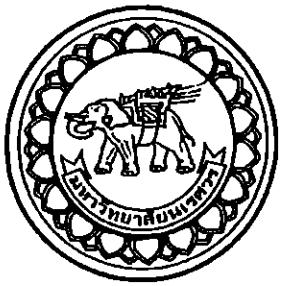


การตรวจจับสีและรูปร่างของวัตถุโดยใช้กล้อง

SHAPE AND COLOR DETECTION BY IMAGE PROCESSING

นายณฐพงศ์ เทพทองดี รหัส 53362655

นายณัฐจักร์ หงสวัฒนานนท์ รหัส 53362679

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 19 พ.ย. 57
เลขทะเบียน 1690010
เลขเรียกหนังสือ ปร.
มหาวิทยาลัยนเรศวร ฉ 119 ร
2556

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2556

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน การตรวจจับสีและรูปร่างของวัตถุโดยใช้กล้อง

ผู้ดำเนินโครงงาน นายณฐพงศ์ เทพทองดี รหัส 53362655

นายณัฐจักร์ หงสวัฒนานนท์ รหัส 53362679

ที่ปรึกษาโครงงาน ดร.มุฑิตา สงฆ์จันทร์

สาขาวิชา วิศวกรรมไฟฟ้า

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2556

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

.....ที่ปรึกษาโครงงาน

(ดร.มุฑิตา สงฆ์จันทร์)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ศุภวรรณ พลพิทักษ์ชัย)

.....กรรมการ

(อาจารย์เศรษฐา ตั้งค้าวานิช)

ชื่อหัวข้อโครงงาน	การตรวจจับสีและรูปร่างของวัตถุโดยใช้กล้อง
ผู้ดำเนินโครงงาน	นายณฐพงศ์ เทพทองดี รหัส 53362655
	นายณัฐจักร์ หงสวัฒนานนท์ รหัส 53362679
ที่ปรึกษาโครงงาน	ดร.มุฑิตา สงฆ์จันทร์
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2556

.....

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอระบบการจำแนกสีและรูปร่างของวัตถุโดยใช้กล้องซึ่งควบคุมการทำงานจากโปรแกรมแมทแลป (Matlab) และไมโครคอนโทรลเลอร์อาดูยโน่ (Arduino) ระบบนี้จะใช้หลักการของการประมวลผลภาพ (Image Processing) มาวิเคราะห์หาสีและรูปร่างของวัตถุโดยนำเอาภาพที่ได้มาประมวลผลผ่านทางคอมพิวเตอร์และนำเอาสีแต่ละจุดมาพิจารณาเพื่อจำแนกสีและรูปร่างของวัตถุ ซึ่งจะทำการจำแนกสีทั้งหมด 3 สี คือ สีแดง สีเขียว และสีน้ำเงิน และรูปร่างทั้งหมด 3 แบบ คือ ทรงกลม ทรงสี่เหลี่ยมจัตุรัส และทรงสามเหลี่ยมด้านเท่า จากผลการทดลองระบบการจำแนกสีและรูปร่างของวัตถุโดยใช้กล้องแสดงให้เห็นว่า ระบบนี้มีความสามารถในการจำแนกสีและรูปร่างของวัตถุ พร้อมทั้งสามารถระบุตำแหน่งของวัตถุได้ตามที่ตั้งไว้

Project title	Shape and Color Detection by Image Processing		
Name	Mr. Nattaphong	Thepthongdee	ID. 53362655
	Mr. Nattajuk	Hongsawatnanon	ID. 53362679
Project advisor	Miss Mutita	Songjun, Ph.D.	
Major	Electrical Engineering		
Department	Electrical and Computer Engineering		
Academic year	2013		

Abstract

This project presents the classification of the shape and the color of the object by using the digital camera. The operational system is controlled by microcontroller Arduino and processed via program Matlab. The principle of the image processing is used to analyze the shape and the color of the object. The image is firstly brought into computer and the color at each point is considered to classify the color and the shape of the object. There are only 3 colors which are classified; red green and blue. The shapes which the system can be classified are sphere pyramid and cube. The experiments show that this system can detect and classify the color and the shape of the object. Also the position of the object can be identified correctly.

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดีเป็นเพราะได้รับความกรุณาเป็นอย่างยิ่งจาก ดร.มุฑิตา สงฆ์จันทร์ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงาน อีกทั้งยังให้ความกรุณาในการตรวจทานปริญญานิพนธ์ คณะดำเนินโครงงานขอกราบขอบพระคุณเป็นอย่างสูงและขอระลึกถึงความกรุณาของท่านไว้ตลอดไป และขอขอบคุณคณาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้ให้กับคณะผู้ดำเนินงาน นอกจากนี้ต้องขอขอบคุณภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ที่ให้ยืมอุปกรณ์และเครื่องมือวัดมาใช้งาน จนทำให้โครงงานนี้สำเร็จลุล่วงไปได้

เหนือสิ่งอื่นใด คณะผู้ดำเนินโครงงานขอกราบขอบพระคุณคุณพ่อ คุณแม่ ผู้มอบความรัก ความเมตตา สติปัญญา รวมทั้งเป็นผู้ให้ทุกสิ่งทุกอย่างตั้งแต่วัยเยาว์จวบจนถึงปัจจุบัน คอยเป็นกำลังใจทำให้ได้รับความสำเร็จอย่างทุกวันนี้ และขอขอบคุณทุกๆคนในครอบครัวของคณะผู้ดำเนินโครงงานที่ไม่ได้กล่าวไว้ ณ ที่นี้ด้วย

นายณฐพงศ์ เทพทองดี

นายณัฐจักร์ หงสวัฒนานนท์

สารบัญ

หน้า

ใบรับรองปริญญานิพนธ์ ..... ก
บทคัดย่อภาษาไทย ..... ข
บทคัดย่อภาษาอังกฤษ ..... ค
กิตติกรรมประกาศ ..... ง
สารบัญ ..... จ
สารบัญตาราง ..... ช
สารบัญรูป ..... ฌ

บทที่ 1 บทนำ 1
1.1 ที่มาและความสำคัญของโครงงาน 1
1.2 วัตถุประสงค์ของโครงงาน 1
1.3 ขอบเขตของโครงงาน 2
1.4 ขั้นตอนและแผนการดำเนินงาน 2
1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน 4
1.6 งบประมาณของโครงงาน 4

บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง 5
2.1 ขั้นตอนการประมวลผลภาพดิจิตอล 5
2.1.1 ขั้นตอนการประมวลผลภาพดิจิตอล (Digital Image Processing) 5
2.1.2 กระบวนการก่อนการประมวลผลภาพ (Pre-processing) 6
2.2 ภาพดิจิตอล 6
2.3 ประเภทของภาพ (Image types) 7
2.3.1 ภาพสี (Color Image) 7
2.3.2 ภาพระดับความเข้มเทา (Gray Scale Image) 7
2.3.3 ภาพไบนารี (Binary Image) 8
2.3.4ภาพแบบดัชนี (Index Image) 8
2.4 ขนาดของไฟล์ภาพ (Image File Sizes) 8
2.5 การประมวลผลภาพแบบจุด (Point Image Processing) 9
2.5.1 การกระทำทางคณิตศาสตร์ (Arithmetic Operations) 10
2.5.2 ภาพแบบเนกาทีฟ (Image Negative) 11

สารบัญ (ต่อ)

หน้า

2.6 การประมวลผลภาพสี (Color Image Processing) 11

บทที่ 3 วิธีการดำเนินโครงงาน....14

3.1 การออกแบบส่วนลำตัวของสายพานลำเลียง....14

3.1.1 โครงสร้างหลักของสายพานลำเลียง....15

3.1.2 โรลเลอร์ (Roller)....15

3.1.3 สายพาน....16

3.1.4 มอเตอร์....16

3.1.5 ตัวต้านทานชนิดไวแสง....17

3.1.6 กล้องเวปแคม (Webcam)....17

3.2 วัตถุที่นำมาตรวจสอบ....19

3.2.1 จำแนกวัตถุตามสี....19

3.2.2 จำแนกวัตถุตามรูปร่าง....20

3.3 ขั้นตอนการทำงานของระบบตรวจจับวัตถุบนสายพานลำเลียงโดยใช้กล้อง....20

3.3.1 ระบบสายพานลำเลียง....20

3.3.2 ระบบประมวลผลภาพ....21

บทที่ 4 ผลการทดลอง....24

4.1 การทดลองใช้งานโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ (GUI)....24

4.1.1 ส่วนแสดงผลด้วยภาพ (Image)....25

4.1.2 ส่วนแสดงผลด้วยข้อมูล (Data)....25

4.2 การทดลองการหาตำแหน่งของวัตถุ....27

4.2.1 ส่วนแสดงค่าตำแหน่งของวัตถุผ่านโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ .27

4.2.2 การคำนวณหาค่าตำแหน่งของวัตถุ....28

4.3 การทดลองการจำแนกวัตถุในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์....29

4.3.1 การทดลองการจำแนกสีของวัตถุในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์....30

4.3.2 การทดลองการจำแนกรูปร่างของวัตถุในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์ 32

4.4 การทดลองการจำแนกวัตถุในสภาพแสงจากดวงอาทิตย์....34

4.4.1 การทดลองการจำแนกสีของวัตถุในสภาพแสงจากแสงจากดวงอาทิตย์ 35

สารบัญ (ต่อ)

หน้า

4.4.2 การทดลองการจำแนกรูปร่างของวัตถุในสภาพแสงจากแสงจากดวงอาทิตย์ 36
4.5 การทดลองการจำแนกรูปทรงของวัตถุจากความเข้มแสง....39
4.5.1 การทดลองด้วยการใช้กระดาษไขจำนวน 1 แผ่น 39
4.5.2 การทดลองด้วยการใช้กระดาษไขจำนวน 3 แผ่น 41
4.5.3 การทดลองด้วยการใช้กระดาษไขจำนวน 5 แผ่น 43
4.5.4 การทดลองด้วยการใช้กระดาษไข โดยการเพิ่มครั้งละ 5 แผ่น 45

บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ 47
5.1 สรุปผลการดำเนินโครงงาน 47
5.2 ปัญหาและแนวทางแก้ไข 47
5.3 แนวทางในการพัฒนาโครงงาน....48

เอกสารอ้างอิง....49

ภาคผนวก ก โปรแกรมการประมวลผลภาพโดยการตรวจจับสีของวัตถุ 50
ภาคผนวก ข โปรแกรมควบคุมระบบตรวจจับวัตถุบนสายพานลำเลียงโดยใช้กล้อง....57
ภาคผนวก ค รายละเอียดของไอซีหมายเลข ATMEGA328 60

ประวัติผู้ดำเนินโครงงาน....66

สารบัญตาราง

ตารางที่ หน้า

1.1 แผนการดำเนินงานโครงงาน 3

4.1 ผลการทดลองการจำแนกสีของวัตถุในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์ 32

4.2 ผลการทดลองการจำแนกรูปร่างในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์ 34

4.3 ผลการทดลองการจำแนกสีของวัตถุในสภาพแสงจากดวงอาทิตย์ 36

4.4 ผลการทดลองการจำแนกรูปร่างของวัตถุในสภาพแสงจากดวงอาทิตย์ 38

4.5 ผลการทดลองการจำแนกรูปทรงวัตถุจากความเข้มแสงโดยใช้กระดาษไข 1 แผ่น 41

4.6 ผลการทดลองการจำแนกรูปทรงวัตถุจากความเข้มแสงโดยใช้กระดาษไข 3 แผ่น 43

4.7 ผลการทดลองการจำแนกรูปทรงวัตถุจากความเข้มแสงโดยใช้กระดาษไข 5 แผ่น 45

4.8 ผลการทดลองการจำแนกรูปทรงวัตถุจากความเข้มแสงโดยเพิ่มกระดาษไขครั้งละ 5 แผ่น 46

สารบัญรูป

รูปที่ หน้า

2.1 การกระทำการกับภาพแบบจุด 9
2.2 การประมวลผลภาพสีวิธีที่ 1 12
2.3 การประมวลผลภาพสีวิธีที่ 2 13
3.1 แผนภาพแสดงภาพรวมของระบบ (System Overview Diagram) 14
3.2 สายพานลำเลียง 14
3.3 โครงสร้างส่วนลำตัวของสายพานลำเลียง 15
3.4 โครงสร้างของโรลเลอร์ 15
3.5 ลักษณะของสายพาน 16
3.6 มอเตอร์ไฟฟ้ากระแสตรงขนาด 12 โวลต์ 16
3.7 วงจรตัวต้านทานชนิดไวแสง 17
3.8 ตัวต้านทานชนิดไวแสง 17
3.9 ลักษณะของกล้องเวปแคมยี่ห้อโลจิเทค รุ่น C170 18
3.10 ลักษณะของระบบตรวจจับวัตถุบนสายพานลำเลียงโดยใช้กล้องด้านหน้า 18
3.11 ลักษณะของระบบตรวจจับวัตถุบนสายพานลำเลียงโดยใช้กล้องด้านบน 18
3.12 วัตถุที่นำมาตรวจสอบจำแนกตามสี 19
3.13 วัตถุที่นำมาตรวจสอบจำแนกตามรูปร่าง 20
3.14 แผนภาพแสดงขั้นตอนการทำงานของระบบสายพานลำเลียง 21
3.15 แผนภาพแสดงขั้นตอนการทำงานของระบบประมวลผลภาพ 22
4.1 หน้าต่างแสดงการใช้งานโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ 24
4.2 ภาพแสดงผลลัพธ์ของการประมวลผลภาพผ่านทางโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ 26
4.3 รูปภาพแสดงผลของการประมวลผลภาพ 27
4.4 รูปภาพแสดงผลของการประมวลผลภาพที่ระบุตำแหน่งของวัตถุ 27
4.5 รูปภาพแสดงการเปรียบเทียบระยะทางจริงในภาพกับระยะภาพในหน่วยพิกเซล 28
4.6 ภาพแสดงตัวอย่างพิกัดในแกนนอนและแกนตั้งในหน่วยพิกเซล 29
4.7 ภาพการทดลองการจำแนกสีในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์ 30
4.8 ภาพแสดงการจำแนกวัตถุสีแดงในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์ 30
4.9 ภาพแสดงการจำแนกวัตถุสีน้ำเงินในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์ 31
4.10 ภาพแสดงการจำแนกวัตถุสีเขียวในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์ 31

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.11 ภาพแสดงการจำแนกวัตถุทรงสามเหลี่ยมด้านเท่าในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์....32
4.12 ภาพแสดงการจำแนกวัตถุทรงสี่เหลี่ยมจัตุรัสในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์....33
4.13 ภาพแสดงการจำแนกวัตถุทรงกลมในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์....33
4.14 ภาพแสดงการทดลองการจำแนกวัตถุในสภาพแสงจากดวงอาทิตย์....34
4.15 ภาพแสดงการจำแนกวัตถุสีแดงในสภาพแสงจากดวงอาทิตย์....35
4.16 ภาพแสดงการจำแนกวัตถุสีเขียวในสภาพแสงจากดวงอาทิตย์....35
4.17 ภาพแสดงการจำแนกวัตถุสีน้ำเงินในสภาพแสงจากดวงอาทิตย์....36
4.18 ภาพแสดงการจำแนกวัตถุทรงสามเหลี่ยมด้านเท่าในสภาพแสงจากดวงอาทิตย์....37
4.19 ภาพแสดงการจำแนกวัตถุทรงสี่เหลี่ยมจัตุรัสในสภาพแสงจากดวงอาทิตย์....37
4.20 ภาพแสดงการจำแนกวัตถุทรงกลมในสภาพแสงจากดวงอาทิตย์....38
4.21 ภาพแสดงการทดลองโดยใช้กระดาษไข....39
4.22 ภาพแสดงการจำแนกวัตถุทรงสามเหลี่ยมด้านเท่าโดยใช้กระดาษไข 1 แผ่น....40
4.23 ภาพแสดงการจำแนกวัตถุทรงสี่เหลี่ยมจัตุรัสโดยใช้กระดาษไข 1 แผ่น....40
4.24 ภาพแสดงการจำแนกวัตถุทรงกลมโดยใช้กระดาษไข 1 แผ่น....40
4.25 ภาพแสดงการจำแนกวัตถุทรงสามเหลี่ยมด้านเท่าโดยใช้กระดาษไข 3 แผ่น....41
4.26 ภาพแสดงการจำแนกวัตถุทรงสี่เหลี่ยมจัตุรัสโดยใช้กระดาษไข 3 แผ่น....42
4.27 ภาพแสดงการจำแนกวัตถุทรงกลมโดยใช้กระดาษไข 3 แผ่น....42
4.28 ภาพแสดงการจำแนกวัตถุทรงสามเหลี่ยมด้านเท่าโดยใช้กระดาษไข 5 แผ่น....43
4.29 ภาพแสดงการจำแนกวัตถุทรงสี่เหลี่ยมจัตุรัสโดยใช้กระดาษไข 5 แผ่น....44
4.30 ภาพแสดงการจำแนกวัตถุทรงกลมโดยใช้กระดาษไข 5 แผ่น....44

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ปัจจุบันเทคโนโลยีเข้ามามีบทบาทและความสำคัญเป็นอย่างมากในชีวิตประจำวันของมนุษย์ในหลากหลายด้าน ซึ่งการประยุกต์ต่างๆเหล่านี้ได้นำไปใช้ผ่านระบบคอมพิวเตอร์อิเล็กทรอนิกส์ไม่ว่าจะเป็นด้านอุตสาหกรรม ทางการแพทย์ การกีฬา เป็นต้น ทำให้การประยุกต์ใช้งานด้านนี้ได้มีการใช้งานแพร่หลายและเพิ่มมากขึ้นอย่างรวดเร็ว เพื่ออำนวยความสะดวกสบายให้กับการทำงานให้กับมนุษย์ รวมไปถึงทำให้คุณภาพชีวิตของคนในปัจจุบันดีขึ้นอย่างมากด้วย

การประมวลผลสัญญาณภาพ เป็นอีกเรื่องหนึ่งที่ได้รับความนิยมอย่างแพร่หลายในปัจจุบัน เช่น การสแกนลายนิ้วมือ การทำเอ็มอาร์ไอ (Magnetic Resonance Imaging) ในทางการแพทย์ ภาพถ่ายดาวเทียม รวมไปถึงการประชุมผ่านเทเลคอนเฟอเรนซ์ เป็นต้น ซึ่งต่างก็มีพื้นฐานมาจากการประมวลสัญญาณภาพทั้งสิ้น ทำให้การประมวลผลสัญญาณภาพเป็นประโยชน์อย่างมากในปัจจุบันทั้งในด้านการช่วยลดต้นทุนด้านแรงงาน และประสิทธิภาพการทำงานของโรงงาน ช่วยเพิ่มศักยภาพในการรักษาผ่านเครื่องมือแพทย์ที่มีความแม่นยำสูง การช่วยในการจัดการประชุมผ่านเครือข่ายอินเตอร์เน็ตที่ไม่ต้องถูกจำกัดด้วยสถานที่ รวมไปถึงการคัดแยกหรือตรวจสอบด้วย การประมวลผลสัญญาณภาพก็เป็นที่นิยมอย่างมาก เนื่องจากประยุกต์ใช้งานได้หลากหลาย

ดังนั้นโครงงานนี้จึงได้จัดทำโครงงานการตรวจจับรูปร่างและสีของวัตถุที่ควบคุมการทำงานโดยใช้การประมวลผลภาพเพื่อเป็นประโยชน์ต่อการค้นคว้า ตลอดจนสามารถนำไปประยุกต์ใช้งานในรูปแบบต่างๆได้

1.2 วัตถุประสงค์ของโครงงาน

เพื่อศึกษาการทำงานและสร้าง ระบบตรวจจับสีและรูปร่างของวัตถุ โดยใช้กล้อง และควบคุมการทำงานโดยไมโครคอนโทรลเลอร์

1.3 ขอบเขตของโครงงาน

1. ติดตั้งกล้องรับภาพไว้ที่ตำแหน่งปลายสุดของหุ่นยนต์แขนกลเพื่อจับภาพวัตถุ
2. ตรวจสอบวัตถุโดยการจำแนกสี 3 สี คือ แดง เขียว และ น้ำเงิน
3. ตรวจสอบวัตถุโดยการจำแนกรูปทรงต่างๆ คือ ทรงกลม สี่เหลี่ยม และสามเหลี่ยม
4. ขนาดของวัตถุที่นำมาตรวจจับต้องมีขนาดไม่เกิน 10 × 10 ซม.
5. ระบุพิกัดของวัตถุในพิกัด x , y

1.4 ขั้นตอนและแผนการดำเนินงาน

ขั้นตอนและแผนการดำเนินงานสามารถแสดงได้ดังตารางที่ 1.1

มหาวิทยาลัยนเรศวร

ตารางที่ 1.1 แผนการดำเนินงานโครงงาน

ลำดับ	การดำเนินการ	ปี 2556								
		เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.
1	ศึกษาวิธีการตรวจจับวัตถุโดยใช้การบวนการวิธีต่างๆ	←	—	→						
2	เขียนโปรแกรมในการตรวจจับวัตถุโดยใช้กล้อง			←	—	→				
3	ตรวจสอบการทำงานของโปรแกรมในการจำแนกสีและรูปร่าง					←	—	→		
4	ติดตั้งกล้องกับหุ่นยนต์แขนกลและตรวจสอบการทำงานอีกครั้ง						←	—	→	
5	สรุปผลการทดลองจัดรูปเล่มโครงงาน								←	—

1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน

1.ได้ระบบตรวจจับวัตถุใช้กล้อง ซึ่งสามารถตรวจสอบและจำแนกสีและรูปร่างของวัตถุ ที่สามารถใช้งานได้จริง

2. สามารถนำระบบตรวจจับวัตถุโดยใช้กล้อง ไปพัฒนาเพื่อเป็นประโยชน์ในเชิงพาณิชย์

3. สามารถนำระบบตรวจจับวัตถุโดยใช้กล้องไปประยุกต์กับงานคัดแยกวัตถุในทางอุตสาหกรรมได้

1.6 งบประมาณของโครงงาน

1.ค่าอุปกรณ์ไฟฟ้าและอิเล็กทรอนิกส์	1,000	บาท
2.ค่าเอกสาร	1,000	บาท
รวมเป็นเงินทั้งสิ้น (สองพันบาทถ้วน)	2,000	บาท

(หมายเหตุ: ถัวเฉลี่ยทุกรายการ)

บทที่ 2

หลักการและทฤษฎี

ในบทนี้จะกล่าวถึงหลักการและทฤษฎีที่เกี่ยวข้องกับการตรวจจับวัตถุซึ่งจำแนกโดยใช้สีและรูปร่างของวัตถุ โดยหลักการที่นำมาใช้เป็นเรื่องของการประมวลผลภาพ(Image Processing) โดยที่การประมวลผลภาพคือกระบวนทำการอย่างใดอย่างหนึ่งกับภาพต้นฉบับ (Input Image) เพื่อให้ได้ภาพผลลัพธ์ (Output Image) ที่มีลักษณะของภาพเป็นไปตามที่ต้องการ ซึ่งการกระทำการกับภาพที่ใช้ในการประมวลผลภาพดิจิตอลมีอยู่หลากหลายวิธีด้วยกัน ความเข้าใจเกี่ยวกับคุณลักษณะและการแยกแยะประเภทของการกระทำการกับภาพจะช่วยให้สามารถคาดคะเนภาพผลลัพธ์ที่จะได้จากการกระทำการแต่ละแบบ หรือประมาณความซับซ้อนของการกระทำการกับภาพที่จะนำไปใช้ได้ ซึ่งหลักการของการประมวลผลภาพสามารถอธิบายได้ดังหัวข้อต่อไปนี้

2.1 ขั้นตอนการประมวลผลภาพดิจิตอล

2.1.1 ขั้นตอนการประมวลผลภาพดิจิตอล

ประกอบไปด้วย 3 ส่วนคือ

1. ขั้นตอนการนำข้อมูลภาพเข้าสู่เครื่องคอมพิวเตอร์ (Input)
2. ขั้นตอนการนำคอมพิวเตอร์เพื่อประมวลผลภาพผ่านอัลกอลิทึมต่างๆ (Processing)
3. ขั้นตอนการแสดงผลภาพ (Output)

โดยทั่วไปอุปกรณ์พื้นฐานในการประมวลผลภาพตามขั้นตอนที่กล่าวมาข้างต้นจะประกอบด้วย เครื่องไมโครคอมพิวเตอร์ที่มีอุปกรณ์ตรวจจับภาพ (Frame grabber card) ซึ่งจะต่อกับกล้องวิดีโอตัวจับภาพ ขั้นตอนการนำภาพเข้าสู่เครื่องคอมพิวเตอร์เป็นการแปลงสัญญาณภาพทั่วไปที่เป็นสัญญาณอนาล็อกให้เป็นสัญญาณภาพดิจิตอล เพื่อให้เครื่องคอมพิวเตอร์สามารถประมวลผลภาพได้ ทั้งนี้สามารถทำได้โดยนำข้อมูลภาพจากตัวจับภาพ ซึ่งโดยทั่วไปแล้วภาพดิจิตอลที่ได้จากการแปลงสัญญาณจะถูกเก็บในหน่วยความจำที่มีอยู่ในอุปกรณ์ตัวจับภาพและสามารถนำมาประมวลผลได้โดยการเขียนโปรแกรมภาษาระดับสูง เช่น ภาษาซี อุปกรณ์ตัวจับภาพที่นิยมใช้จะสามารถเก็บภาพขนาด 512 × 512 จุดภาพ (pixel) และแต่ละจุดภาพสามารถแสดงระดับเทาได้อย่างน้อย 256 ระดับ ส่วนกล้องวิดีโอที่ใช้ในการเก็บภาพมักจะเป็นกล้องซีซีดี (Charge Coupled Device (CCD) camera)

2.1.2 ก่อนการประมวลผลภาพ(Pre-processing)

ต้องเตรียมข้อมูลภาพซึ่งมีขั้นตอนดังนี้

1. **ปรับปรุงข้อมูลภาพในส่วนที่ไม่คมชัด**

2. **กำจัดสัญญาณรบกวน** (Noise)

3. **การปรับคอนทราสต์** (Contrast) **หรือปรับปรุงคุณสมบัติของการมองเห็น เช่น การปรับค่าความเข้มของจุดภาพ การกลับข้อมูลภาพ เป็นต้น**

4. **การแบ่งแยกข้อมูลภาพออกจากสีพื้นโดยใช้ค่าเธสโฮลด์** (Threshold)

5. การแปลงข้อมูลภาพในทางเรขาคณิต เช่น การหมุนภาพ การเปลี่ยนแปลงขนาดภาพ และการแก้ไขตำแหน่งของจุดภาพ

6. การแก้ไขข้อมูลภาพในส่วนที่มีการผิดเพี้ยน

2.2 ภาพดิจิตอล

ข้อมูลภาพดิจิตอลจะมีระดับความเข้มเทาที่สามารถเขียนให้อยู่ในรูป**ของฟังก์ชันสองมิติของค่าระดับความเข้มแสง** ซึ่งระดับความเข้มเทา (Gray Level) ก็คือค่าระดับ**ของสัญญาณที่ผ่านกระบวนการควอนไทซ์** (Quantization) ที่จะเป็นตัวบ่งบอกความมืดหรือ**ความสว่างของแต่ละจุดภาพ** (Pixel) โดยค่าของฟังก์ชัน $f(x, y)$ จะแสดงถึงค่าความเข้มแสงที่จุดพิกัด**สปาเชียล** (Spatial) x, y, ใดๆ โดยทั่วไปนิยมเขียนแทนภาพดิจิตอลสองมิติ $f(x, y)$ ขนาด $M \times N$ ให้**อยู่ในรูปของเมทริกซ์สองมิติ**ดังในสมการที่ (2.1)

$$f(x,y)=\begin{bmatrix} f(0,0) & f(0,1) & f(0,2) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & f(1,2) & \cdots & f(1,N-1) \\ f(2,0) & f(2,1) & f(2,2) & \cdots & f(2,N-1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,1) & f(M-1,2) & \cdots & f(M-1,N-1) \end{bmatrix} \quad (2.1)$$

โดยที่ $f(x, y)$ **แทนค่าระดับความเข้มเทาที่จุดพิกัด** x, y ใดๆ และ x, y **คือจุดพิกัดของลำดับแถวและหลักของภาพตามลำดับ**

คุณภาพของรูปภาพดิจิตอลจะขึ้นอยู่กับจำนวนพิกเซลและจำนวนค่าระดับความเข้มเทาคือจำนวนพิกเซลในแต่ละแถวและแต่ละหลักของรูปภาพดิจิตอลเป็นจำนวนค่าสุ่มตัวอย่างทั้งหมดหรือเป็นความละเอียดของสัญญาณภาพดิจิตอล ดังนั้นการลดจำนวนพิกเซลลงจะเป็นการลดความ

ละเอียดของรูปภาพดิจิตอล จึงมีผลทำให้รูปภาพผลลัพธ์ที่ได้อาจผิดเพี้ยนไปจากรูปภาพต้นฉบับได้ ในทำนองเดียวกัน จำนวนค่าระดับของสัญญาณดิจิตอลจะถูกกำหนดด้วยจำนวนค่าระดับความเข้มเทาที่ใช้ในการแสดงผลภาพดิจิตอลนั่นคือคุณภาพของรูปภาพดิจิตอลจะขึ้นอยู่กับจำนวนค่าระดับความเข้มเทาที่ใช้ แต่อย่างไรก็ตาม การเพิ่มจำนวนพิกเซลหรือจำนวนค่าระดับความเข้มเทาในบางครั้งก็อาจจะไม่เพิ่มคุณภาพของรูปภาพดิจิตอลหรืออาจจะเพิ่มน้อยมาก

โดยปกติแล้ว ภาพจะดูคมชัดเท่าไรขึ้นอยู่กับว่าภาพนั้นมีจำนวนพิกเซลอยู่มากเท่าใด ซึ่งมักจะบอกความละเอียดของภาพ(Resolution) โดยดูจากจำนวนพิกเซลต่อนิ้ว (Pixel/inch: PPI) หรือพิกเซลต่อตารางนิ้ว (Pixels/inch^2) หากภาพยิ่งมีจำนวนพิกเซลต่อหนึ่งตารางนิ้วมาก ภาพก็ยิ่งมีความละเอียดเหมือนจริงมากขึ้น โดยภาพที่มีความละเอียดมากๆ จะเรียกว่าภาพนี้มีเรสโซลูชั่นสูง ส่วนภาพที่มีความละเอียดน้อยก็จะเรียกว่าภาพนี้มีเรสโซลูชั่นต่ำ เช่น ภาพขนาดกว้าง 1 นิ้ว ยาว 1 นิ้ว ที่มีความละเอียด 72 พิกเซลต่อนิ้ว จะมีจำนวนพิกเซลทั้งหมด 5184 จุดพิกเซล (72 × 72 = 5184)

2.3 ประเภทของภาพ (Image types)

โดยทั่วไปสามารถแบ่งประเภทของภาพบิตแมป (Bitmap image) หรือคือภาพที่ประกอบขึ้นจากจุดขนาดเล็กๆ หรือพิกเซลจำนวนมากที่เรียงต่อกันจนเป็นภาพ ตามคุณสมบัติการแสดงผลของสีภาพได้เป็น 4 ประเภทดังนี้

2.3.1 ภาพสี (Color Image)

ภาพชนิดนี้ แต่ละพิกเซลของภาพจะเก็บค่าระดับความเข้มเทาของแต่ละแถบแสงของแม่สีหลัก 3 สีที่ซ้อนกันอยู่คือ สีแดง (Red) สีเขียว (Green) สีน้ำเงิน (Blue) ซึ่งในแต่ละพิกเซลนั้นๆ จะแสดงผลของค่าสีของแต่ละพิกเซลตามระดับความเข้มในแต่ละแถบแสงสีนั้น

2.3.2 ภาพระดับความเข้มเทา (Intensity Image or Gray Scale Image)

ลักษณะของภาพชนิดนี้ในแต่ละพิกเซลจะมีค่าความเข้มของแสงในแต่ละระดับที่แตกต่างกันไป ตั้งแต่ระดับเทาดำไปยังระดับสีขาว ซึ่งสามารถกำหนดระดับความเข้มของแสงได้โดยใช้ค่าระดับความเข้มเทา (Gray scale หรือ Gray level) โดยปกติภาพแบบระดับสีเทาจะมีค่าระดับความเข้มเทาเท่ากับ 8 บิต ดังนั้นค่าความเข้มแสงจะถูกแบ่งออกเป็น 256 ระดับ เมื่อค่าระดับความเข้มเทามีค่าเป็น 0 จะหมายถึงจุดภาพนั้นมีค่าความเข้มของแสงต่ำ จะทำให้จุดภาพเป็นสีดำ ในทางกลับกัน หากค่าระดับความเข้มเทาเป็น 255 จะหมายถึงจุดภาพนั้นมีค่าความเข้มของแสงมาก

จะทำให้จุดภาพเป็นสีขาว ซึ่งสีขาวจะถูกแทนด้วยค่าความเข้มเทาเท่ากับ 255 (11111111) และสีดำจะถูกแทนด้วยค่าระดับความเข้มเทาเท่ากับ 0 (00000000) ส่วนค่าระหว่าง 0 – 255 จะมีค่าไล่เฉดสีจากสีดำไปหาสีขาว

2.3.3 ภาพไบนารี (Binary Image)

ภาพไบนารีจะแสดงลักษณะของข้อมูลภาพในรูปแบบขาวดำ กล่าวคือในแต่ละพิกเซลของภาพจะถูกแสดงด้วยค่าแบบไบนารี (Binary) คือมี 1 บิต ซึ่งประกอบไปด้วยค่า 1 และ 0 โดยที่ 1 หมายถึงจุดภาพสีขาว และ 0 หมายถึงจุดภาพสีดำ ภาพประเภทนี้เหมาะสำหรับภาพที่เกี่ยวกับตัวอักษร (Text) ภาพลายนิ้วมือ (Finger print) เป็นต้น

2.4 ขนาดของไฟล์ภาพ (Image File Sizes)

ขนาดของไฟล์ภาพขึ้นอยู่กับปัจจัยหลักๆ 2 ส่วนคือ ขนาดของภาพ (Size) และจำนวนบิตที่ใช้ในการแสดงค่าสีหรือระดับความเข้มของแสงในแต่ละพิกเซลของภาพ เมื่อกำหนดให้ $M \times N$ คือขนาดสัญญาณภาพและ L คือจำนวนบิตที่ใช้ควอนไทซ์ในแต่ละจุดภาพ ดังนั้นขนาดของไฟล์ภาพ (S) หรือจำนวนบิตทั้งหมดที่ต้องใช้สำหรับหน่วยความจำในการเก็บข้อมูลภาพมีค่าดังสมการต่อไปนี้

$$S = M \times N \times L \qquad (2.2)$$

ถ้าพิจารณาภาพขาวดำขนาด 512 × 512 พิกเซล ขนาดของไฟล์ภาพจะสามารถคำนวณได้จาก

$512 \times 512 \times 1$ = 262,144 บิต

= 32,768 ไบต์

= 0.033 เมกะไบต์

กรณีที่ภาพเป็นภาพแบบใช้ระดับความเข้มเทาที่มีระดับความเข้มเทาอยู่ระหว่าง 0 – 255 (8 บิต) และมีขนาด 512 × 512 พิกเซล ขนาดของไฟล์ภาพสามารถคำนวณได้จาก

$512 \times 512 \times 8$ = 2,097,152 บิต

= 262,144 ไบต์

= 0.262 เมกะไบต์

แต่ถ้าภาพที่พิจารณาเป็นภาพสีขนาด 512 × 512 พิกเซล ความละเอียด 8 บิต ขนาดของไฟล์ภาพจะสามารถคำนวณได้จาก

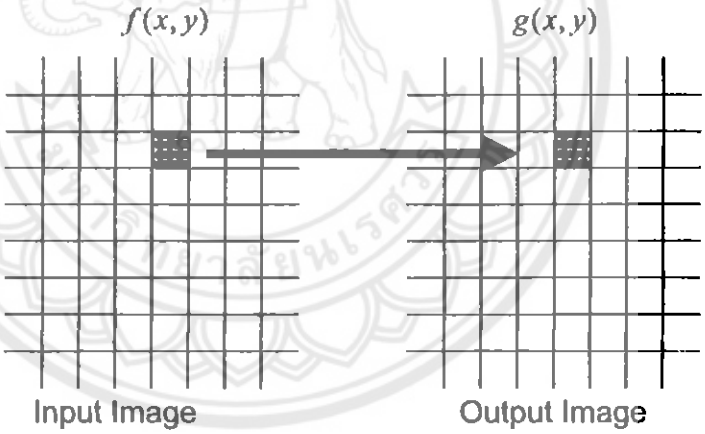
$512 \times 512 \times 8 \times 3$ = 6,291,456 บิต

= 786,432 ไบต์

= 0.786 เมกะไบต์

2.5 การประมวลผลภาพแบบจุด (Point Image Processing)

การประมวลผลภาพแบบจุดเป็นวิธีการกระทำการกับภาพต้นฉบับที่ค่าระดับความเข้มเทาที่แสดงในแต่ละพิกเซลของภาพผลลัพธ์จะขึ้นอยู่กับค่าระดับความเข้มเทาของพิกเซลในภาพต้นฉบับพิกเซลต่อพิกเซลในตำแหน่งที่สมนัยกัน โดยที่ค่าเปลี่ยนแปลงของพิกเซลของภาพผลลัพธ์ไม่ได้ขึ้นกับค่าพิกเซลที่อยู่บริเวณใกล้เคียงของภาพต้นแบบโดยรูปที่ 2.1 แสดงลักษณะการกระทำการกับภาพแบบจุดต่อจุด



รูปที่ 2.1 การกระทำการกับภาพแบบจุด

กำหนดให้ $f(x, y)$ และ $g(x, y)$ เป็นภาพต้นฉบับและภาพผลลัพธ์ตามลำดับ สมการที่ใช้ในการประมวลผลภาพแบบจุดต่อจุดสามารถแสดงได้ดังสมการที่ (2.3)

$$g(x,y) = M\left[f(x,y)\right] \qquad (2.3)$$

เมื่อ $M[\ ]$ เป็นการกระทำกับภาพแบบจุดใดๆ หรือเป็นการแทนข้อมูลภาพด้วยฟังก์ชันแมปปิ้ง (Mapping Function) ใดๆ โดยค่าระดับความเข้มเทาใหม่ที่ได้ของแต่ละพิกเซลของภาพจะถูกแทนที่ในค่าของแต่ละพิกเซลของภาพต้นฉบับที่พิกัด (x, y) เดิมของภาพ

ตัวอย่างของการประมวลผลภาพทางดิจิตอลแบบนี้ได้แก่ การปรับความสว่างและคอนทราสต์ของภาพดิจิตอลด้วยการบวก ลบ คูณ และหารด้วยจำนวนค่าใดๆ กับภาพดิจิตอลต้นแบบ หรือการกระทำทางตรรกศาสตร์ต่างๆ เป็นต้น

2.5.1 การกระทำทางคณิตศาสตร์ (Arithmetic Operations)

การกระทำทางคณิตศาสตร์นี้เป็นการกระทำการแบบจุดกับภาพต้นฉบับอย่างง่ายที่สุด ซึ่งวิธีการนี้สามารถที่จะนำเลขจำนวนใดๆ ก็ตามไปบวก ลบ คูณ หรือหารกับค่าระดับความเข้มเทาเดิมของภาพต้นฉบับ ผลลัพธ์ของการกระทำทางคณิตศาสตร์มีผลต่อการปรับเพิ่มหรือลดค่าระดับความสว่าง (Brightness) และค่าคอนทราสต์ (Contrast) ของภาพแบบเป็นเชิงเส้น โดยค่าระดับความสว่างของภาพจะปรับเพิ่มหรือลดได้โดยนำค่า b ไปบวกหรือลบกับภาพต้นฉบับ ส่วนค่าคอนทราสต์ของภาพจะปรับเพิ่มหรือลดได้โดยนำค่า a ไปคูณหรือหารกับภาพต้นฉบับ ดังสมการที่ (2.4)

$$g(x,y) = (a \cdot f(x,y)) + b \qquad (2.4)$$

เมื่อ $f(x, y)$ คือค่าระดับความเข้มเทาในแต่ละจุดพิกัด (x, y) ใดๆ ของภาพต้นฉบับ $g(x, y)$ คือค่าระดับความเข้มเทาในแต่ละจุดพิกัด (x, y) ใดๆ ของภาพผลลัพธ์ a คือตัวแปรสำหรับใช้ปรับค่าคอนทราสต์ และ b คือตัวแปรสำหรับใช้ปรับค่าความสว่าง

ในกระบวนการกระทำกับภาพทางคณิตศาสตร์นั้น มีสิ่งจำเป็นที่ควรจะต้องพิจารณาคือ เมื่อทำการบวก ลบ คูณ หรือหารกับภาพต้นฉบับ เมื่อผลลัพธ์ของค่าระดับความเข้มเทาใหม่ที่ได้มีค่าไม่เป็นจำนวนเต็ม จะต้องทำการปัดเศษเพื่อให้ค่าระดับความเข้มเทาใหม่ที่ได้เป็นจำนวนเต็ม แต่อย่างไรก็ตาม ในบางครั้งและมีความเป็นไปได้เสมอที่ค่าผลลัพธ์ของค่าระดับความเข้มเทาใหม่มีค่าน้อยกว่า 0 หรือมากกว่า 255 ในกรณีนี้จำเป็นต้องทำการตัดค่าระดับความเข้มเทาที่ได้ตามกฎเกณฑ์ดังสมการต่อไปนี้

$$y = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } 0 \le x \le 255 \\ 255 & \text{if } x > 255 \end{cases} \qquad (2.5)$$

เมื่อ x คือค่าระดับความเข้มเทาที่ผ่านกระบวนการกระทำกับภาพทางคณิตศาสตร์ และ y คือค่าระดับความเข้มเทาใหม่ซึ่งผ่านการตัดค่าแล้ว ดังนั้นค่าระดับความเข้มเทาที่มีค่ามากกว่า 255 และน้อยกว่า 0 ค่าระดับความเข้มเทาเหล่านั้นจะถูกกำหนดให้มีค่าเป็น 255 และ 0 ตามลำดับ การทำ

เช่นนี้ก็เพื่อให้ภาพผลลัพธ์ที่ได้มีค่าระดับความเข้มเทาอยู่ในช่วง 0 ถึง 255 แต่ข้อเสียของวิธีการนี้ก็คือ หากค่าระดับความเข้มเทาใหม่มีค่ามากกว่า 255 มากๆ เช่น มีค่าเป็นจำนวนเต็มหลักพันถึงหลักล้าน ก็จะมีผลทำให้ภาพที่ได้มีความผิดเพี้ยนไป

2.5.2 ภาพแบบเนกาทีฟ (Image Negative)

การทำภาพแบบเนกาทีฟคือการแปลงกลับของข้อมูลภาพ โดยที่ในแต่ละพิกเซลของภาพที่ค่าระดับความเข้มเทาสูงสุดจะถูกแปลงให้มีค่าระดับความเข้มเทาต่ำสุด และในทางตรงกันข้าม จุดพิกเซลที่มีค่าระดับความเข้มเทาต่ำสุดจะถูกแปลงให้จุดพิกเซลนั้นมีค่าระดับความเข้มเทาสูงสุด ยกตัวอย่างเช่น ถ้าข้อมูลภาพมีขนาด 8 บิต ค่าสูงสุดของข้อมูลคือ 255 และค่าข้อมูลต่ำสุดคือ 0 ดังนั้นค่าแต่ละพิกเซลของภาพผลลัพธ์สามารถคำนวณได้จากสมการที่ (2.6)

$$g(x,y) = 255 - f(x,y) \qquad (2.6)$$

เมื่อ $f(x, y)$ คือค่าระดับความเข้มเทาในแต่ละจุดพิกเซล (x, y) ใดๆ ของภาพต้นฉบับ และ $g(x, y)$ คือค่าระดับความเข้มเทาในแต่ละจุดพิกเซล (x, y) ใดๆ ของภาพผลลัพธ์

2.6 การประมวลผลภาพสี (Color Image Processing)

การประมวลผลภาพสีนั้นสามารถนำเทคนิคและวิธีการประมวลผลภาพระดับความเข้มเทาที่ได้กล่าวมาแล้วนำมาใช้ในการประมวลผลภาพสีได้ เช่น การประมวลผลภาพแบบจุด การปรับเพิ่มหรือลดความสว่าง การขยายฮีสโทแกรมภาพ เป็นต้น แต่เนื่องจากภาพสีในแบบจำลองสี RGB ประกอบไปด้วยข้อมูลภาพของแม่สีหลัก 3 สีที่ซ้อนกันอยู่คือ ข้อมูลภาพของแม่สีแดง สีเขียว และสีน้ำเงิน ดังนั้นในการประมวลผลภาพสีนั้น จะทำการประมวลผลทีละแบนด์ของข้อมูลภาพของแต่ละแม่สีหลัก กล่าวคือ ขั้นตอนของการประมวลผลภาพในแต่ละแม่สีหลักนั้นจะมีขั้นตอนการประมวลผลภาพแยกออกจากกัน จากนั้นจึงนำผลลัพธ์ของภาพที่ประมวลผลได้ในแต่ละข้อมูลภาพของแม่สีหลักมารวมกันแสดงผลเป็นภาพผลลัพธ์ที่ได้จากการประมวลผลภาพสี ในบางกรณี ภาพผลลัพธ์ที่ได้จากการประมวลผลภาพสีในบางครั้งอาจมีผลทำให้ภาพผลลัพธ์มีความไม่สมบูรณ์หรือมีความผิดเพี้ยนของสีในภาพผลลัพธ์ได้ ดังนั้นเพื่อแก้ไขความผิดเพี้ยนของสีที่อาจจะเกิดขึ้นได้จากการทำการประมวลผลภาพทีละข้อมูลภาพหรือแบนด์ย่อยของแต่ละแม่สีหลักแยกจากกัน ก็สามารถที่จะทำการเปลี่ยนแปลงภาพสีในรูปแบบจำลอง RGB ให้แสดงผลอยู่ในรูปของแบบจำลองสีแบบ

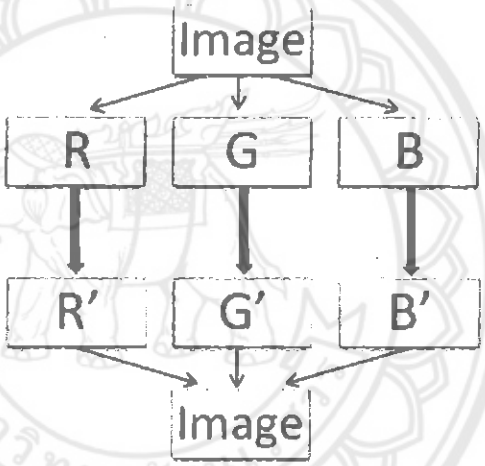
อื่นๆ จากนั้นจะทำการประมวลผลภาพเฉพาะส่วนของข้อมูลภาพที่แสดงถึงค่าความเข้มแสงหรือความสว่างของสีเท่านั้น

การประมวลผลภาพสีสามารถแบ่งออกได้เป็น 2 วิธีคือ

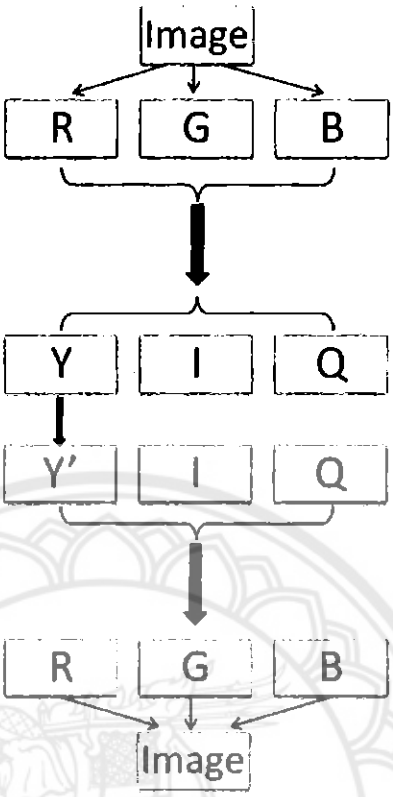
1. ทำการประมวลผลภาพในแต่ละแบนด์ย่อยของแบบจำลองสี RGB

2. ทำการเปลี่ยนแบบจำลองสี RGB เป็นแบบจำลองสี YIQ แล้วทำการประมวลผลภาพเฉพาะในแบนด์ย่อย Y เท่านั้น (แบนด์ I และ Q ไม่ต้องทำการประมวลผลภาพ) จากนั้นทำการเปลี่ยนแบบจำลองสี YIQ เป็นแบบจำลองสี RGB)

รูปแบบการประมวลผลภาพสีทั้ง 2 วิธีนี้ถูกแสดงไว้ในรูปที่ 2.4 และรูปที่ 2.5



รูปที่ 2.4 การประมวลผลภาพสีวิธีที่ 1

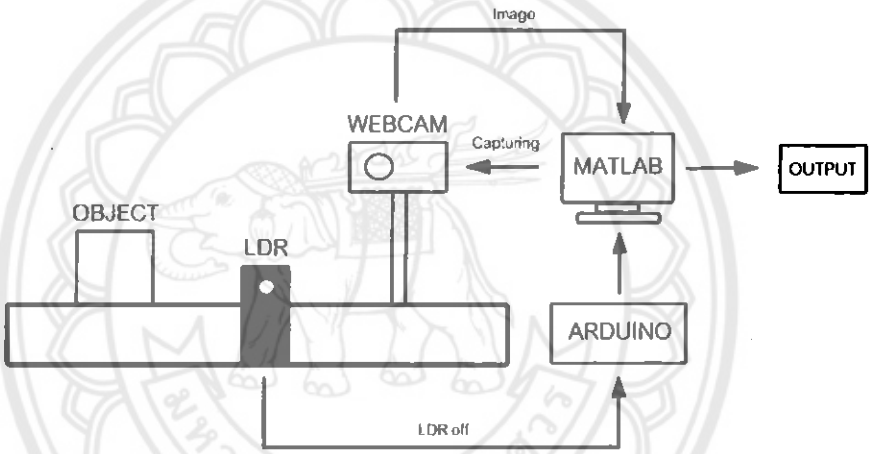


รูปที่ 2.5 การประมวลผลภาพสีวิธีที่ 2

บทที่ 3

วิธีดำเนินโครงงาน

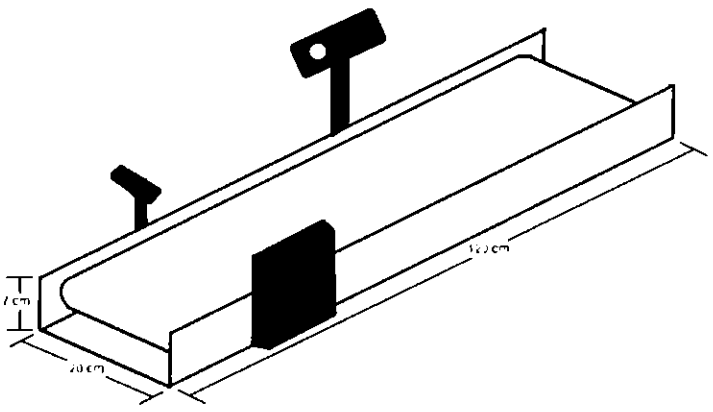
ในกรณีศึกษานี้ เป็นเรื่องการตรวจจับวัตถุบนสายพานลำเลียง ที่ควบคุมการทำงานของสายพานด้วยตัวต้านทานชนิดไวต่อแสง (LDR) จากนั้นระบบประมวลผลภาพจะทำการเริ่มกระบวนการประมวลผล ที่ได้จากการจับภาพนิ่งเพื่อทำภาพเข้าสู่กระบวนการประมวลผลผ่านโปรแกรมคอมพิวเตอร์คือ แมทแลป (Matlab) และสุดท้ายก็จะทำการส่งค่าที่ได้จากการประมวลผลในกระบวนการอื่นต่อไป



รูปที่ 3.1 แผนภาพแสดงภาพรวมของระบบ (System Overview Diagram)

3.1 การออกแบบส่วนลำตัวของสายพานลำเลียง

การออกแบบโครงสร้างของสายพานลำเลียงประกอบด้วยโครงสร้างหลักดังรูปที่ 3.2



รูปที่ 3.2 สายพานลำเลียง

3.1.1 โครงสร้างหลักของสายพานลำเลียง

โครงสร้างหลักของสายพานลำเลียงเป็นโครงสร้างรูปทรงสี่เหลี่ยมผืนผ้า ที่ทำจากอะคริลิค ทำให้ได้โครงสร้างที่ค่อนข้างมีความแข็งแรง และมีน้ำหนักไม่มาก เมื่อเทียบกับวัสดุที่มีความแข็งแรงชนิดอื่นๆ โดยที่ลักษณะการออกแบบโครงสร้างดังรูปที่ 3.3

รูปที่ 3.3 โครงสร้างส่วนลำตัวของสายพานลำเลียง

3.1.2 โรลเลอร์(Roller)

โครงสร้างของโรลเลอร์ในรางเลื่อนนั้นทำจากท่อพีวีซี โดยมีลักษณะเป็นทรงกระบอก โดยโรลเลอร์ที่วางไว้ที่ตำแหน่งแรกกับที่ตำแหน่งสุดท้ายจะใช้แผ่นยางมาพันให้รอบเพื่อให้โรลเลอร์มีความเหนียว สามารถยึดเกาะกับสายพานได้ ดังรูปที่ 3.4

รูปที่ 3.4 โครงสร้างของโรลเลอร์

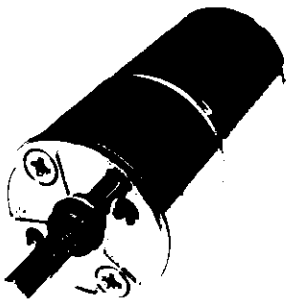
3.1.3 สายพาน

ส่วนของสายพานจะใช้ไวนิลสีขาวเป็นอุปกรณ์ในการทำมีลักษณะเป็นรูปสี่เหลี่ยมผืนผ้ามีขนาดความกว้าง 20 เซนติเมตร และความยาว 250 เซนติเมตร ดังรูปที่ 3.5

รูปที่ 3.5 ลักษณะของสายพาน

3.1.4 มอเตอร์

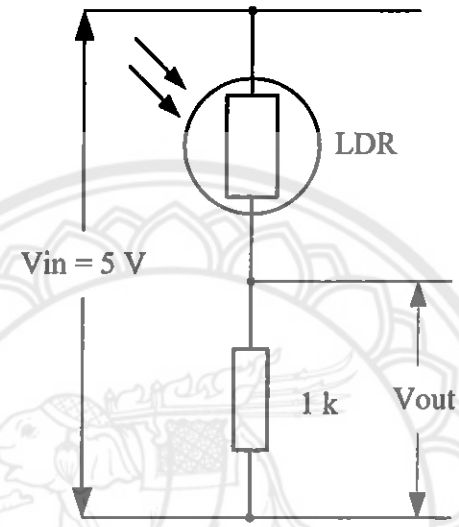
ส่วนของการขับเคลื่อนสายพานจะใช้มอเตอร์ไฟฟ้ากระแสตรงขนาด 12 โวลต์ โดยสายพานจะเริ่มเคลื่อนที่ทันทีเมื่อระบบเริ่มทำงาน และจะสายพานจะหยุดเคลื่อนที่เมื่อระบบสามารถตรวจจับว่ามีวัตถุเลื่อนเข้ามาในระบบ และมอเตอร์จะหยุดทำงานครู่หนึ่งทำให้สายพานหยุดเลื่อนเพื่อให้กล้องสามารถจับภาพวัตถุขณะอยู่นิ่งได้ จากนั้นมอเตอร์จะกลับมาทำงานขับเคลื่อนสายพานต่อไป



รูปที่ 3.6 มอเตอร์ไฟฟ้ากระแสตรงขนาด 12 โวลต์

3.1.5 ตัวต้านทานชนิดไวแสง (LDR)

ตัวต้านทานชนิดไวแสงเป็นอุปกรณ์ที่นำมาใช้ในการควบคุมการทำงานของมอเตอร์ โดยเมื่อตัวต้านทานแสงทำงานคือมีแสงมาตกกระทบที่ตัวต้านทานชนิดไวแสงจะส่งผลให้มอเตอร์ทำงาน แต่ถ้าไม่มีแสงมาตกกระทบที่ตัวต้านทานชนิดไวแสงนี้ มอเตอร์จะหยุดหมุนและส่งผลให้สายพานหยุดทำงาน วงจรการทำงานของตัวต้านทานชนิดไวแสงแสดงในรูปที่ 3.7



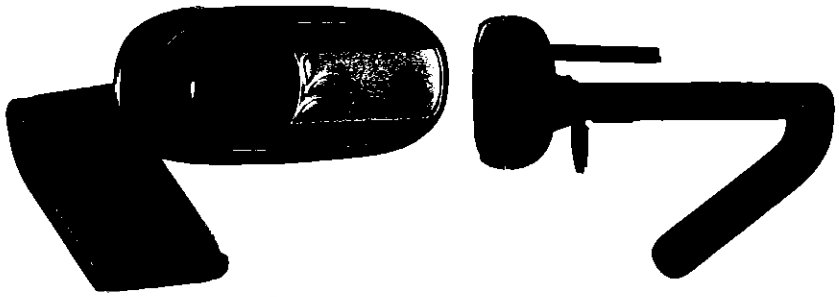
รูปที่ 3.7 วงจรตัวต้านทานชนิดไวแสง

รูปที่ 3.8 ตัวต้านทานชนิดไวแสง

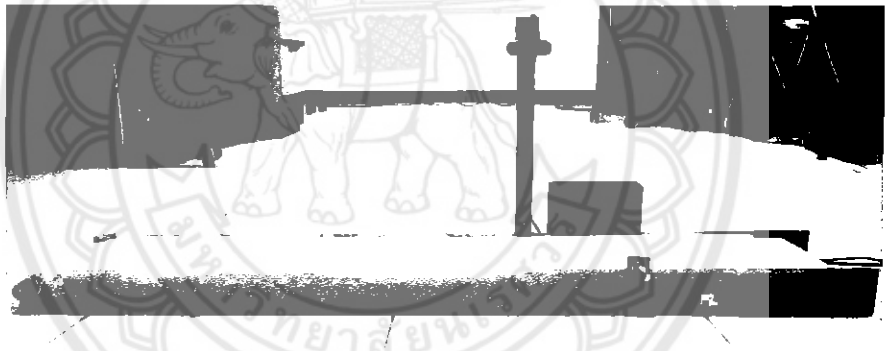
3.1.6 กล้องเว็บแคม (Webcam)

กล้องที่ใช้ในการจับภาพวัตถุในโครงงานนี้จะเป็นกล้องดิจิตอลยี่ห้อโลจิเทค (Logitech) รุ่น C170 ซึ่งมีความละเอียด 5 ล้านพิกเซล ดังแสดงในรูปที่ 3.9 โดยกล้องดิจิตอลนี้จะถูกติดตั้งอยู่ที่ส่วนปลายสุดของหุ่นยนต์แขนกล โดยที่หุ่นยนต์แขนกลจะนำมาวางไว้ติดกับรางของสายพานลำเลียงสิ่งของ

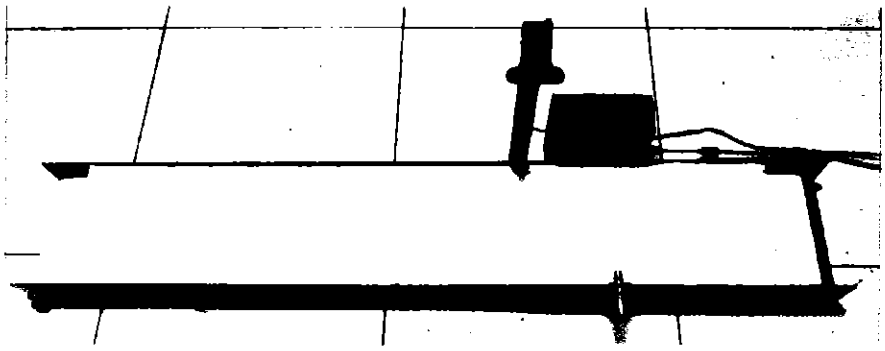
หมายเหตุ : แต่ในการทำการทดลองการประมวลผลภาพเพื่อจำแนกสีและจำแนกวัตถุนั้นจะนำกล้องไปติดไว้ที่จุดกึ่งกลางของรางของสายพานลำเลียงดังแสดงในรูปที่ 3.10 และรูปที่ 3.11



รูปที่ 3.9 ลักษณะของกล้องเวปแคมยี่ห้อโลจิเทค รุ่น C170



รูปที่ 3.10 ลักษณะของระบบตรวจจับวัตถุบนสายพานลำเลียงโดยใช้กล้องด้านหน้า



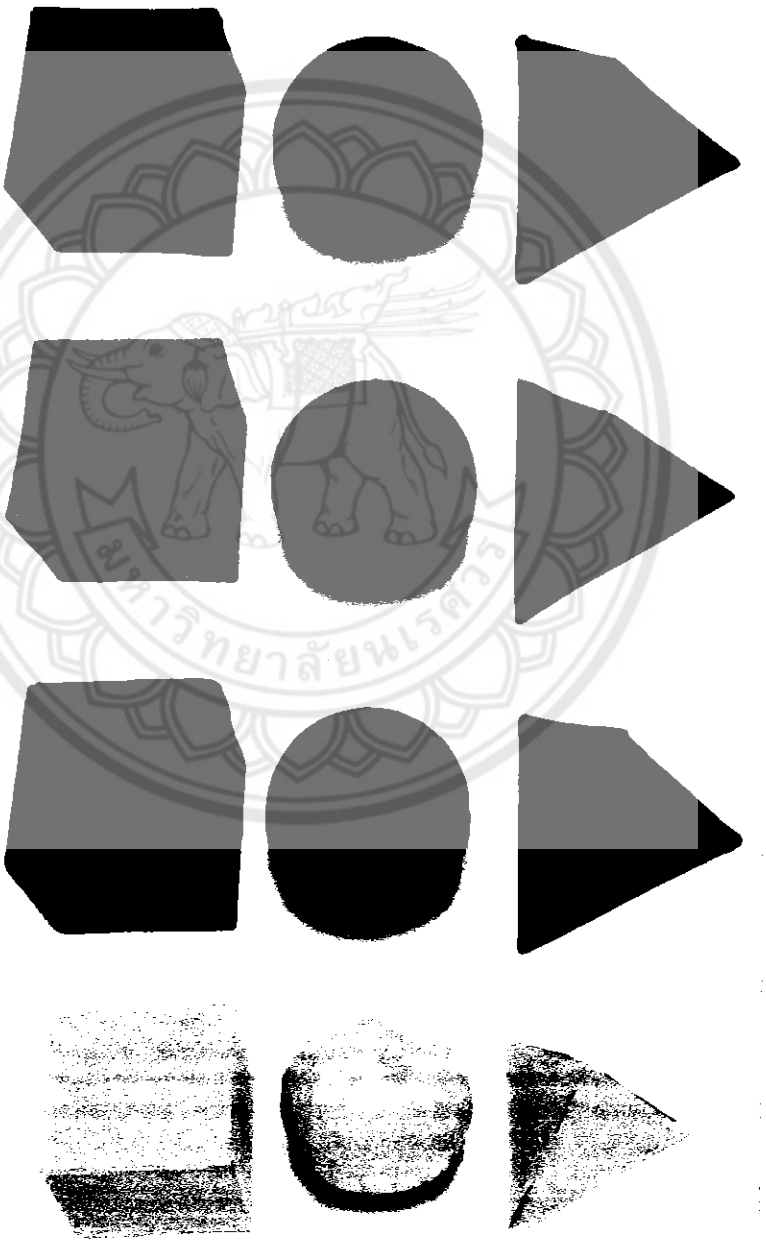
รูปที่ 3.11 ลักษณะของระบบตรวจจับวัตถุบนสายพานลำเลียงโดยใช้กล้องด้านบน

3.2 วัตถุที่นำมาตรวจสอบ

วัตถุที่นำมาตรวจสอบ สามารถแบ่งได้เป็น 2 กลุ่ม

3.2.1 จำแนกวัตถุตามสี

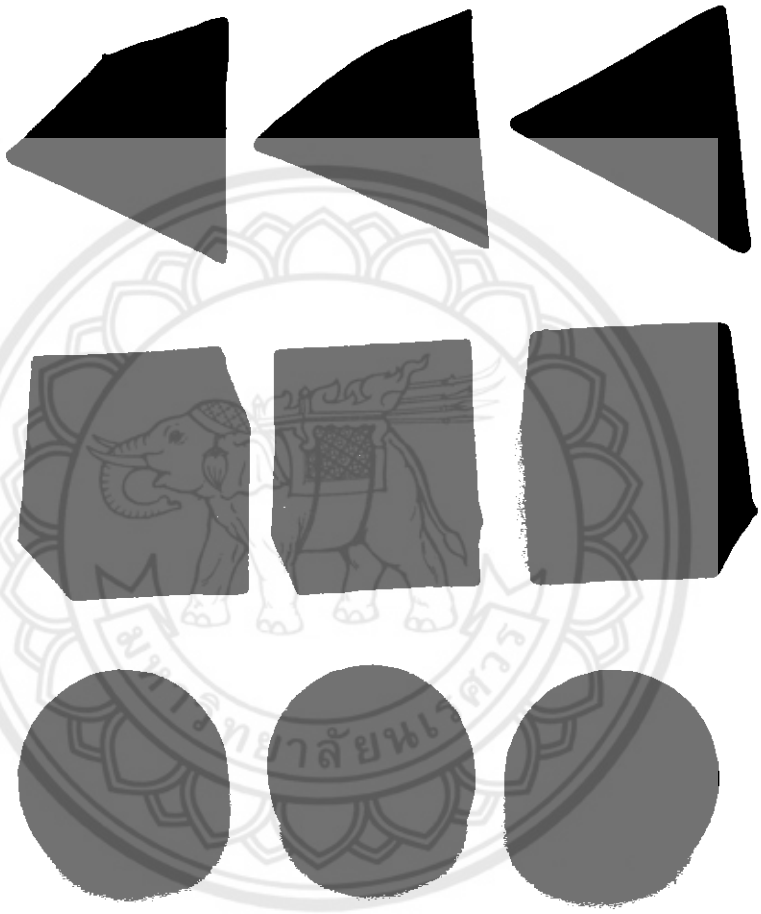
วัตถุในกลุ่มนี้ ประกอบไปด้วยวัตถุที่มีสีแตกต่างกันทั้งหมด 4 สี คือ สีแดง สีเขียว สีน้ำเงิน และสีเหลือง ดังแสดงในรูปที่ 3.12



รูปที่ 3.12 วัตถุที่นำมาตรวจสอบจำแนกตามสี

3.2.2 จำแนกวัตถุตามรูปร่าง

วัตถุที่เลือกนำมาใช้ในการจำแนก ประกอบไปด้วยวัตถุที่มีรูปร่างแตกต่างกันทั้งหมด 3 รูปร่าง คือ ทรงกลม ทรงสามเหลี่ยมด้านเท่า (ทรงปิรามิด) และทรงสี่เหลี่ยมจัตุรัส ดังแสดงในรูปที่ 3.13



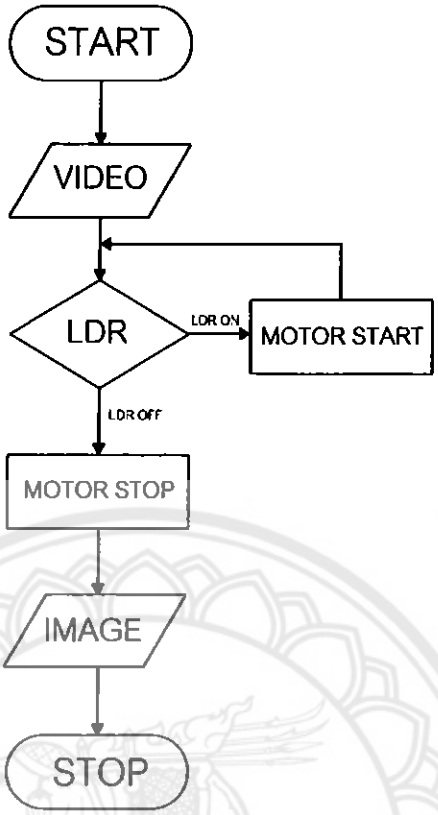
รูปที่ 3.13 วัตถุที่นำมาตรวจสอบจำแนกตามรูปร่าง

3.3 ขั้นตอนการทำงานของระบบตรวจจับวัตถุบนสายพานลำเลียงโดยใช้กล้อง

การทำงานของระบบตรวจจับวัตถุบนสายพานลำเลียงโดยใช้กล้องแบ่งออกเป็น 2 ส่วนหลัก คือส่วนของสายพานลำเลียง และส่วนของการประมวลผลภาพผ่านคอมพิวเตอร์

3.3.1 ระบบสายพานลำเลียง

ระบบการทำงานของสายพานลำเลียงสามารถแสดงได้ดังแผนภาพในรูปที่ 3.14

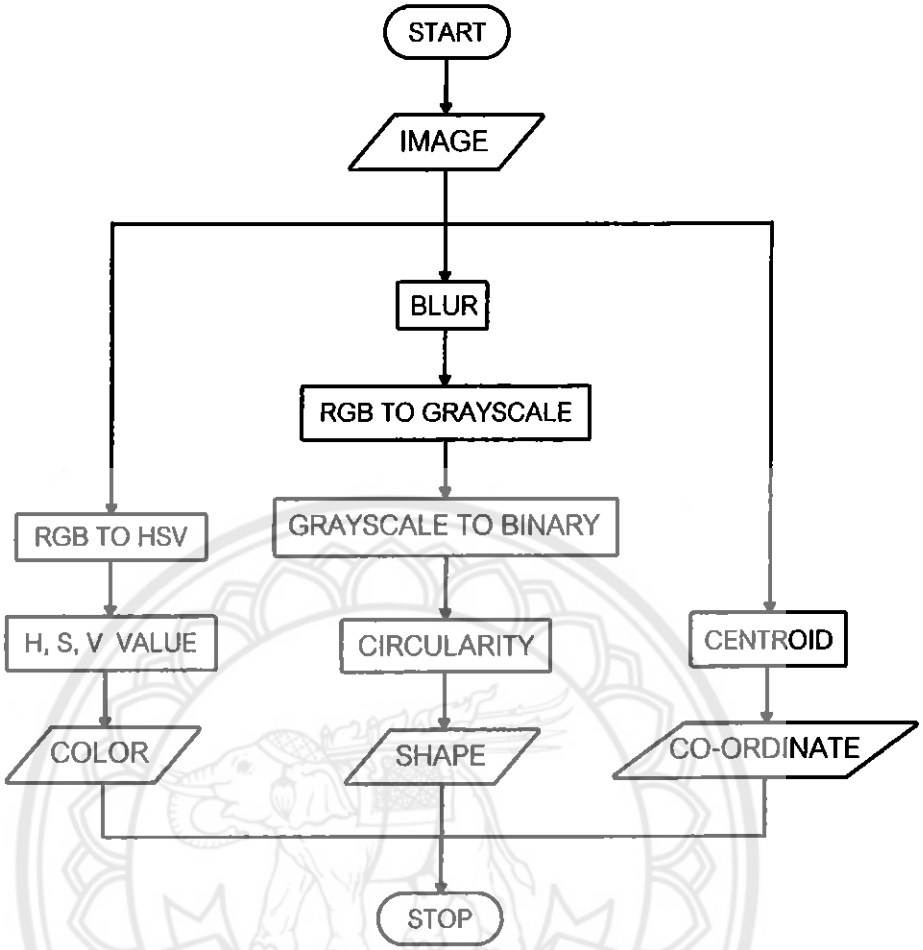


รูปที่ 3.14 แผนภาพแสดงขั้นตอนการทำงานของระบบสายพานลำเลียง

จากแผนภาพในรูปที่ 3.14 มีขั้นตอนการทำงานดังนี้ เมื่อเปิดเครื่องให้ระบบสายพานลำเลียงเริ่มทำงาน มอเตอร์จะเริ่มทำงานและทำให้ส่วนของสายพานเคลื่อนที่ไปข้างหน้า สายพานจะเคลื่อนที่ไปเรื่อยๆ จนกระทั่งระบบตรวจจับวัตถุด้วยตัวต้านทานชนิดไวแสงตรวจพบวัตถุและทำให้ตัวต้านทานชนิดไวแสงหยุดทำงานและส่งผลให้มอเตอร์หยุดการทำงานลงเช่นกันกัน พร้อมกับที่ตัวต้านทานชนิดไวแสง ส่งสัญญาณเข้าไปที่ระบบประมวลภาพให้ทำการจับภาพนิ่งเพื่อจะนำภาพวัตถุเข้าสู่กระบวนการการประมวลผลภาพต่อไป

3.3.2 ระบบประมวลผลภาพ

จากระบบสายพานลำเลียง เมื่อมีวัตถุเข้ามาและกล้องได้ทำการจับภาพวัตถุไว้แล้ว ภาพของวัตถุที่ได้จะถูกส่งเข้าสู่ระบบการประมวลผลภาพ จากนั้นระบบประมวลผลภาพก็จะเริ่มการประมวลผลจากภาพที่ได้ผ่านอัลกอริทึมภายใน และสุดท้ายจะได้เป็นตำแหน่งของวัตถุที่เข้ามาในสายพานลำเลียงสามารถแสดงได้ดังแผนภาพในรูปที่ 3.15



รูปที่ 3.15 แผนภาพแสดงขั้นตอนการทำงานของระบบประมวลผลภาพ

จากแผนภาพในรูปที่ 3.15 สามารถอธิบายกระบวนการย่อยของการประมวลผลภาพได้ดังต่อไปนี้ ซึ่งกระบวนการย่อยนี้จะมีทั้งหมด 3 กระบวนการย่อยคือ

1. กระบวนการจำแนกสี

- แปลงรูปแบบสีของภาพจากรูปแบบ RGB ให้เป็นรูปแบบ HSV และแยกพิจารณารูปภาพทั้งสามระนาบคือ ระนาบ H ระนาบ S และระนาบ V

- กำหนดค่าเทรชโฮลด์ (Threshold) สีของวัตถุสำหรับ 3 สี คือ สีแดง สีน้ำเงิน และสีเขียว บนระนาบ H ระนาบ S และระนาบ V เพื่อจำแนกสีที่ต้องการ

2. ขั้นตอนการจำแนกรูปร่าง

- ปรับปรุงคุณสมบัติของภาพโดยการทำให้ภาพเบลอ (blurred) เพื่อลดรายละเอียดของภาพ

- แปลงรูปแบบสีของภาพจากรูปแบบ RGB เป็นรูปแบบเกรย์ (Gray scale)

- แปลงรูปแบบสีของภาพจากรูปแบบเกรย์เป็นรูปแบบไบนารี (Binary)

- กำหนดเงื่อนไขการพิจารณาในการจำแนกรูปร่าง โดยจะจำแนกรูปร่างตามค่าความกลม (Circularity) โดยจะใช้สมการในการคำนวณดังนี้

$$\text{Circularity} = \frac{R^2}{4\pi A} \qquad (3.1)$$

เมื่อคำนวณค่าความกลมแล้ว จะนำค่าความกลมมาหารูปร่างของวัตถุโดยกำหนดให้มีรูปทรงวัตถุทั้งหมด 3 รูปแบบ คือ ทรงสามเหลี่ยมด้านเท่า หรือทรงพีระมิด ทรงสี่เหลี่ยมจัตุรัส และทรงกลม โดยจะพิจารณาค่าความกลมตามลำดับจากค่าความกลมน้อยไปหาค่าความกลมมาก

บทที่ 4

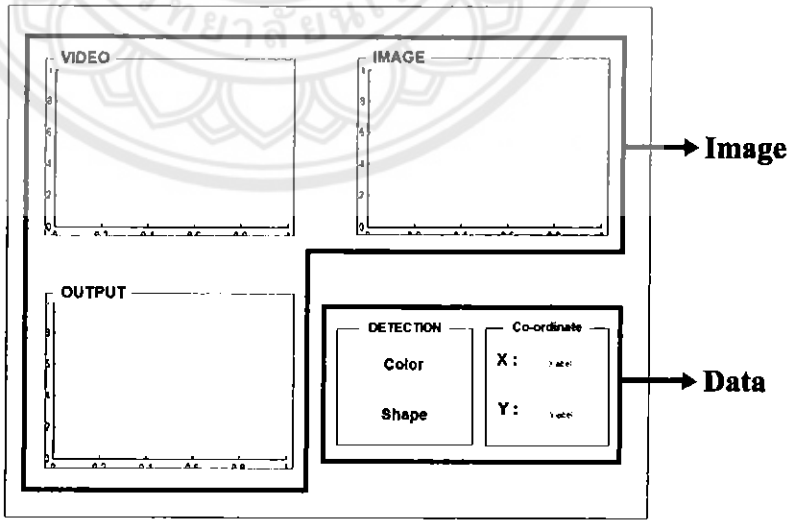
ผลการทดลอง

หลังจากสร้างระบบตรวจจับวัตถุบนสายพานลำเลียงโดยใช้กล้อง ผู้ดำเนินโครงงานได้ทำการทดสอบการทำงานของระบบตรวจจับวัตถุบนสายพานลำเลียงโดยใช้กล้อง โดยแบ่งการทดลองออกเป็น 5 การทดลองดังนี้

1. การทดลองการใช้งานโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ (GUI)
2. การทดลองการหาตำแหน่งของวัตถุ
3. การทดลองการจำแนกวัตถุในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์
4. การทดลองการจำแนกวัตถุในสภาพแสงจากแสงอาทิตย์
5. การทดลองการจำแนกวัตถุจากปริมาณความเข้มแสง

4.1 การทดลองการใช้งานโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ (GUI)

โปรแกรมส่วนต่อประสารกราฟิกกับผู้ใช้นี้ จะทำการแสดงผลจากกระบวนการการประมวลภาพตลอดทั้งกระบวนการทำงานของระบบ โดยจะแบ่งการใช้งานเป็น 2 ส่วน คือส่วนแสดงผลด้วยภาพและส่วนแสดงผลข้อมูล ดังแสดงในรูปที่ 4.1



รูปที่ 4.1 หน้าต่างแสดงการใช้งานของโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้

4.1.1 ส่วนแสดงผลด้วยภาพ (Image)

ในส่วนนี้จะทำแสดงผลของการประมวลผลภาพในรูปแบบของภาพเท่านั้น โดยจะแบ่งย่อยได้ 3 ส่วนด้วยกัน

1. การแสดงผลด้วยภาพเคลื่อนไหว (Video)

การแสดงผลด้วยภาพเคลื่อนไหว จะทำการแสดงภาพของระบบในเวลาปัจจุบัน ทำให้เห็นการเคลื่อนไหวของระบบตลอดการทำงาน

2. การแสดงผลด้วยภาพนิ่ง (Image)

เมื่อระบบสามารถตรวจจับได้ว่ามีวัตถุเข้ามาในระบบ กล้องจะทำการจับภาพนิ่งจากภาพเคลื่อนไหว เพื่อจะรอการนำไปประมวลผลต่อไป

3. การแสดงผลด้วยภาพขาวดำ (Binary Image)

ระบบจะทำการประมวลผลภาพที่ได้จากการจับภาพนิ่ง และแสดงภาพการจำแนกวัตถุออกจากพื้นหลัง พร้อมทั้งแสดงจุดศูนย์กลางรูปเราขาคณิต (Centroid) ของวัตถุ

4.1.2 ส่วนแสดงผลด้วยข้อมูล (Data)

ในส่วนนี้จะทำการแสดงผลของระบบประมวลภาพ โดยจะแบ่งย่อยได้เป็น 3 ส่วนด้วยกัน

1. การแสดงผลการจำแนกสี

ในการแสดงผลการจำแนกสีหลังจากผ่านกระบวนการประมวลผลภาพของวัตถุแล้ว จะมีการแสดงผลอยู่ทั้งหมด 4 รูปแบบด้วยกันคือ

- สีแดง (Red)
- สีเขียว (Green)
- สีน้ำเงิน (Blue)
- ไม่สามารถระบุสีได้ (Unidentified)

2. การแสดงผลการจำแนกรูปร่าง

ในการแสดงผลการจำแนกรูปร่าง หลังจากผ่านกระบวนการประมวลผลภาพของวัตถุแล้ว จะมีการแสดงผลอยู่ทั้งหมด 4 รูปแบบด้วยกันคือ

- รูปสามเหลี่ยม (Triangle)

- รูปสี่เหลี่ยม (Rectangular)

- รูปวงกลม (Circle)

- ไม่สามารถระบุรูปร่างได้ (*Unidentified*)

3. การแสดงผลของตำแหน่งวัตถุ

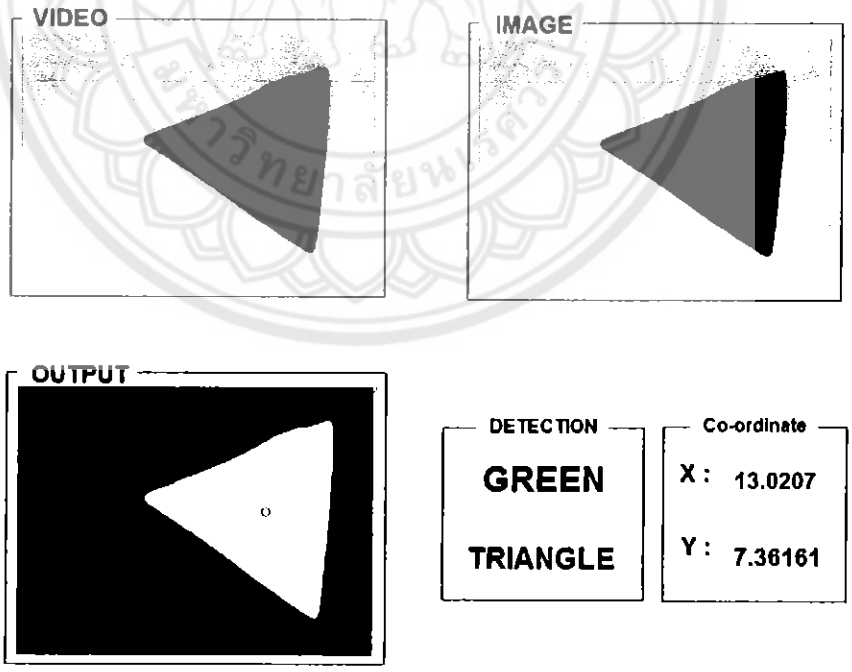
ในการแสดงผลการระบุตำแหน่งของวัตถุ จะทำการแสดงข้อมูลตัวเลข 2 จำนวนที่ระบุตำแหน่งของวัตถุที่วางบนระนาบ

- ตำแหน่งของวัตถุในแกนนอน (X-axes)

- ตำแหน่งของวัตถุในแกนตั้ง (Y-axes)

หมายเหตุ : โดยที่ตำแหน่ง (x,y) ที่หาได้จะเทียบกับจุดกำเนิดที่ตำแหน่งมุมบนซ้ายสุดของภาพ

และผลการประมวลผลภาพโดยผ่านทางโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้สามารถแสดงให้ดูเป็นตัวอย่างได้ในรูปที่ 4.2

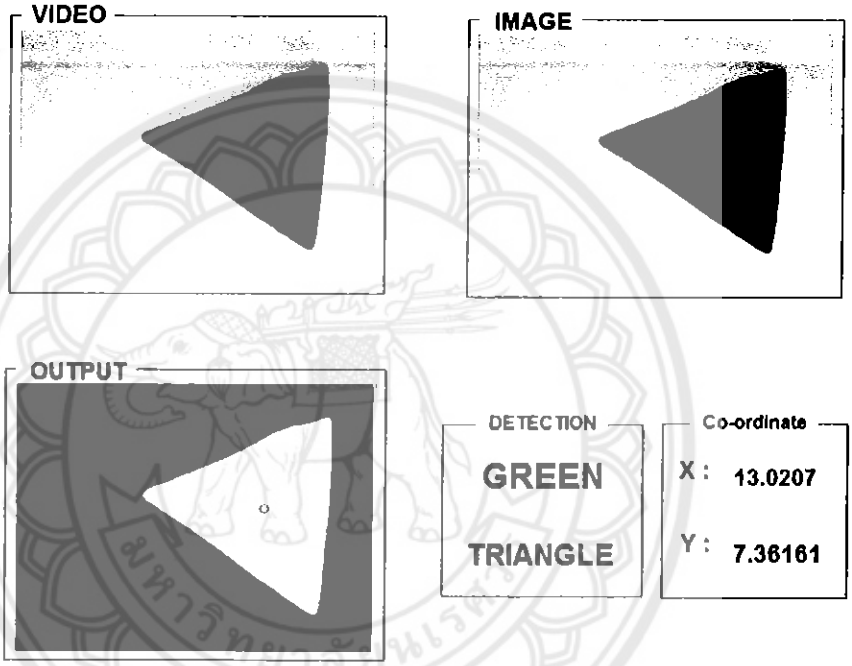


รูปที่ 4.2 ภาพแสดงผลลัพธ์ของการประมวลผลภาพผ่านทางโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้

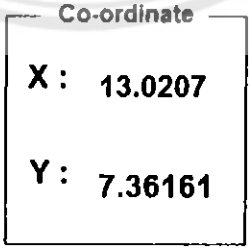
4.2 การทดลองการหาตำแหน่งของวัตถุ

4.2.1 ส่วนแสดงค่าตำแหน่งของวัตถุผ่านโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้

ผลการทดลองการหาตำแหน่งของวัตถุ จะดูแสดงผลอยู่ในรูปพิกัด แกนนอน(X-axes) และพิกัดแกนตั้ง (Y-axes) โดยจะแสดงผลในหน่วยเซนติเมตรในทั้งสองแกนดังแสดงในรูปที่ 4.3 และรูปที่ 4.4



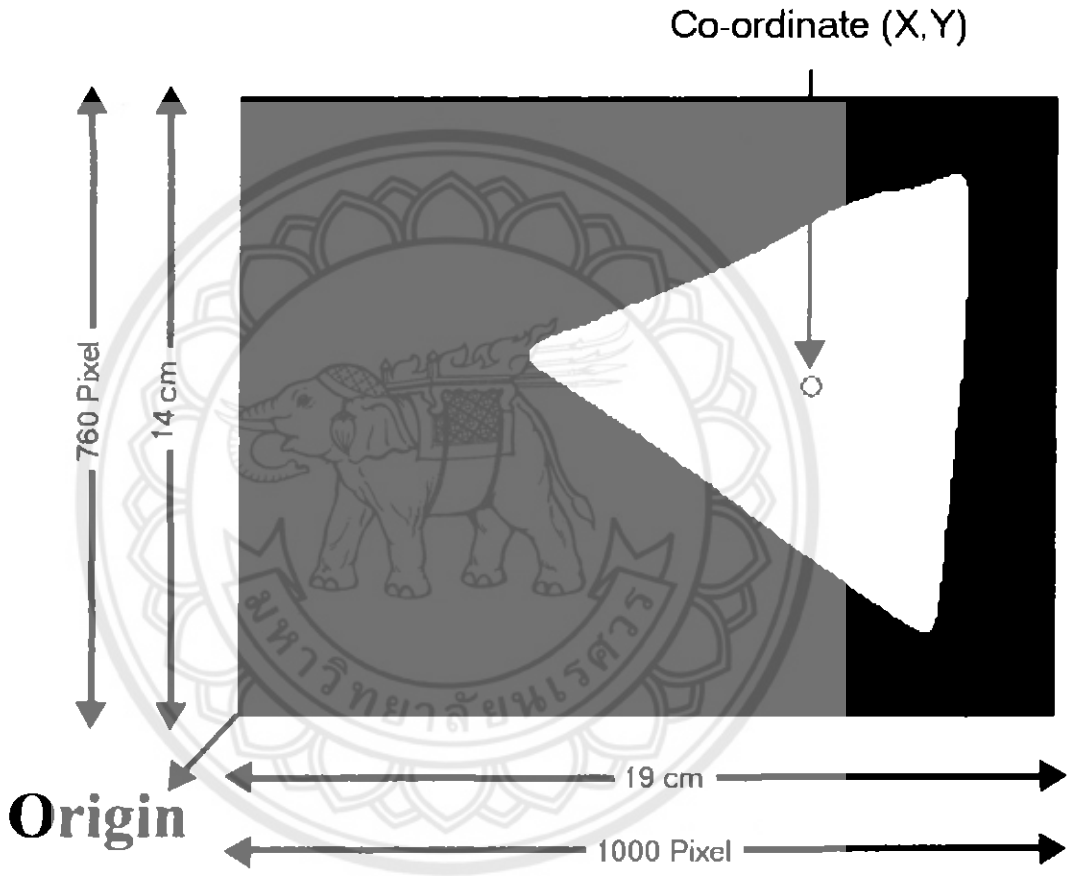
รูปที่ 4.3 รูปภาพแสดงผลของการประมวลผลภาพ



รูปที่ 4.4 รูปภาพแสดงผลของการประมวลผลภาพ ที่ระบุตำแหน่งของวัตถุ

4.2.2 การคำนวณหาค่าตำแหน่งของวัตถุ

การคำนวณหาตำแหน่งของวัตถุ เริ่มจากการแปลงจากหน่วยพิกเซลเป็นเซนติเมตร จากเงื่อนไขเริ่มต้นที่ภาพมีขนาด 1000 พิกเซลในแกนนอน และ 760 พิกเซลในการตั้ง และระยะทางจริงของวัตถุกับกล้องคือ 19 เซนติเมตรในแกนนอน และ 14 เซนติเมตรในแกนตั้ง ดังแสดงในรูป



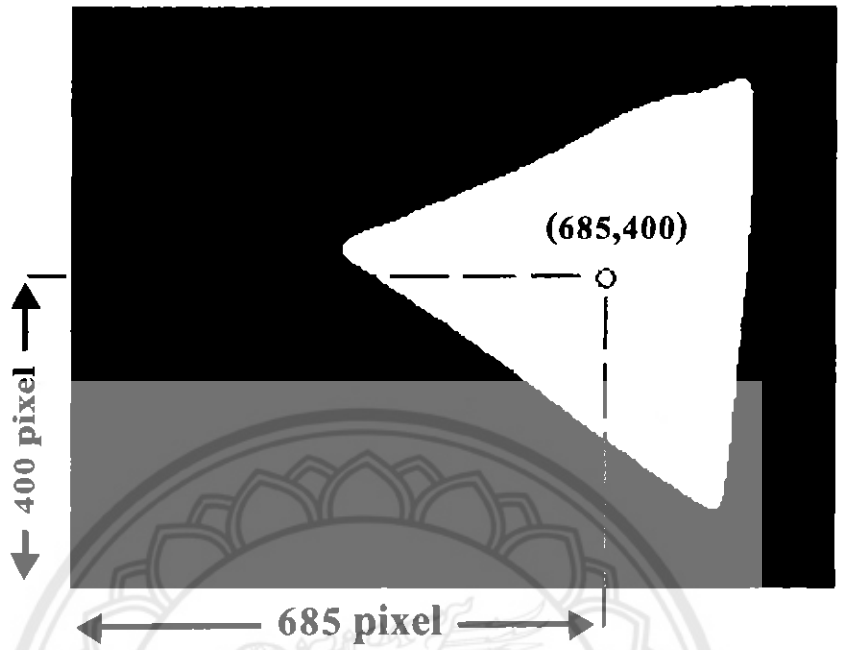
รูปที่ 4.5 รูปภาพแสดงการเปรียบเทียบระยะทางจริงในภาพกับระยะภาพในหน่วยพิกเซล

เพราะฉะนั้นจะสามารถเทียบระยะทางจริงของภาพจากขนาดของภาพได้คือ

$$\text{ระยะทางจริงในแกนนอน} = \frac{19}{1000} \times \text{ระยะทางในภาพในหน่วยพิกเซล}$$

$$\text{ระยะทางจริงในแกนตั้ง} = \frac{14}{760} \times \text{ระยะทางในภาพในหน่วยพิกเซล}$$

ตัวอย่างผลการทดลองการระบุตำแหน่งของวัตถุ

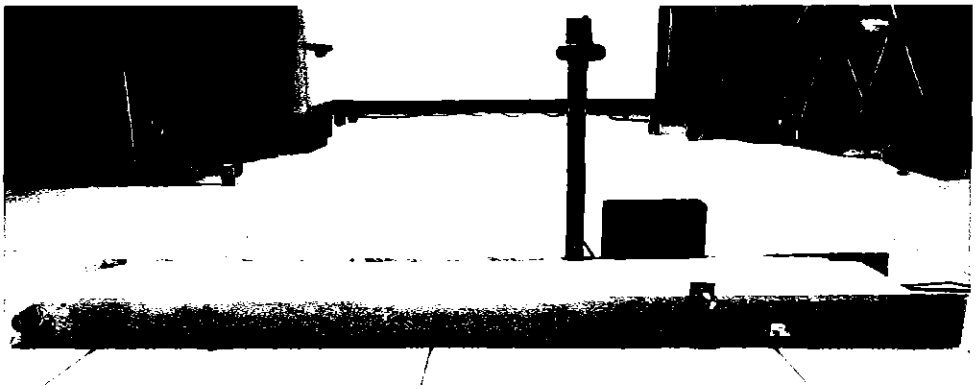


รูปที่ 4.6 ภาพแสดงตัวอย่างพิกัดในแกนนอนและแกนตั้งในหน่วยพิกเซล

มีระยะทางในภาพในหน่วยพิกเซลมีค่า 685 พิกเซลในแกนนอน และ 400 พิกเซลในแกนตั้ง เพราะฉะนั้นเมื่อนำค่าทั้งสองแทนในสมการ จะได้ตำแหน่งพิกัดของวัตถุในในแกนตั้งและแกนนอน คือ 13.02 เซนติเมตร และ 7.36 เซนติเมตร

4.3 การทดลองการจำแนกวัตถุในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์

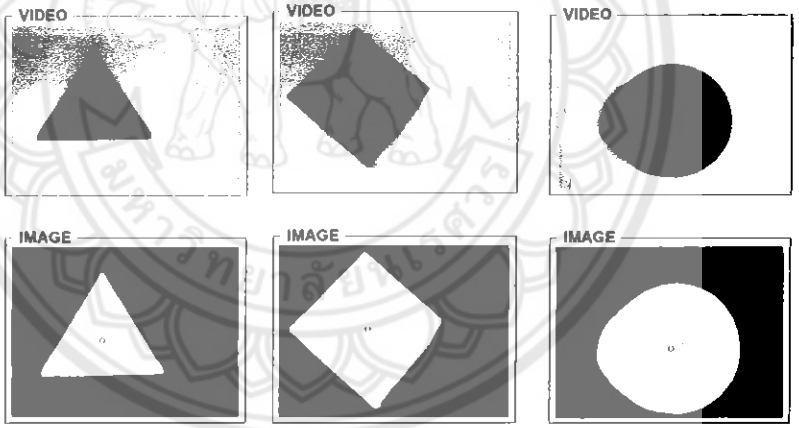
ในการทดลองการจำแนกวัตถุในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์ จะทดสอบโดยการจำแนกสีประกอบด้วย สีแดง สีเขียว สีน้ำเงินตามลำดับ และการจำแนกรูปร่างของวัตถุประกอบด้วยสามเหลี่ยม สี่เหลี่ยม วงกลมตามลำดับ ซึ่งทำการทดลองโดยวางอุปกรณ์การทดลองทั้งหมดภายใต้แสงของหลอดฟลูออเรสเซนต์ ขนาด 28 วัตต์ จำนวน 2 หลอดโดยทำการทดลอง 5 ครั้ง ซึ่งภาพการทดลองแสดงในรูปที่ 4.7



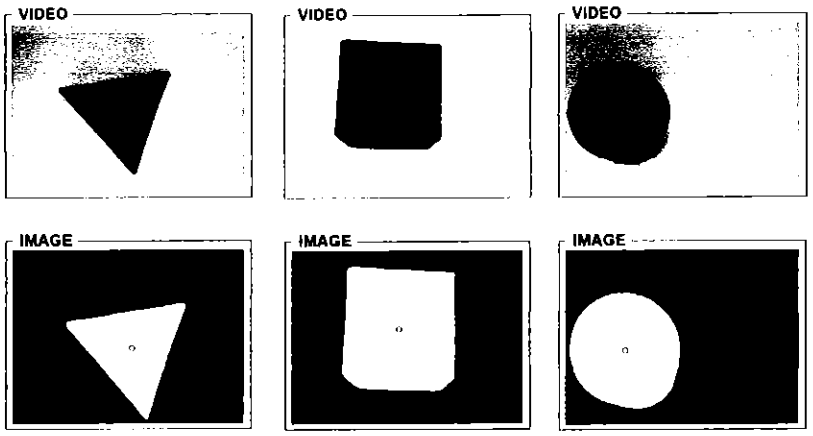
รูปที่ 4.7 ภาพการทดลองการจำแนกสีในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์

4.3.1 การทดลองการจำแนกสีของวัตถุในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์

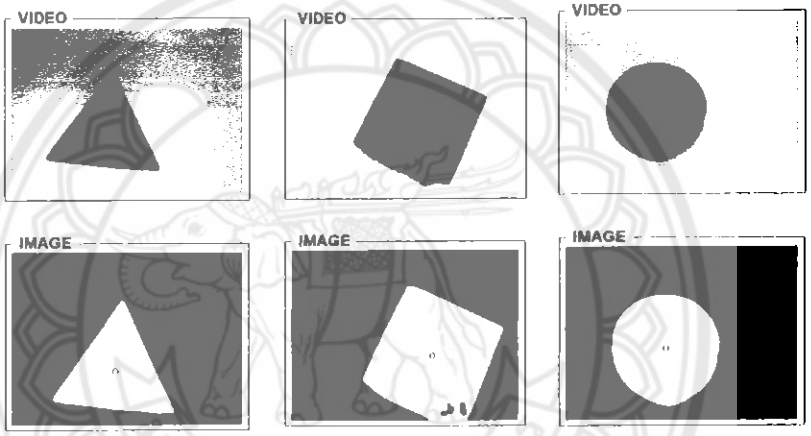
ในส่วนของการทดลองการจำแนกสีของวัตถุในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์จะทดลองทั้งหมด 3 สี คือ สีแดง สีน้ำเงินและสีเขียว ดังภาพผลการทดลองที่ได้แสดงในรูปที่ 4.8 รูปที่ 4.9 และรูปที่ 4.10 ตามลำดับ



รูปที่ 4.8 ภาพแสดงการจำแนกวัตถุสีแดงในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์



รูปที่ 4.9 ภาพแสดงการจำแนกวัตถุสีน้ำเงินในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์



รูปที่ 4.10 ภาพแสดงการจำแนกวัตถุสีเขียวในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์

ในการจำแนกสีต่างๆ แต่ละสีนั้นจะมีการทดลองทั้งหมด 5 ครั้ง และได้ผลการ**ทดลองดังแสดงใน ตารางที่** 4.1

ตารางที่ 4.1 ผลการทดลองการจำแนกสีของวัตถุในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์

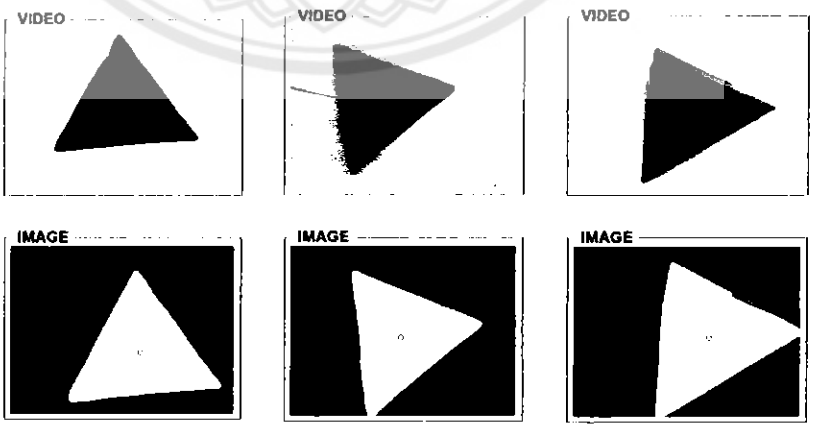
ครั้งที่	สีแดง			สีเขียว			สีน้ำเงิน		
	ทรงสามเหลี่ยม	ทรงสี่เหลี่ยม	ทรงกลม	ทรงสามเหลี่ยม	ทรงสี่เหลี่ยม	ทรงกลม	ทรงสามเหลี่ยม	ทรงสี่เหลี่ยม	ทรงกลม
1	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	✓	✓	✓	✓	✓	✓	✓	✓	✓
4	✓	✓	✓	✓	✓	✓	✓	✓	✓
5	✓	✓	✓	✓	✓	✓	✓	✓	✓

หมายเหตุ : สัญลักษณ์ในตารางแสดงการจำแนกสีดังนี้

✓ หมายถึง สามารถจำแนกได้ และ ✗ หมายถึง ไม่สามารถจำแนกได้

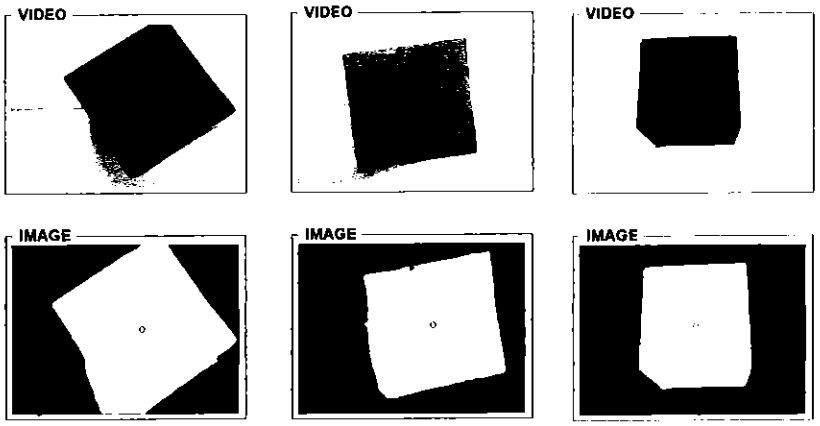
4.3.2 การทดลองการจำแนกรูปร่างของวัตถุในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์

ในส่วนของการทดลองการจำแนกรูปร่างของวัตถุในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์จะทำการทดลองโดยใช้วัตถุทั้งหมด 3 รูปร่าง คือ ทรงสามเหลี่ยมด้านเท่า ทรงสี่เหลี่ยมจัตุรัส และทรงกลม ดังภาพผลการทดลองที่ได้แสดงในรูปที่ 4.11 รูปที่ 4.12 และรูปที่ 4.13 ตามลำดับ

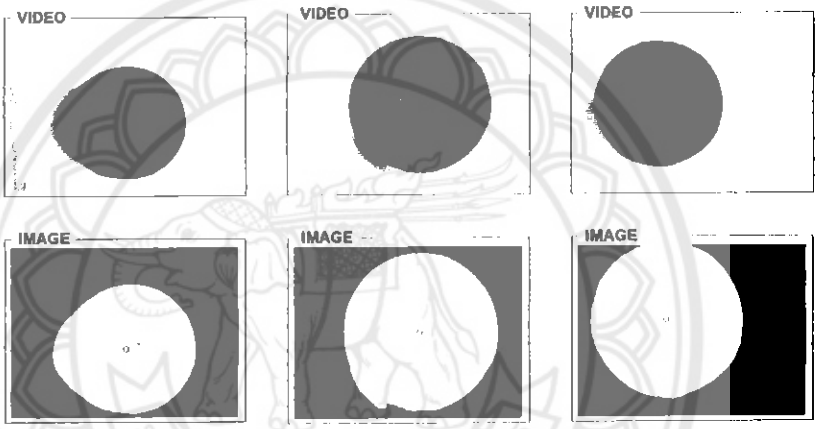


รูปที่ 4.11 ภาพแสดงการจำแนกวัตถุทรงสามเหลี่ยมด้านเท่า

ในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์



รูปที่ 4.12 ภาพแสดงการจำแนกวัตถุทรงสี่เหลี่ยมจัตุรัสในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์\



รูปที่ 4.13 ภาพแสดงการจำแนกวัตถุทรงกลมในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์

ในการจำแนกรูปร่างต่างๆ ของวัตถุนั้นจะมีการทดลองทั้งหมด 5 ครั้ง และได้ผลการทดลองดังแสดงในตารางที่ 4.2

ตารางที่ 4.2 ผลการทดลองการจำแนกรูปร่างในสภาพแสงจากหลอดไฟฟลูออเรสเซนต์

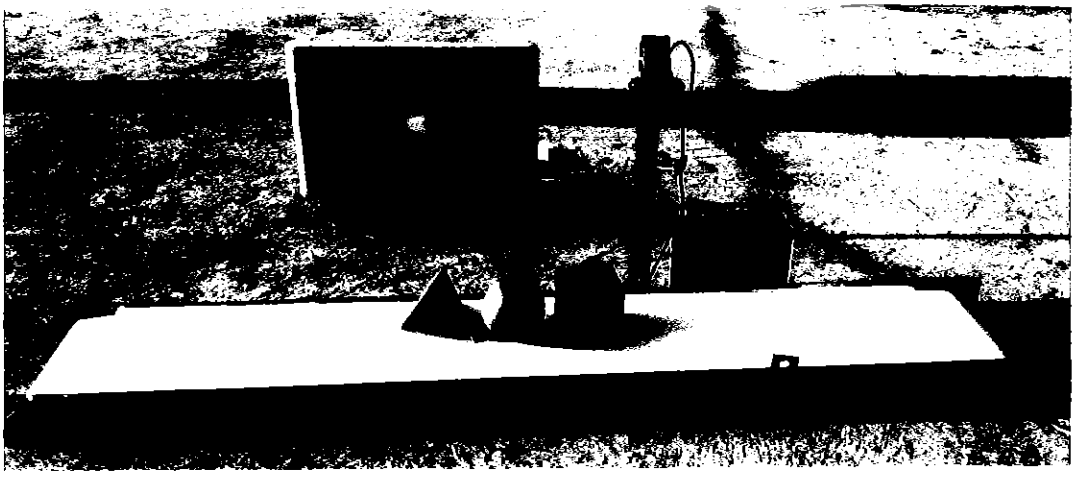
ครั้งที่	ทรงสามเหลี่ยม			ทรงสี่เหลี่ยม			ทรงกลม		
	สีแดง	สีเขียว	สีน้ำเงิน	สีแดง	สีเขียว	สีน้ำเงิน	สีแดง	สีเขียว	สีน้ำเงิน
1	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	✓	✓	✓	✓	✓	✓	✓	✓	✓
4	✓	✓	✓	✓	✓	✓	✓	✓	✓
5	✓	✓	✓	✓	✓	✓	✓	✓	✓

หมายเหตุ : สัญลักษณ์ในตารางแสดงการจำแนกรูปร่างดังนี้

✓ หมายถึง สามารถจำแนกได้ และ ✗ หมายถึง ไม่สามารถจำแนกได้

4.4 การทดลองการจำแนกวัตถุในสภาพแสงจากดวงอาทิตย์

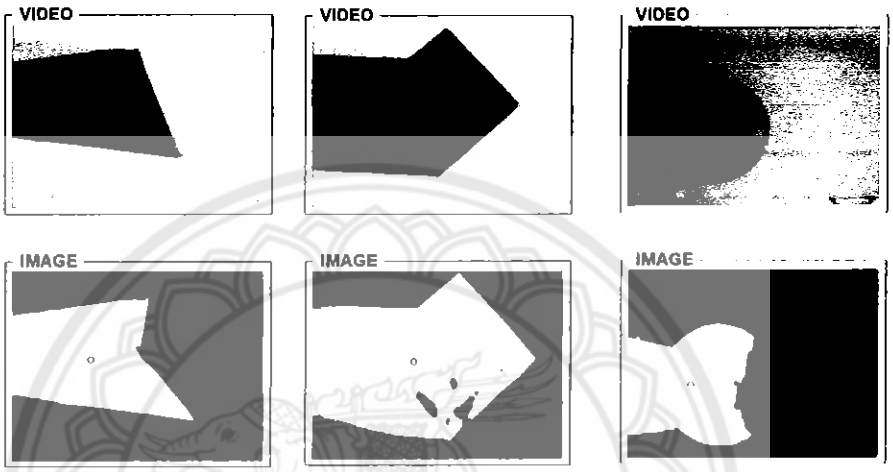
ในการทดลองการจำแนกวัตถุในสภาพแสงจากดวงอาทิตย์จะทดสอบโดยการจำแนกสี ประกอบด้วย สีแดง สีเขียว สีน้ำเงินตามลำดับ และจำแนกรูปร่างของวัตถุซึ่งประกอบด้วยทรงสี่เหลี่ยมจัตุรัส ทรงสามเหลี่ยมด้านเท่า และทรงกลมตามลำดับ ซึ่งทำการทดลองโดยวางอุปกรณ์ทุกอย่างไว้กลางแสงแดด และภาพการทดลองเมื่อทำการทดลองในสภาพแสงจากดวงอาทิตย์แสดงได้ในรูปที่ 4.14



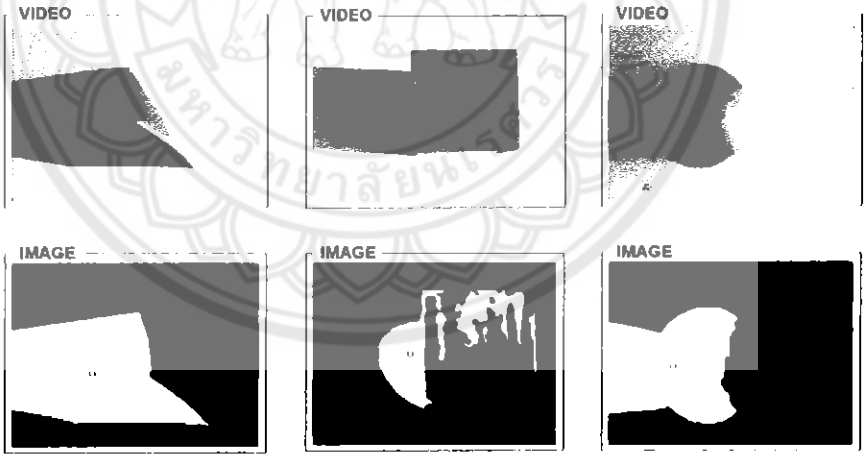
รูปที่ 4.14 ภาพแสดงการทดลองการจำแนกวัตถุในสภาพแสงจากดวงอาทิตย์

4.4.1 การทดลองการจำแนกสีของวัตถุในสภาพแสงจากแสงจากดวงอาทิตย์

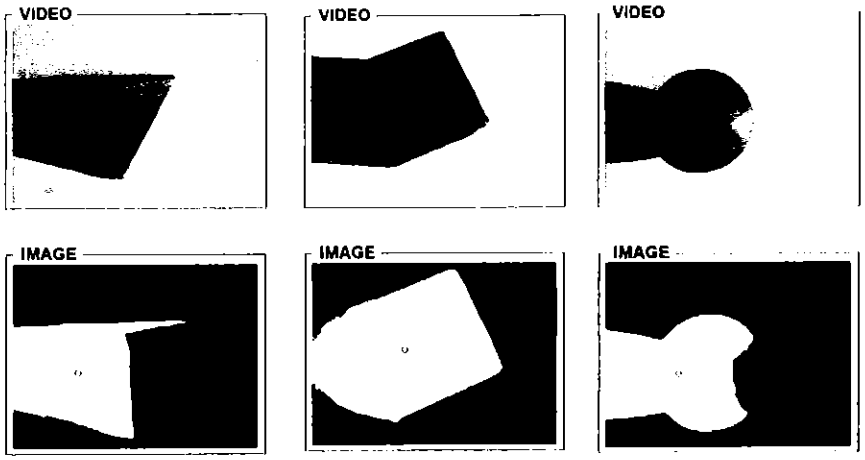
ในส่วนของการทดลองการจำแนกสีของวัตถุในสภาพแสงจากดวงอาทิตย์ จะทำการทดลองทั้งหมด 3 สี คือ สีแดง สีน้ำเงินและสีเขียว ดังภาพผลการทดลองที่ได้แสดงในรูปที่ 4.15 รูปที่ 4.16 และรูปที่ 4.17 ตามลำดับ



รูปที่ 4.15 ภาพแสดงการจำแนกวัตถุสีแดงในสภาพแสงจากดวงอาทิตย์



รูปที่ 4.16 ภาพแสดงการจำแนกวัตถุสีเขียวในสภาพแสงจากดวงอาทิตย์



รูปที่ 4.17 ภาพแสดงการจำแนกวัตถุสีน้ำเงินในสภาพแสงจากดวงอาทิตย์

ในการจำแนกสีต่างๆ แต่ละสีนั้นจะมีการทดลองทั้งหมด 5 ครั้ง และได้ผลการทดลองดังแสดงในตารางที่ 4.3

ตารางที่ 4.3 ผลการทดลองการจำแนกสีของวัตถุในสภาพแสงจากดวงอาทิตย์

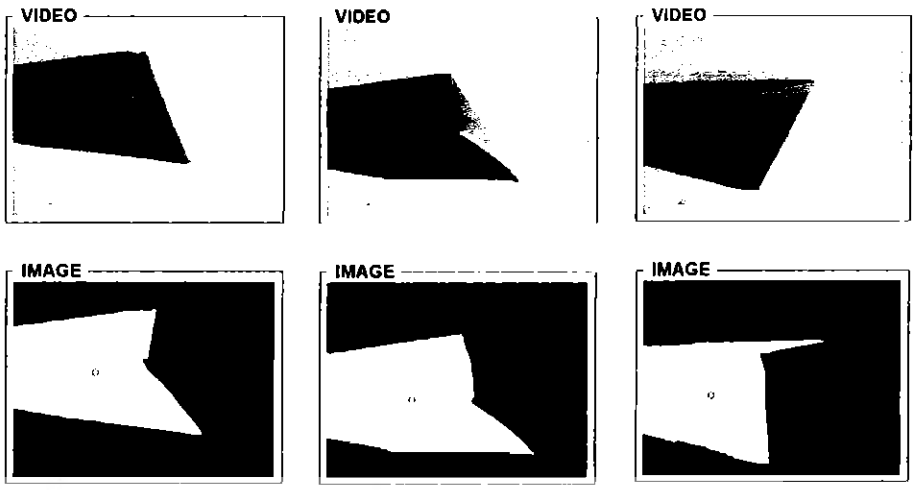
ครั้งที่	สีแดง			สีเขียว			สีน้ำเงิน		
	ทรงสามเหลี่ยม	ทรงสี่เหลี่ยม	ทรงกลม	ทรงสามเหลี่ยม	ทรงสี่เหลี่ยม	ทรงกลม	ทรงสามเหลี่ยม	ทรงสี่เหลี่ยม	ทรงกลม
1	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	✓	✓	✓	✓	✓	✓	✓	✓	✓
4	✓	✓	✓	✓	✓	✓	✓	✓	✓
5	✓	✓	✓	✓	✓	✓	✓	✓	✓

หมายเหตุ : สัญลักษณ์ในตารางแสดงการจำแนกสีดังนี้

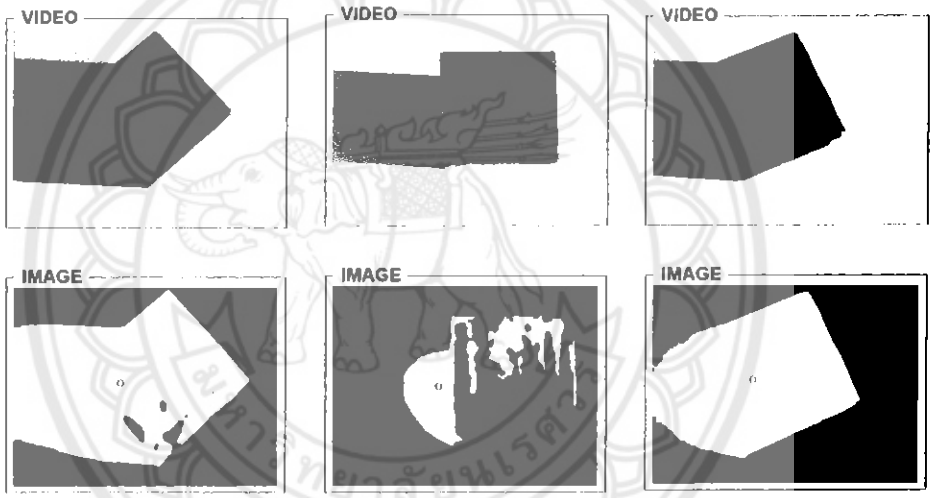
✓ หมายถึง สามารถจำแนกได้ และ ✗ หมายถึง ไม่สามารถจำแนกได้

4.4.2 การทดลองการจำแนกรูปร่างของวัตถุในสภาพแสงจากแสงจากดวงอาทิตย์

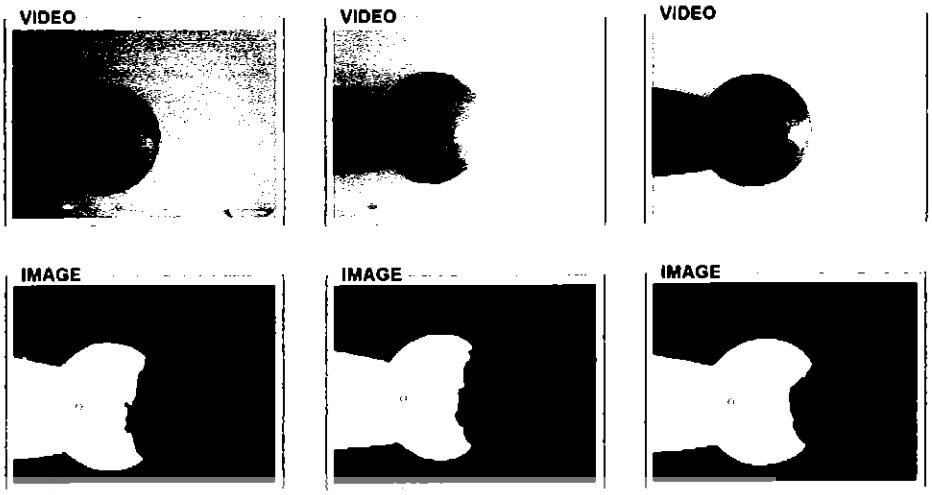
ในส่วนของการทดลองการจำแนกรูปร่างของวัตถุในสภาพแสงจากดวงอาทิตย์ จะทำการทดลองโดยใช้วัตถุทั้งหมด 3 รูปร่าง คือ ทรงสามเหลี่ยมด้านเท่า ทรงสี่เหลี่ยมจัตุรัส และทรงกลม ดังภาพผลการทดลองที่ได้แสดงในรูปที่ 4.18 รูปที่ 4.19 และรูปที่ 4.20 ตามลำดับ



รูปที่ 4.18 ภาพแสดงการจำแนกวัตถุทรงสามเหลี่ยมด้านเท่าในสภาพแสงจากดวงอาทิตย์



รูปที่ 4.19 ภาพแสดงการจำแนกวัตถุทรงสี่เหลี่ยมจัตุรัสในสภาพแสงจากดวงอาทิตย์



รูปที่ 4.20 ภาพแสดงการจำแนกวัตถุทรงกลมในสภาพแสงจากดวงอาทิตย์

ในการจำแนกรูปร่างต่างๆ ของวัตถุนั้นจะมีการทดลองทั้งหมด 5 ครั้ง และได้ผลการทดลองดังแสดงในตารางที่ 4.4

ตารางที่ 4.4 ผลการทดลองการจำแนกรูปร่างของวัตถุในสภาพแสงจากดวงอาทิตย์

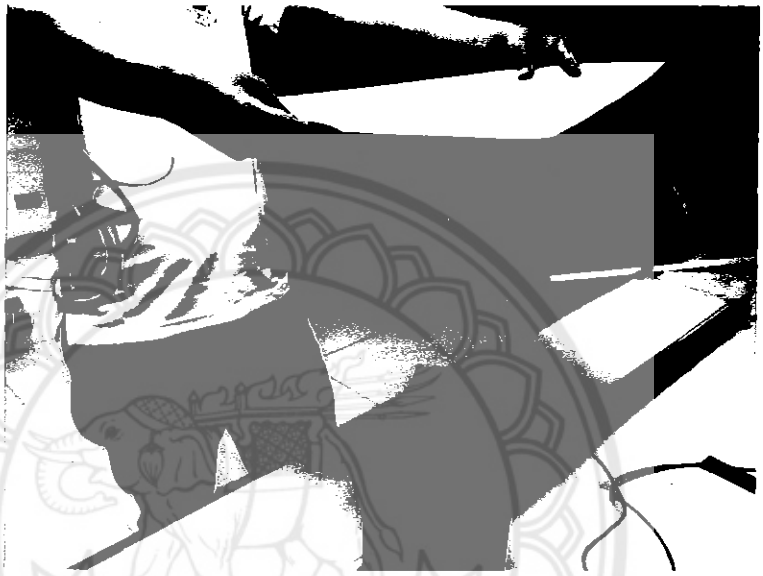
ครั้งที่	ทรงสามเหลี่ยม			ทรงสี่เหลี่ยม			ทรงกลม		
	สีแดง	สีเขียว	สีน้ำเงิน	สีแดง	สีเขียว	สีน้ำเงิน	สีแดง	สีเขียว	สีน้ำเงิน
1	×	×	×	×	×	×	×	×	×
2	×	×	×	×	×	×	×	×	×
3	×	×	×	×	×	×	×	×	×
4	×	×	×	×	×	×	×	×	×
5	×	×	×	×	×	×	×	×	×

หมายเหตุ : สัญลักษณ์ในตารางแสดงการจำแนกรูปร่างดังนี้

✓ หมายถึง สามารถจำแนกได้ และ × หมายถึง ไม่สามารถจำแนกได้

4.5 การทดลองการจำแนกรูปทรงของวัตถุจากความเข้มแสง

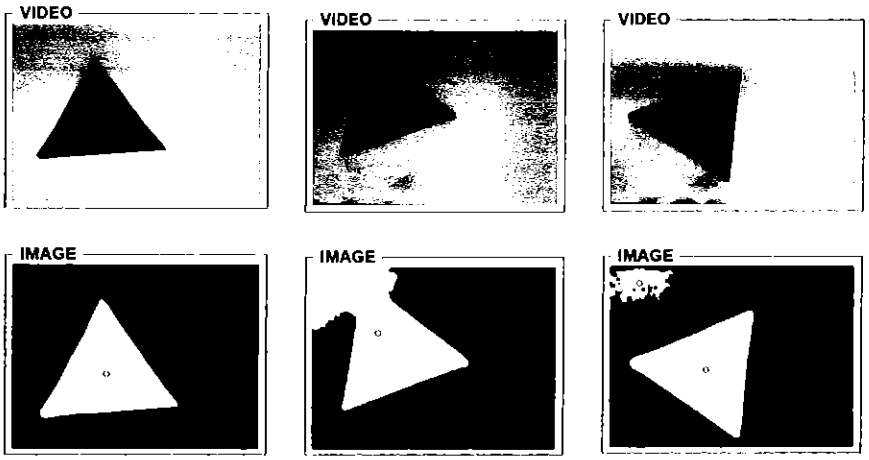
ในการทดลองการจำแนกวัตถุจากความเข้มแสง จะทดสอบโดยการใช้กระดาษไขในการกรองแสงเพื่อลดปริมาณความเข้มแสงที่จะเข้ามาในระบบ และจะทำการเพิ่มจำนวนกระดาษไขเพื่อทำให้แสงผ่านเข้ามาในระบบได้น้อยลง โดยจำนวนกระดาษไขที่ใช้จะมี 1 แผ่น 3 แผ่น 5 แผ่น และเพิ่มครั้งละ 5แผ่น



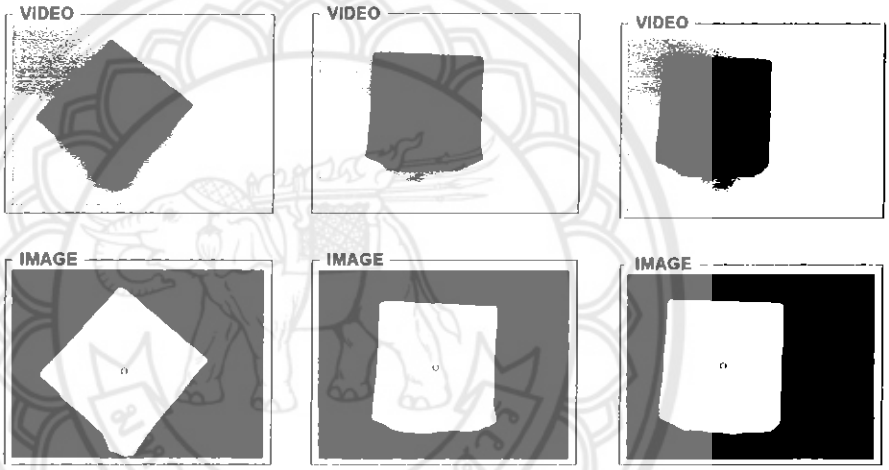
รูปที่ 4.21 ภาพแสดงการทดลองโดยใช้กระดาษไข

4.5.1 การทดลองด้วยการใช้กระดาษไข จำนวน 1 แผ่น

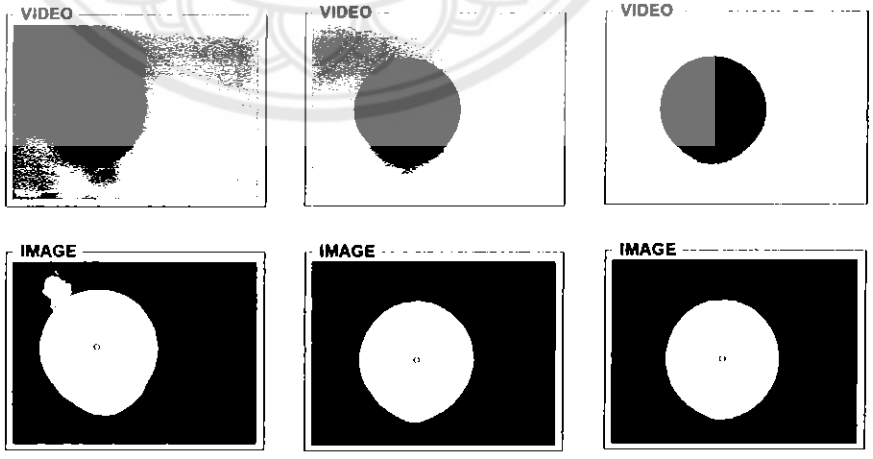
การใช้กระดาษไขจำนวน 1 แผ่น สามารถแสดงผลการทดลองในการจำแนกรูปทรงของวัตถุสำหรับวัตถุทรงสามเหลี่ยมด้านเท่า รูปทรงสี่เหลี่ยมจัตุรัส และทรงกลม ดังในรูปที่ 4.22 รูปที่ 4.23 และรูปที่ 4.24 ตามลำดับ



รูปที่ 4.22 ภาพแสดงการจำแนกวัตถุทรงสามเหลี่ยมด้านเท่าโดยใช้กระดาษไข 1 แผ่น



รูปที่ 4.23 ภาพแสดงการจำแนกวัตถุทรงสี่เหลี่ยมจัตุรัสโดยใช้กระดาษไข 1 แผ่น



รูปที่ 4.24 ภาพแสดงการจำแนกวัตถุทรงกลมโดยใช้กระดาษไข 1 แผ่น

ในการจำแนกรูปทรงต่างๆ ของวัตถุเมื่อใช้กระดาษไขจำนวน 1 แผ่น นั้นจะมีการทดลองทั้งหมด 5 ครั้ง และได้ผลการทดลองดังแสดงในตารางที่ 4.5

ตารางที่ 4.5 ผลการทดลองการจำแนกรูปทรงวัตถุจากความเข้มแสงโดยใช้กระดาษไข 1 แผ่น

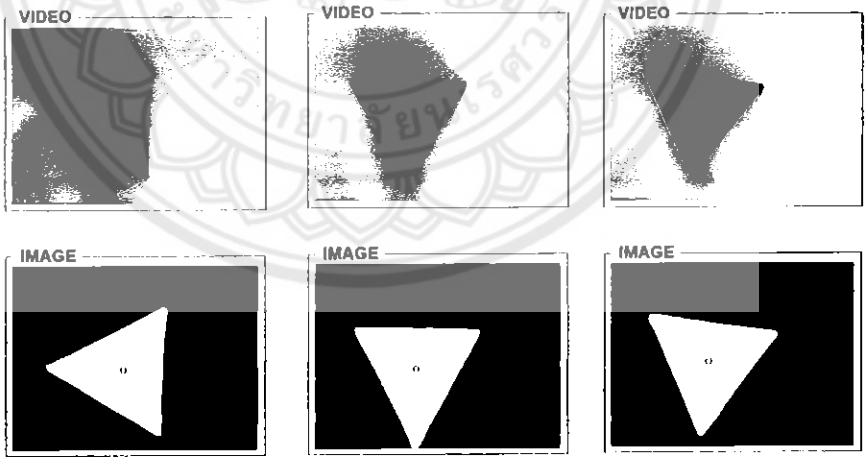
ครั้งที่	ทรงสามเหลี่ยม			ทรงสี่เหลี่ยม			ทรงกลม		
	สีแดง	สีเขียว	สีน้ำเงิน	สีแดง	สีเขียว	สีน้ำเงิน	สีแดง	สีเขียว	สีน้ำเงิน
1	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	✓	✓	✓	✓	✓	✓	✓	✓	✓
4	✓	✓	✓	✓	✓	✓	✓	✓	✓
5	✓	✓	✓	✓	✓	✓	✓	✓	✓

หมายเหตุ : สัญลักษณ์ในตารางแสดงการจำแนกรูปร่างดังนี้

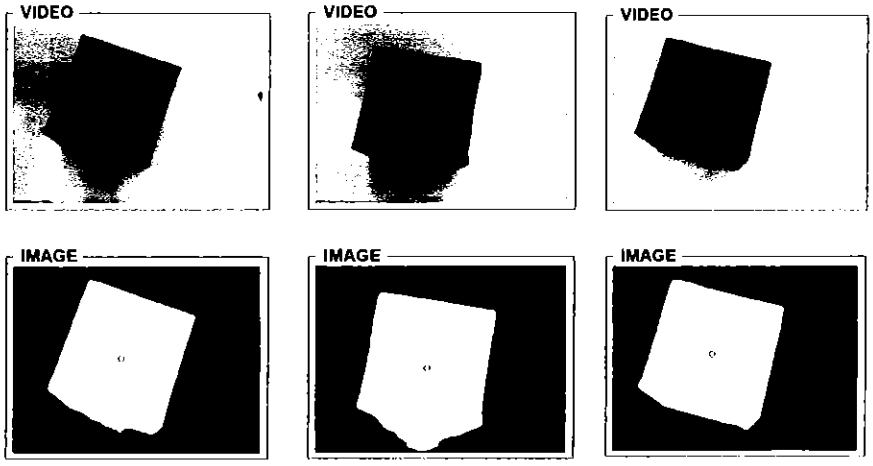
✓ **หมาย**ถึง สามารถจำแนกได้ และ ✗ หมายถึง ไม่สามารถจำแนกได้

4.5.2 การทดลองด้วยการใช้กระดาษไข จำนวน 3 แผ่น

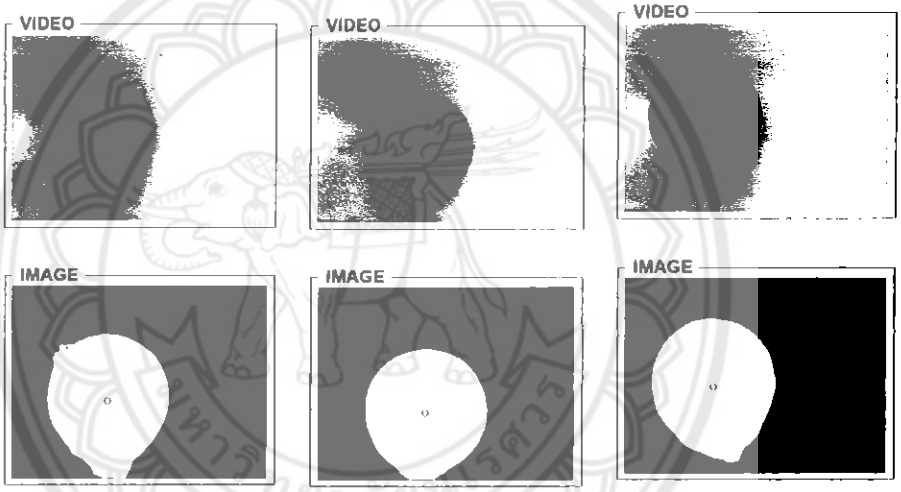
การใช้กระดาษไขจำนวน 3 แผ่น สามารถแสดงผลการทดลองในการจำแนกรูปทรงของวัตถุสำหรับวัตถุทรงสามเหลี่ยมด้านเท่า รูปทรงสี่เหลี่ยมจัตุรัส และทรงกลม ดังในรูปที่ 4.25 รูปที่ 4.26 และรูปที่ 4.27 ตามลำดับ



รูปที่ 4.25 ภาพแสดงการจำแนกวัตถุทรงสามเหลี่ยมด้านเท่าโดยใช้กระดาษไข 3 แผ่น



รูปที่ 4.26 ภาพแสดงการจำแนกวัตถุทรงสี่เหลี่ยมจัตุรัสโดยใช้กระดาษไข 3 แผ่น



รูปที่ 4.27 ภาพแสดงการจำแนกวัตถุทรงกลมโดยใช้กระดาษไข 3 แผ่น

ในการจำแนกรูปทรงต่างๆ ของวัตถุเมื่อใช้กระดาษไขจำนวน 3 แผ่น นั้นจะมีการทดลองทั้งหมด 5 ครั้ง และได้ผลการทดลองดังแสดงในตารางที่ 4.6

ตารางที่ 4.6 ผลการทดลองการจำแนกรูปทรงวัตถุจากความเข้มแสงโดยใช้กระดาษไข 3 แผ่น

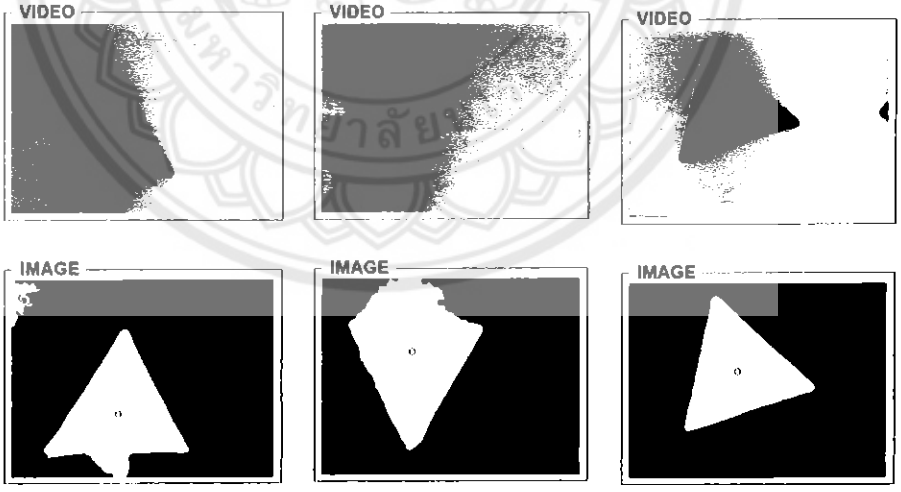
ครั้งที่	ทรงสามเหลี่ยม			ทรงสี่เหลี่ยม			ทรงกลม		
	สีแดง	สีเขียว	สีน้ำเงิน	สีแดง	สีเขียว	สีน้ำเงิน	สีแดง	สีเขียว	สีน้ำเงิน
1	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	✓	✓	✓	✓	✓	✓	✓	✓	✓
4	✓	✓	✓	✓	✓	✓	✓	✓	✓
5	✓	✓	✓	✓	✓	✓	✓	✓	✓

หมายเหตุ : สัญลักษณ์ในตารางแสดงการจำแนกรูปร่างดังนี้

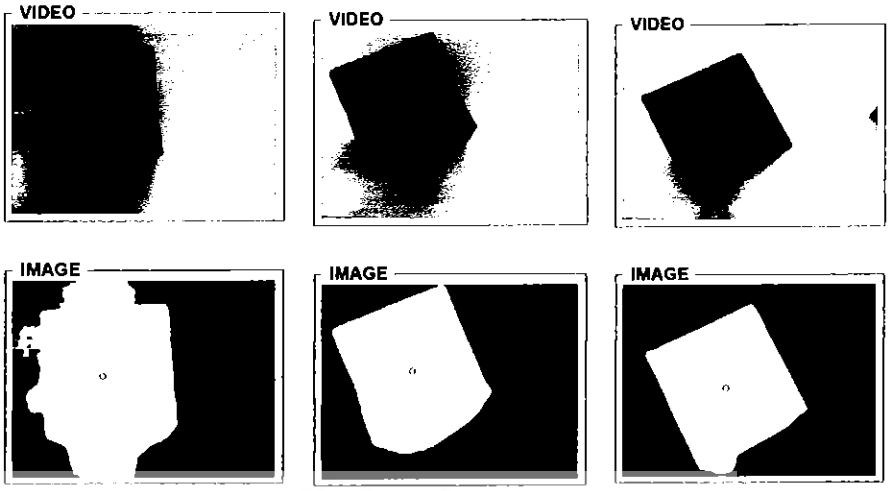
✓ **หมาย**ถึง สามารถจำแนกได้ และ ✗ หมายถึง ไม่สามารถจำแนกได้

4.5.3 การทดลองด้วยการใช้กระดาษไข จำนวน 5 แผ่น

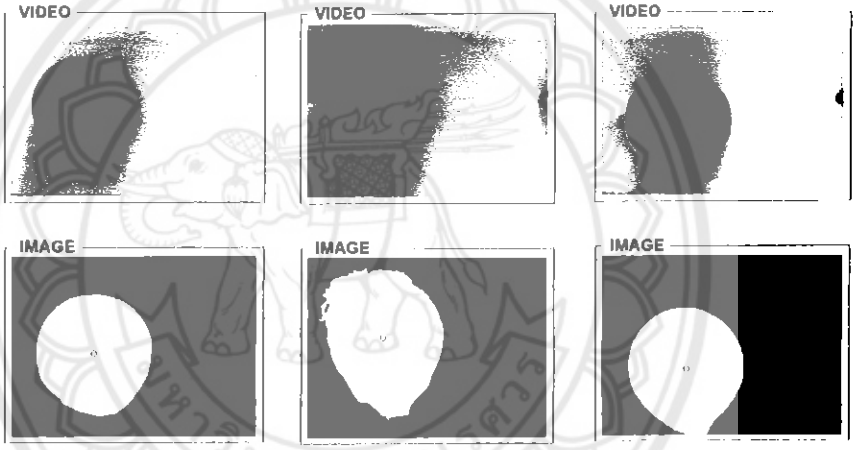
การใช้กระดาษไขจำนวน 5 แผ่น สามารถแสดงผลการทดลอง**ในการจำแนกรูปทรงของวัตถุ**สำหรับวัตถุทรงสามเหลี่ยมด้านเท่า รูปทรงสี่เหลี่ยมจัตุรัส และทรงกลม **ดังในรูปที่** 4.28 **รูปที่** 4.29 และรูปที่ 4.30 ตามลำดับ



รูปที่ 4.28 ภาพแสดงการจำแนกวัตถุทรงสามเหลี่ยมด้านเท่าโดยใช้กระดาษไข 5 แผ่น



รูปที่ 4.29 ภาพแสดงการจำแนกวัตถุทรงสี่เหลี่ยมจัตุรัสโดยใช้กระดาษไข 5 แผ่น



รูปที่ 4.30 ภาพแสดงการจำแนกวัตถุทรงกลมโดยใช้กระดาษไข 5 แผ่น

ในการจำแนกรูปทรงต่างๆ ของวัตถุเมื่อใช้กระดาษไขจำนวน 5 แผ่น นั้นจะมีการทดลองทั้งหมด 5 ครั้ง และได้ผลการทดลองดังแสดงในตารางที่ 4.7

ตารางที่ 4.7 ผลการทดลองการจำแนกรูปทรงวัตถุจากความเข้มแสงโดยใช้กระดาษไข 5 แผ่น

ครั้งที่	ทรงสามเหลี่ยม			ทรงสี่เหลี่ยม			ทรงกลม		
	สีแดง	สีเขียว	สีน้ำเงิน	สีแดง	สีเขียว	สีน้ำเงิน	สีแดง	สีเขียว	สีน้ำเงิน
1	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	✓	✓	✓	✓	✓	✓	✓	✓	✓
4	✓	✓	✓	✓	✓	✓	✓	✓	✓
5	✓	✓	✓	✓	✓	✓	✓	✓	✓

หมายเหตุ : สัญลักษณ์ในตารางแสดงการจำแนกรูปร่างดังนี้

✓ หมายถึง สามารถจำแนกได้ และ ✗ หมายถึง ไม่สามารถจำแนกได้

4.5.4 การทดลองด้วยการใช้กระดาษไข โดยการเพิ่มครั้งละ 5 แผ่น

ในการจำแนกรูปทรงต่างๆ ของวัตถุเมื่อทำการเพิ่มกระดาษไขครั้งละ 5 แผ่น นั้นจะได้ผลการทดลองดังแสดงในตารางที่ 4.8

ตารางที่ 4.8 ผลการทดลองการจำแนกรูปทรงวัตถุจากความเข้มแสงโดยเพิ่มกระดาษไขครั้งละ 5 แผ่น

จำนวนกระดาษไข	ทรงสามเหลี่ยม			ทรงสี่เหลี่ยม			ทรงกลม		
	สีแดง	สีเขียว	สีน้ำเงิน	สีแดง	สีเขียว	สีน้ำเงิน	สีแดง	สีเขียว	สีน้ำเงิน
5	✓	✓	✓	✓	✓	✓	✓	✓	✓
10	✓	✓	×	✓	✓	×	✓	✓	✓
15	✓	×	✓	×	✓	✓	×	×	✓
20	×	×	×	×	×	×	×	×	×
25	×	×	×	×	×	×	×	×	×

หมายเหตุ : สัญลักษณ์ในตารางแสดงการจำแนกรูปร่างดังนี้

✓ หมายถึง สามารถจำแนกได้ และ × หมายถึง ไม่สามารถจำแนกได้

จากผลการทดลองพบว่าในการจำแนกทั้งสีและรูปร่างของวัตถุนั้น แสงเป็นปัจจัยสำคัญที่มีผลต่อการบวนการประมวลผลภาพ จะเห็นได้จากผลการทดลองในสภาพแสงฟลูออเรสเซนต์ถึงแม้จะสามารถทำการจำแนกทั้งสีและรูปร่างของวัตถุได้อย่างถูกต้องและมีประสิทธิภาพ แต่เมื่อทำการทดลองในสภาพแสงจากดวงอาทิตย์นั้นจะไม่สามารถจำแนกสีและรูปร่างของวัตถุได้ ประกอบกับการทดลองด้วยกระดาษไขที่มีจำนวนความหนาแตกต่างกันก็ได้ผลการทดลองที่สอดคล้องกับการทดลองทั้งสองการทดลองข้างต้น

เพราะฉะนั้นในการประมวลผลภาพที่ดีและมีประสิทธิภาพ ต้องมีการควบคุมปริมาณของแสงในระบบอย่างดีและมีประสิทธิภาพด้วยเช่นกัน จึงจะทำให้ได้ผลลัพธ์จากการประมวลผลภาพที่ถูกต้อง และเป็นไปตามที่เราต้องการ

บทที่ 5

สรุปผลและข้อเสนอแนะ

จากการดำเนินโครงงานสามารถสรุปผล ชี้แจงปัญหาที่เกิดขึ้นในระหว่างการดำเนินงาน รวมทั้งแนวทางแก้ปัญหา พร้อมให้ข้อเสนอแนะในการนำโครงงานไปพัฒนาต่อไป

5.1 สรุปผลการดำเนินโครงงาน

โครงงานระบบตรวจจับวัตถุบนสายพานลำเลียงโดยใช้กล้อง เป็นการออกแบบและสร้างต้นแบบระบบตรวจจับวัตถุบนสายพานลำเลียงโดยใช้กล้อง เพื่อทำการศึกษาหลักการของการประมวลผลภาพระบบประมวลผลภาพที่ทำงานสัมพันธ์กับระบบสายพานลำเลียง โดยระบบตรวจจับวัตถุบนสายพานลำเลียงโดยใช้กล้อง สามารถตรวจจำแนกวัตถุได้ทั้ง การจำแนกสีของวัตถุทั้งหมด 3 สี คือ สีแดง สีเขียว และสีน้ำเงิน และจำแนกรูปร่างของวัตถุทั้งหมด 3 รูปร่างคือ ทรงสามเหลี่ยมด้านเท่า ทรงสี่เหลี่ยมจตุรัส และทรงกลม

โดยการดำเนินการได้ทำการเขียนโปรแกรม 2 ส่วนคือ ส่วนของระบบสายพานลำเลียง ทำการควบคุมการทำงานโดยใช้ตัวต้านทานชนิดความไวแสงติดต่อกับไมโครคอนโทรลเลอร์อาดุยโน่ และทำการส่งสัญญาณต่อไปยังส่วนที่ 2 คือ ระบบประมวลผลภาพ ซึ่งทำการจับถ่ายนิ่งและส่งภาพไปทำการประมวลผลภาพผ่านโปรแกรมแมทแลป ให้กับไมโครคอนโทรลเลอร์เพื่อควบคุมการทำงาน

ผลการดำเนินโครงงานระบบตรวจจับวัตถุบนสายพานลำเลียงโดยใช้กล้องสามารถจำแนกได้ทั้งสีและรูปร่างภายใต้แสงจากหลอดไฟฟลูออเรสเซนต์และการใช้กระดาษไข 1 3 และ 5 แผ่น แต่ภายใต้แสงอาทิตย์ระบบตรวจจับวัตถุบนสายพานลำเลียงโดยใช้กล้องสามารถจำแนกได้เพียงสี แต่ไม่สามารถจำแนกรูปร่างได้

5.2 ปัญหาและแนวทางแก้ไข

1. ความสว่างมีผลต่อการจำแนกรูปร่างของวัตถุ โดยถ้าแสงสว่างมากหรือน้อยเกินไปจะทำให้ไม่สามารถจำแนกรูปร่างของวัตถุออกจากฉากหลังได้

แนวทางการแก้ไขปัญหา ควรควบคุมปริมาณของแสงในสถานที่ที่จะทำการจำแนกรูปร่างของวัตถุหรือจำเพราะสถานที่ที่จะดำเนินการ

2. ในการสื่อสารระหว่างโปรแกรมมีความจำกัดของชุดคำสั่งที่ใช้ในการสื่อสารกัน ทำให้เกิดการสื่อสารระหว่างโปรแกรมไม่ได้ประสิทธิภาพเต็มที่

แนวทางการแก้ไขปัญหา ควรศึกษาชุดคำสั่งของโปรแกรมให้มากขึ้นหรือลดการสื่อสารระหว่างโปรแกรมให้น้อยลง

3. ในการทำงานของระบบตรวจจับวัตถุบนสายพานลำเลียงโดยใช้กล้องนั้น เมื่อมีการให้ระบบทำงานเป็นเวลานานๆ และมอเตอร์เกิดการทำงานหนักทำให้แรงดันของมอเตอร์ตกไปในบางช่วงของการทำงาน

แนวทางแก้ไขปัญหา เปลี่ยนหม้อแปลงแหล่งจ่ายไฟให้ใหญ่ขึ้นเพื่อแก้ไขไม่ให้แรงดันของมอเตอร์ตก และไม่ให้มอเตอร์รับโหลดมากจนเกินไป

5.3 แนวทางในการพัฒนาโครงงาน

1. ระบบตรวจจับวัตถุบนสายพานลำเลียงโดยใช้กล้องจะต้องได้รับการพัฒนาในการตรวจจับวัตถุได้หลากหลายขนาดขึ้น เพื่อให้สามารถนำไปใช้ประโยชน์ในเชิงพาณิชย์ได้ดียิ่งขึ้น

2. การตรวจจับวัตถุจะต้องได้รับการพัฒนาให้สามารถจำแนกสีและรูปร่างของวัตถุได้หลากหลายขึ้น

3. ระบบตรวจจับวัตถุบนสายพานลำเลียงควรมีการพัฒนาให้สายพานสามารถคัดแยกวัตถุได้ด้วยตนเอง

4. ในการจำแนกวัตถุโดยใช้ค่าความกลมอาจจะเกิดข้อผิดพลาดได้ จึงควรมีการตรวจสอบวัตถุโดยใช้ค่าอื่นๆด้วยเช่น ค่าความเป็นสี่เหลี่ยม และค่าความเป็นสามเหลี่ยม

ทั้งนี้แนวทางในการพัฒนาเพื่อเพิ่มศักยภาพการทำงานของระบบตรวจจับวัตถุบนสายพานลำเลียงโดยใช้กล้องเพื่อเป็นต้นแบบในการพัฒนาระบบควรคำนึงถึงประโยชน์ที่ได้รับและความคุ้มค่าในการลงทุน แต่เนื่องจากโครงงานนี้สามารถนำไปเป็นตัวอย่างทั้งในเรื่องของการศึกษา และการใช้งานจริงได้จึงควรมีการศึกษาและพัฒนาสร้างต่อเพื่อให้เกิดประโยชน์ต่อไป

เอกสารอ้างอิง

[1] ขจร อนุดิตย์ (2550) "การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ MC51 ด้วยภาษา C" นนทบุรี สำนักพิมพ์คอร์ฟังก์ชั่น

[2] บุญธรรม ภัทราจารุกุล (2556) "การประมวลผลภาพดิจิตอลเบื้องต้น" กรุงเทพฯ ซีเอ็ดยูเคชั่น

[3] สมเกียรติ อุดมหรรษากุล (2554) "การประมวลผลภาพดิจิตอลเบื้องต้น" กรุงเทพฯ ท้อป

[4] A Koschan and M. Abidi (2005) "Detection and Classification of Edges in Color Images" Signal Processing Magazine, Special Issue on Color Image Processing, Vol. 22, No. 1, pp. 64-73

[5] Jumping into Digital Image Processing (วีดีทัศน์) (2555) ปทุมธานี ห้างหุ้นส่วนจำกัด ไอบุคเอนจิเนียร์ริ่ง

[6] Shu-Yu Zhu (1998) "Comprehensive analysis of edge detection in color image processing" University of Toronto Department of Electrical and Computer Engineering Toronto, Ontario M5S 3G4Canada

[7] Soumya Dutta and Bidyut B. Chaudhuri (2009) "A Color Edge Detection Algorithm in RGB Color Space" International Conference on Advances in Recent Technologies in Communication and Computing

ภาคผนวก ก

โปรแกรมการประมวลผลภาพโดยการตรวจจับสีและรูปร่างของวัตถุ

```
function varargout = GUIVii(varargin)
% GUIVII MATLAB code for GUIVii.fig
%      GUIVII, by itself, creates a new GUIVII or raises the existing
%      singleton*.
%
%      H = GUIVII returns the handle to a new GUIVII or the handle to
%      the existing singleton*.
%
%      GUIVII('CALLBACK',hObject,eventData,handles,...) calls the
local
%      function named CALLBACK in GUIVII.M with the given input
arguments.
%
%      GUIVII('Property','Value',...) creates a new GUIVII or raises
the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before GUIVii_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to GUIVii_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUIVii

% Last Modified by GUIDE v2.5 13-Dec-2013 01:31:08

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUIVii_OpeningFcn, ...
                   'gui_OutputFcn',  @GUIVii_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before GUIVii is made visible.
function GUIVii_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUIVii (see VARARGIN)
```

```
% Choose default command line output for GUIVii
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUIVii wait for user response (see UIRESUME)
% uiwait(handles.figure1);


                    %% ====== Import Video ====== %%
                obj    = videoinput('winvideo',1);
                vidRes = get(obj, 'VideoResolution');     % Resolution
of device
                nBands = get(obj, 'NumberOfBands');       % number of
channels
                hImage = image( zeros(vidRes(2), vidRes(1), nBands) );
                preview(obj, hImage);
                axes(handles.axes2);
                handles.objVideo=obj;
                guidata(hObject, handles);
                    % ============================== %

                %% ====== Connect to Arduino ====== %%
                    y   = zeros(5000);
                    s1 = serial('COM6');      %define serial port
                    s1.BaudRate = 9600;       %define baud rate
                    fopen(s1);                %open serial port
                    clear data;
                % ================================== %

                 %% ====== Caputuring Video ====== %%
             for i = 1:500                          %acquisition of
500 points
              data = fscanf(s1);                    %read sensor
               A = str2double(data)/100;

            if A <= 1
               pause(8)
               disp('Caputering');
               axes(handles.axes1);
               I=getsnapshot(handles.objVideo);
               handles.image=I;
               imshow(handles.image)
               guidata(hObject, handles);
               %fclose(s1);                          % close the serial
port
                % ================================ %

                %% ====== Color Detection ======= %%
                 Ihsv = rgb2hsv(I);
                    h = Ihsv(:,:,1);
                    s = Ihsv(:,:,2);
                    v = Ihsv(:,:,3);

                color = 0;
              [ r c ] = size(h) ;
        for i=1:r ;
```

```
            for j=1:c;
               if h(i,j)>=0.95 && h(i,j)<=1.00 && s(i,j)>=0.60 &&
s(i,j)<=0.70 && v(i,j)>=0.50 && v(i,j)<=0.60;
                  color=1;
              elseif h(i,j)>=0.32 && h(i,j)<=0.37 && s(i,j)>=0.30 &&
s(i,j)<=0.66 && v(i,j)>=0.50 && v(i,j)<=0.60;
                  color=2;
               else h(i,j)>=0.62 && h(i,j)<=0.65 && s(i,j)>=0.40 &&
s(i,j)<=0.66 && v(i,j)>=0.45 && v(i,j)<=1.0;
                  color=3;
            %elseif h(i,j)>=0.09 && h(i,j)<=0.19 && s(i,j)>=0.26 &&
s(i,j)<=0.73 && v(i,j)>=0.75 && v(i,j)<=1.0;
                %color=4;
               end
            end
         end
               % ================================== %

               %% ====== Shape Detection ======= %%
                  H = fspecial('disk',10);
               blurred = imfilter(I,H,'replicate');
                Igray = rgb2gray(blurred);
               Ibinary = Igray<150;
               Ibinary = bwareaopen(Ibinary,500);

    [labeledImage numberOfObjcts] = bwlabel(Ibinary);
                blobMeasurements =
regionprops(labeledImage,'Perimeter','Area','Centroid');
                        Icen = [blobMeasurements.Centroid];
                circularities = [blobMeasurements.Perimeter.^2]
./ (4 * pi * [blobMeasurements.Area]);

            for blobNumber = 1 : numberOfObjcts;
               if circularities(blobNumber) >=1.0 &&
circularities(blobNumber) <=1.19;
                  cc=1;
              elseif circularities(blobNumber) >=1.2 &&
circularities(blobNumber) <=1.40;
                  cc=2;
               else circularities(blobNumber) >=1.45 &&
circularities(blobNumber) <=1.85;
                  cc=3;
               end
            end
               % ================================== %

                %% ====== Identify Color ====== %%
                  if color == 1;
                      a='RED';
                      set(handles.color,'String',a);
                elseif color == 2;
                      b='GREEN';
                      set(handles.color,'String',b);
                elseif color == 3;
                      c='BLUE';
                      set(handles.color,'String',c);
                  else
                      d='UNIDENTIFY';
                      set(handles.color,'String',d);
                   end
```

```
                % ================================ %

                  %% ====== Identify Shape ====== %%
                  if cc == 1;
                      sa='CIRCULAR';
                      set(handles.shapee,'String',sa);
                elseif cc == 2;
                      sb='RECTANGULAR';
                      set(handles.shapee,'String',sb);
                elseif cc == 3;
                      sc='TRIANGLE';
                      set(handles.shapee,'String',sc);
                  else
                      sd='UNIDENTIFY';
                      set(handles.shapee,'String',sd);
                  end
                % ================================== %

                %% ====== Identify Co-ordinate ====== %%
                    centroidX = [Icen(1)];
                    set(handles.xLabel,'String',centroidX);
                    centroidY = [Icen(2)];
                    set(handles.yLabel,'String',centroidY);
                % ================================== %

            for x = 1: numel(blobMeasurements)
                axes(handles.axes3);

plot(blobMeasurements(x).Centroid(1),blobMeasurements(x).Centroid(2),
'ro');
                handles.image=Ibinary;
                imshow(handles.image)
                guidata(hObject, handles);
            end


    else
        disp('Continous');
    end
          end
    imshow(I)


% --- Outputs from this function are returned to the command line.
function varargout = GUIVii_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function color_Callback(hObject, eventdata, handles)
% hObject    handle to color (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of color as text
```

```
%        str2double(get(hObject,'String')) returns contents of color
as a double


% --- Executes during object creation, after setting all properties.
function color_CreateFcn(hObject, eventdata, handles)
% hObject    handle to color (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function shapee_Callback(hObject, eventdata, handles)
% hObject    handle to shapee (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of shapee as text
%        str2double(get(hObject,'String')) returns contents of shapee
as a double

% --- Executes during object creation, after setting all properties.
function shapee_CreateFcn(hObject, eventdata, handles)
% hObject    handle to shapee (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function xLabel_Callback(hObject, eventdata, handles)
% hObject    handle to xLabel (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of xLabel as text
%        str2double(get(hObject,'String')) returns contents of xLabel
as a double


% --- Executes during object creation, after setting all properties.
function xLabel_CreateFcn(hObject, eventdata, handles)
% hObject    handle to xLabel (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
```

```
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function yLabel_Callback(hObject, eventdata, handles)
% hObject    handle to yLabel (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of yLabel as text
%        str2double(get(hObject,'String')) returns contents of yLabel
as a double


% --- Executes during object creation, after setting all properties.
function yLabel_CreateFcn(hObject, eventdata, handles)
% hObject    handle to yLabel (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: place code in OpeningFcn to populate axes1

% --- Executes during object creation, after setting all properties.
function axes2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: place code in OpeningFcn to populate axes2

% --- Executes during object creation, after setting all properties.
function axes3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: place code in OpeningFcn to populate axes3
```

ภาคผนวก ข

โปรแกรมควบคุมระบบตรวจจับวัตถุบนสายพานลำเลียง

```
int LDR = 0;       //analog pin to which LDR is connected, here we set it to 0 so it means A0

int LDRValue = 0;  //that's a variable to store LDR values

int light_sensitivity = 100;  //This is the approx value of light surrounding your LDR


void setup()

 {

  Serial.begin(9600);    //start the serial monitor with 9600 buad

  pinMode(13, OUTPUT);   //we mostly use13 because there is already a built in yellow LED
in arduino which shows output when 13 pin is enabled

 }


void loop()

 {

  LDRValue = analogRead(LDR);     //reads the ldr's value through LDR which we have set to
Analog input 0 "A0"

  Serial.println(LDRValue);      //prints the LDR values to serial monitor

  delay(1000);                   //This is the speed by which LDR sends value to arduino



 digitalWrite(13,HIGH);


    if (LDRValue < light_sensitivity)

   {

    digitalWrite(7,HIGH);

     digitalWrite(8,LOW);

   }

    else

   {
```

```
    digitalWrite(8,HIGH);
    digitalWrite(7,LOW);
  }
  if (LDRValue < light_sensitivity)
   {
    digitalWrite(13,HIGH);
    delay(5000);
    digitalWrite(13,LOW);
    delay(2000);
    digitalWrite(13,HIGH);
    delay(1000);
   }
  }
```

ภาคผนวก ค

รายละเอียดไอซีหมายเลข ATMEGA 328

Features

- High Performance, Low Power AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions - Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20 MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 4/8/16/32K Bytes of In-System Self-Programmable Flash progam memory (ATmega48PA/88PA/168PA/328P)
  - 256/512/512/1K Bytes EEPROM (ATmega48PA/88PA/168PA/328P)
  - 512/1K/1K/2K Bytes Internal SRAM (ATmega48PA/88PA/168PA/328P)
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C(1)
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - Programming Lock for Software Security
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Six PWM Channels
  - 8-channel 10-bit ADC in TQFP and QFN/MLF package
    - Temperature Measurement
  - 6-channel 10-bit ADC in PDIP Package
    - Temperature Measurement
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Byte-oriented 2-wire Serial Interface (Philips I²C compatible)
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
  - 23 Programmable I/O Lines
  - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
  - 1.8 - 5.5V for ATmega48PA/88PA/168PA/328P
- Temperature Range:
  - -40°C to 85°C
- Speed Grade:
  - 0 - 20 MHz @ 1.8 - 5.5V
- Low Power Consumption at 1 MHz, 1.8V, 25°C for ATmega48PA/88PA/168PA/328P:
  - Active Mode: 0.2 mA
  - Power-down Mode: 0.1 µA
  - Power-save Mode: 0.75 µA (Including 32 kHz RTC)

8-bit AVR® Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash

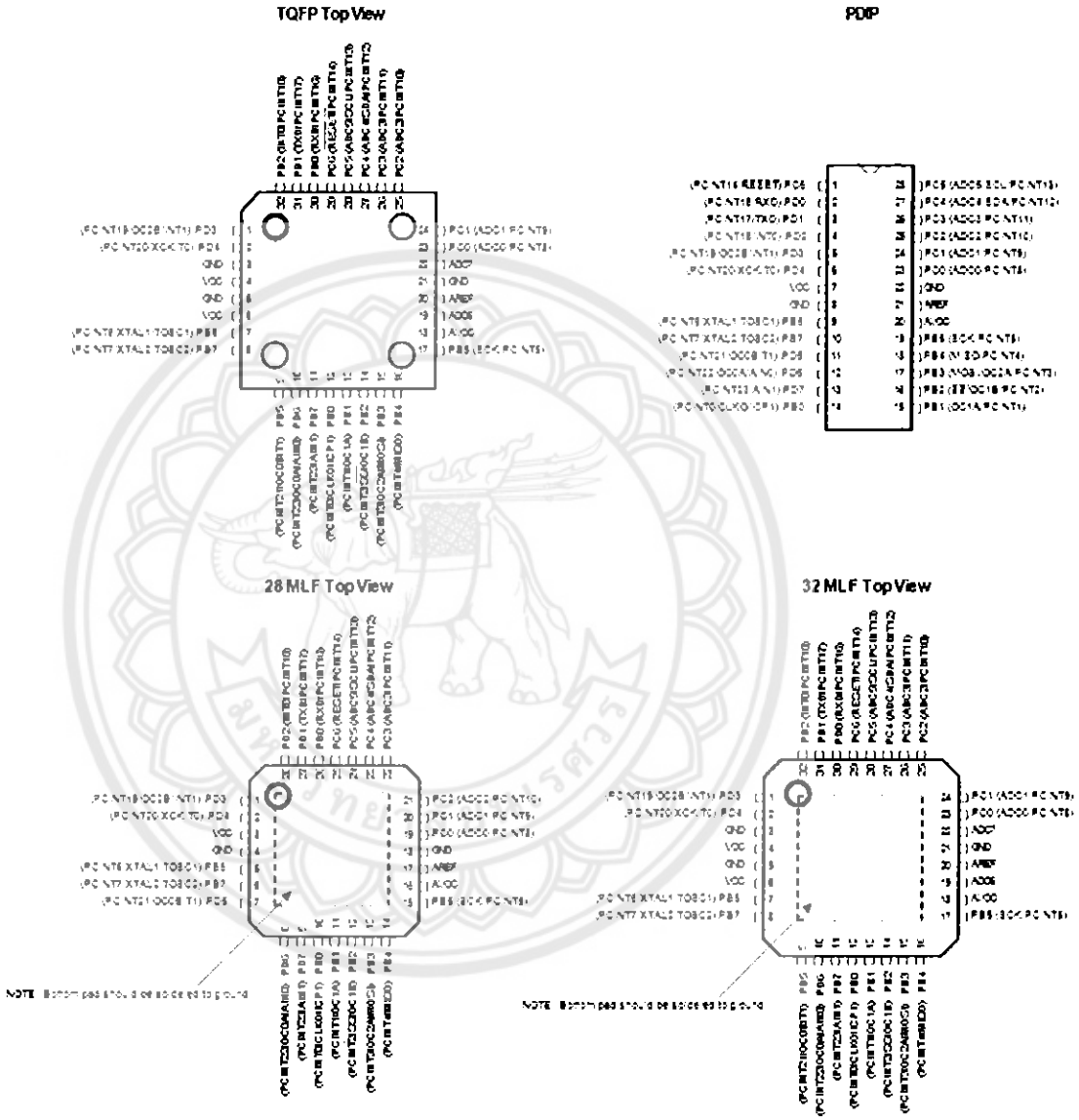
ATmega48PA
ATmega88PA
ATmega168PA
ATmega328P

Summary

Rev. 8161DS-AVR-10/09

1. Pin Configurations

Figure 1-1. Pinout ATmega48PA/88PA/168PA/328P



1.1 Pin Descriptions

1.1.1 VCC

Digital supply voltage.

1.1.2 GND

Ground.

1.1.3 Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..6 is used as TOSC2..1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

The various special features of Port B are elaborated in "Alternate Functions of Port B" on page 76 and "System Clock and Clock Options" on page 26.

1.1.4 Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5..0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

1.1.5 PC6/RESET

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in Table 28-3 on page 308. Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in "Alternate Functions of Port C" on page 79.

1.1.6 Port D (PD7:0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

The various special features of Port D are elaborated in "Alternate Functions of Port D" on page 82

1.1.7 AV_{CC}

AV_{CC} is the supply voltage pin for the A/D Converter, PC3:0, and ADC7:6. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter. Note that PC6..4 use digital supply voltage, V_{CC}.

1.1.8 AREF

AREF is the analog reference pin for the A/D Converter.

1.1.9 ADC7:6 (TQFP and QFN/MLF Package Only)

In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

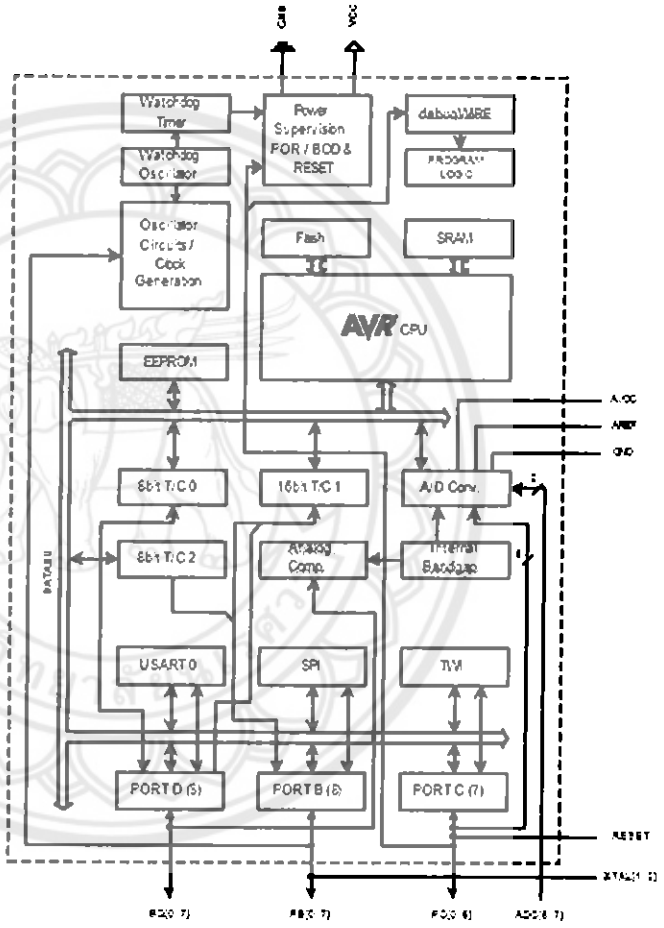
มหาวิทยาลัยนเรศวร

2. Overview

The ATmega48PA/88PA/168PA/328P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega48PA/88PA/168PA/328P achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block Diagram

Figure 2-1. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting