กล้องติดตามด้วย Image Processing

Tracking Camera with Image Processing

นายศุภวิชญ์ รังสิธาดา
Mr.Supawich Rungsithada
นายปัณณวิชญ์ หรรษา
Mr.Punnawich Hansa

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษาที่ 2560

กล้องติดตามด้วย Image Processing

Tracking Camera with Image Processing

จัดทำโดย

นายศุภวิชญ์ รังสิธาดา

นายปัณณวิชญ์ หรรษา

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ พลศาสตร์ เลิศประเสริฐ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษาที่ 2560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษาที่ 2560

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

เรื่อง กล้องติดตามด้วย Image Processing

Tracking Camera with Image Processing

ผู้จัดทำ นาย ศุภวิชญ์ รังสิธาดา **รหัสนักศึกษา** 57011275

นาย ปัณณวิชญ์ หรรษา **รหัสนักศึกษา** 57011565

ปริญญานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาเรียบร้อยแล้ว

(ผู้ช่วยศาสตราจารย์ พลศาสตร์ เลิศประเสริฐ)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	กล้องติดตามด้วย Image Processing
นักศึกษา	นายศุภวิชญ์ รังสิธาดา รหัสประจำตัว 57011275
	นายปัณณวิชญ์ หรรษา รหัสประจำตัว 57011565
ปริญญา	วิศวกรรมศาสตรบัณฑิต
ภาควิชา	วิศวกรรมอิเล็กทรอนิกส์
ปีการศึกษา	2560
อาจารย์ที่ปรึกษาโครงการ	ผศ.พลศาสตร์ เลิศประเสริฐ

บทคัดย่อ

ปริญญานิพนธ์นี้อธิบายเกี่ยวกับการเขียนโปรแกรมและใช้งานโปรแกรมทางด้าน Image Processing ในการจับภาพบุคคลหรือภาพวัตถุที่เราสนใจโดยอิงสีแดงซึ่งเป็นสีของอุณหภูมิเป็นที่หลัก การใช้งานทำได้โดยการเชื่อมต่อกล้องเว็บแคมเข้ากับโปรแกรม MATLAB แล้วทำการเขียนโค้ดเพื่อรันให้เกิดโปรแกรมการทำงานตามลำดับขั้นตอนคือ การทำการจับภาพบุคคลหรือภาพวัตถุที่ต้องการแล้วคงตำแหน่งของภาพนั้นให้อยู่ที่กึ่งกลางเมื่อเทียบกับภาพที่กล้องที่จับได้ โดยใช้บอร์ดไมโครคอนโทรลเลอร์เพื่อที่จะควบคุมเซอร์โวมอเตอร์เพื่อที่จะใช้ปรับมุมการเคลื่อนที่หรือการหมุนของกล้องให้มีการเคลื่อนที่ตามที่กำหนดไว้ โดยในที่นี้เราต้องการใช้กล้องที่สามารถเคลื่อนไหวตามอุณหภูมิที่เรากำหนดไว้ในการเคลื่อนที่กระบอกปืนใหญ่รถถังให้สามารถเล็งเป้าหมายไปยังจุดที่มีอุณหภูมิสูงหรือรถถังฝ่ายตรงข้ามซึ่งมีเครื่องยนต์ที่เป็นส่วนประกอบที่มีความร้อนสูง ทำให้การเล็งเป้าหมายเป็นไปด้วยความถูกต้องและแม่นยำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Tracking Camera with Image Processing	
Student	Mr.Supawich Rungsithada	Student ID 57011275
	Mr.Punnawich Hansa	Student ID 57011565
Degree	Bachelor of Engineering	
Program	Electronics Engineering	
Year	2017	
Project Advisor	Asst. Prof. Polsart Lertprasert	

ABSTRACT

This thesis is represent about how to create and apply with Image Processing to captured, detect and tracking something we interested in. In this time, the main color we really need to detect is color that related with temperature and red is the answer of color we used. Webcam is connected to the computer's USB port, then make the communication sequentially with Matlab by systems programing. Photo with object is captured at the center of webcam, while the object is moving we use single-board microcontroller to control servo motor that can make webcam rotate to face the object on every degrees and moving. In this time, we need to set up the camera with tank cannon then tracking with the opposite tank to aim objective that consists of high temperature engine. All we need is accurate and precise.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา ผศ.พลศาสตร์ เลิศประเสริฐ ที่ให้คำปรึกษาชี้แนะแนวทางและการแก้ไขปัญหาที่เกิดขึ้นในงานโครงงาน ให้อุปกรณ์ที่จำเป็นบาง ส่วน ปรับเปลี่ยนอุปกรณ์ที่ใช้การไม่ได้ ตลอดจนการให้ความรู้ จนบรรลุตามเป้าหมายที่ตั้งใจไว้ด้วยดีทุกประการ

ขอขอบคุณพี่แบรนด์ Thai Summit Harness พี่ภูมิ ECONTECH และครูขัน กองทัพไทย ที่ถ่ายทอดความรู้เกี่ยวกับการทำงานของ Image Processing ที่สำคัญรวมทั้งให้คำแนะนำในด้านวิชาการที่มีประโยชน์ต่อโครงงานนี้ และขอขอบคุณรุ่นพี่และเพื่อนๆทุกคนที่ให้ความช่วยเหลือในทุกๆด้าน

สุดท้ายนี้ขอกราบขอบพระคุณมารดา ผู้ที่ทำให้มีแรงกายแรงใจในการทำงานโครงงานชิ้นนี้ ข้าวทุกมื้อทุกจาน ที่นอนที่อบอุ่นให้ข้าพเจ้าได้นอนเต็มอิ่มตื่นเต็มตา คอยช่วยเหลือและแนะนำสิ่งต่างๆในชีวิต ค่าใช้จ่ายต่างๆในการทำโครงงานชิ้นนี้จนสำเร็จ รวมถึงบุคคลอื่นๆ ที่ไม่ได้กล่าวนามทุกท่านที่ให้คำแนะนำช่วยเหลือในการจัดทำโครงงานฉบับนี้สำเร็จไปได้ด้วยดี ข้าพเจ้าขอขอบคุณไว้ ณ ที่นี้

สำหรับคุณประโยชน์อันพึงเกิดจากโครงงานเล่มนี้ ขอมอบให้แก่คณาจารย์ที่เคารพทุกท่าน ภาควิชาวิศวกรรมอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ซึ่งเป็นสถานที่ให้ความรู้และประสบการณ์ในการทำโครงงานฉบับนี้

ศุภวิชญ์ รังสิธาดา

ปัณณวิชญ์ หรรษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
บทคัดย่อภาษาไทย		I
บทคัดย่อภาษาอังกฤษ		II
กิตติกรรมประกาศ		III
สารบัญ		IV
สารบัญตาราง		VI
สารบัญรูป		VII
บทที่ 1	บทนำ	1
	1.1 ความเป็นมาและความสำคัญ	1
	1.2 วัตถุประสงค์ของโครงงาน	2
	1.3 ขอบเขตการวิจัยและขั้นตอนการศึกษา	2
	1.4 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน	3
บทที่ 2	ทฤษฎีและหลักการ	4
	2.1 กล้องเว็บแคม OKER 177	5
	2.1.1 คุณสมบัติเด่น	5
	2.2 การประมวลผลภาพ	6
	2.2.1 ภาพและความหมายของพิกเซล	7
	2.2.2 ระบบสีแบบ RGB	8
	2.2.3 การเปลี่ยนภาพจากระบบ RGB ไปเป็นระบบ Grayscale	10
	2.2.4 การแปลงภาพระดับสีเทาให้เป็นภาพขาว-ดำ	11
	2.2.5 กระบวนการปิดล้อมวัตถุ (Bounding box)	12
	2.3 โปรแกรม MATLAB	13
	2.3.1 การรับอินพุต	13
	2.3.2 การลบรูปภาพ	13
	2.3.3 การแปลงภาพ RGB เป็นภาพ Grayscale	14
	2.3.4 การแปลงภาพ Grayscale เป็นภาพ Binary	14
	2.3.5 การลบวัตถุเล็กออกจากภาพขาว-ดำ	14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

		หน้า
	2.4 โปรแกรม Arduino IDE	15
	2.5 Servo Motor	15
	2.5.1 หลักการทำงานของ Servo Motor	16
	2.5.2 ควบคุม Servo Motor ด้วยArduino	16
	2.6 บอร์ด Arduino UNO R2	17
บทที่ 3	กระบวนการสร้างและการทดลอง	18
	3.1 กระบวนการสร้างและการทดลอง	18
	3.1.1 อุปกรณ์ส่วนฮาร์ดแวร์	18
	3.1.2 ซอฟต์แวร์หลัก	19
	3.1.3 ขั้นตอนการเตรียมโปรแกรม	19
	3.1.4 ขั้นตอนการใช้งานจริง	24
	3.1.5 การรับค่าจากโปรแกรม MATLAB ส่งไปยัง Arduino UNO R3	32
บทที่ 4	ผลการทดลอง	39
	4.1 ผลการตรวจจับวัตถุ	39
	4.1.1 ผลการตรวจจับวัตถุในตอนกลางวัน	39
	4.1.2 ผลการตรวจจับวัตถุในตอนกลางคืน	40
	4.1.3 ผลการตรวจจับวัตถุในสภาพแวดล้อมปกติ	41
	4.2 ผลลการติดตามวัตถุ	43
บทที่ 5	สรุปผลการทดลอง	47
	เอกสารอ้างอิง	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
4.1 ค่าพิกัด x, y ในแต่ละเฟรม	44
4.2 ค่าพิกัด x, y ในแต่ละเฟรม (วัตถุเคลื่อนที่ช้า)	46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
1.1	การประมวลผลภาพและสัญญาณ	1
1.2	ผลลัพธ์ของการถ่ายภาพจากกล้องถ่ายภาพความร้อน	2
2.1	Human vision เปรียบเทียบกับ Computer vision	4
2.2	กล้องเว็บแคม OKER177	5
2.3	Block diagram การ Image Processing	6
2.4	การสแกนนิ้วมือจากการประมวลผลภาพ	7
2.5	พิกเซลภายในรูป	8
2.6	ระบบวงจรสีของแสงในระบบ RGB	9
2.7	ไดอะแกรมของลูกบาศก์ในระบบ RGB	9
2.8	ภาพแสดงระดับสีเทา	10
2.9	การแปลงภาพระบบ RGB เป็นภาพระบบ Grayscale	11
2.10	การแปลงภาพ RGB เป็น Grayscale และเป็น B&W	12
2.11	การสร้างสี่เหลี่ยมปิดล้อมวัตถุ	12
2.12	โปรแกรม MATLAB	13
2.13	โปรแกรม Arduino IDE	15
2.14	Block Diagram การทำงานของ Servo Motor	16
2.15	Servo Motor	16
2.16	Arduino UNO R3	17
3.1	อุปกรณ์ฮาร์ดแวร์หลัก	19
3.2	ภาพสีที่นำมาเป็นตัวอย่างในการวิเคราะห์	20
3.3	ค่า Intensity values ของภาพสี	20
3.4	Red plane	21
3.5	ค่า Red intensity values ในระนาบของสีแดง	21
3.6	ภาพ Grayscale	22
3.7	ค่า Intensity values ของภาพ Grayscale	22
3.8	ภาพเลเยอร์สีแดงที่ทำการลบภาพเทา	23
3.9	ค่า Intensity values ของภาพเลเยอร์แดงที่ทำการลบภาพเทา	23
3.10	ภาพขาวดำ	24
3.11	ภาพวัตถุที่มีค่าสีแสงใกล้เคียงกัน	25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
3.12	กราฟฮิสโตแกรมของภาพวัตถุที่มีค่าสีแดงใกล้เคียงกัน	25
3.13	ภาพเลเยอร์สีแดง	26
3.14	กราฟฮิสโตแกรมของภาพเลเยอร์สีแดง	26
3.15	ภาพเทา	27
3.16	กราฟฮิสโตแกรมของภาพเทา	27
3.17	ภาพเลเยอร์แดงที่ทำการลบภาพเทา	28
3.18	กราฟฮิสโตแกรมของภาพเลเยอร์แดงที่ทำการลบภาพเทา	28
3.19	ภาพขาวดำ	29
3.20	กราฟฮิสโตแกรมของภาพขาวดำ	29
3.21	ภาพขาวดำที่ทำการตีกรอบ	30
3.22	ภาพขาวดำที่ผ่านการเลือกค่าพิกเซล	31
3.23	กราฟฮิสโตแกรมของภาพขาวดำที่ทำการเลือกค่าพิกเซล	31
3.24	โปรแกรมส่วนขยาย MATLAB Support Package fo Arduino Hardware	32
3.25	ภาพจำลองการออกแบบและติดตั้งเซอร์โวมอเตอร์เพื่อใช้ในการหมุนกล้อง	34
3.26	การเชื่อมต่ออุปกรณ์	34
3.27	ขนาดของภาพที่ใช้งาน	35
3.28	การคำนวณจุดศูนย์กลางจาก Bounding box	36
3.29	พิกัดแกน x มีค่าน้อยกว่า 160	37
3.30	พิกัดแกน x มีค่ามากกว่า 160	37
3.31	พิกัดแกน y มีค่าน้อยกว่า 120	38
3.32	พิกัดแกน y มีค่ามากกว่า 120	38
4.1	วัตถุที่มีค่าสีแดงใกล้เคียงกัน	39
4.2	ผลลัพธ์การตรวจจับ	40
4.3	จับภาพวัตถุในตอนกลางคืน	40
4.4	ผลลัพธ์การตรวจจับในตอนกลางคืน	41
4.5	จับภาพวัตถุในห้อง	41
4.6	ค่า Intensity ของวัตถุสีแดงแต่ละตำแหน่งในห้อง	42
4.7	ผลลัพธ์การตรวจจับโดยการตัดพิกเซลขนาดเล็กออก	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.8	กราฟแสดงค่าพิกัด x, y ในแต่ละเฟรม	44
4.9	กราฟแสดงค่าพิกัด x, y ในแต่ละเฟรม (วัตถุเคลื่อนที่ช้า)	46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

Tracking Camera เป็นการนำทฤษฎีของ Image Processing มารวมกันกับการเขียนภาษา C, C++ ในโปรแกรม MATLAB และ Simulink โดยเป็นกล้องตรวจจับการเคลื่อนไหวของวัตถุที่ผ่านมาในรัศมีการรับภาพของกล้องเว็บแคม

นิยามของการประมวลผลภาพ (Image Processing) คือ การประยุกต์ใช้งานการประมวลสัญญาณบนสัญญาณ 2 มิติ เช่น ภาพนิ่ง ภาพถ่าย ภาพวิดีโอ หรือภาพวีดีทัศน์ ครอบคลุมไปถึงสัญญาณอื่นๆที่ไม่ใช่เพียงแค่ภาพถ่ายแต่เป็นสัญญาณ 2 มิติอีกด้วย

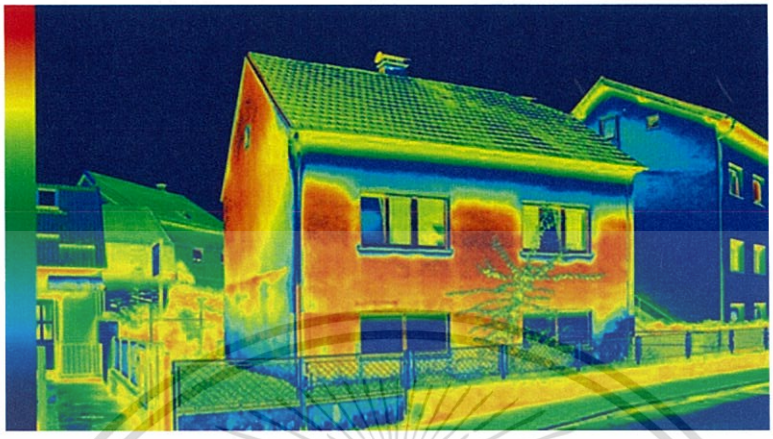
การมองเห็นของมนุษย์เป็นสิ่งที่สำคัญและเป็นกลไกการรับภาพที่ซับซ้อนอย่างหนึ่ง ซึ่งจะให้ข้อมูลที่มีความจำเป็นสำหรับใช้งานง่ายๆ เช่น การจดจำวัตถุ และสำหรับงานที่มีความซับซ้อน เช่น การวางแผน การตัดสินใจ การค้นคว้าทางวิทยาศาสตร์ การพัฒนาทางด้านความคิด เป็นต้น ดังคำสุภาษิตจีนที่กล่าวไว้ว่า “รูปเพียงหนึ่งรูปสามารถแทนคำพูดได้เป็นพันๆคำ” รูปภาพมีบทบาทมากสำหรับองค์กรต่างๆ เช่น วงการหนังสือพิมพ์ วงการโทรทัศน์ และวงการภาพยนตร์ ที่มีการใช้ภาพนิ่งหรือภาพเคลื่อนไหวในการเป็นสื่อกลางในการนำเสนอข้อมูลข่าวสารต่างๆ สิ่งที่น่าสนใจของข้อมูลที่เกี่ยวกับการมองเห็นหรือข้อมูลภาพนั้นก็คือกระบวนการประมวลผลภาพ (Image Processing) โดยใช้ดิจิตอลคอมพิวเตอร์

รูปที่ 1.1 การประมวลผลภาพและสัญญาณ

โดยปัจจุบันแล้วทฤษฎีการประมวลผลภาพมักจะปรากฏตัวแบบผสมผสานอยู่ในรูปแบบของคำสั่งในการทำภาพนิ่งหรือภาพเคลื่อนไหวต่างๆ โดยจะสอดแทรกอยู่ในสื่อต่างๆที่เรารับรู้ได้จาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปภาพ โดยในที่นี้เราจะเน้นไปที่การออกคำสั่งในการตรวจสอบและตรวจจับการเคลื่อนไหวของวัตถุบางอย่างที่เราต้องการจะศึกษา ซึ่งในที่นี้คือวัตถุใดๆที่เคลื่อนไหวโดยมีอุณหภูมิภายใน โดยจะเป็นการศึกษาเพิ่มเติมโดยใช้พิกเซลสีแดงแทนอุณหภูมิภายในวัตถุเนื่องด้วยอุปกรณ์กล้องที่ใช้การตรวจวัดอุณหภูมินั้นมีราคาสูงเกินกว่าจะนำมาใช้งานได้จริงในโครงงานนี้



รูปที่ 1.2 ผลลัพธ์ของการถ่ายภาพจากกล้องถ่ายภาพความร้อน

1.2 วัตถุประสงค์ของโครงงาน

โครงงานนี้นำเสนอความสามารถและความสำคัญของอุปกรณ์ Tracking Camera ซึ่งมีอิทธิพลต่อโลกดิจิตอลในปัจจุบันมาก อีกทั้งเพื่อนำไปต่อยอดความรู้ในการออกแบบและเป็นจุดเริ่มต้นทางความคิดที่จะผลิตอุปกรณ์การตรวจสอบหรือจับภาพโดยอาศัยเทคนิคและปัจจัยต่างๆในการค้นหาสิ่งที่สนใจที่มีลักษณะไปในทางเดียวกัน เพื่อให้สอดคล้องกับโลกในปัจจุบันที่นิยมใช้อุปกรณ์ดิจิตอลในการตรวจสอบเอกลักษณ์หรือลักษณะเฉพาะของสิ่งต่างๆ เพื่อให้ระบุออกมาเป็นผลลัพธ์ที่ต้องการ

1. ศึกษากระบวนการใช้งานเบื้องต้นของกล้องเว็บแคมในการตรวจสอบ
2. ศึกษาการใช้งานโปรแกรม MATLAB และ Simulink ที่ได้มีการศึกษาและนำไปประยุกต์ใช้ในวิชา Industrial Training in Summer ไปบ้างแล้ว
3. ศึกษาข้อมูลและทฤษฎีของ Image Processing
4. ศึกษาภาษา PYTHON, C, C++, C# และอื่นๆเพิ่มเติม
5. เข้าใจการเชื่อมต่อแบบ Serial

1.3 ขอบเขตการวิจัยและขั้นตอนการศึกษา

จากวัตถุประสงค์ของโครงงานนี้ต้องการศึกษาการทำงานของการประมวลผลภาพ (Image Processing) ในการนำไปตรวจจับการเคลื่อนไหววัตถุที่มีการเคลื่อนไหว โดยจะใช้โปรแกรม MATLAB และ Simulink ที่นำภาษา C, C++ เป็นหลักในการดำเนินการเขียนโปรแกรมคำสั่งเพื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุมการจับตำแหน่งจากภาพที่กล้องเว็บแคมเก็บภาพและแสดงผลออกมาเป็นพิกเซลสีแดง ซึ่งเป็นสีที่มีองค์ประกอบที่เกี่ยวข้องกับความร้อนและสามารถนำไปใช้ในกล้อง Thermal ได้ เพื่อที่จะนำมาประมวลผลต่อในการควบคุมทิศทางของกล้องให้สอดคล้องกับสิ่งที่สนใจ ในที่นี้คือพิกเซลสีแดงที่สุดหรือจุดที่มีความร้อนมากที่สุด โดยให้กล้องนั้นจับภาพอยู่ในจุดกึ่งกลาง (Centroid)

1.4 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน

1. ความรู้ความเข้าใจเกี่ยวกับการประมวลผลภาพ (Image processing) โดยภาพรวมทั้งหมด
2. ความรู้ความเข้าใจในอุปกรณ์ทั้งหมด
3. ความเข้าใจคำสั่งที่ป้อนให้อุปกรณ์ทำงาน
4. มีความรู้ในการเขียนโค้ดในภาษาต่างๆเพื่อในไปต่อยอดในการใช้งานและการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

โครงงานชิ้นนี้ได้ทำการศึกษากระบวนการเขียนโปรแกรมคำสั่งในการสร้างอุปกรณ์อำนวยความสะดวกเพื่อใช้อุปกรณ์อิเล็กทรอนิกส์ (กล้องเว็บแคม) เพื่อการตรวจจับวัตถุโดยการใช้ทฤษฎีการประมวลผลภาพ (Image Processing) ในการประกอบการเข้าใจในการทำโครงงานชิ้นนี้ และยังใช้โปรแกรม MATLAB และ Simulink ในการวิเคราะห์เพื่อจะป้อนโปรแกรมคำสั่งลงไปให้เกิดการทำงานของกล้องเว็บแคมที่ถูกต้องเป็นไปตามผลลัพธ์ที่ต้องการ ในบทนี้จะกล่าวถึงทฤษฎีและหลักการที่เกี่ยวข้องกับโครงงานดังนี้

แนวคิดและความพยายามทางด้านการประมวลผลภาพได้เริ่มขึ้นเมื่อปี 1964 ณ ห้องแลป Jet Propulsion (Pasasena California) ซึ่งได้นำกระบวนการในการประมวลผลภาพ (Image Processing) มาใช้ในการพิจารณาภาพถ่ายดาวเทียมของดวงจันทร์ ต่อมาได้มีการตั้งสาขาทางวิทยาศาสตร์สาขาใหม่ที่มีชื่อว่า Digital Image Processing หลังจากนั้นงานด้านการประมวลผลภาพก็ได้มีการพัฒนามาอย่างต่อเนื่องและใช้กันอย่างกว้างขวางสำหรับงานหลายๆอย่างหลายๆด้าน ตัวอย่างเช่น ทางด้านการสื่อสารโทรคมนาคม การสื่อสารทางโทรทัศน์ ทางด้านการพิมพ์ ทางด้านกราฟฟิค ทางด้านการแพทย์ และการค้นคว้าทางวิทยาศาสตร์รวมไปถึงการพัฒนาทางด้านเทคโนโลยีดิจิตอล[2] โดยการวิเคราะห์ข้อมูลทางภาพนั้นจะเกี่ยวกับวิธีการอธิบายและการจดจำข้อมูลภาพดิจิตอลซึ่งอินพุทของระบบจะเป็นข้อมูลภาพดิจิตอลและเอาท์พุทจะเป็นเครื่องหมายที่ใช้แทนข้อมูลภาพดิจิตอลเหล่านั้น ในการวิเคราะห์ภาพนั้นมีอยู่หลายวิธีด้วยกัน ซึ่งนำมาจากการจำลองการทำงานของตามนุษย์ (Human vision) โดยการมองเห็นของ

รูปที่ 2.1 Human vision เปรียบเทียบกับ Computer vision

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มนุษย์นั้นนับว่าเป็นกระบวนการที่ซับซ้อนซึ่งเป็นหนึ่งในกระบวนการที่ซับซ้อนและลักษณะโดยทั่วๆไปในกระบวนการวิเคราะห์ภาพ (Digital Image Processing) ซึ่งจะมีความค่อนข้างซับซ้อน[3]

2.1 กล้องเว็บแคม OKER177

รูปที่ 2.2 กล้องเว็บแคม OKER177

กล้องเว็บแคม OKER177 เป็นโมดูลกล้องเว็บแคมความละเอียดสูงขนาด 2-16 ล้านพิกเซล (ขึ้นอยู่กับซอฟต์แวร์ที่ใช้) เป็นเวอร์ชันล่าสุดที่ปรับปรุงประสิทธิภาพและการเชื่อมต่อให้ใช้งานได้ง่ายขึ้นด้วย CMOS เซนเซอร์ความละเอียดสูง 2-16 ล้านพิกเซล มีขนาดเล็ก และทำการเชื่อมต่อได้ง่ายขึ้นโดยใช้พอร์ท USB เท่านั้น มาพร้อมกับไลบรารี่พร้อมใช้งาน[3] โดย OKER177 นั้นสามารถนำไปใช้งานได้ในหลายแพลทฟอร์ม เช่น Arduino, Raspberry PI, Maple, Chipkit, Beaglebone black[4]

2.1.1 คุณสมบัติเด่น

- ความละเอียดสูง 16 ล้านพิกเซล
- ความละเอียดปรับเปลี่ยนได้ตั้งแต่ 2-16 ล้านพิกเซลขึ้นอยู่กับซอฟต์แวร์ที่ใช้
- ใช้เลนส์แก้ว ให้ความคมชัดของภาพระดับ Hi-End
- เฟรมเรทอัตราการแสดงภาพเคลื่อนไหว = 60 fps แสดงผลลัพธ์แบบวิดิโอที่ความละเอียดสูง
- มีระบบปรับความสมดุลของภาพเพื่อให้ได้สีที่เป็นธรรมชาติที่สุด
- ตัวโมดูลทำขึ้นด้วยโลหะ มีความทนทาน
- สามารถหมุนปรับโฟกัสได้ที่หน้าเลนส์กล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถปรับขาให้ใช้ได้กับจอภาพ (หนีบกับอุปกรณ์อื่น) หรือใช้ตั้งวางได้ (ดึงที่หนีบออกเพื่อใช้วางในแนวราบ)
- ปรับแสงได้เองโดยอัตโนมัติ ลดปัญหาเรื่องการขาดแสงในที่มืดและการย้อนแสง
- ใช้สำหรับอัดวิดีโอหรือถ่ายภาพนิ่งได้
- สามารถนำไปใช้ทำกล้องวงจรปิดได้ดี
- สามารถติดตั้งใช้งานกับโน้ตบุ้คหรือคอมพิวเตอร์ตั้งโต๊ะได้ทันที
- Info noise rate : 48dB
- Focus range : 30mm-infinitive
- ขนาดภาพความละเอียดสูงสุดระดับ HD 1600×1200 Pixel

2.2 การประมวลผลภาพ (Image Processing)

การประมวลผลภาพ (Image Processing) หมายถึง การนำภาพมาประมวลผลหรือคิดคำนวณด้วยคอมพิวเตอร์ เพื่อให้ได้ข้อมูลที่เราต้องการทั้งในเชิงคุณภาพและเชิงปริมาณ

รูปที่ 2.3 Block diagram การ Image Processing

โดยมีขั้นตอนต่างๆที่สำคัญ คือ การทำให้ภาพมีความคมชัดขึ้น การกำจัดสัญญาณรบกวนออกจากภาพ การแบ่งส่วนของวัตถุที่เราสนใจออกมาจากภาพ เพื่อนำภาพวัตถุที่ได้ไปวิเคราะห์หาข้อมูลเชิงปริมาณ เช่น ขนาด รูปร่าง และทิศทางการเคลื่อนไหวของวัตถุในภาพ จากนั้นเราสามารถนำข้อมูลเชิงปริมาณเหล่านี้ไปวิเคราะห์และสร้างเป็นระบบ[5] เพื่อใช้ประโยชน์ในงานด้านต่างๆ เช่น ระบบจดจำลายนิ้วมือเพื่อตรวจสอบว่าภาพลายนิ้วมือนั้นเป็นเอกลักษณ์ทางบุคคลของใคร ระบบตรวจสอบคุณภาพของผลิตภัณฑ์ในกระบวนการผลิตของโรงงานอุตสาหกรรม ระบบคัดแยกเกรดหรือคุณภาพของพืชผลทางการเกษตร ระบบอ่านรหัสไปรษณีย์ที่อยู่บนซอง ระบบเก็บข้อมูลรถที่เข้าและออกอาคารโดยใช้ภาพถ่ายของป้ายทะเบียนรถเพื่อประโยชน์ในด้านความปลอดภัย ระบบดูแลและตรวจสอบสภาพการจราจรบนท้องถนนโดยการนับจำนวนรถบนท้องถนนในภาพถ่ายด้วยกล้องวงจรปิดในแต่ละช่วงเวลา และระบบจดจำใบหน้าเพื่อเฝ้าระวังผู้ก่อการร้ายในอาคารสถานที่สำคัญๆหรือในเขตคนเข้าเมือง เป็นต้น[6] จะเห็นได้ว่าระบบเหล่านี้จำเป็นต้องมีการประมวลผลภาพเป็นจำนวนมากและเป็นกระบวนการที่ต้องทำซ้ำๆกันในรูปแบบเดิมเป็นส่วนใหญ่ ซึ่งงานในลักษณะเหล่านี้หากให้มนุษย์เป็นผู้วิเคราะห์เอง มักจะต้องใช้เวลามาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

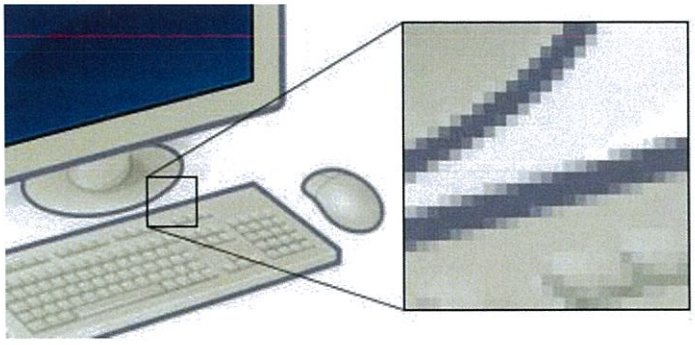
และใช้แรงงานสูง อีกทั้งถ้าหากจำเป็นจะต้องวิเคราะห์รูปภาพเป็นจำนวนมาก ผู้วิเคราะห์ภาพเองอาจเกิดอาการล้า ส่งผลให้เกิดความผิดพลาดขึ้นได้ ดังนั้นคอมพิวเตอร์จึงมีบทบาทสำคัญในการทำหน้าที่เหล่านี้แทนมนุษย์[7] อีกทั้งเป็นที่ทราบกันโดยทั่วไปว่า คอมพิวเตอร์มีความสามารถในการคำนวณและประมวลผลข้อมูลจำนวนมหาศาลได้ในเวลาอันสั้น ซึ่งมีประโยชน์อย่างมากในการเพิ่มประสิทธิภาพการประมวลผลภาพและวิเคราะห์ข้อมูลที่ได้จากภาพในระบบต่างๆดังกล่าวข้างต้น[5]

รูปที่ 2.4 การสแกนนิ้วมือจากการประมวลผลภาพ

2.2.1 ภาพและความหมายของพิกเซล

พิกเซลมาจากคำว่า ภาพ (Picture) กับคำว่า พื้นฐาน (Element) คือ หน่วยพื้นฐานซึ่งเล็กที่สุดของภาพดิจิตอล เทียบได้กับจุดสีของภาพ 1 จุด หลากหลายสีหลายๆจุดที่เรียงชิดติดกันถูกรวมกันทำให้เกิดเป็นภาพนั่นเอง แต่ 1 พิกเซลจะเป็นสีใดเพียงสีหนึ่งเท่านั้นจะมีสีอื่นไม่ได้ เนื่องจากว่าเป็นส่วนที่เล็กที่สุดของการแสดงผลพิกเซลนั้นมีความสำคัญต่อการสร้างกราฟฟิคของคอมพิวเตอร์มาก เพราะทุกๆส่วนของกราฟฟิค เช่น จุด เส้น แบบลายและสีของภาพ ล้วนเกิดจากพิกเซลทั้งสิ้น[9] ตัวอย่างกล้องถ่ายรูปความละเอียด 5 ล้านพิกเซล นั้นหมายความว่าเมื่อถ่ายภาพที่ความละเอียดสูงสุดที่กล้องสามารถจะถ่ายได้นั้นจะได้เม็ดพิกเซลออกมาทั้งหมด 5 ล้านเม็ด ดังนั้นจึงสรุปได้อย่างง่ายๆว่า ค่าพิกเซลยิ่งมีค่าสูงมากขึ้นเท่าไรภาพที่ได้ก็ยิ่งมีความละเอียดสูงมากไปด้วย[8]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



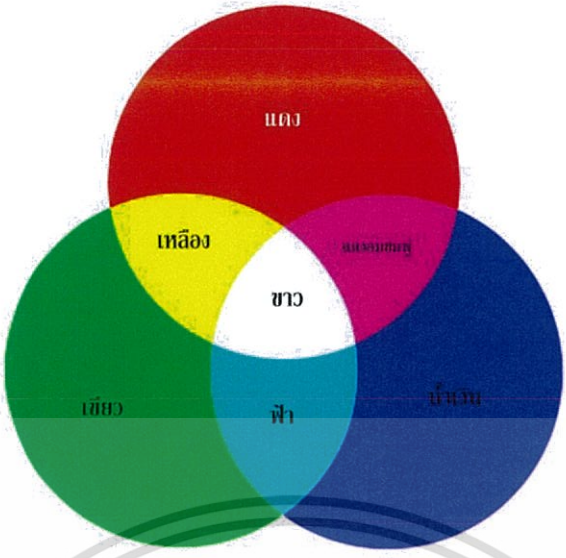
รูปที่ 2.5 พิกเซลภายในรูป

นอกเหนือจากหน่วยของการแสดงที่เป็นพิกเซลยังมีคำว่า bpp ย่อมากจากคำว่า Bits per pixel คือค่าของจำนวนสีที่สามารถจะแสดงได้ใน 1 พิกเซล ดังที่บอกไปในข้างต้นแล้วว่า 1 พิกเซลจะสามารถแสดงผลได้เพียงแค่สีเดียวเท่านั้น[8] ดังนั้นค่า bpp จะเป็นตัวที่บ่งบอกว่าภายใน 1 พิกเซลนั้นจะสามารถเปลี่ยนสีได้กี่สี ซึ่งมีตั้งแต่ 1-32bpp โดยปัจจุบันที่เราใช้ในการแสดงผลในคอมพิวเตอร์นั้นจะอยู่ที่ 32bpp[9]

2.2.2 ระบบสีแบบ RGB

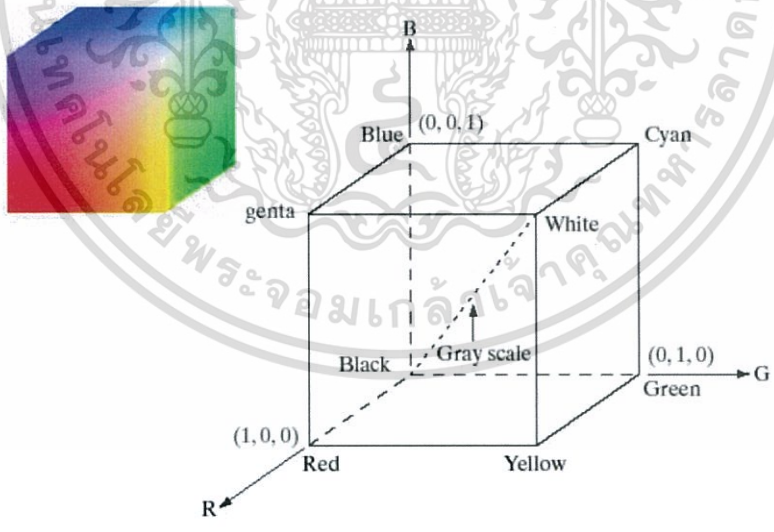
ระบบสีแบบ RGB คำว่า RGB นั้นย่อมาจาก Red, Green, Blue นั่นก็คือระบบสีของแสง ซึ่งเกิดจากการหักเหของแสงกลายเป็นสีรุ้งด้วยกัน 7 สี ซึ่งเป็นช่วงความยาวคลื่นแสงที่ตาคนเราสามารถจะมองเห็นได้ แสงสีม่วงจะมีความถี่ที่สูงที่สุดเรียกว่า อัลตร้าไวโอเลต (Ultraviolet) และแสงสีแดงจะมีความต่ำที่สุดเรียกว่า อินฟราเรด (Infrared)[10] คลื่นแสงที่มีความถี่สูงกว่าสีม่วงและต่ำกว่าสีแดงนั้นสายตาของมนุษย์ไม่สามารถมองเห็นหรือรับได้ แสงสีทั้งหมดจึงเกิดจากแสงสี 3 สี คือ สีแดง (Red) สีน้ำเงิน (Blue) และสีเขียว (Green) ทั้ง 3 สีถือเป็นแม่สีของแสง ซึ่งเมื่อแต่ละแม่สีมารวมกันนั้นก็จะได้สีอื่นๆ ต่างกันดังนี้ สีแดงกับสีเขียว = สีเหลือง (Yellow) สีเขียวกับสีน้ำเงิน = สีฟ้า (Cyan) และสีแดงกับสีน้ำเงิน = สีแดงอมชมพู (Magenta)[11] เมื่อนำแม่สีของทั้ง 3 สีมาผสมกันในปริมาณแสงสว่างที่เท่ากันก็จะได้แสงสีขาวออกมา แต่ถ้าผสมกันระหว่างแสงที่มีระดับความสว่างต่างกันก็จะได้ผลออกมาเป็นแสงสีที่มีมากมายหลายล้านสี ส่วนใหญ่การใช้สีลักษณะนี้จะใช้ในอุปกรณ์ที่เกี่ยวกับแสง เช่น จอภาพ สแกนเนอร์ และกล้องดิจิตอลหรือกล้องวิดีโอ เป็นต้น[10] ซึ่งระบบสีแบบ RGB นั้นเหมาะกับการใช้ในการออกแบบต่างๆ ที่แสดงผลบนหน้าจอเพราะจะมีความสวยงามใกล้เคียงกับสีที่ตาเรามองเห็นปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 ระบบวงจรสีของแสงในระบบ RGB

โมเดลระบบสี RGB สามารถแทนได้ด้วยแบบจำลองลูกบาศก์ ค่าของสีแทนด้วยค่าพิกัดในสเปซ RGB ซึ่งแกน X แทนแสงสีแดง แกน Y แทนแสงสีน้ำเงิน และแกน Z แทนแสงสีเขียว โดยในการใช้อุปกรณ์ดิจิตอลนั้นจะใช้ระบบนี้เป็นหลักเพราะว่าสีเหล่านี้คือสีพื้นฐานกายภาพ อุปกรณ์ดิจิตอลที่ใช้ เช่น รูปถ่ายและวีดิโอจากกล้องดิจิตอลหรือโทรศัพท์มือถือ[10]

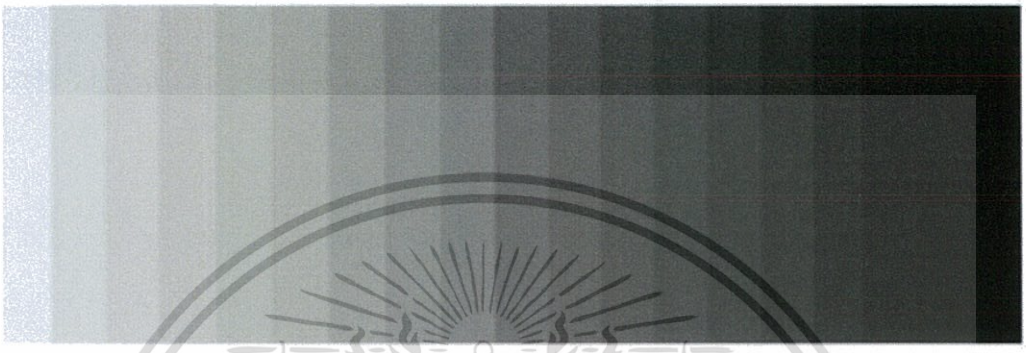


รูปที่ 2.7 ไดอะแกรมของลูกบาศก์ในระบบ RGB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 การเปลี่ยนภาพจากระบบ RGB ไปเป็นระบบ Grayscale

ภาพระดับสีเทา (Grayscale Image) เป็นภาพซึ่งค่าในแต่ละจุดภาพคือค่าความเข้มของสีแต่ละตำแหน่งของจุดภาพนั้น ซึ่งค่าที่เป็นไปได้ของภาพระดับสีเทาทั้งหมดขึ้นอยู่กับจำนวนบิตที่ใช้ ตัวอย่างเช่น ภาพระดับสีเทา 8 บิตที่ระดับสีทั้งหมด 256 ระดับ โดยนิยมระบุในช่วง 0-1 หรือ 0-255 แสดงระดับสีเทาดังในรูปที่ 2.8[14]



รูปที่ 2.8 ภาพแสดงระดับสีเทา

ภาพระดับสีเทาจะมีการไล่ระดับความอ่อนแก่ของสีซึ่งอยู่ระหว่างสีขาวและสีดำ Halftone Image อย่างต่อเนื่อง โดยที่ค่าแต่ละพิกเซลของภาพจะหมายถึงความเข้มแสงแต่ละตำแหน่งของพิกเซลที่อยู่ในรูประดับสีเทา การเปลี่ยนภาพจากระบบสี RGB เป็น ระดับสีเทา)Gray Scaleจะใช้สมการในการ (เปลี่ยนดังนี้[15]

$$Y = 0.3R + 0.59G + 0.11B$$

โดย Y แทนค่าระดับสีเทา ณ จุดพิกเซลที่เราต้องการหา

R แทนค่าสีแดง ณ จุดที่ต้องการหา

G แทนค่าสีเขียว ณ จุดที่ต้องการหา

B แทนค่าสีน้ำเงิน ณ จุดที่ต้องการหา

เพราะฉะนั้น Weight ที่เราออกมาได้นั้นจะอยู่ที่ R33%, G33%, B33%[12] โดยอาจจะมีปัญหาเกิดขึ้นจากความยาวคลื่นของแสงแต่ละสีนั้นมีค่าไม่เท่ากัน แต่สามารถแก้ได้โดยการ Weight ระดับของแสงใหม่ ตัวอย่างที่นำมาให้ชมจะเป็นตัวอย่างที่ Weight ระดับแสงมาเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.9 การแปลงภาพระบบ RGB เป็นภาพระบบ Grayscale

ภาพที่รับเข้ามาในขั้นตอนแรกเป็นภาพที่อยู่ในระบบปริภูมิสีแบบ RGB ดังนั้นแต่ละพิกัดของภาพจะประด้วยค่าของเซตที่แสดงถึงค่าของ R ค่าของ G และค่าของ B ระบบทำการเปลี่ยนให้เป็นภาพระดับสีเทา (Grayscale) เพื่อทำให้สามารถวิเคราะห์ภาพได้ง่ายขึ้นเพราะเมื่อทำการแปลงภาพเป็นระดับสีเทาแล้วจะทำให้แต่ละจุดของภาพเหลือเพียงแค่ค่าความเข้มของสีที่มีค่าตั้งแต่ 0 ถึง 255 เมื่อแปลงจากภาพสีเป็นภาพระดับสีเทา[13]

2.2.4 การแปลงภาพระดับสีเทาให้เป็นภาพขาว-ดำ

หมายถึง ภาพที่ประกอบด้วยสีขาวและสีดำเป็นหลัก ในทางดิจิทัลหมายความว่ามีเพียง 2 สถานะ คือ 0 และ 1 ถ้าพิกเซลใดมีค่าเป็น 0 หมายความว่าพิกเซลนั้นมีสีดำ พิกเซลใดมีค่าเป็น 1 หมายความว่าพิกเซลนั้นมีสีขาว การแปลงภาพระดับสีเทาให้เป็นภาพขาว-ดำ ต้องกำหนดค่าความเข้มที่ต้องการอ้างอิงหรือค่าขีดแบ่ง (Threshold Value) โดยผู้ใช้สามารถใช้อัลกอริธึมในการหาค่าหรือกำหนดได้เอง จาก

$(B)m, n = 1$ ถ้า $(f)m, n \geq T$

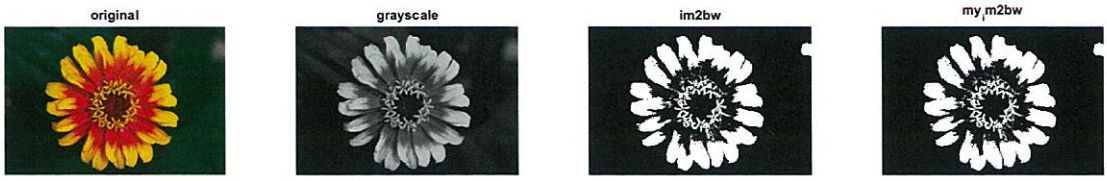
$= 0$ ถ้า $(f)m, n < T$

โดยที่ m คือพิกเซลในแนวแกน Y

n คือพิกเซลในแนวแกน X

T คือค่าขีดแบ่ง (Threshold Value) ที่เรากำหนดไว้[12]

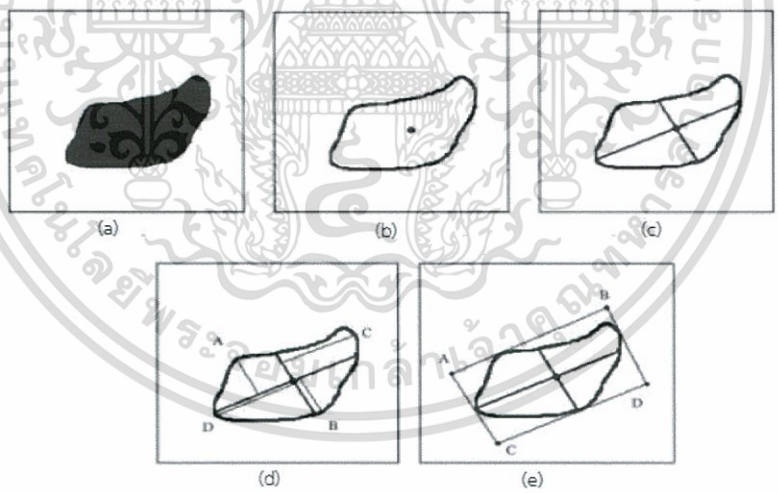
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 การแปลงภาพ RGB เป็น Grayscale และเป็น B&W

2.2.5 กระบวนการปิดล้อมวัตถุ (Bounding box)

คือกล่องสี่เหลี่ยมสำหรับปิดล้อมรอบบริเวณที่เราสนใจ ซึ่งกล่องสี่เหลี่ยมนี้จะช่วยในการคำนวณจุดศูนย์กลาง พื้นที่ และความยาวแกนเอก – แกนโทของบริเวณที่เราสนใจ ตัวอย่างเช่น จากรูปที่ 2.5(a) คือภาพวัตถุสีขาว – ดำ ส่วนในรูป 2.5(b) คือขอบและจุดศูนย์กลางของวัตถุ รูปที่ 2.5(c) คือการหาแกนเอกและแกนโทของวัตถุ รูปที่ 2.5(d) เป็นการหาจุดที่มีระยะห่างจากจุดศูนย์กลางวัตถุถึงขอบวัตถุที่ยาวที่สุดโดยพิจารณาจากส่วนด้านบนของแกนเอกและส่วนล่างของแกนเอก (A และ B) รวมทั้งการหาจุดที่มีระยะห่างจากจุดศูนย์กลางวัตถุถึงขอบวัตถุที่ยาวที่สุดโดยพิจารณาจากบริเวณส่วนซ้ายของแกนโทและส่วนขวาของแกนโท (C และ D) สุดท้ายจะได้รูป 2.5(e) ที่ได้พิกัดในการสร้างกล่องสี่เหลี่ยมในการปิดล้อมวัตถุ[13]



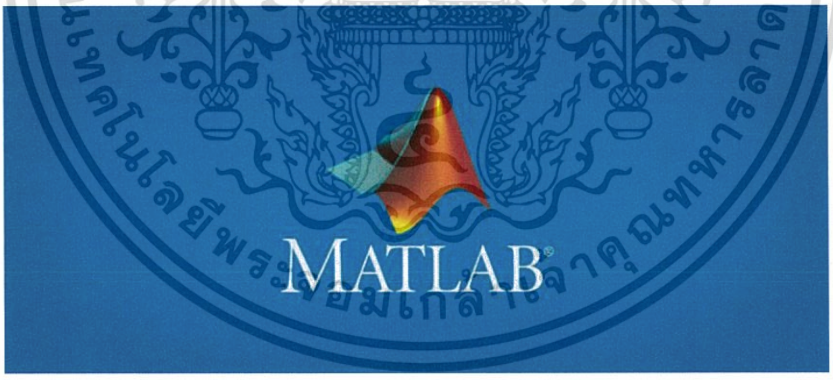
รูปที่ 2.11 การสร้างสี่เหลี่ยมปิดล้อมวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 โปรแกรม MATLAB

Matlab เป็น ภาษาคอมพิวเตอร์ระดับสูงที่มาพร้อมด้วยสภาพแวดล้อมการทำงานเชิงโต้ตอบ (คล้ายเครื่องคิดเลข) ซึ่งสามารถคำนวณคณิตศาสตร์ที่ซับซ้อนได้อย่างรวดเร็วมากกว่าภาษาคอมพิวเตอร์สมัยก่อน เช่น ภาษา C, C++ หรือ Fortran[16]

Matlab เป็นภาษาคอมพิวเตอร์ระดับสูงที่ใช้สำหรับคำนวณเชิงตัวเลข (Numerical Computing) แสดงผลกราฟฟิกและเขียนแอพพลิเคชั่น ทำให้เราสามารถคำนวณผลลัพธ์ พัฒนาอัลกอลิทึ่ม สร้างแบบจำลอง และแอพพลิเคชั่นได้ง่ายและรวดเร็วมาก ภายในตัว Matlab ประกอบด้วยภาษาคอมพิวเตอร์ ทูลบ็อกซ์ (Toolbox : กลุ่มฟังก์ชันสำเร็จรูปในแต่ละสาขาวิชา) และฟังก์ชันพื้นฐานจำนวนมาก ทำให้การวิเคราะห์ทำได้หลากหลายวิธี พร้อมกับคำตอบที่รวดเร็วกว่าโปรแกรมตารางคำนวณ (Spreadsheet) หรือภาษาคอมพิวเตอร์สมัยก่อน เช่น C, C++, Fortran, Java และอื่นๆ เราสามารถนำ Matlab ไปประยุกต์ใช้งานได้หลายสาขามาก ทั้ง การประมวลผลสัญญาณ (Signal Processing) การสื่อสาร (Communication) การประมวลผลภาพและวิดีโอ (Image and Video Processing) ระบบควมคุม (Control System) การวัดและควบคุม (Instruments and Control) การคำนวณทางเศรษฐศาสตร์ (Economic) การคำนวณทางชีววิทยา (Biology) และอื่นๆ มีนักวิทยาศาสตร์และวิศวกรหลายล้านคนทั้งในแวดวงการศึกษาและอุตสาหกรรมที่ใช้ Matlab ในการคำนวณเชิงตัวเลข [17]



รูปที่ 2.12 โปรแกรม MATLAB

2.3.1 การรับอินพุต

```
obj = videoinput(adaptorname,deviceID,format)
```

เป็นการรับค่า video input มาเก็บไว้ในตัวแปรที่ชื่อ obj โดยที่ adaptorname คือ ชื่อของอุปกรณ์ที่ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

deviceID คือ ตัวเลขที่ระบุตำแน่งของอุปกรณ์ที่เชื่อมต่อ
format คือ ค่าที่ระบุรูปแบบการกำหนดพิกเซลของภาพ[12]

2.3.2 การลบรูปภาพ

```
Z = imsubtract(X,Y)
```

เป็นการลบองค์ประกอบของอาเรย์ Y ออกจากส่วนของอาเรย์ X และได้ค่าความแตกต่างเป็นผลลัพธ์ในตัวแปร Z โดยที่ X และ Y ต้องเป็นตัวแปรที่เป็นชนิดและขนาดเดียวกัน[11] เช่นถ้าหาก X เป็นรูปภาพที่มีขนาด 320 x 240 Y ก็จะต้องเป็นรูปภาพขนาด 320 x 240 ด้วย[14]

2.3.3 การแปลงภาพ RGB เป็นภาพ Grayscale

```
I = rgb2gray(RGB)
```

เป็นการแปลงภาพสีของตัวแปรที่ชื่อว่า RGB เป็นภาพเทา (Grayscale)[11] ในตัวแปรที่ชื่อ I โดยมีอัลกอริทึ่มที่สามารถคำนวณแปลงค่าคือ

```
0.2989 * R + 0.5870 * G + 0.1140 * B
```

โดย R คือค่าส่วนประกอบของสีแดง G คือค่าส่วนประกอบของสีเขียว และ B คือค่าส่วนประกอบของสีน้ำเงิน[14]

2.3.4 การแปลงภาพ Grayscale เป็นภาพ Binary

```
BW = im2bw(I, level)
```

เป็นการแปลงภาพสีเทาของตัวแปรที่ชื่อว่า I เป็นภาพขาว - ดำ (Binary Image)[11] ในตัวแปรที่ชื่อ BW โดยค่า level คือค่าที่กำหนดขีด (Threshold) ของรูปภาพ ซึ่งมีค่าอยู่ที่ช่วง 0 – 1 หากพิกเซลใดในรูปภาพมีค่าต่ำกว่าค่า level ที่กำหนดจะถูกแทนที่เป็นสีดำ ส่วนพิกเซลที่มีค่าสูงกว่าจะถูกแทนที่เป็นสีขาว[13]

2.3.5 การลบวัตถุเล็กออกจากภาพขาว - ดำ

```
BW2 = bwareaopen(BW,P)
```

เป็นการกำจัดพิกเซลในรูปภาพที่ชื่อว่า BW2 ที่มีการเชื่อมต่อกันน้อยกว่าค่า P ที่กำหนดจากภาพในรูปแบบ Binary Image[17]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 โปรแกรม Arduino IDE

ซอฟต์แวร์ที่ใช้ในการพัฒนางานสำหรับบอร์ด Arduino นั่นคือโปรแกรมที่มีชื่อเรียกว่า Arduino IDE ในการเขียนโปรแกรมและคอมไพล์ลงบอร์ด คำว่า IDE นั้นย่อมาจากคำว่า (Integrated Development Environment) คือ ส่วนเสริมของระบบการพัฒนาหรือตัวช่วยต่างๆที่จะคอยช่วยเหลือนักพัฒนา (Developer) หรือช่วยเหลือคนพัฒนา Application เพื่อเสริมให้เกิดความรวดเร็วและถูกต้องแม่นยำ ตรวจสอบระบบที่จัดทำได้ ทำให้การพัฒนางานต่างๆมีความรวดเร็วมากยิ่งขึ้น[20]

รูปที่ 2.13 โปรแกรม Arduino IDE

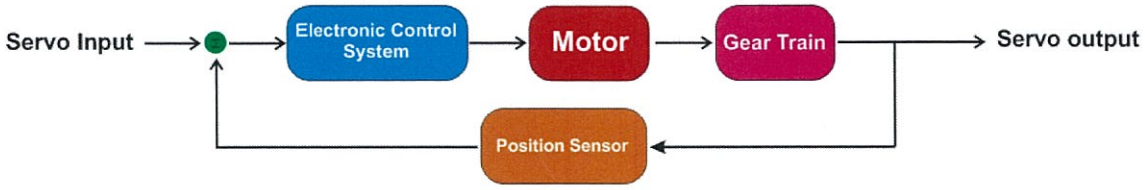
2.5 Servo Motor

Servo เป็นคำศัพท์ที่ใช้กันทั่วไปในระบบควบคุมอัตโนมัติ มาจากภาษาละตินคำว่า Sevus หมายถึง “ทาส” (Slave) ในเชิงความหมายของ Servo Motor ก็คือ Motor ที่เราสามารถสั่งงานหรือตั้งค่า แล้วตัว Motor จะหมุนไปยังตำแหน่งองศาที่เราสั่งได้เองอย่างถูกต้อง โดยใช้การควบคุมแบบป้อนกลับ (Feedback Control) ในบทความนี้จะกล่าวถึง RC Servo Motor ซึ่งนิยมนำมาใช้ในเครื่องเล่นที่บังคับด้วยคลื่นวิทยุ (RC = Radio Controlled) เช่น เรือบังคับวิทยุ รถบังคับวิทยุ เฮลิคอปเตอร์บังคับวิทยุ เป็นต้น

Feedback Control คือ ระบบควบคุมที่มีการวัดค่าเอาต์พุตของระบบนำมาเปรียบเทียบกับค่าอินพุตเพื่อควบคุมและปรับแต่งให้ค่าเอาต์พุตของระบบให้มีค่า เท่ากับ หรือ ใกล้เคียงกับค่าอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1 หลักการทำงานของ Servo Motor



รูปที่ 2.14 Block Diagram การทำงานของ Servo Motor

เมื่อจ่ายสัญญาณพัลซ์เข้ามายัง RC Servo Motor ส่วนวงจรควบคุม (Electronic Control System) ภายใน Servo จะทำการอ่านและประมวลผลค่าความกว้างของสัญญาณพัลซ์ที่ส่งเข้ามาเพื่อแปลค่าเป็นตำแหน่งองศาที่ต้องการให้ Motor หมุนเคลื่อนที่ไปยังตำแหน่งนั้น แล้วส่งคำสั่งไปทำการควบคุมให้ Motor หมุนไปยังตำแหน่งที่ต้องการ โดยมี Position Sensor เป็นตัวเซ็นเซอร์คอยวัดค่ามุมที่ Motor กำลังหมุน เป็น Feedback กลับมาให้วงจรควบคุมเปรียบเทียบกับค่าอินพุตเพื่อควบคุมให้ได้ตำแหน่งที่ต้องการอย่างถูกต้องแม่นยำ

2.5.2 ควบคุม Servo Motor ด้วย Arduino

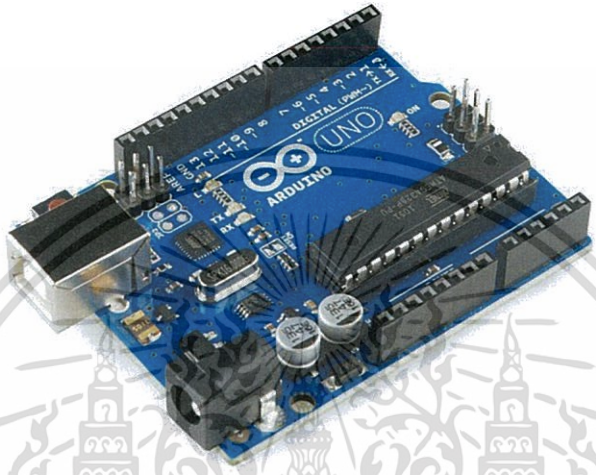
Arduino มีไลบรารี่สำหรับสั่งงาน RC Servo Motor มาให้ใช้งานอยู่แล้วเป็นฟังก์ชั่นสำเร็จรูปและใช้งานได้ง่ายในหน้าเว็บไซต์ http://arduino.cc/en/reference/servo ได้ให้ข้อมูลไว้ว่า Servo Library ของ Arduino สามารถสั่งงาน RC Servo Motor ได้ทั้งแบบหมุนไป-กลับได้ 0-180 องศา และแบบต่อเนื่องที่หมุนครบรอบได้เรียกว่าเป็น Continuous Rotation Servo โดยสามารถรองรับการเชื่อมต่อ Servo Motor ได้ถึง 12 ตัวกับบอร์ด Arduino UNO และรองรับสูงสุดถึง 48 ตัวหากใช้บอร์ด Arduino Mega[21]

รูปที่ 2.15 Servo Motor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 บอร์ด Arduino UNO R3

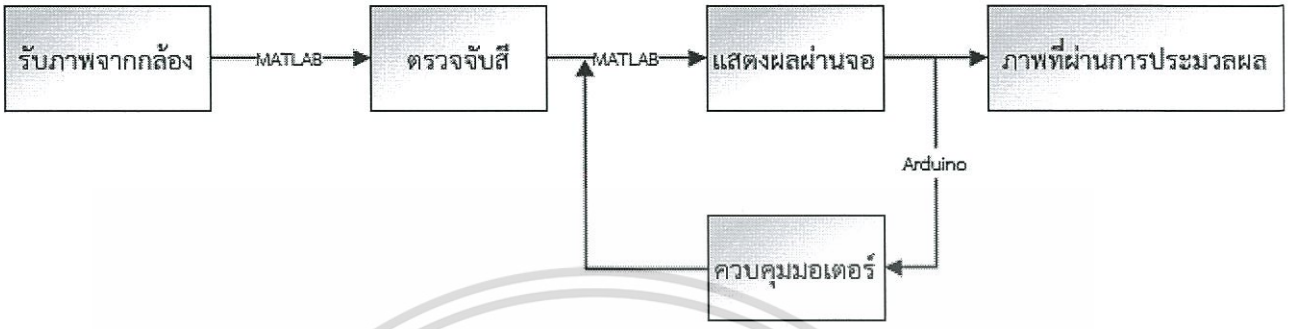
Arduino UNO R3 บอร์ดไมโครคอนโทรลเลอร์แบบ Open-source บนแพลตฟอร์ม Arduino ใช้ชิพ ATmega328 รันที่ความถี่ 16 MHz หน่วยความจำแฟลช 32 KB แรม 2 KB บอร์ดใช้ไฟเลี้ยง 7 ถึง 12 V มีระดับแรงดันไฟฟ้าในการทำงานและขาสัญญาณอยู่ที่ 5 V (TTL) มี Digital Input / Output 14 ขา (เป็น PWM ได้ 6 ขา) มี Analog Input 6 ขา Serial UART 1 ชุด I2C 1 ชุด SPI 1 ชุด[22]



รูปที่ 2.16 Arduino UNO R3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
กระบวนการสร้างและการทดลอง



Input คือ สัญญาณภาพจากโมดูลกล้อง

Output คือ ภาพที่ผ่านการประมวลผลจากกระบวนการขั้นต้นที่สามารถใช้ในการ Tracking ได้

แหล่งจ่ายไฟ คือ พอร์ท USB Drive ในตัว Computer หรือ Laptop ที่มีแรงดันไฟขนาด 5V.

ในบทนี้จะพูดถึงการสร้างผลลัพธ์ตามที่แบบแผนโครงงานที่กำหนดไว้คืออุปกรณ์การตรวจจับวัตถุที่มีความร้อนหรือมีอุณหภูมิภายใน โดยจะมีการเล็งเป้าไปที่วัตถุเพื่อระบุตำแหน่งที่ชัดเจนเพื่อการติดตามอย่างแม่นยำและเพื่อศึกษาค้นคว้าข้อมูลการเขียนโปรแกรม MATLAB และ Arduino ซึ่งมีความสำคัญมากในอนาคตอันใกล้

3.1 กระบวนการสร้างและการทดลอง

ก่อนอื่นเราจะต้องมีกระบวนการเตรียมทั้งในขั้นซอฟต์แวร์และฮาร์ดแวร์นำทั้งสองขั้นตอนนั้นมาประกอบกันจึงจะได้เป็นชิ้นงานที่มีรูปร่างสมบูรณ์ทั้งซอฟต์แวร์และฮาร์ดแวร์ โดยจะมีวิธีการแยกย่อยไปในแต่ละด้าน ดังที่จะกล่าวต่อไปนี้

3.1.1 อุปกรณ์ส่วนฮาร์ดแวร์

อุปกรณ์แรกที่เป็นหัวใจหลักที่ขาดไม่ได้เลยของโครงงานนี้ ขาดไปไม่ได้เลยสำหรับหัวใจในการรับภาพ นั่นก็คือ กล้องเว็บแคม ในที่นี้เราใช้เป็นยี่ห้อ OKER รุ่น 177 เป็นกล้องสารพัดประโยชน์ เป็นกล้องเว็บแคมที่มีความละเอียดสูงถึง 16 ล้านพิกเซล แต่ในที่นี้เพื่อการประมวลผลที่เร็วขึ้นเราจึงทำการลดความละเอียดลงเพื่อให้อุปกรณ์มีความเร็วในการประมวลผลมากขึ้น อุปกรณ์ต่อมาคือเซอร์โวมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อใช้ในการควบคุมตำแหน่งของวัตถุที่สนใจ และอีกอุปกรณ์ที่ขาดไม่ได้สำหรับการประมวลผลภาพนั่นคือ แลปท็อปหรือคอมพิวเตอร์ตั้งโต๊ะ เราใช้เพื่อโปรแกรมคำสั่งควบคุมการทำงานของสิ่งที่กล้องมองเห็น

โดยรวมอุปกรณ์ฮาร์ดแวร์มี 3 ชิ้น หลักๆคือ

1) กล้องเว็บแคม OKER177
2) แลปทอปหรือคอมพิวเตอร์
3) เซอร์โวมอเตอร์

รูปที่ 3.1 อุปกรณ์ฮาร์ดแวร์หลัก

3.1.2 ซอฟแวร์หลัก

ในการเริ่มต้นสร้างรากฐานระบบการทำงานขึ้นมาจะต้องเริ่มต้นด้วยการใช้งาน MATLAB โดยเราจะสร้างสภาพแวดล้อมในคอมพิวเตอร์ของเราให้มีความพร้อมที่จะเริ่มป้อนคำสั่งลงไปเพื่อที่จะนำไปใช้งานจริงในการล็อคเป้าหมายหรือตรวจจับวัตถุที่เราสนใจ และโปรแกรม Arduino ที่ใช้งานในการป้อนคำสั่งเพื่อควบคุมการหมุนของเซอร์โวมอเตอร์ เพื่อให้กล้องหันไปหาวัตถุที่สนใจ

โดยรวมอุปกรณ์ซอฟต์แวร์มี 2 ชิ้น หลักๆคือ

1) โปรแกรม MATLAB
2) โปรแกรม Arduino

โดยจะมีขั้นตอนในการเตรียมการโดยสังเขปดังนี้

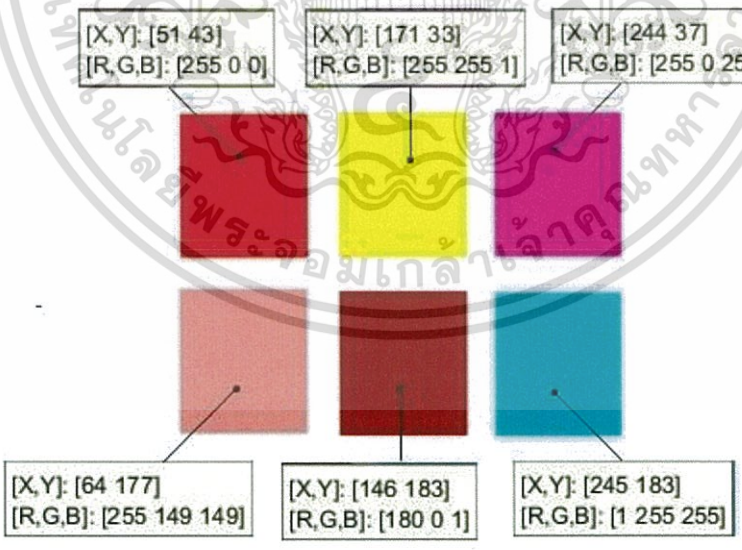
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 ขั้นตอนการเตรียมโปรแกรม

3.1.3.1 การตรวจจับวัตถุสีแดง

1) ทำการ detect รูปภาพโดยในที่นี้จะ detect สีแดงซึ่งเป็นสีที่แสดงถึงบริเวณที่มีอุณหภูมิสูง เช่น ความร้อนในร่างกาย ความร้อนจากเครื่องยนต์ เป็นต้น

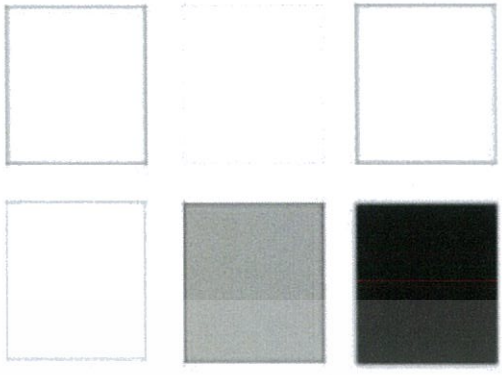
รูปที่ 3.2 ภาพสีที่นำมาเป็นตัวอย่างในการวิเคราะห์



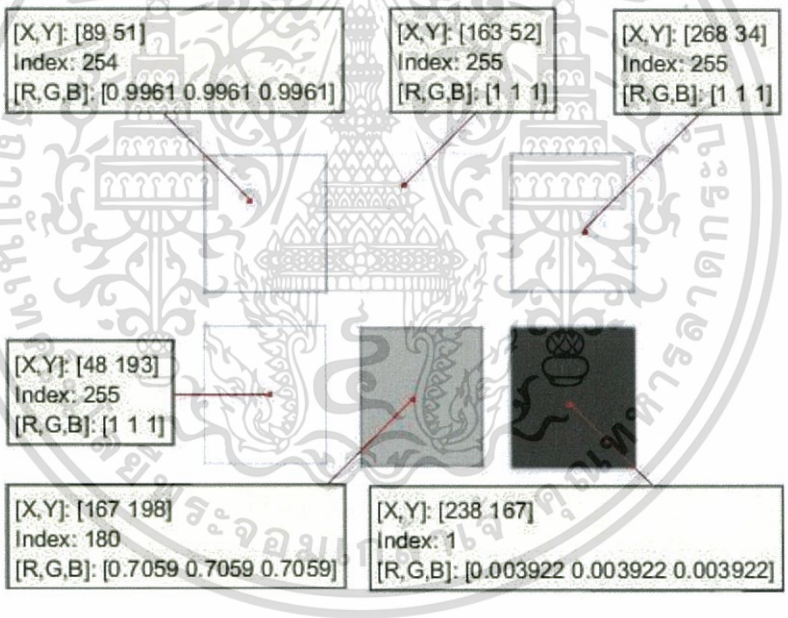
รูปที่ 3.3 ค่า Intensity values ของภาพสีที่นำมาเป็นตัวอย่างในการวิเคราะห์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ทำการแยกระนาบสีแดงออกจากรูป จะได้รูปที่แสดงถึงค่า red intensity values



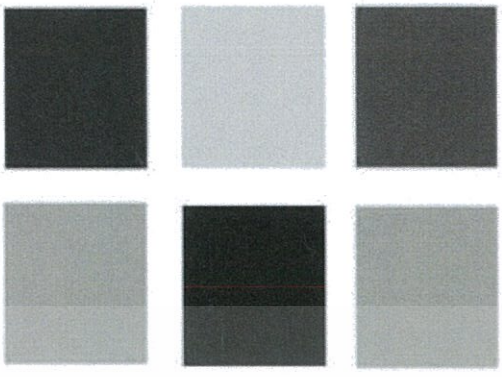
รูปที่ 3.4 ระนาบสีแดงที่ได้จากการแยก



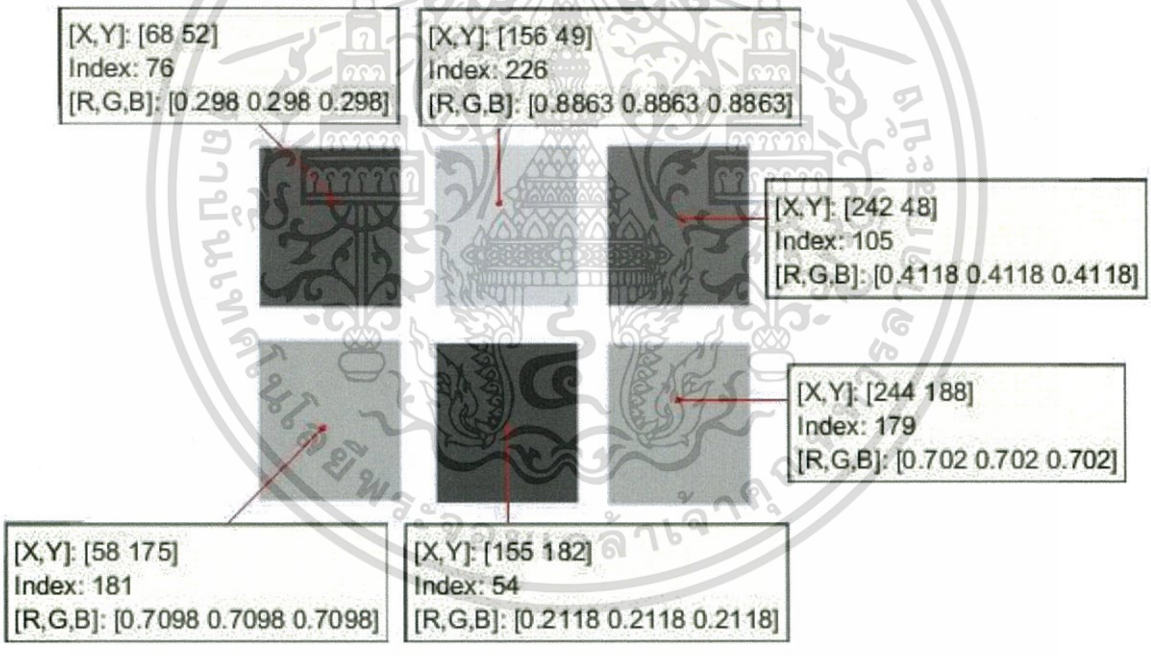
รูปที่ 3.5 ค่า Red intensity values ในระนาบของสีแดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ทำการแปลงรูปให้อยู่ในภาพเทา (Grayscale)



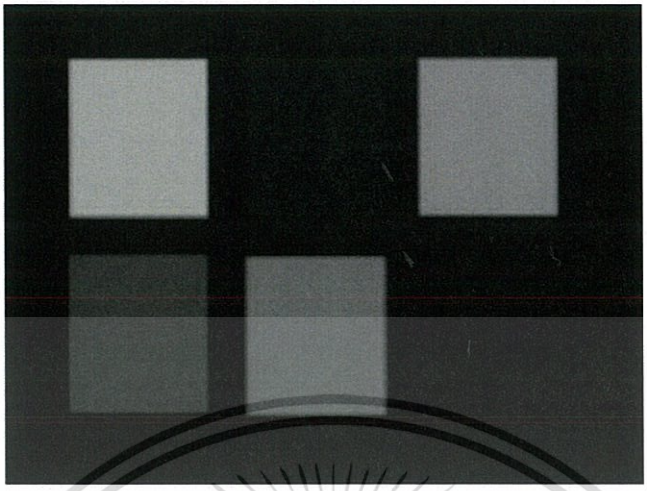
รูปที่ 3.6 ภาพ Grayscale



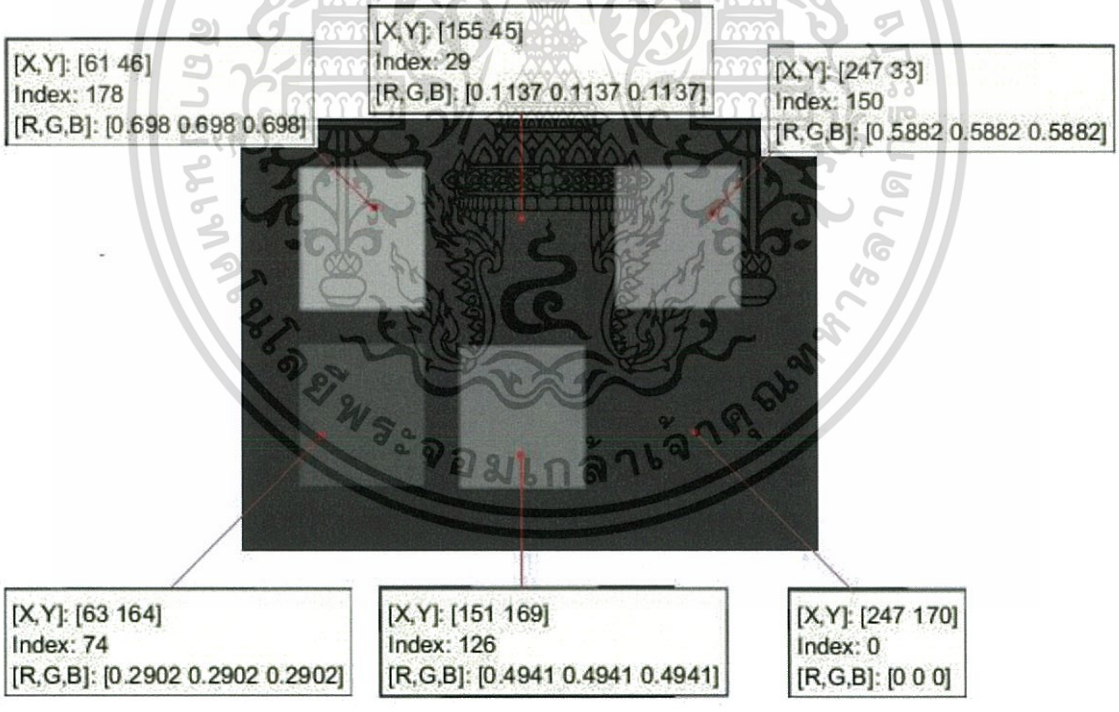
รูปที่ 3.7 ค่า Intensity values ของภาพ Grayscale

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ทำการลบภาพ Grayscale ออกจากภาพระนาบของสีแดง



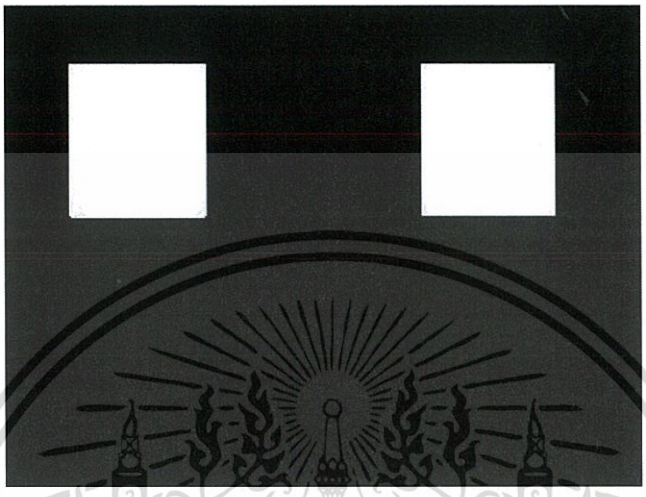
รูปที่ 3.8 ภาพเลเยอร์สีแดงที่ทำการลบภาพเทา



รูปที่ 3.9 ค่า Intensity values ของภาพเลเยอร์สีแดงที่ทำการลบภาพเทา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) ทำการแยกบริเวณรูปภาพโดยใช้ค่า Threshold กำหนด โดยดูจาก Intensity ของค่าพิกเซล กล่าวคือหากค่า Intensity ต่ำกว่าค่า Threshold ที่กำหนด พิกเซลนั้นจะถูกปรับให้เป็นสีดำ ส่วนพิกเซลที่มากกว่านั้นจะถูกปรับให้เป็นสีขาว ในที่นี้กำหนดค่า Threshold ให้มีค่าเท่ากับ 0.55



รูปที่ 3.10 ภาพขาวดำ

จะเห็นได้ว่ารูปที่มีค่า Intensity เท่ากับ 0.1137 ,0.2902 ,0.4941 และ 0 จะถูกปรับเป็นสีดำ ส่วนรูปที่มีค่า Intensity เท่ากับ 0.698 และ 0.5882 จะถูกปรับเป็นสีขาว

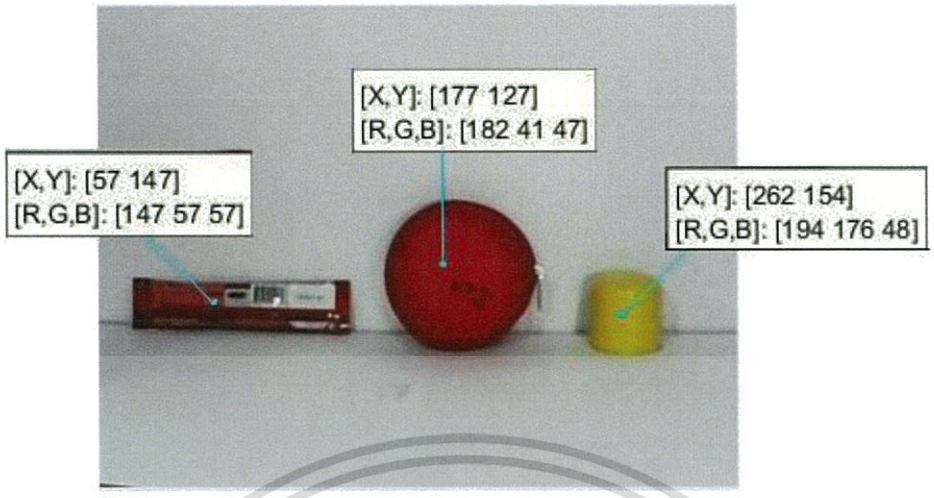
3.1.4 ขั้นตอนการใช้งานจริง

ทดลองใช้กล้อง webcam ในการบันทึกภาพแบบ real-time โดยทดลองกับวัตถุ 3 ชนิดที่มีค่าสีใกล้เคียงกัน โดยนำเข้ากล้อง webcam จากคำสั่ง

```
vid = videoinput('winvideo', 1, 'RGB24_320x240');

set(vid, 'FramesPerTrigger', Inf);
set(vid, 'ReturnedColorspace', 'rgb')
vid.FrameGrabInterval = 4;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



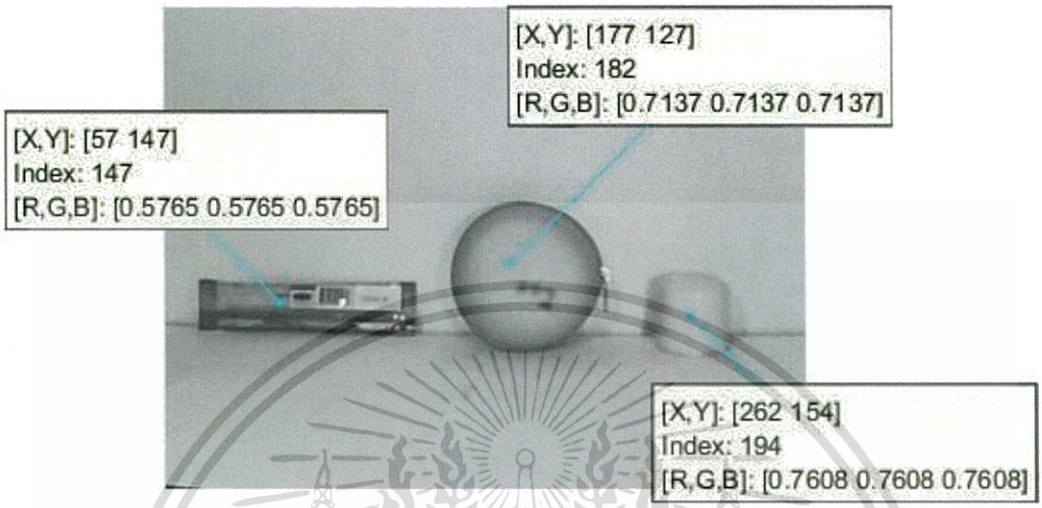
รูปที่ 3.11 ภาพวัตถุที่มีค่าสีแดงใกล้เคียงกัน

รูปที่ 3.12 กราฟฮิสโตแกรมของภาพวัตถุที่มีค่าสีแดงใกล้เคียงกัน

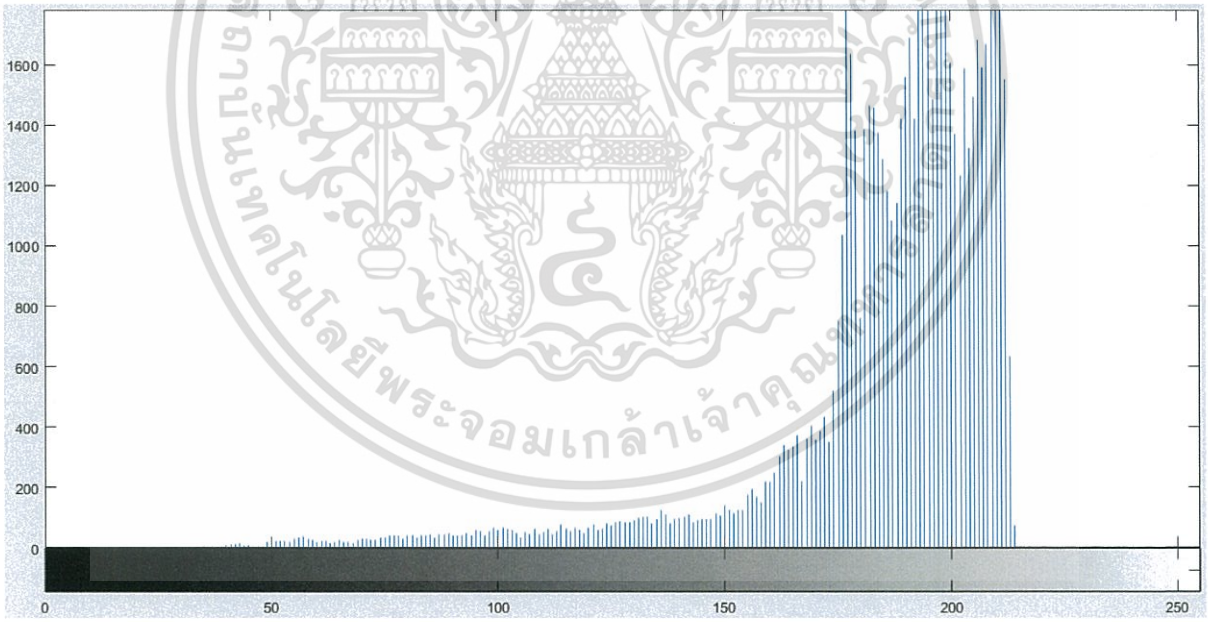
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการแยกเลเยอร์สีแดงออกจากรูป จะได้รูปที่แสดงถึงค่า red intensity values

```
data(:,:,1)
```



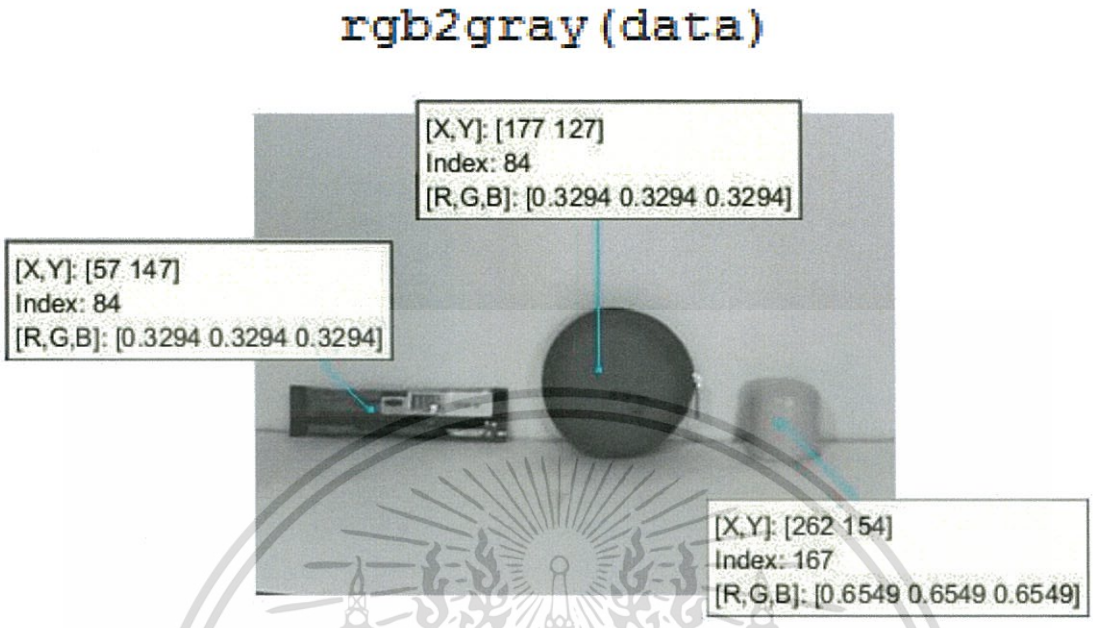
รูปที่ 3.13 ภาพเลเยอร์สีแดง



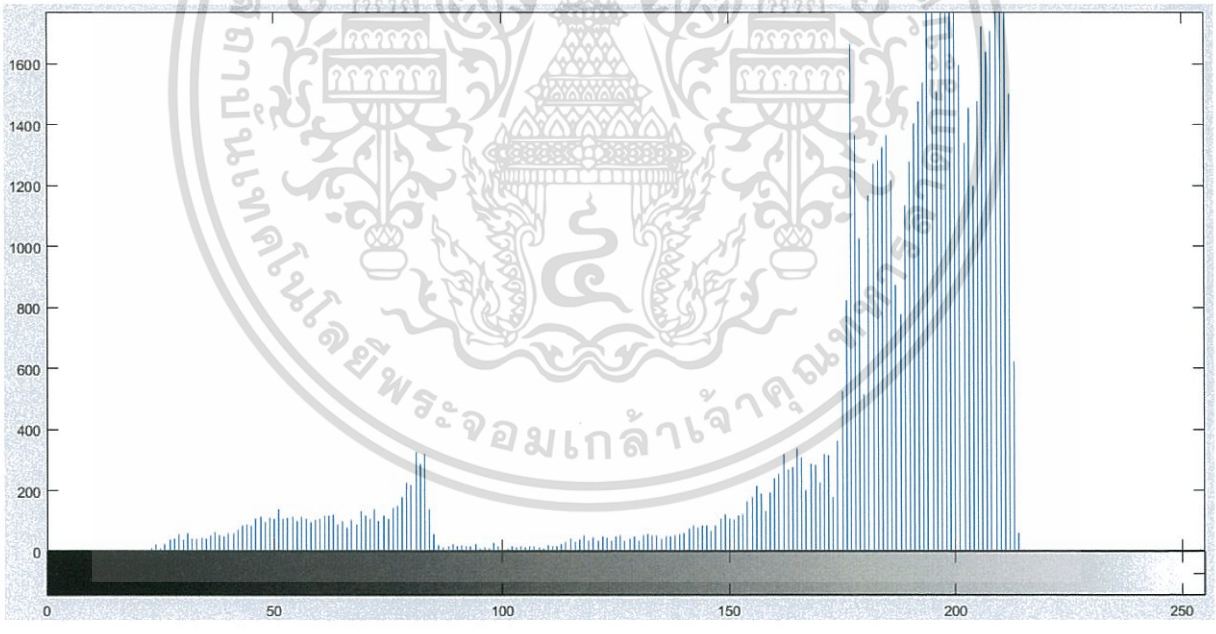
รูปที่ 3.14 กราฟฮิสโตแกรมของภาพเลเยอร์สีแดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการแปลงรูปให้อยู่ในภาพเทา (Grayscale) โดยใช้คำสั่ง rgb2gray



รูปที่ 3.15 ภาพเทา



รูปที่ 3.16 กราฟฮิสโตแกรมของภาพเทา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการลบภาพ Grayscale ออกจากภาพระนาบของสีแดง โดยใช้คำสั่ง imsubtract

```
redplane = imsubtract(data(:,:,1), rgb2gray(data));
```

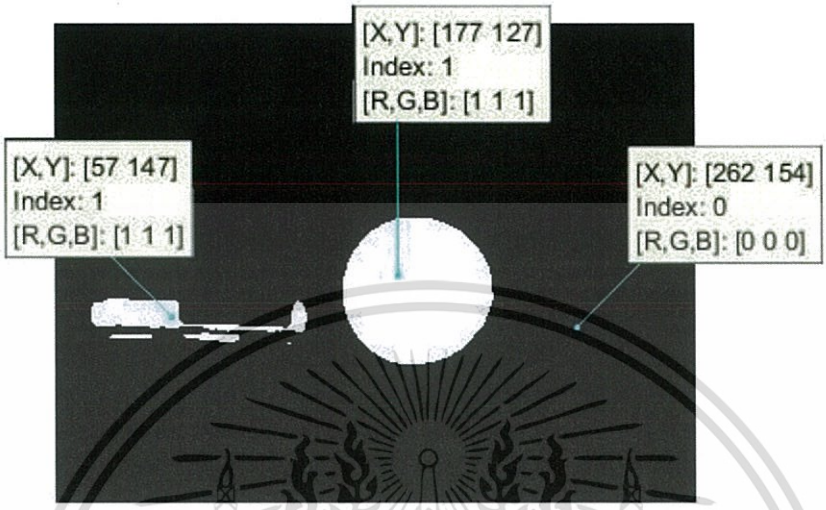
รูปที่ 3.17 ภาพเลเยอร์แดงที่ทำการลบภาพเทา

รูปที่ 3.18 กราฟฮิสโตแกรมของภาพเลเยอร์แดงที่ทำการลบภาพเทา

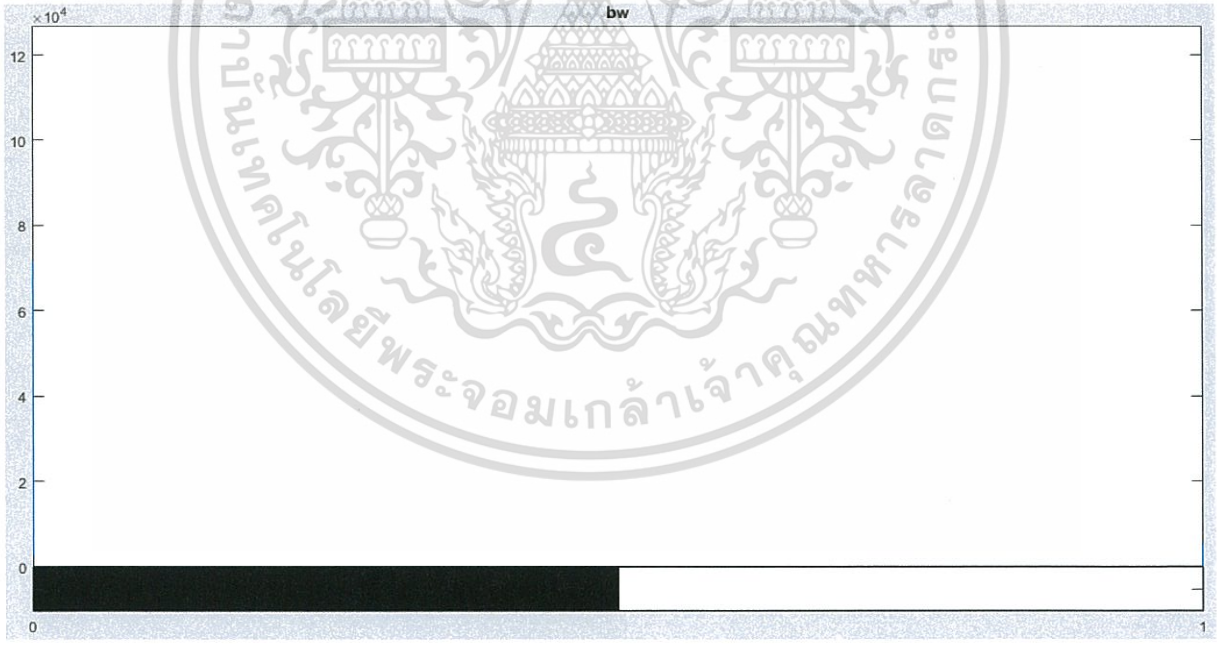
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการแยกบริเวณรูปภาพโดยใช้ค่า Threshold ในที่นี้กำหนดให้มีค่าเท่ากับ 0.14

```
redbw = im2bw(redfil,0.14);
```



รูปที่ 3.19 ภาพขาวดำ



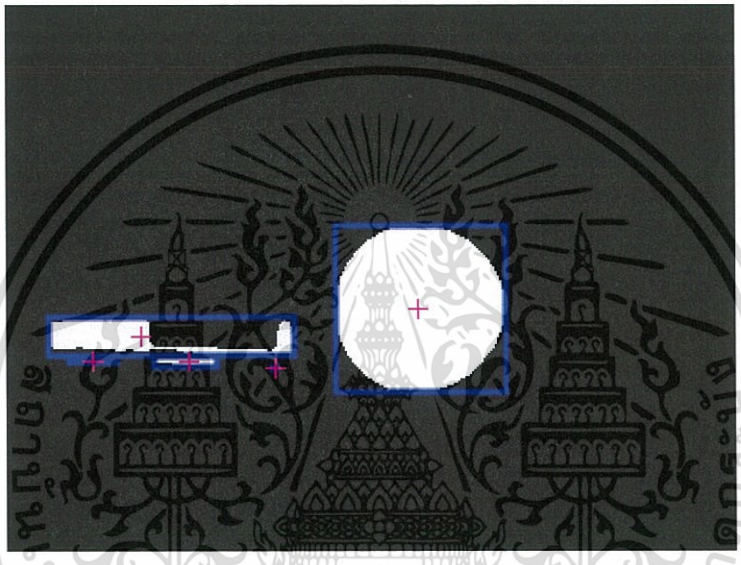
รูปที่ 3.20 กราฟฮิสโตแกรมของภาพขาวดำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการตีกรอบวัตถุโดยใช้คำสั่ง BoundingBox

```
stats = regionprops(bw, 'BoundingBox', 'Centroid');

for object = 1:length(stats)
    bb = stats(object).BoundingBox;
    bc = stats(object).Centroid;
    rectangle('Position',bb,'EdgeColor','r','LineWidth',2)
    plot(bc(1),bc(2), '-m+')
end
```



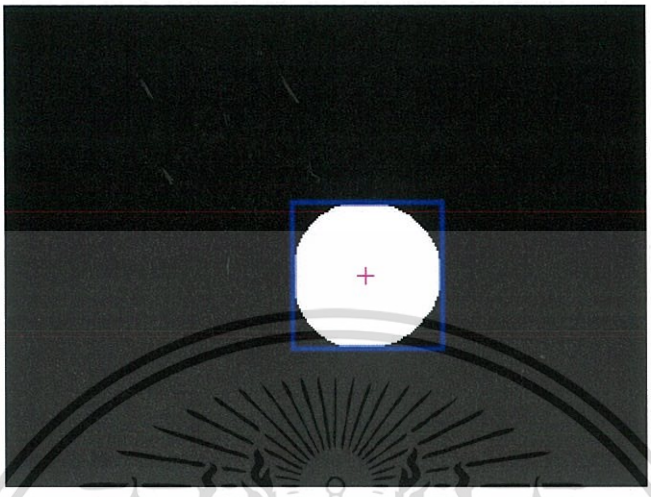
รูปที่ 3.21 ภาพขาวดำที่ทำการตีกรอบ

ในที่นี้เราสนใจวัตถุทรงกลมเท่านั้น กรณีที่ต้องการตัดวัตถุอื่นออกสามารถทำได้โดยการใช้คำสั่ง bwareaopen ในการเลือกวัตถุที่มีค่าพิกเซลสูงกว่าค่าที่กำหนด และจะทำการลบวัตถุที่มีค่าพิกเซลต่ำกว่าค่าที่กำหนด

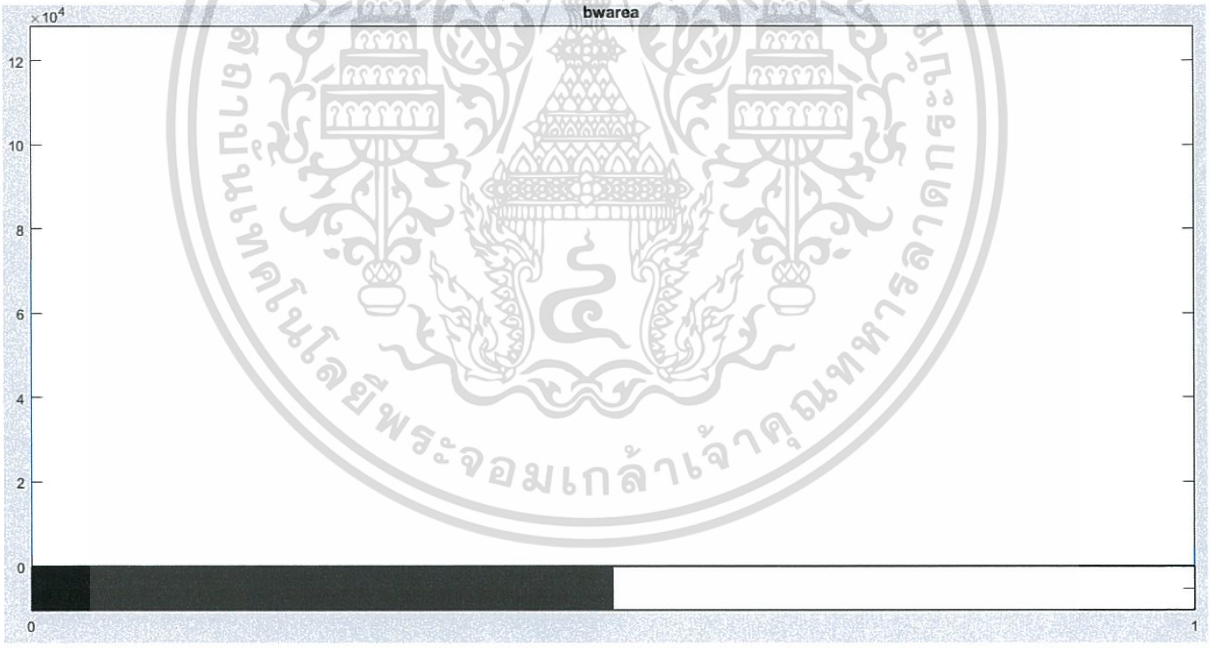
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในที่นี้กำหนดให้ค่าพิกเซลมีค่า 1000 พิกเซล

```
detred = bwareaopen(redbw,1000);
```



รูปที่ 3.22 ภาพขาวดำที่ผ่านการเลือกค่าพิกเซล



รูปที่ 3.23 กราฟฮิสโตแกรมของภาพขาวดำที่ทำการเลือกค่าพิกเซล

สุดท้ายจะได้รูปภาพที่อยู่ในรูป Binary Image และมีการ detect โดยมี Bounding box ล้อมรอบวัตถุที่สนใจไม่ว่าจะเคลื่อนที่ไปไหน ทั้งนี้ความเสถียรจะขึ้นอยู่กับสภาพแวดล้อมด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5 การรับค่าจากโปรแกรม MATLAB ส่งไปยัง Arduino UNO R3

เมื่อทำการตรวจจับวัตถุสีแดง แล้วได้ค่าจุดศูนย์กลาง (Centroid) จากการคำนวณโดย Bounding Box แล้ว จะทำการส่งค่าจุดศูนย์กลางในรูปพิกัด (x,y) ไปยังไมโครคอนโทรลเลอร์ Arduino UNO R3 โดยจำเป็นที่จะต้องติดตั้งส่วนขยายในโปรแกรม MATLAB ที่ชื่อว่า MATLAB Support Package fo Arduino Hardware เพื่อที่จะใช้ในการรับส่งข้อมูลระหว่างกัน

รูปที่ 3.24 โปรแกรมส่วนขยาย MATLAB Support Package fo Arduino Hardware

3.1.4.1 การส่งข้อมูลโดยโปรแกรม MATLAB

```
arduino=serial('COM3','BaudRate',9600);
```

คำสั่งนี้เป็นการเชื่อมต่อ MATLAB ในการติดต่อกับ Arduino ในที่นี้ COM3 คือ พอร์ตที่ Arduino ต่อกับคอมพิวเตอร์ โดยมีอัตราความเร็วในการรับส่งข้อมูล (Baud rate) เท่ากับ 9600 บิตต่อวินาที

```
fprintf(arduino,'%s',char(centx));
fprintf(arduino,'%s',char(centy));
```

คำสั่งนี้เป็นส่งข้อมูลพิกัดของจุดศูนย์กลาง (x,y) ไปยัง Arduino ในฐานะตัวแปร string โดยที่ centx และ centy คือค่าพิกัดที่แกน x และแกน y ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4.2 การรับข้อมูลโดย Arduino

```
Serial.begin(9600);
```

คำสั่งนี้เป็นการกำหนดอัตราความเร็วในการรับส่งข้อมูล (Baud rate) เท่ากับ 9600 บิตต่อวินาที ซึ่งค่าความเร็วนี้จะต้องมีอัตราเร็วเท่ากับคอมพิวเตอร์ที่เชื่อมต่อด้วย

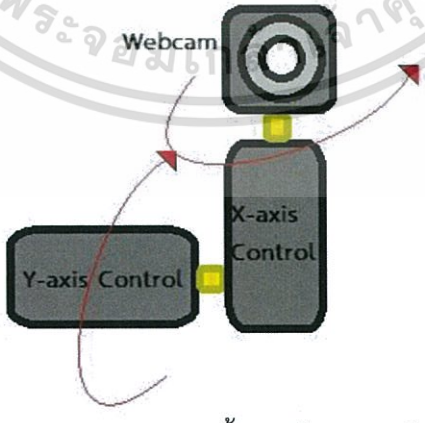
```
LSB = Serial.read();
valx = int(LSB);
delay(5);

LSB = Serial.read();
valy = int(LSB);
delay(5);
```

คำสั่งนี้เป็นการอ่านค่าที่รับมาจากพอร์ตอนุกรม (ในที่นี้คือค่าพิกัดแกน x และค่าพิกัดแกน y จากโปรแกรม MATLAB)

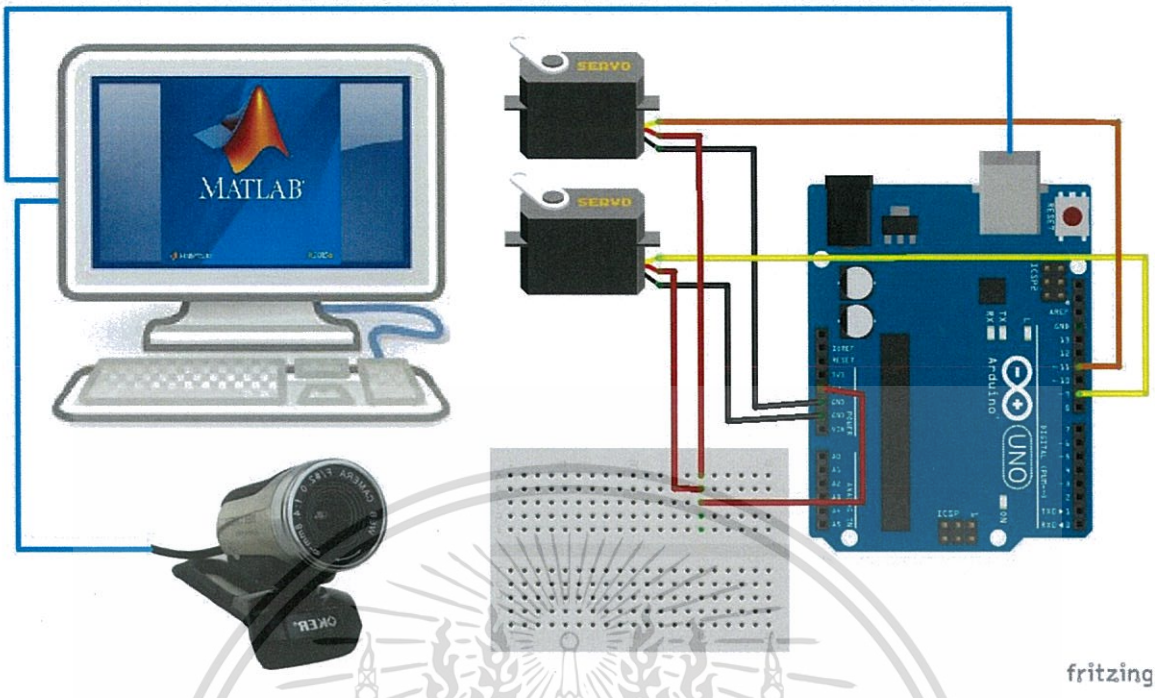
3.1.4.3 การควบคุมการหมุนของกล้อง

ในการควบคุมการหมุนของกล้องนั้น ได้ใช้เซอร์โวมอเตอร์ 2 ตัว โดยให้ตัวหนึ่งเคลื่อนที่กล้องในแนวราบ (แกน x) และอีกตัวหนึ่งเคลื่อนที่กล้องในแนวตั้ง (แกน y) ดังแสดงในรูป



รูปที่ 3.25 ภาพจำลองการออกแบบและติดตั้งเซอร์โวมอเตอร์เพื่อใช้ในการหมุนกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.26 การเชื่อมต่ออุปกรณ์

ในการใช้งานนี้เลือกใช้ความคมชัดของกล้องเป็นขนาด 320 x 240 พิกเซล เนื่องจากการใช้งานเป็นกล้องเทอร์มอลเราสนใจแค่สีที่แสดงถึงการแผ่ความร้อน จึงไม่จำเป็นที่จะต้องใช้ความคมชัดมากนัก

ขนาด 320 x 240 หมายถึง พิกเซลเล็กๆหลายพิกเซลที่ประกอบขึ้นมาเป็นรูปภาพ โดยมีรูปแบบการเก็บข้อมูลอยู่ในรูปเมทริกซ์ ซึ่งมีระยะทางในแนวแกนตั้งที่วัดจากจุดกำเนิดเท่ากับ 320 และมีระยะทางในแนวแกนนอนที่วัดจากจุดกำเนิดเท่ากับ 240

รูปที่ 3.27 ขนาดของภาพที่ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มขั้นตอนการประมวลผลภาพ เมื่อเริ่มการทำงานกล้องจะจับภาพสีแดง และทำการตีกรอบสี่เหลี่ยม (Bounding Box) ล้อมรอบบริเวณสีแดงนั้น และจะได้ค่าจุดศูนย์กลาง (Centroid) ซึ่งแสดงอยู่ในรูปพิกัด x,y

```
centx=bb(1) + (bb(3)/2);
centy=bb(2) + (bb(4)/2);
```

จากคำสั่งที่เขียนใน MATLAB นี้ เป็นคำสั่งในการคำนวณค่าจุดศูนย์กลางในแต่ละแกน โดยขั้นแรกเริ่มจากการหาขนาดความยาวในแต่ละแกนของกรอบสี่เหลี่ยม Bounding Box ก่อน ยกตัวอย่างเช่นในรูปที่ 3.28 โดยที่ bb(1) คือพิกัดแกน x ที่มุมบนซ้าย มีค่าเท่ากับ 115, bb(2) คือพิกัดแกน y ที่มุมบนซ้าย มีค่าเท่ากับ 85, bb(3) คือระยะห่างระหว่างพิกัดแกน x ที่มุมบนขวาและพิกัดแกน x ที่มุมบนซ้าย มีค่าเท่ากับ 185 – 115 เท่ากับ 70 และ bb(4) คือระยะห่างระหว่างพิกัดแกน y ที่มุมล่างซ้ายและมุมบนซ้าย มีค่าเท่ากับ 161 – 85 เท่ากับ 76

จากนั้นนำค่าที่ได้ มาคำนวณเพื่อใช้ในการหาจุดศูนย์กลาง ตามที่เขียนในสมการ จุดศูนย์กลางที่แกน x จะได้ 115 + 70/2 = 150 และที่แกน y จะได้ 85 + 76/2 = 123

รูปที่ 3.28 การคำนวณจุดศูนย์กลางจาก Bounding box

โดยใช้ไมโครคอนโทรลเลอร์ Arduino UNO R3 ในการควบคุมองศาการหมุนของเซอร์โวมอเตอร์ เมื่อได้รับค่าพิกัดแกน x และค่าพิกัดแกน y จากโปรแกรม MATLAB แล้ว Arduino ก็จะทำการสั่งให้เซอร์โวมอเตอร์หมุนตามคำสั่งที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(valx < 160){
    posx += distancex;
   }
   else if(valx > 160){
    posx -=distancex;
   }
```

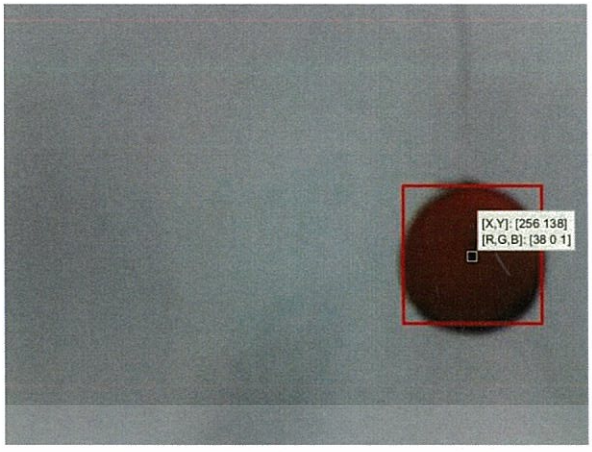
จากคำสั่งนี้ กำหนดให้ตัวแปร valx คือ ค่าพิกัดแกน x ที่รับมาจากโปรแกรม MATLAB และ distancex คือระดับองศาการหมุนของเซอร์โวมอเตอร์ โดยกำหนดให้หมุนครั้งละ 1 องศา

ถ้าหากค่าพิกัด valx มีค่าน้อยกว่า 160 (ตำแหน่งจุดศูนย์กลางอยู่บริเวณด้านซ้าย) Arduino จะสั่งให้มอเตอร์หมุนจนกว่าค่าจุดศูนย์กลางจะมีค่ามากกว่า 160 ซึ่งก็คือหมุนมาทางขวา

รูปที่ 3.29 พิกัดแกน x มีค่าน้อยกว่า 160

ถ้าหากค่าพิกัด valx มีค่ามากกว่า 160 (ตำแหน่งจุดศูนย์กลางอยู่บริเวณด้านขวา) Arduino จะสั่งให้มอเตอร์หมุนจนกว่าค่าจุดศูนย์กลางจะมีค่าน้อยกว่า 160 ซึ่งก็คือหมุนมาทางซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



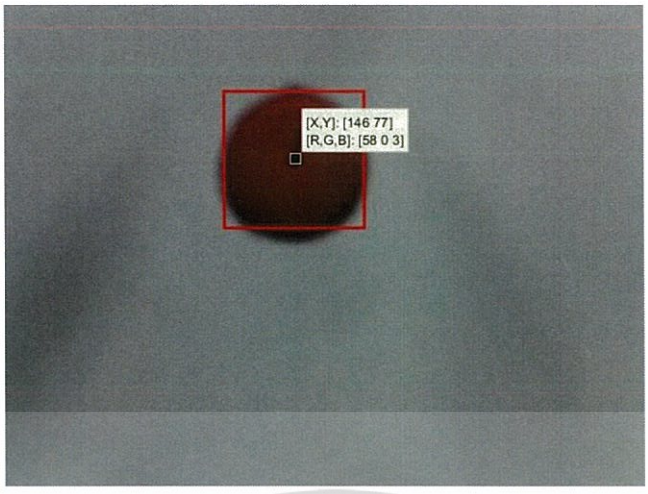
รูปที่ 3.30 พิกัดแกน x มีค่ามากกว่า 160

```
if(valy < 120){
    posy += distancey;
   }
   else if(valy > 120){
    posy -= distancey;
   }
```

จากคำสั่งนี้ กำหนดให้ตัวแปร valy คือ ค่าพิกัดแกน y ที่รับมาจากโปรแกรม MATLAB และ distancey คือระดับองศาการหมุนของเซอร์โวมอเตอร์ โดยกำหนดให้หมุนครั้งละ 1 องศา

ถ้าหากค่าพิกัด valy มีค่าน้อยกว่า 120 (ตำแหน่งจุดศูนย์กลางอยู่บริเวณด้านบน) Arduino จะสั่งให้มอเตอร์หมุนจนกว่าค่าจุดศูนย์กลางจะมีค่ามากกว่า 120 ซึ่งก็คือหมุนมาด้านล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.31 พิกัดแกน y มีค่าน้อยกว่า 120

ถ้าหากค่าพิกัด valy มีค่ามากกว่า 120 (ตำแหน่งจุดศูนย์กลางอยู่บริเวณด้านล่าง) Arduino จะสั่งให้มอเตอร์หมุนจนกว่าค่าจุดศูนย์กลางจะมีค่าน้อยกว่า 120 ซึ่งก็คือหมุนมาด้านบน

รูปที่ 3.32 พิกัดแกน y มีค่ามากกว่า 120

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

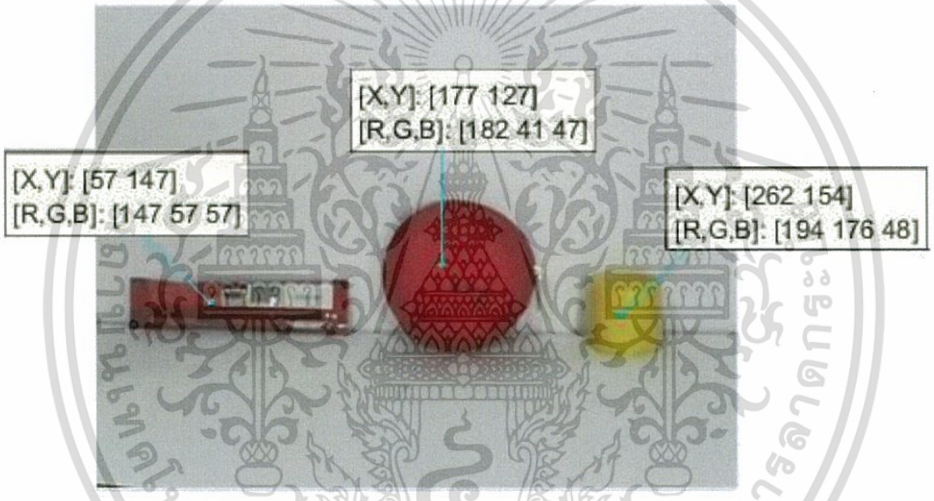
ผลการทดลอง

ในการทดลองนี้ได้ทำการศึกษาการใช้กล้องเว็บแคมในการจับกรอบไปที่ภาพที่มีความเป็นสีแดงมากซึ่งคล้ายกับการตรวจจับอุณหภูมิ การทดลองได้ผลการทดลองออกมาเป็นดังนี้

4.1 ผลการตรวจจับวัตถุ

4.1.1 ผลการตรวจจับวัตถุในตอนกลางวัน

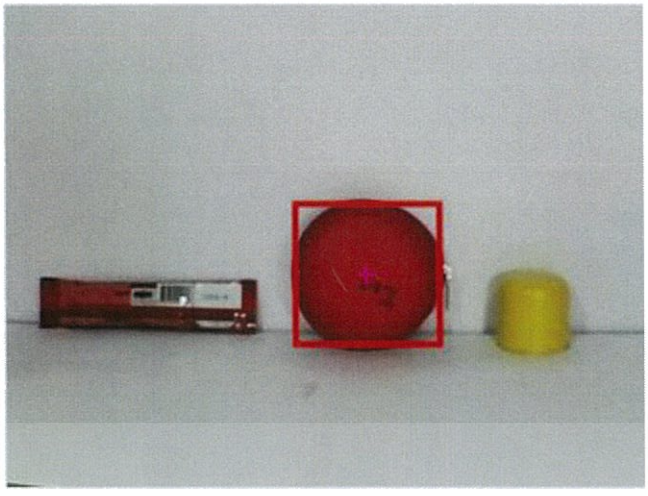
เมื่อทำการเขียนโปรแกรม MATLAB เพื่อทำการตรวจจับวัตถุที่สนใจ ในที่นี้คือวัตถุกลมสีแดง



รูปที่ 4.1 วัตถุที่มีค่าสีแดงใกล้เคียงกัน

จะเห็นได้ว่าในรูปมีวัตถุอีกสองชิ้นที่มีค่า R ใกล้เคียงวัตถุที่สนใจคือซองกาแฟสีแดงซึ่งมีค่าสีแดงเท่ากับ 147 และฝาขวดสีเหลืองที่มีค่าสีแดง เท่ากับ 194 เพื่อตัดวัตถุที่ไม่สนใจจึงใช้หลักการลบภาพเทาออกจากภาพระนาบสีแดง จะลดความสว่างของสีเหลือง และทำให้สีแดงเด่นชัดขึ้น จากนั้นทำการเปลี่ยนภาพเทาให้เป็นภาพขาว - ดำ โดยเลือกใช้ค่า Threshold ที่เหมาะสม จะเหลือภาพของวัตถุสีแดง 2 ชิ้น ซึ่งต้องการเฉพาะวัตถุกลมสีแดงเท่านั้น จึงใช้หลักการ bwopenarea เพื่อตัดพิกเซลที่ต่ำกว่าค่าที่เรากำหนดออกไป ซองกาแฟซึ่งมีพื้นที่พิกเซลของสีแดงที่ต่ำกว่าวัตถุกลมจึงถูกตัดออกไป สุดท้ายใช้ Bounding box เพื่อตีกรอบวัตถุที่เราสนใจ จะได้ผลลัพธ์ดังรูปที่ 4.2

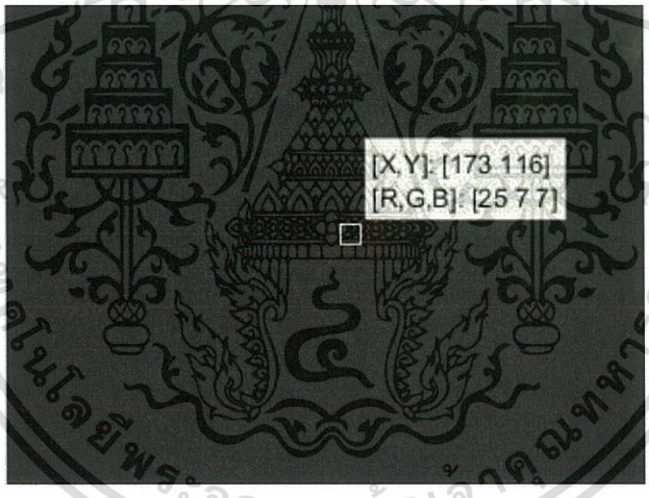
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 ผลลัพธ์การตรวจจับ

4.1.2 ผลการตรวจจับวัตถุในตอนกลางคืน

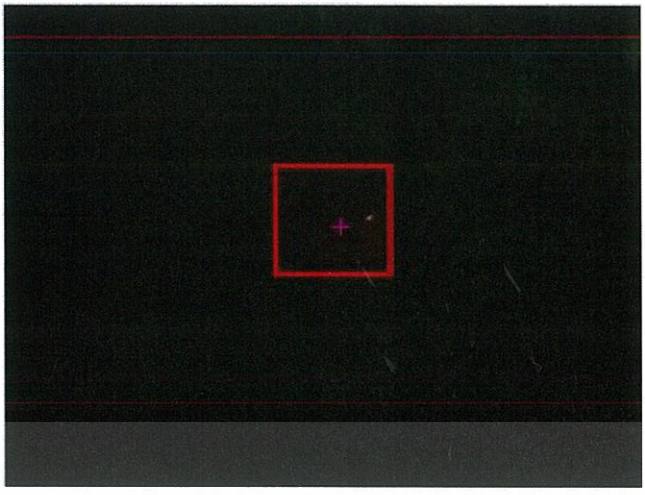
ทดลองตรวจจับในสภาพแวดล้อมที่มีความสว่างน้อย



รูปที่ 4.3 จับภาพวัตถุในตอนกลางคืน

เมื่อทดลองตอนกลางคืนที่มีค่าความเข้มของแสง (Contract) น้อย พบว่ามีค่าสีแดงที่น้อยมาก คือมีค่า 25 ซึ่งถ้ามองด้วยตาเปล่าจะสังเกตเห็นสีแดงได้ยากมาก สามารถทดลองได้โดยการใช้หลักการเดิมคือทำการลบภาพเทาออกจากเลเยอร์ของสีแดง จากนั้นเปลี่ยนเป็นภาพขาว – ดำ โดยหลักการสำคัญในการตรวจจับภาพในที่มืดคือค่า Threshold จะต้องมีค่าน้อยในระดับ 0.0x เนื่องจากมีความต่างของภาพระดับเทาน้อยมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



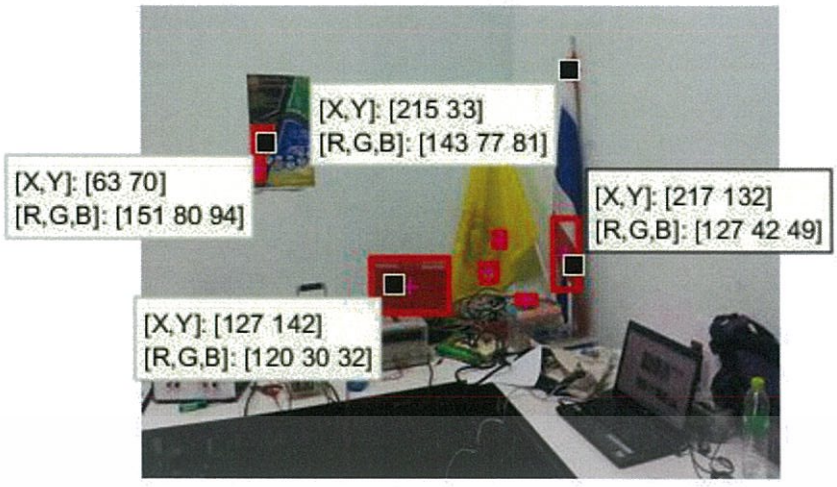
รูปที่ 4.4 ผลลัพธ์การตรวจจับในตอนกลางคืน

4.1.3 ผลการตรวจจับวัตถุในสภาพแวดล้อมปกติ

ทดลองตรวจจับวัตถุในสภาพแวดล้อมปกติ โดยทดลองตรวจจับสีแดงในห้อง จะพบว่ามีวัตถุสีแดงอยู่หลายตำแหน่ง เลยทำการตรวจสอบค่า Intensity ของสีแดงที่เห็นได้ชัด

รูปที่ 4.5 จับภาพวัตถุในห้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 ค่า Intensity ของวัตถุสีแดงแต่ละตำแหน่งในห้อง

ในที่นี้จะทำการตรวจจับไปที่กล่องเครื่องมือสีแดง จากภาพค่า Intensity ของกล่องเครื่องมือมีค่าต่ำกว่าวัตถุที่เหลือ ทำให้ใช้ค่า Threshold ในการตัดวัตถุอื่นออกไปไม่ได้ สังเกตว่ากล่องเครื่องมือมีพื้นที่ของพิกเซลมากกว่าวัตถุอื่น จึงใช้คำสั่ง bwareaopen ในการตัดวัตถุที่มีพื้นที่พิกเซลขนาดเล็กออก

รูปที่ 4.7 ผลลัพธ์การตรวจจับโดยการตัดพิกเซลขนาดเล็กออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการติดตามวัตถุ

เมื่อทำการตรวจจับวัตถุสีแดงได้ จากนั้นจะทำการติดตามวัตถุนั้น โดยเมื่อเคลื่อนที่วัตถุแล้ว เซอร์โวมอเตอร์ที่ติดตั้งในแนวแกน x และแกน y จะควบคุมการหมุนของกล้องเพื่อให้วัตถุอยู่บริเวณกึ่งกลางตามคำสั่งที่เขียนไว้ ซึ่งจุดศูนย์กลางของภาพขนาด 320 x 240 จะมีพิกัดอยู่ที่ 160, 120 ดังนั้นเซอร์โวมอเตอร์จะหมุนจนกว่าพิกัดแกน x และแกน y จะมีค่าอยู่ในช่วง 160 และ 120 ตามลำดับ ในที่นี้จะตัดช่วงหนึ่งของวิดีโอมาสังเกต ซึ่งวิดีโอก็คือรูปภาพจำนวนหลายๆเฟรมมาแสดงต่อกัน มาจับค่าพิกัดในแต่ละเฟรมเป็นจำนวน 30 เฟรม

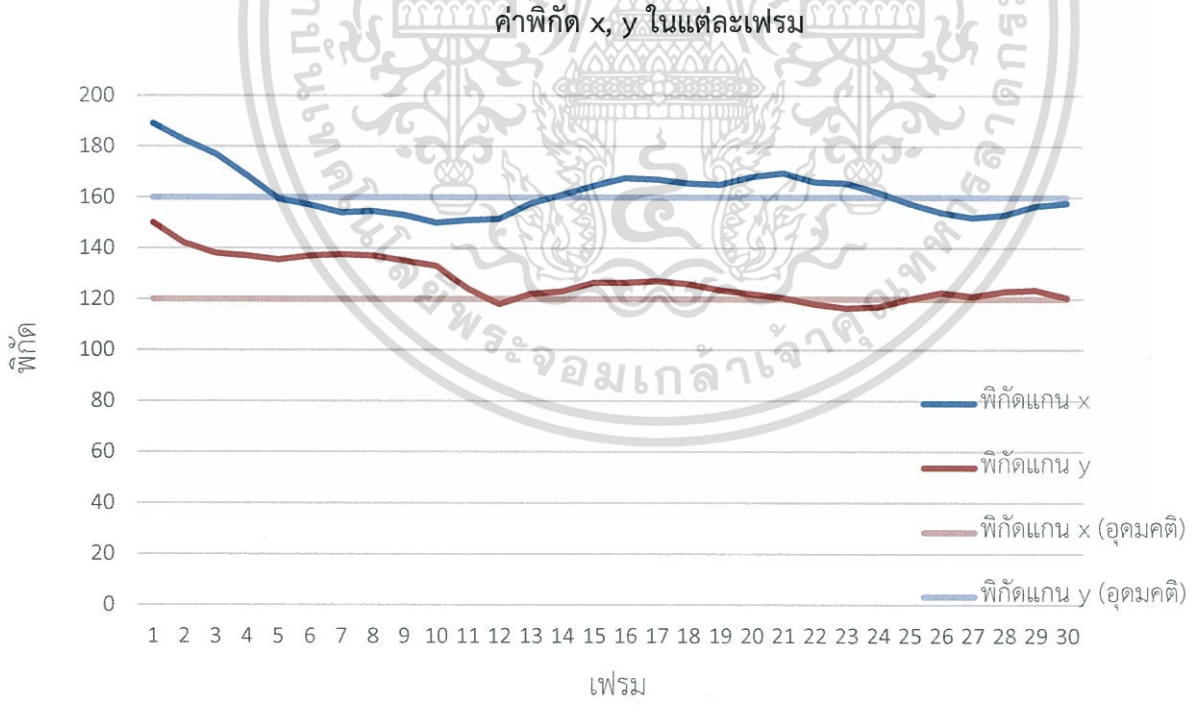
เฟรม	พิกัดแกน x	พิกัดแกน y
1	189	150
2	182.5	142
3	177	138
4	168.5	137
5	159.5	135.5
6	157	137
7	154	137.5
8	154.5	137
9	153	135
10	150	133
11	151	124
12	151.5	118
13	157.5	122
14	161	123
15	164.5	126.5
16	167.5	126.5
17	167	127
18	165.5	126
19	165	123.5
20	168	122

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

21	169.5	120.5
22	166	118
23	165.5	116.5
24	162	117
25	157.5	120
26	154	122.5
27	152	121
28	153	123
29	156.5	123.5
30	157.7	120.5

ตารางที่ 4.1 ค่าพิกัด x, y ในแต่ละเฟรม (วัตถุเคลื่อนที่ปกติ)

เมื่อนำมาสร้างกราฟและเทียบกับค่าจุดศูนย์กลางของรูปภาพในอุดมคติ (จุดศูนย์กลางอยู่ที่พิกัด 160, 120) พบว่าค่าพิกัดแกน x และแกน y อยู่ในช่วง 150 ถึง 170 และช่วง 110 ถึง 140 ตามลำดับ ซึ่งมีความใกล้เคียงกับค่าในอุดมคติ



รูปที่ 4.8 กราฟแสดงค่าพิกัด x, y ในแต่ละเฟรม (วัตถุเคลื่อนที่ปกติ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.8 เราสามารถกำหนดความแม่นยำของการติดตามได้ โดยการปรับความละเอียดองศาของเซอร์โวมอเตอร์ให้มีความละเอียดขึ้น แต่ต้องแลกด้วยการติดตามวัตถุที่มีการเคลื่อนที่ช้าลง ดังเช่นตารางข้างล่างที่ได้ติดตามวัดถุที่มีการเคลื่อนที่ช้า โดยตัดช่วงหนึ่งของวิดีโอมาเป็นจำนวน 30 เฟรม

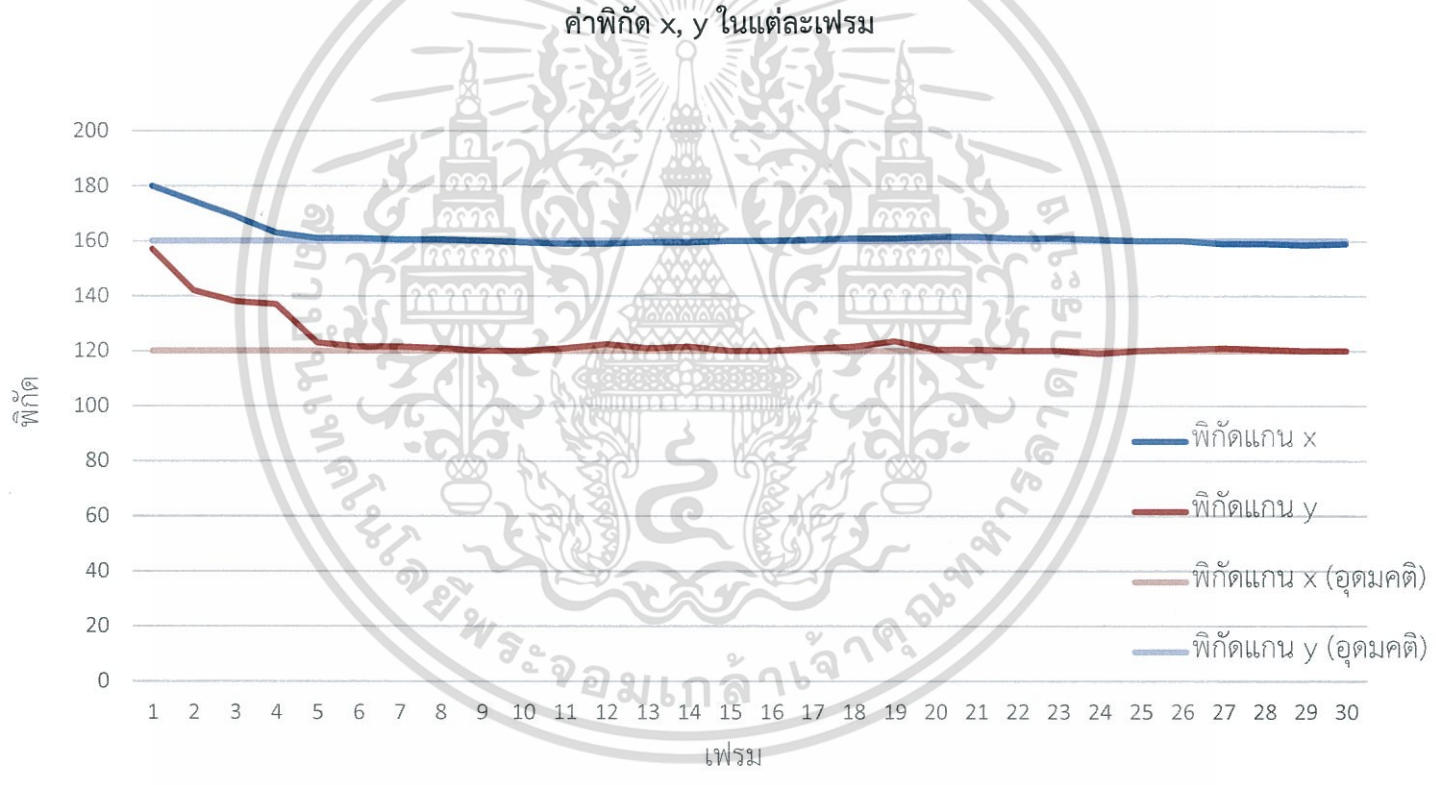
เฟรม	พิกัดแกน x	พิกัดแกน y
1	180	157
2	174.5	142
3	169	138
4	163	137
5	161	123
6	161	121.5
7	160.5	121.5
8	160.5	121
9	160	120
10	159.5	120
11	159	121
12	159	122.5
13	159.5	121
14	159.5	121.5
15	160	120
16	160	120
17	160.5	121
18	161	121.5
19	161	123.5
20	161.5	120.5
21	161.5	120.5
22	161	120
23	161	120
24	160.5	119
25	160	120

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

26	160	120.5
27	159	121
28	159	120.5
29	158.5	120
30	159	120

ตารางที่ 4.2 ค่าพิกัด x, y ในแต่ละเฟรม (วัตถุเคลื่อนที่ช้า)

เมื่อนำมาสร้างกราฟและเทียบกับค่าจุดศูนย์กลางของรูปภาพในอุดมคติ พบว่าค่าพิกัดแกน x และแกน y อยู่ใกล้หรือเทียบเท่ากับค่าในอุดมคติขึ้นมากขึ้น ดังนั้นจึงควรเลือกใช้ให้เหมาะสมกับสถานการณ์



รูปที่ 4.9 กราฟแสดงค่าพิกัด x, y ในแต่ละเฟรม (วัตถุเคลื่อนที่ช้า)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

ในการทดลองได้ทำการวิจัย ศึกษา และทดลองการใช้งานโปรแกรม MATLAB ในการใช้งานด้าน Image Processing เกี่ยวกับการตรวจจับวัตถุโดยเลือกให้ตรวจจับพิกเซลสีแดง ซึ่งเป็นสีที่แสดงถึงอุณหภูมิสูงในย่านรังสีอินฟราเรดเพื่อที่จะประยุกต์ใช้กับกล้องเทอร์มอล พบว่าสามารถตรวจจับพิกเซลสีแดงได้ถูกต้อง และการนำไมโครคอนโทรลเลอร์ Arduino มาใช้ในการควบคุมตำแหน่งของวัตถุที่สนใจ พบว่าจับตำแหน่งใกล้เคียงกับกึ่งกลางของภาพ การศึกษา วิจัยครั้งนี้อาจนำไปต่อยอดในการลดต้นทุนของกล้องเทอร์มอลได้ในอนาคต และการตรวจจับติดตามโดยใช้กล้อง

5.2 ข้อเสนอแนะ

การศึกษาด้าน Image Processing ในที่นี้ได้เลือกใช้โปรแกรม MATLAB เนื่องจากเป็นโปรแกรมที่อำนวยความสะดวกในการคำนวณในเรื่อง Matrix และมี Toolbox สำหรับการใช้งานในด้าน Image Processing โดยเฉพาะ แต่ข้อเสียคือ เนื่องจากมี Toolbox ที่หลากหลาย ทำให้หาฟังก์ชั่นที่ต้องการยาก และใช้เวลาประมวลผลสูง

ด้านการทำงานของตัวโปรแกรมตรวจจับนั้น เมื่อนำไปใช้จริงจะพบกับสิ่งรบกวนมากมาย เช่นความสว่าง ความร้อนอาจจะทำให้ค่า Intensity ของสีผิดเพี้ยนไป มีวัตถุอื่นๆที่มีค่าสีและพื้นที่ใกล้เคียงกับวัตถุที่สนใจทำให้การตรวจจับผิดพลาด

ด้านการติดตามวัตถุนั้น ในการใช้งานจริงนั้นความเร็วในการตอบสนองของกล้องยังมีความหน่วงอยู่ ซึ่งสามารถลดความหน่วงนี้ได้โดยการเพิ่มความเร็วของเซอร์โวมอเตอร์ แต่จะทำให้พิกัดที่ได้เบี่ยงเบนไปจากเดิมหลายตำแหน่ง เนื่องจากเวลาเพิ่มความเร็วจะทำให้ช่วงของพิกัดกว้างขึ้นด้วย ดังนั้นเพื่อให้ใกล้เคียงพิกัดที่ต้องการที่สุด จึงใช้ความเร็วค่าปกติของเซอร์โวมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] ดร.วรากรณ์ สามโกเศศ. 2017. **“Image Processing”.** [Online].
Available : https://th.wikipedia.org/wiki/image-processing

[2] ธนศรสวรรค์ วรธนมงคลชัย. 2016 **“การประมวลผลภาพดิจิตอล"**. [Online].
Available : https://kb.psu.ac.th/psukb/bitstream/2553/2731/ch_3.pdf

[3] Veedvil, 2017. **“กล้องเว็บแคม OKER177”** [Online].
Available : http://www.webcam2home.com/webcam-oker-177

[4] Oker. 2017. **“Webcam OKER177”**. [Online].
Available : http://www.advice.co.th/webcam/oker-177

[5] Crescere. 2014. **“Image Processing คืออะไร”.** [Online].
Available : https://www.crescere.co.th/blogs/content/B1047

[6] ChaowatX. 2015. **“การประมวลผลภาพ”**. [Online].
Available : https://sites.google.com/chaowatx/image-processing

[7] Jaratcyberu. 2009. **“Image Processing เทคโนโลยีการประมวลผลภาพ ”**. [Online].
Available : http://jaratcyberu.blogspot.com/image-processing

[8] TECHTALKTHAI. 2017. **“พิกเซลคืออะไร และมีความสำคัญอย่างไร”**. [Online].
Available : https://www.mindphp.com/2122-พิกเซล-คืออะไร/

[9] วายส์มีดู. 2017. **“พิกเซลใช้ทำอะไร”**. [Online].
Available : http://sites.google.com/cctvandsecuritysystem/pixel

[10] TECHTALKTHAI. 2017. **“RGB คือ ระบบสีของแสง”**. [Online].
Available : http://www.mindphp.com/2172-what-is-rgb/

[11] ธนศรสวรรค์ วรธนมงคลชัย. 2017. **“RGB คืออะไร”**. [Online].
Available : http://www.dpu.ac.th/techno/what-is-rgb-2017/

[12] BSTSoft. 2015. **“การ Convert รูปภาพทั่วไปให้เป็น Grayscale”**. [Online].
Available : http://www.blogspot.com/image-processing-convert-grayscale/

[13] Nextsoftwares. 2014. **“ความรู้เบื้องต้นในการประมวลผลภาพ”**. [Online].
Available : http://nextsoftwares.wordpress.com/2014/22/digital-image-processing/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[14] Zunvoo. 2013. **"Matlab คืออะไร"**. [Online].
Available : https://matlabthai.blogspot.com/matlab/

[15] Amornthep Qwave. 2016. **"Matlab กับการใช้งานจริง"**. [Online].
Available : https://www.qwaveshop.com/th/product/228737/product-228737?category_id=40983

[18] ทันพงษ์ ภู่รักษ์. 2016. **"โครงสร้างโปรแกรม Matlab". [Online].**
Available : http://www.sbt.ac.th/new/sites/default

[19] อมรเทพ ผันสิน. 2016. **"Image Processing พัฒนาด้วย Matlab และภาษา C#". [Online].**
Available :
http://www.thailandindustry.com/onlinemag/view2.php?id=31&section=30&issues=4&extend=semi

[20] Giampiero Camba. 2016. **"Legacy MATLAB and Simulink". [Online].**
Available : https://www.mathworks.com/matlabcentral/27843-legacy-matlab

[21] Marium Malik. 2014. **"Bounding Box". [Online].**
Available :
https://www.mathworks.com/matlabcentral/answers/?term=bounding%20box

[22] Chang. 2015. **"สื่อสารด้วย Serial". [Online].**
Available :
http://www.ayarafun.com/2011/02/arduino-tutorial-2-how-to-use-serial/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้