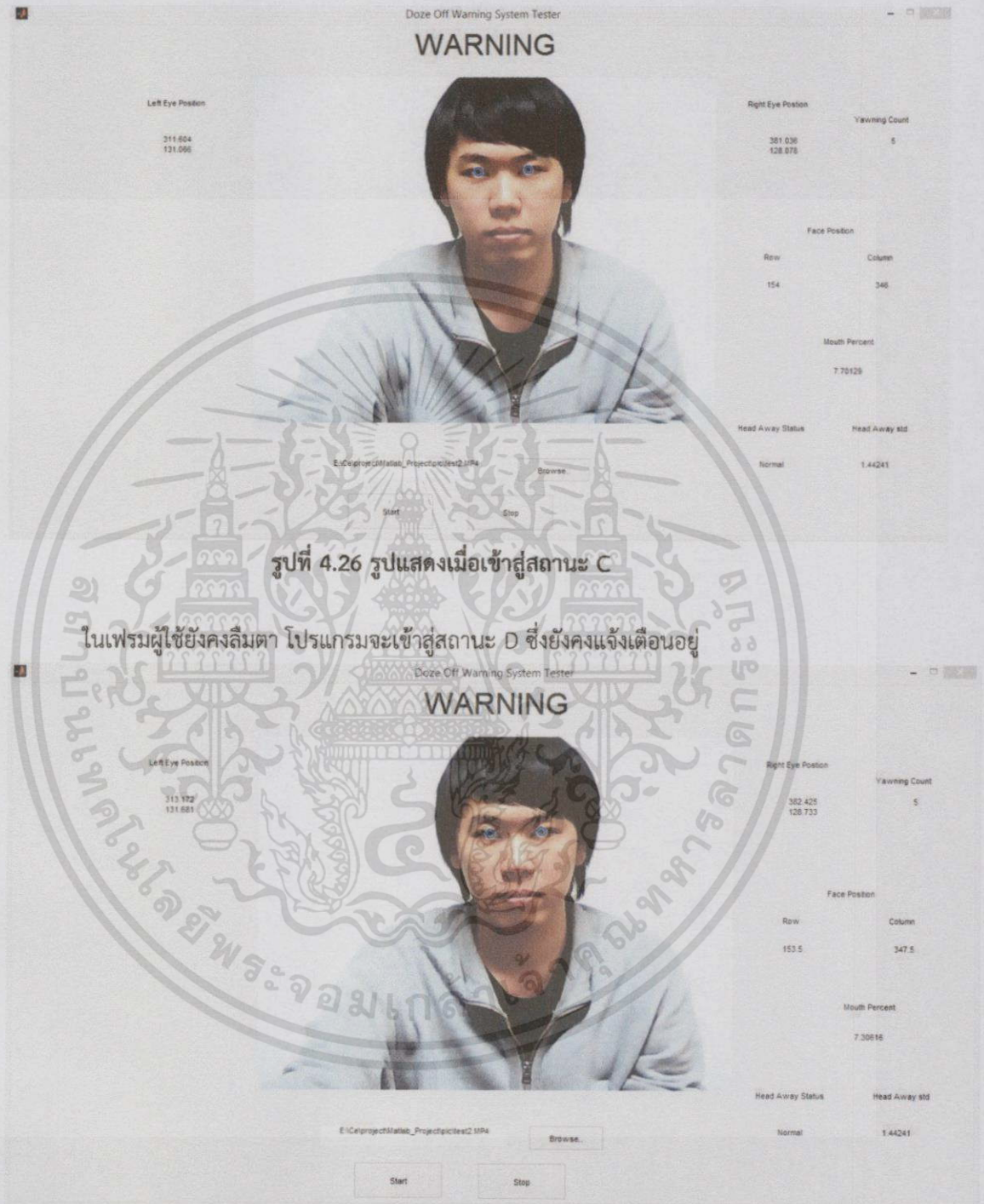
ในเฟรมถัดมา โปรแกรมก็ยังไม่สามารถตรวจหาลูกตาดำพบ จึงเข้าสู่สถานะแจ้งเตือน



รูปที่ 4.25 รูปแสดงเมื่อเข้าสู่สถานะแจ้งเตือน

ในเฟรมถัดมา ผู้ใช้ลืมตาจึงตรวจหาดวงตาพบ จึงเข้าสู่สถานะ C และยังคงแจ้งเตือนอยู่

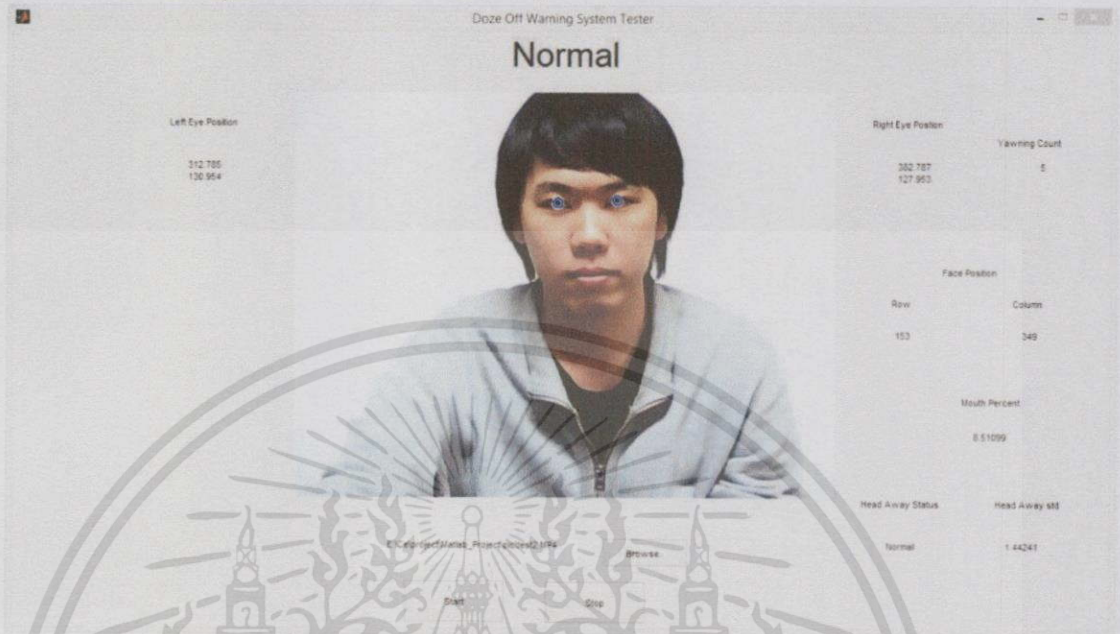
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.27 รูปแสดงสถานะ D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฟรมถัดไป ผู้ใช้ยังคงลืมตา โปรแกรมจึงเข้าสู่สถานะ Normal



รูปที่ 4.28 รูปแสดงเมื่อโปรแกรมกลับสู่สถานะปกติ

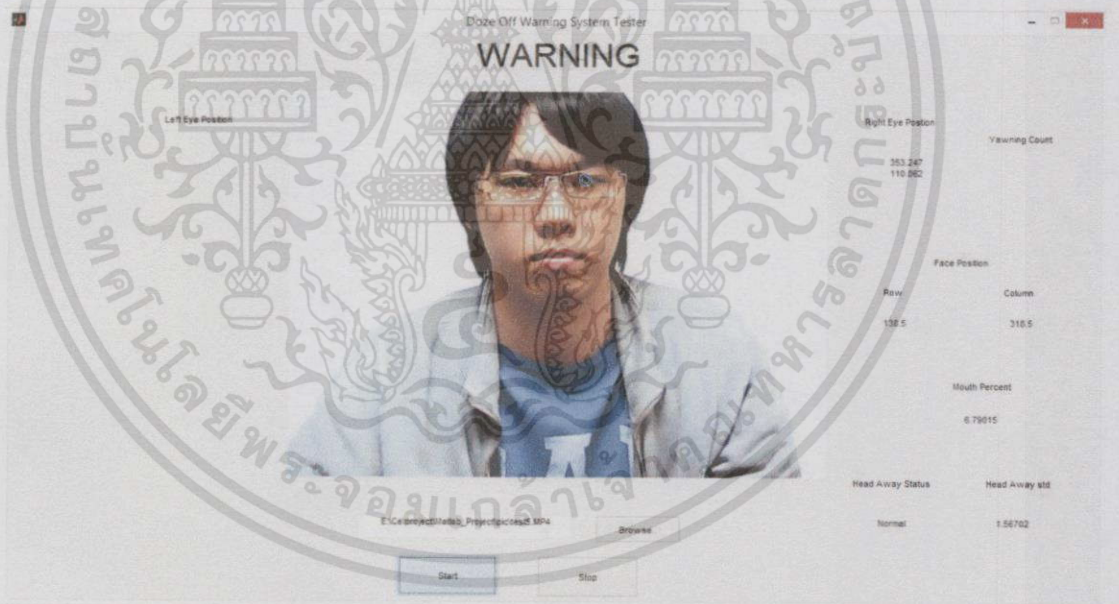
4.3.4 การตรวจจับดวงตาและหาลูกตาคำเมื่อผู้ใช้สวมแว่นตา

การสวมแว่นตาของผู้ใช้ จะสร้างปัญหาให้กับโปรแกรมทำให้ความแม่นยำในการหาลูกตาคำของโปรแกรมลดลง เพราะบริเวณเลนส์ของแว่นตาอาจมีแสงสะท้อนทำให้โปรแกรมทำงานได้อย่างไม่ถูกต้องเท่าที่ควร แต่หากกรณีที่มีแสงสะท้อนมีไม่มาก โปรแกรมก็ยังคงสามารถทำงานได้ในระดับหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.29 รูปแสดงการทดลองเมื่อผู้ใช้สวมใส่แว่นตา



รูปที่ 4.30 รูปแสดงการทดลองเมื่อผู้ใช้สวมใส่แว่นตา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.32 รูปแสดงการทดลองเมื่อผู้ใช้สวมใส่แว่นตา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.33 รูปแสดงการทดลองเมื่อผู้ใช้สวมใส่แว่นตา

4.4 การทดลองตรวจหาอาการเหนื่อยล้าจากเคลื่อนที่ของศีรษะ

การตรวจหาการเหนื่อยล้าจะประมวลผลจากตำแหน่งศูนย์กลางของใบหน้าในแนวตั้ง โดยจะเก็บข้อมูลจากเฟรมภาพจำนวนหนึ่ง ในการทดลองนี้จะเก็บภาพทั้งหมด 10 เฟรม และประมวลผลหาค่าส่วนเบี่ยงเบนมาตรฐานเพื่อหาอาการเหนื่อยล้า



รูปที่ 4.34 แสดงภาพที่ได้จากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Face Position	
Row	Column
145	348

รูปที่ 4.35 แสดงตำแหน่งจุดศูนย์กลางของภาพในแนวแถวและหลัก (Row , Column)

Head Away Status	Head Away std
headaway	std

รูปที่ 4.36 แสดงส่วนแสดงผลเกี่ยวกับค่าส่วนเบี่ยงเบนมาตรฐาน

เมื่อทำการเก็บข้อมูลจากวิดีโอสีเฟรมแรก ได้ค่าข้อมูลจุดศูนย์กลางดังนี้ 345.5,348,348.5,348.5,348.5,348,349,349,349,350 และนำข้อมูลมาหาส่วนเบี่ยงเบนมาตรฐาน จะได้ดังรูป

Head Away Status	Head Away std
Normal	1.17379

รูปที่ 4.37 แสดงผลลัพธ์ของการหาส่วนเบี่ยงเบนมาตรฐานจากข้อมูลสีเฟรมแรก

4.5 การทดลองตรวจหาอาการทาว

4.5.1 การทดลองตรวจหาริมฝีปาก

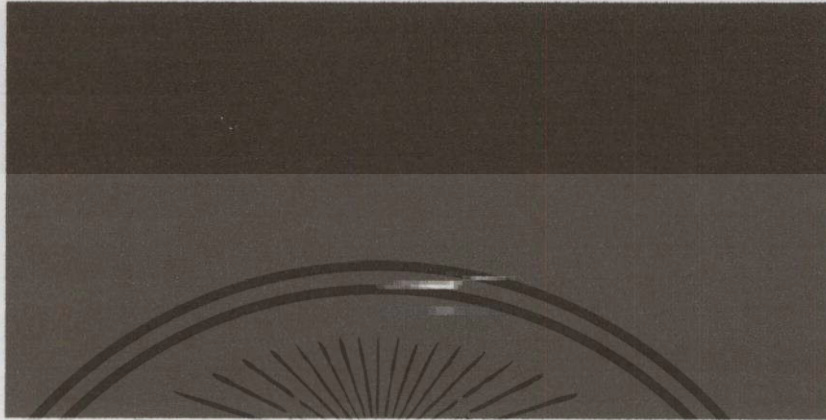
การหาริมฝีปาก เริ่มต้นจากการรับภาพครึ่งล่างของใบหน้าจากการตรวจจับใบหน้าที่ดังรูป



รูปที่ 4.38 ภาพใบหน้าครึ่งล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อมานำภาพครึ่งล่างของใบหน้าแปลงเป็นระบบสี YCbCr จากนั้นนำภาพเข้าสู่สมการ Lip Map จะได้รูปภาพดังรูป



รูปที่ 4.39 ภาพที่ได้จากสมการ Lip Map

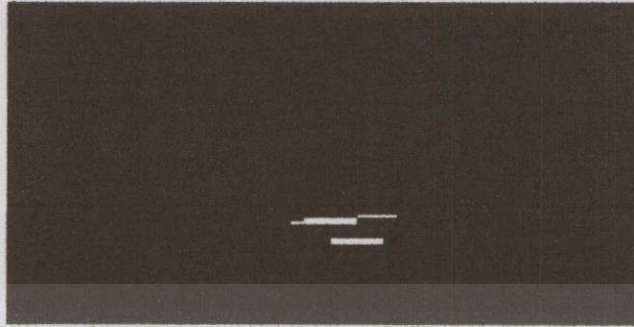
จากนั้นนำภาพที่ได้มาแปลงจาก Grayscale เป็นภาพ Binary



รูปที่ 4.40 รูป Binary ของริมฝีปาก

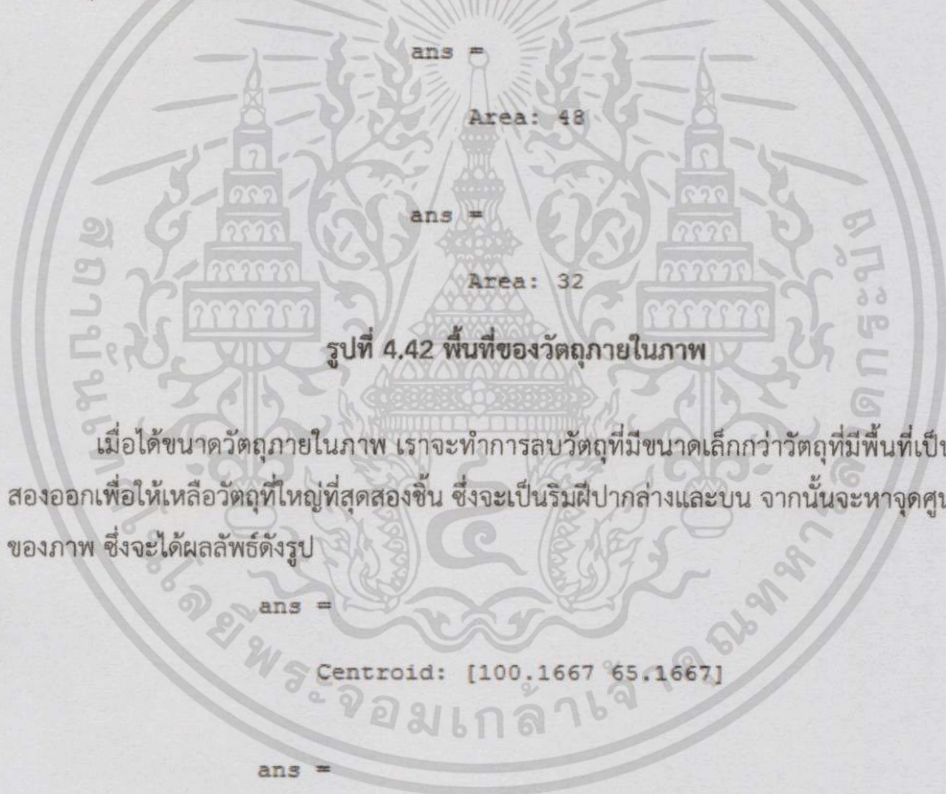
4.5.2 การทดลองหาอาการทวจากภาพริมฝีปากในกรณีริมฝีปากปิดปกติ

จากภาพ Binary ที่ได้จากสมการ Lip map ซึ่งก่อนจะนำไปประมวลผลได้ ต้องทำให้เหลือเพียงริมฝีปากบนและล่างก่อน โดยขั้นตอนแรกคือใช้ Opening Operation เพื่อลดสิ่งรบกวนในภาพ จะได้ผลลัพธ์ดังภาพ



รูปที่ 4.41 รูปปริซึมสี่เหลี่ยมที่ผ่าน Opening Operation

ต่อมาจะนำภาพที่ได้ มาเติมเต็มบริเวณที่เกิดหลุม และหลังจากนั้น จากจะทำการหาพื้นที่ของวัตถุในภาพ ซึ่งได้ผลลัพธ์ดังนี้



รูปที่ 4.42 พื้นที่ของวัตถุภายในภาพ

เมื่อได้ขนาดวัตถุภายในภาพ เราจะทำการลบวัตถุที่มีขนาดเล็กกว่าวัตถุที่มีพื้นที่เป็นอันดับสองออกเพื่อให้เหลือวัตถุที่ใหญ่ที่สุดสองชิ้น ซึ่งจะเป็นรูปสี่เหลี่ยมและบัน จากนั้นจะหาจุดศูนย์กลางของภาพ ซึ่งจะได้ผลลัพธ์ดังรูป

ans =
Centroid: [100.1667 65.1667]

ans =
Centroid: [105.5000 71.5000]

รูปที่ 4.43 รูปแสดงจุดศูนย์กลางวัตถุทั้งสองชิ้น

ต่อมาเราจะนำค่าจุดศูนย์กลางของวัตถุมาลบกันและคำนวณเปรียบเทียบกับความสูงของรูปผลลัพธ์จะแสดงในหน้าจอโปรแกรมดังภาพ

Mouth Percent

6.59722

รูปที่ 4.44 แสดงร้อยละระหว่างระยะห่างของริมฝีปากล่างและบนกับความสูงของรูปครึ่งล่างของใบหน้า

4.5.3 การทดลองหาอาการทวารจากริมฝีปากในกรณีปากอ้ากว้าง

การทวารจะเกิดในกรณีผู้ใช้งานอ้าปากกว้างติดต่อกันเป็นระยะเวลาหนึ่งซึ่งแสดงได้ถึงอาการทวาร ในขั้นตอนนี้ก็จะทำเช่นเดียวกันกับการหาปากในกรณีปากปิดสนิท โดยเริ่มต้นจากรับภาพครึ่งล่างของใบหน้า



รูปที่ 4.45 รูปภาพใบหน้าส่วนล่าง

จากนั้นนำภาพครึ่งล่างที่ได้แปลงเป็นระบบสี YCbCr และนำเข้าสู่สมการ Lip map จะได้ผลลัพธ์ดังภาพ



รูปที่ 4.46 รูปภาพใบหน้าส่วนล่างที่ผ่านสมการ Lip map

ต่อมานำรูป Lip map ที่ได้แปลงเป็น Binary

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.47 รูปภาพครึ่งล่างใบหน้าผ่านสมการ Lip map และแปลงเป็นระบบสี binary

จากนั้น นำรูปที่เป็น Binary ไปทำ Opening Operation จะได้ผลลัพธ์ดังรูป



รูปที่ 4.48 รูปภาพใบหน้าส่วนล่างที่ผ่าน Opening Operation

ต่อมานำภาพริมฝีปากที่ได้ไปเติมหลุมในภาพ และหาพื้นที่ของวัตถุทั้งหมดในภาพ และเลือกเพียงวัตถุที่ใหญ่ที่สุดในภาพสองชิ้น ซึ่งได้ผลลัพธ์ดังรูป

ans =

Area: 68

ans =

Area: 386

ans =

Area: 240

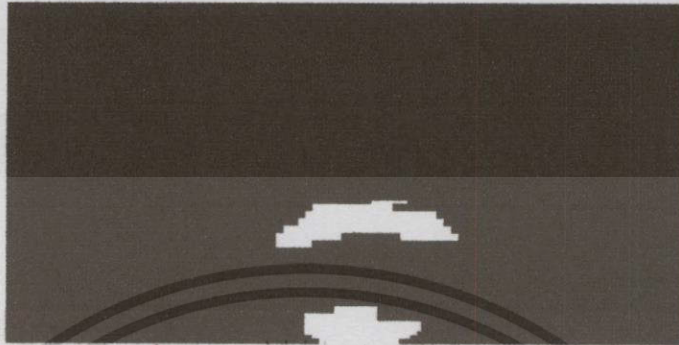
ans =

Area: 32

รูปที่ 4.49 แสดงผลลัพธ์ของพื้นที่ของวัตถุทั้งหมดในรูปภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นเลือกเฉพาะวัตถุที่ใหญ่ที่สุดเพียงสองภาพ เพื่อให้เหลือเฉพาะริมฝีปากบนและล่าง
เอาไว้ ดังรูป



รูปที่ 4.50 แสดงรูปภาพริมฝีปากบนและล่าง

จากนั้นนำจุดศูนย์กลางของวัตถุในแนวระดับของภาพมาหาผลต่างและเทียบกับขนาดความ
สูงของภาพทั้งหมดดังภาพ

ans =

Centroid: [97.4741 61.0207]

ans =

Centroid: [96.0875 89.0958]

รูปที่ 4.51 แสดงผลลัพธ์ของจุดศูนย์กลางของริมฝีปากบนและล่าง

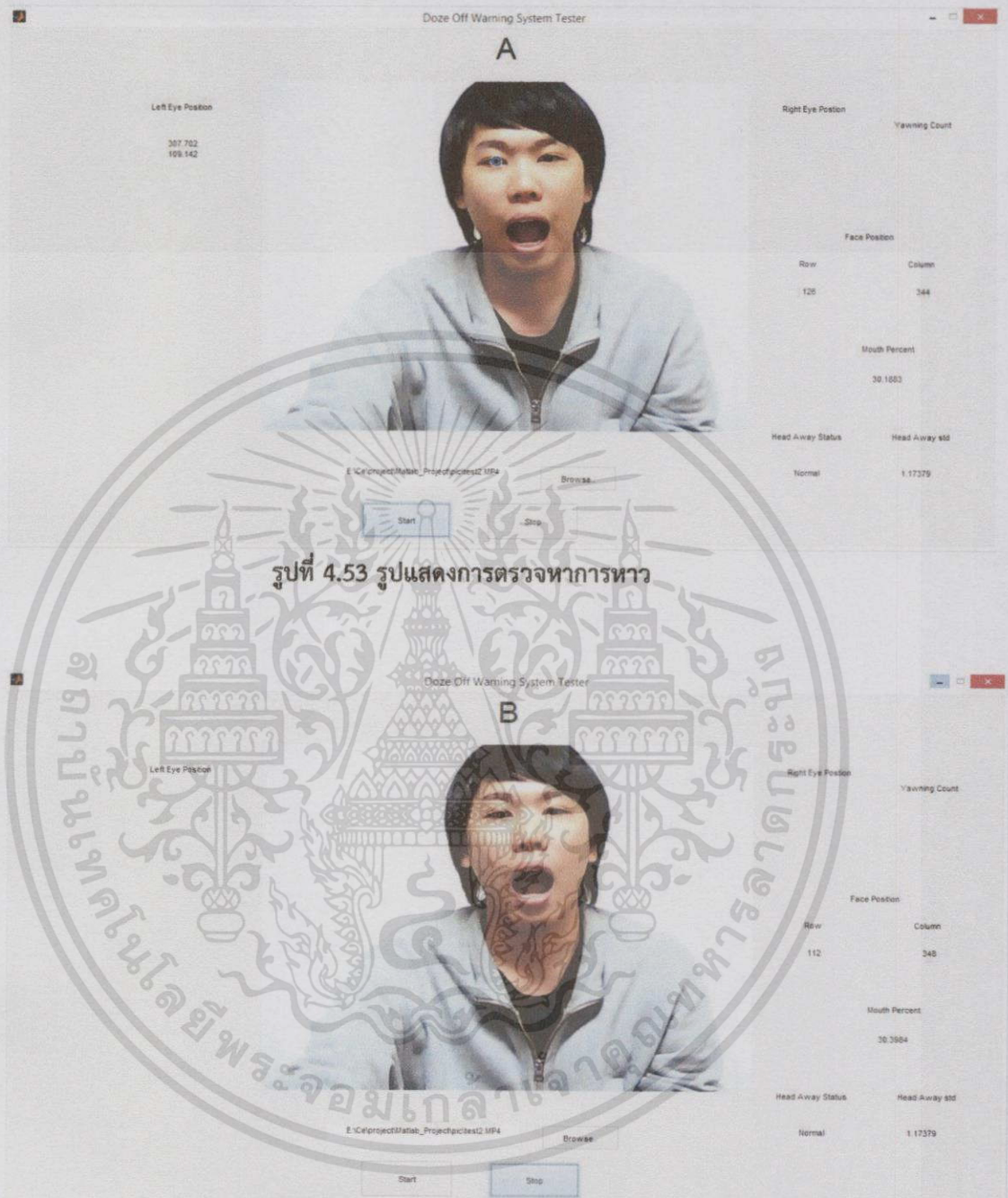
Mouth Percent

30.1883

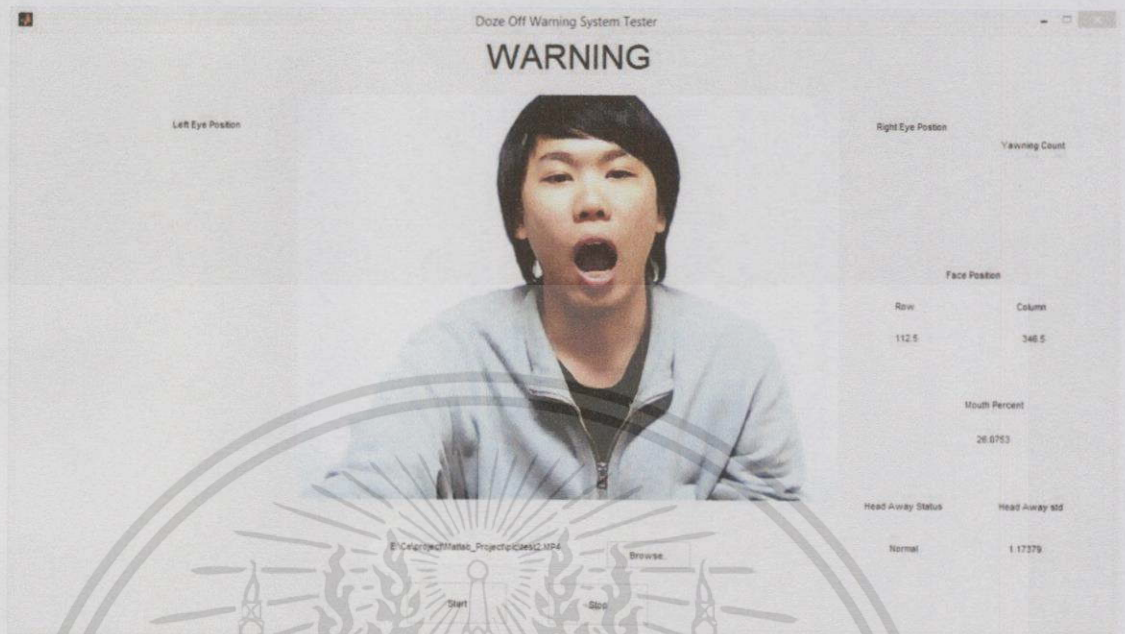
รูปที่ 4.52 แสดงร้อยละระหว่างระยะห่างของริมฝีปากกับความสูงของภาพครึ่งล่างของใบหน้า

4.5.2 การทดลองตรวจสอบการหาในไฟล์วิดีโอทดลอง

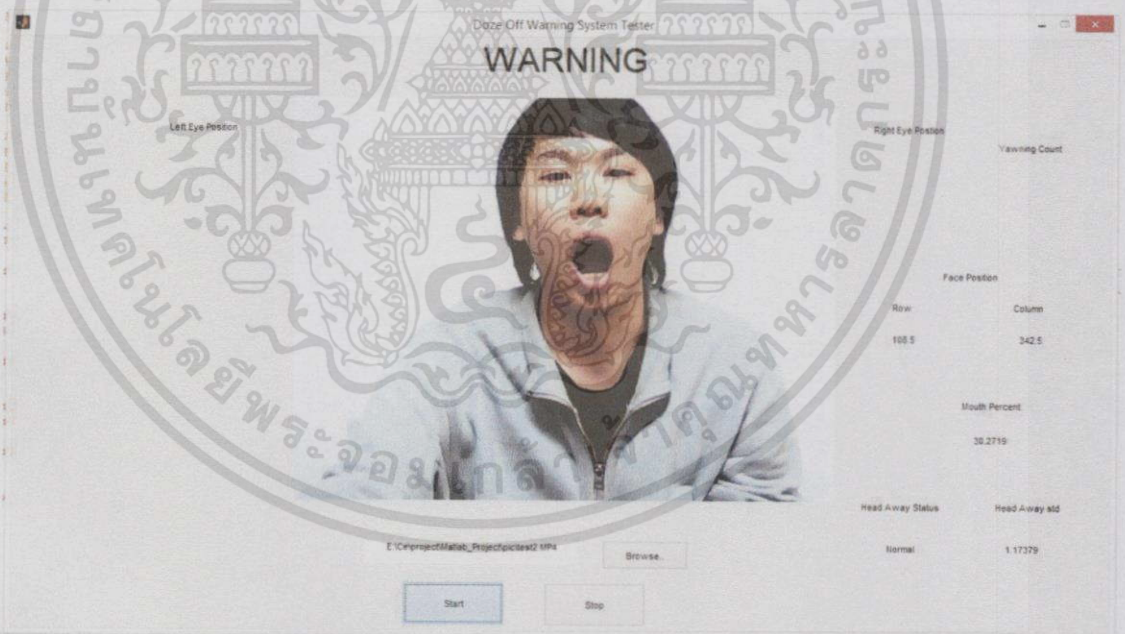
ในการทดลองจะนำวิดีโอซึ่งถ่ายจากกล้องเว็บแคมมาทดลองหาอาการหา ตามขั้นตอนที่ได้
กล่าวไปก่อนในข้างต้นแล้ว ซึ่งได้ผลลัพธ์ที่ได้ จะมีค่า percent มากกว่า 20 เป็นจำนวนเฟรม
ติดต่อกันเป็นจำนวนหนึ่ง และเมื่อตรวจสอบได้ตามจำนวนเฟรม ก็จะตรวจนับได้ว่า มีการหาเกิดขึ้น
1 ครั้ง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

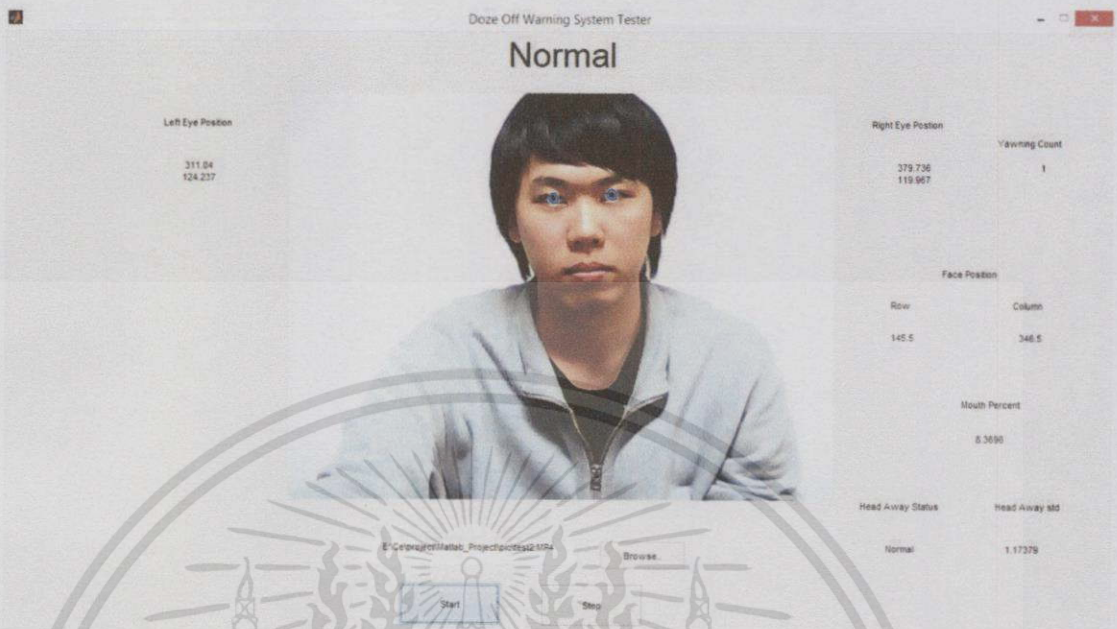


รูปที่ 4.55 รูปแสดงการตรวจหาการหา



รูปที่ 4.56 รูปแสดงการตรวจหาการหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.57 รูปแสดงการตรวจหาการท้าว

4.5.3 การตรวจหาการท้าวกรณีผู้ใช้มีการพูดคุย

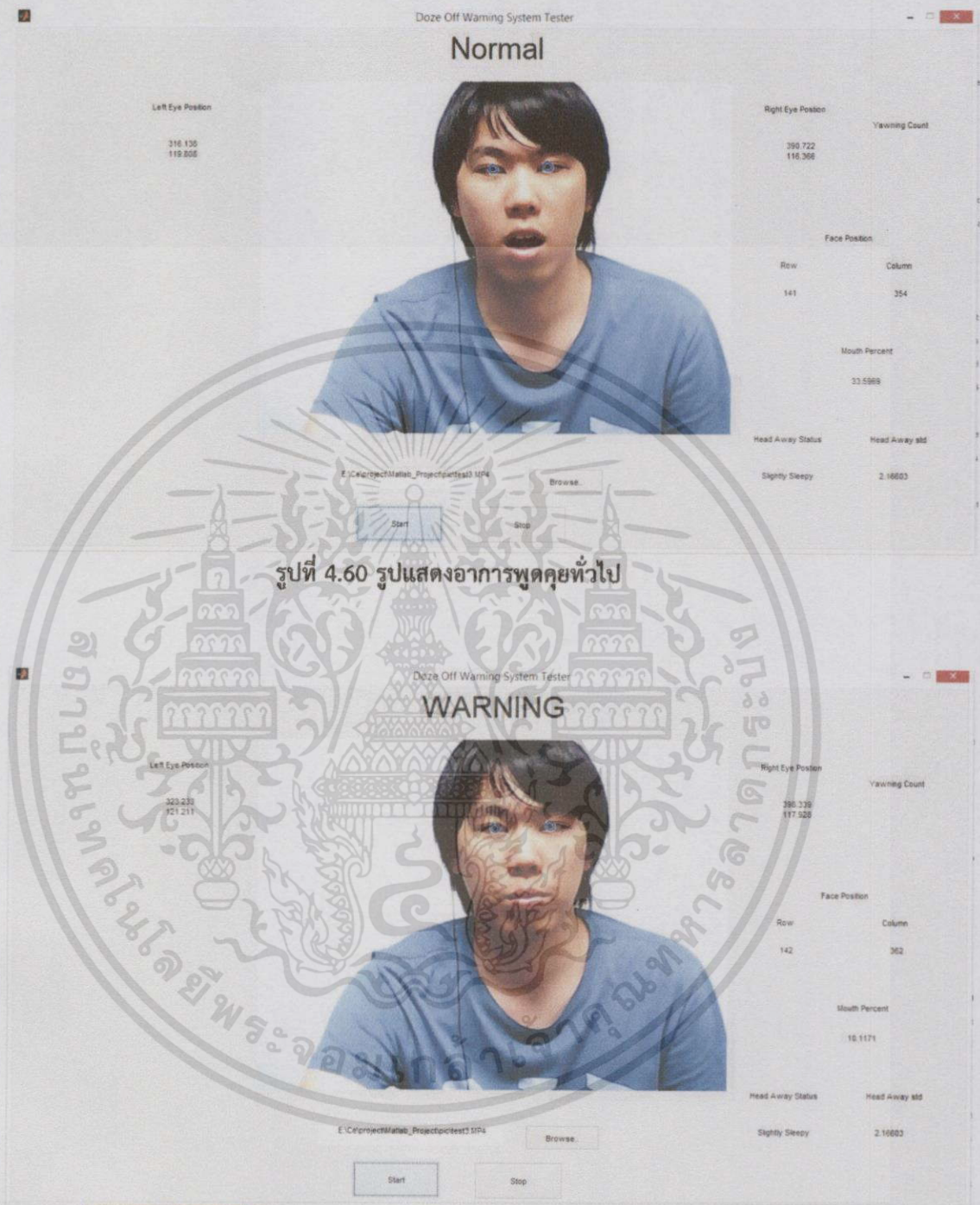
จากการตรวจหาการท้าว โปรแกรมใช้วิธีหาส่วนต่างระหว่างริมฝีปากบนและล่าง เพื่อหาการท้าว ในขณะที่หากผู้ใช้มีการพูดคุย โดยทั่วไป ปากก็จะไม่ได้อ้ากว้างมากนัก และหากมีการอ้ากว้าง ก็จะได้กว้างติดต่อกันนานเหมือนกับเวลาท้าว ซึ่งจะพบค่าร้อยละระหว่างความห่างของริมฝีปากกับความสูงของรูปครึ่งล่างของใบหน้าที่มีค่าสูงอยู่บ้าง แต่ไม่พบติดต่อกันหลายๆเฟรมเหมือนกับอาการท้าว จึงไม่พบอาการท้าว ดังรูป



รูปที่ 4.58 รูปแสดงอาการปกติทั่วไป

รูปที่ 4.59 รูปแสดงอาการผิดปกติทั่วไป

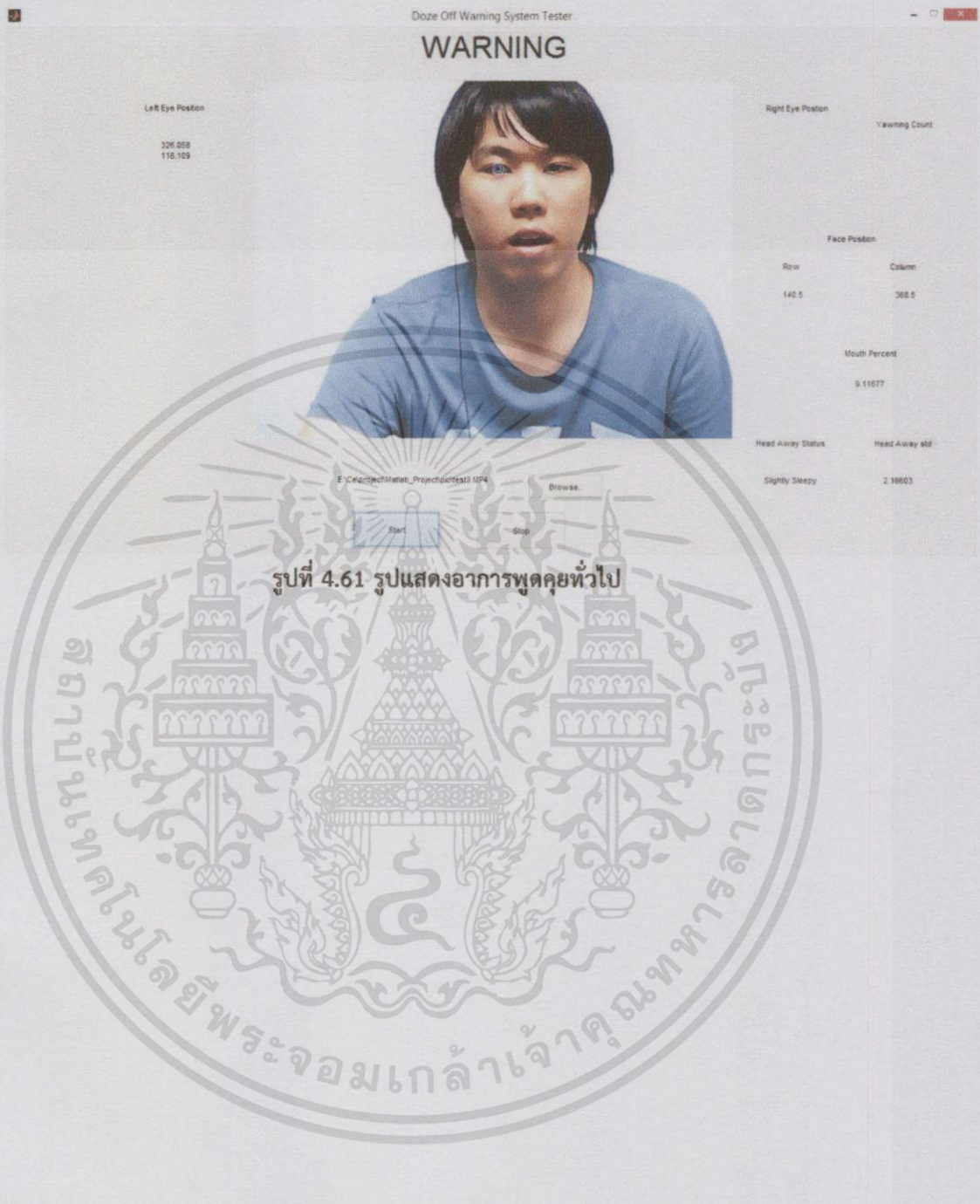
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.60 รูปแสดงอาการปกติทั่วไป

รูปที่ 4.61 รูปแสดงอาการผิดปกติทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 บทสรุปโครงการ

ในการพัฒนาระบบแจ้งเตือนการหลับใน ได้ทำการพัฒนาโดยแบ่งเป็น 3 ส่วนหลักๆคือ

- 1) แจ้งเตือนอาการหลับในจากการทาลูกตาดำ
- 2) ตรวจสอบอาการเหนื่อยล้าจากการเคลื่อนไหวของศีรษะ
- 3) ตรวจสอบอาการหาว

โดยในการพัฒนาโปรแกรมสามารถทำงานบนไฟล์วิดีโอตัวอย่างซึ่งนำมาทดสอบแนวคิดและอัลกอริธึมของโปรแกรม โดยเริ่มจากการตรวจจับใบหน้าของผู้ใช้ และนำภาพใบหน้าของผู้ใช้มาวิเคราะห์เพื่อหาข้อมูลที่ต้องการนำมาประมวลผล

5.2 ปัญหาอุปสรรคและแนวทางการแก้ไข

- 1) โปรแกรมไม่สามารถใช้งานได้จริงเพราะใช้เวลาในการทำงานมากเกินไป โดยปกติกล้องโดยทั่วไปสามารถรับภาพได้ประมาณ 30 ภาพต่อวินาที โดยเราสามารถเลือกภาพมาประมวลผลเพียง 10 ภาพต่อวินาทีได้ แต่โปรแกรมเวลาในการประมวลผลต่อภาพนั้นมากกว่า 1 วินาที จึงทำให้ยังคงไม่สามารถนำไปใช้งานได้ แนวทางการแก้ไขคือ เตรียมเครื่องที่มีประสิทธิภาพให้ดีขึ้น โดยอาจจะใช้อุปกรณ์ที่มีความเหมาะสมในการประมวลผลภาพเช่น GPU เข้ามาช่วยเหลือ หรือปรับปรุงอัลกอริธึมให้มีความซับซ้อนน้อยลง โดยที่ยังให้ผลลัพธ์เหมือนหรือใกล้เคียงกับของเดิมมากที่สุด
- 2) กล้องยังคงมีปัญหากรณีในที่สภาพแสงน้อย ทำให้ภาพที่ได้มี Contrast ไม่เพียงพอที่จะใช้ในการประมวลผล โดยเฉพาะอย่างยิ่งในส่วนของทาลูกตาดำซึ่งต้องใช้การแยกสีดำและขาวออกให้ได้ดีที่สุด โดยแนวทางการแก้ไขคือปรับแก้ไขการตั้งค่าตัวกล้องให้ดีขึ้น
- 3) เนื่องด้วยโปรแกรมยังไม่สามารถทำงานภายใต้สถานการณ์จริงได้ จึงทำให้ไม่สามารถนำไปทดลองในการใช้งานจริงคือการนำกล้องไปติดตั้งในรถยนต์ของผู้ใช้งาน โดยจะต้องแก้ไขในส่วนของเวลาที่ใช้ในการประมวลผลให้ลดลงก่อน
- 4) ความรู้ความเข้าใจในการประมวลผลภาพขั้นสูงมีไม่เพียงพอ ทำให้ในบางจุดโปรแกรมมีความแม่นยำไม่สูงเท่าที่ควรนัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 แนวทางการพัฒนาต่อ

- 1) ศึกษาพฤติกรรมกรรมการบอร์ดของผู้ซับซ้อนเพื่อให้โปรแกรมทำงานได้ดีขึ้น
- 2) ปรับปรุงอัลกอริทึมและการทำงานของโปรแกรมเพื่อลดเวลาในการประมวลผลลง เพื่อให้โปรแกรมสามารถใช้งานจริง
- 3) ศึกษาการเปลี่ยนแปลงของวัตถุที่สนใจเช่น ดวงตา เพื่อตรวจหาอาการกระพริบตา ซึ่งจะมีประโยชน์มากในการพัฒนาส่วนต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] อำนาจ กิตติกรัยฤทธิ “อิมแล้วขับ หลับใน อาจถึงตายได้.” [Online] Available : <http://cpfshe.cpportal.net/Default.aspx?tabid=121&articleType=ArticleView&articleId=192>
- [2] “บทที่ 1 แนะนำสู่การประมวลผลภาพดิจิทัล.” [Online] Available: <http://fivedots.coe.psu.ac.th/~montri/Teaching/image/chap1.htm>
- [3] “บทที่ 8 การปรับปรุงภาพ.” [Online] Available: <http://natres.psu.ac.th/Department/EarthScience/remote1/chapter8.pdf>
- [4] ผศ.ดร. อรฉัตร จิตตโสภักดิ์ . **ทฤษฎีการประมวลผลภาพดิจิทัล**
กรุงเทพมหานคร : บริษัทสงวนกิจ พริน แอนด์ มีเดีย . 2552
- [5] Gary Bradski , Adrian Kaehler **Learning Open CV** United States of America : O’Reilly Media, Inc., 2008
- [6] “Matlab for Thai people.” [Online]. Available: <http://matabthai.blogspot.com/2013/04/matlab.html>. 2013.
- [7] รุสลี สุทธิวีร์กุล , วิไลพร แซ่ลี “การตรวจจับใบหน้าด้วยการพื้นฐานของการจำลองรูปแบบ Haar-like” **วารสารวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ**, ปีที่ 6, ฉบับที่ 2 เดือนกรกฎาคม – ธันวาคม 2554 หน้า 34-43.
- [8] Behnoosh Hariri , Shabnam Abtahi , Shervin Shirmohammadi “Luc Martel .A Yawning Measurement Method to Detect Driver Drowsiness” In: **Distributed and Collaborative Virtual Environments Research Laboratory, University of Ottawa, Ottawa, Canada**
- [9] Haruo Matsuo, Abdelaziz Khiat “Prediction of Drowsy Driving by Monitoring Driver’s Behavior” **21st International Conference on Pattern Recognition (ICPR 2012)** November 11-15, 2012. Tsukuba, Japan pp 3390-3393

