ระบบควบคุมส่วนบันทึกภาพและการตรวจสอบจุดบกพร่อง ในกระบวนการผลิตฮาร์ดดิสก์

A CONTROL MODULE FOR RECORDING IMAGES AND INSPECTION IN HARDDISK MANUFACTURING PROCESS

นายวีรพงษ์ ลิ้มตระกูล รหัส 54361145
นายทศพล ศุภวาร รหัส 54363781
นายสมศักดิ์ พรหมนิ่ม รหัส 54364252

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2557

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน **ระบบควบคุมส่วนบันทึกภาพและการตรวจสอบจุดบกพร่องในกระบวนการผลิตฮาร์ดดิสก์**

ผู้ดำเนินโครงงาน	นายวีรพงษ์	ลิ้มตระกูล	รหัส 54361145
	นายทศพล	ศุภวาร	รหัส 54363781
	นายสมศักดิ์	พรหมนิ่ม	รหัส 54364252
ที่ปรึกษาโครงงาน	รองศาสตราจารย์ ดร. ไพศาล มุณีสว่าง		
สาขาวิชา	วิศวกรรมไฟฟ้า		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2558		

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์**ฉบับนี้เป็นส่วนหนึ่ง**ของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

.....**ที่ปรึกษาโครงงาน**

(รองศาสตราจารย์ ดร. ไพศาล มุณีสว่าง)

.....**กรรมการ**

(รองศาสตราจารย์ ดร. ธนิต มาลากร)

.....**กรรมการ**

(ผู้ช่วยศาสตราจารย์ ดร.พรพิศุทธิ์ วรจิรันตน์)

ชื่อหัวข้อโครงงาน	**ระบบควบคุมส่วนบันทึกภาพและการตรวจสอบจุดบกพร่องในกระบวนการผลิตฮาร์ดดิสก์**
ผู้ดำเนินโครงงาน	**นายวีรพงษ์ ลิ้มตระกูล รหัส 54361145**
	นายทศพล ศุภวาร รหัส 54363781
	นายสมศักดิ์ พรหมนิ่ม รหัส 54364252
ที่ปรึกษาโครงงาน	**รองศาสตราจารย์ ดร. ไพศาล มุณีสว่าง**
สาขาวิชา	**วิศวกรรมไฟฟ้า**
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2557

บทคัดย่อ

โครงงานนี้เป็นการศึกษาและพัฒนาโปรแกรมตรวจจับรอยตำหนิบนจานแม่เหล็กในฮาร์ดดิสก์จากกระบวนการผลิตโดยใช้กล้องเว็บแคม โดยผ่านกระบวนการประมวลผลภาพ ในการวิเคราะห์รอยตำหนิบนจานแม่เหล็กในฮาร์ดดิสก์ การพัฒนาโปรแกรมนี้ ทดสอบและทำงานบนระบบปฏิบัติการไมโครซอฟท์วินโดว์ โดยใช้ไลบรารี่ Aforge .NET ในการเขียนโปรแกรมประมวลผลภาพ โดยผ่านการปรับภาพให้เป็นสีเทา การตรวจหาขอบภาพ การปรับความเข้มของภาพ การสลับสีภาพ เพื่อตรวจจับเฉพาะจานแม่เหล็ก เพื่อทำการตัดภาพให้เหลือเฉพาะจานแม่เหล็กและจะมาทำการเปรียบเทียบภาพเพื่อหาความแตกต่างกับภาพต้นฉบับที่ไม่มีรอยตำหนิ ถ้าจำนวนพิกเซลที่เหลือจากการเทียบกันมากเกินกว่าที่กำหนด โปรแกรมจะตีความว่าพบจุดบกพร่อง ถ้าไม่เกินพิกเซลที่กำหนด โปรแกรมจะตีความว่ามีสภาพดี

ผลที่ได้จากโครงงานนี้ คือ โปรแกรมสามารถตรวจจับรอยตำหนิบนจานแม่เหล็กได้ โดยขนาดที่ตรวจเจอต้องมากกว่าหรือเท่ากับ 1 มิลลิเมตร จากการใช้กล้องเว็บแคมที่ความละเอียด 1.3 ล้านพิกเซล และควรใช้แสงให้เหมาะสม

Project title	A Control Module for Recording Images and Inspection in Harddisk Manufacturing Process
Name	Mr. Weerapong Limtrakul ID. 54361145
	Mr. Tossapon Supawan ID. 54363781
	Mr. Somsak Promnim ID. 54364252
Project advisor	Assc. Prof. Paisarn Muneesawang, Ph.D.
Major	Electrical Engineering
Department	Electrical and Computer Engineering
Academic year	2014

Abstract

This project is the study and development of the defect detecting program on a magnetic disk in a hard disk from a production process by using a web camera via a digital image processing for analyzing defects on a magnetic disk in a hard disk. The development of this program examines and works on Microsoft Windows operating system by using the library Aforge.NET for writing an image processing program via a color adjustment of image to be in grey scale, an edge detection, a contrast adjustment of image, an image inverting. This process is done in order to detect only a magnetic disk, and to crop images to be left only a magnetic disk. After that, an image comparison process is done for finding the differences to the original image with no defect. If the total amount of remaining pixels from this comparison process is more than what is expectedly defined, the program will interpret as "the defect has been found", if not, the program will interpret as "the status is normal".

The result found from this project states that the program is able to detect defects on a magnetic disk only if the size of defects is greater than or equals to 1 millimeter, detected from using a web camera which contains 1.3-million-pixel resolution and using appropriate light, accordingly.

กิตติกรรมประกาศ

ผู้ดำเนินโครงงานขอขอบคุณรองศาสตราจารย์ ดร. ไพศาล มุณีสว่าง ที่ปรึกษาโครงงาน ซึ่งเอาใจใส่ในรายละเอียดทุกขั้นตอนของการดำเนินโครงงาน โดยให้คำปรึกษาและคำแนะนำในการแก้ไขปัญหาต่าง ๆ อย่างต่อเนื่องจนกระทั่งโครงงานสำเร็จลุล่วง รวมถึงแนะนำหลักการเขียนปริญญานิพนธ์และตรวจทานแก้ไขอย่างละเอียดจนได้ปริญญานิพนธ์เป็นรูปเล่มสมบูรณ์

ขอขอบคุณคณาจารย์ทุกท่านที่ให้อบรมสั่งสอนตลอดการศึกษาเล่าเรียนในระดับปริญญาตรี ทำให้สามารถนำความรู้และทักษะในหลาย ๆ ด้านมาประยุกต์ใช้กับการดำเนินโครงงานนี้

ในท้ายที่สุดนี้ เหนือสิ่งอื่นใด ผู้ดำเนินโครงงานขอกราบขอบพระคุณบิดามารดาซึ่งให้การสนับสนุนในทุกด้านเกี่ยวกับการศึกษาของผู้ดำเนินโครงงาน รวมทั้งมอบความรัก ความเมตตา และคอยเป็นกำลังใจให้จนทำให้ประสบกับความสำเร็จในวันนี้

นายวีรพงษ์ ลิ้มตระกูล
นายทศพล ศุภวาร
นายสมศักดิ์ พรหมนิ่ม

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 2
1.3 ขอบเขตของโครงงาน 2
1.4 ขั้นตอนและแผนการดำเนินงาน 3
1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน 3
1.6 งบประมาณ 3

บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง 4
2.1 แผงวงจร Arduino 4
2.2 การประมวลผลภาพ 6
2.2.1 การประมวลผลภาพ 6
2.2.2 การทำงานของ Image Processing 7
2.2.2.1 อุปกรณ์รับภาพ 7
2.2.2.2 อุปกรณ์ในการประมวลผล 7
2.2.2.3 โปรแกรมประมวลผลภาพ 8
2.3 การเตรียมข้อมูลภาพ 8
2.4 โปรแกรม Microsoft Visual Studio 8
2.4.1 Microsoft Visual Studio คือ 10

สารบัญ (ต่อ)

หน้า

2.4.2 ส่วนประกอบของ Microsoft Visual Studio 10
2.5 ภาษา C# 12
2.5.1 C# คืออะไร 12
2.5.2 เปรียบเทียบภาษา C# กับภาษาอื่นๆ 13
2.5.3 จุดเด่นหลักๆ ของภาษา C# 13
2.6 ภาพระดับสีเทา 14
2.7 การหาขอบภาพ 15
2.8 การแปลงภาพสีให้เป็นภาพขาว-ดำ 16
2.9 การเข้าคู่รูปแบบ 18

บทที่ 3 ขั้นตอนการดำเนินงาน 19
3.1 การออกแบบโครงงาน 19
3.1.1 ส่วนขยายของระบบการทำงานโดยรวม 21
3.2 อุปกรณ์ที่ใช้ทำโครงงาน 23
3.3 การสร้างโปรแกรม 26
3.4 การออกแบบหน้าต่างการใช้งานโปรแกรม 26
3.4.1 หน้าต่างการใช้งานโปรแกรมหลัก 26

บทที่ 4 ผลการทดลอง 28
4.1 ขั้นตอนการทดสอบการเชื่อมต่อและทำงานของระบบ 28
4.1.1 ทดสอบการเชื่อมต่อกับกล้องเว็บแคม 28
4.1.2 ทดสอบฟังก์ชั่นการทำงานของโปรแกรม 29
4.2 ทดสอบขอบเขตของการใช้งาน 31
4.2.1 ทดสอบการทำงานของโปรแกรมโดยมีการควบคุมแสงสว่าง 31
4.2.2 การทดลองหาจำนวนพิกเซลทั้งหมดที่ยอมรับได้ 32
4.2.3 การทดลองความแม่นยำของการทำงานโปรแกรม 33
4.2.4 การทดลองหาข้อจำกัดของขนาดตำหนิที่ตรวจพบ 36

สารบัญ (ต่อ)

หน้า

4.2.5 การทดลองการทำงานโปรแกรมทั้งหมด 39
บทที่ 5 สรุปผลและข้อเสนอแนะ 43
5.1 สรุปผลการทดลอง 43
5.2 ข้อเสนอแนะ 43

เอกสารอ้างอิง 45

ภาคผนวก ก โปรแกรมในส่วนของ Microsoft Visual Studio 2010 47
ภาคผนวก ข โปรแกรมในส่วนของ Arduino ควบคุมมอเตอร์ 58
ภาคผนวก ค วิธีการติดตั้ง Aforge.net 60

ประวัติผู้ดำเนินโครงงาน 62

สารบัญตาราง

ตารางที่ หน้า

1.1 ขั้นตอนและแผนการดำเนินโครงงาน 3
4.1 ตารางผลการทดลองการปรับค่าระดับสีเทา (Threshold) ที่สภาวะแสงต่างกัน 32
4.2 ตารางการทดลองหาจำนวนพิกเซลที่ผ่านจากกระบวนการ Template Matching 33
4.3 ตารางผลการทดลองของรูปที่ผ่านการลบกัน 34
4.4 ตารางบันทึกค่าพิกเซลที่นับได้แต่ละการทดลอง 35
4.5 ตารางผลการทดลองของรูปที่ผ่านการลบกันโดยที่กำหนดรอยตำหนิขนาดต่างๆ กัน 36
4.6 ตารางผลการทดลองของรูปที่ผ่านการลบในกรณีที่ตำหนิเป็นรอยขีดข่วน 38
4.7 ตารางผลการทดลองของรูปที่ผ่านวิธีการ Template Matching 39
4.8 ตารางบันทึกค่าพิกเซลที่นับได้แต่ละการทดลอง 41

สารบัญรูป

รูปที่ หน้า

2.1 รูปของแผงวงจร Arduino รุ่น Uno r3 4
2.2 โครงสร้างไมโครคอนโทรลเลอร์ รุ่นATmega328P-PU 6
2.3 กระบวนการประมวลผลภาพ 6
2.4 หน้าต่างแสดงส่วนประกอบของ Microsoft Visual Studio 10
2.5 ภาพแสดงระดับสีเทา 14
2.6 การแปลงภาพอาร์จีบีเป็นภาพระดับสีเทา 15
2.7 ตัวอย่างภาพผลลัพธ์ที่ได้จากการหาขอบภาพด้วยวิธีต่างๆ 15
2.8 แสดงฮิสโตแกรมที่วัตถุและพื้นหลังมีค่าความเข้มแสงแยกออกจากกัน 16
3.1 แสดงระบบการทำงานโดยรวม 19
3.2 แสดงแผนผังการทำงานโดยรวมของระบบ 21
3.3 กล้องเว็บแคม 23
3.4 สายพานลำเลียง 23
3.5 วงจรตัวแปลงไฟฟ้า 23
3.6 มอเตอร์ควบคุม 23
3.7 อินฟราเรดเซ็นเซอร์ 23
3.8 คอมพิวเตอร์ 23
3.9 สวิตซ์ 24
3.10 หลอดไฟ LED 24
3.11 ภาพรวมของโครงงาน 24
3.12 วงจรในโครงงาน 25
3.12 หน้าต่างโปรแกรมหลัก 26
4.1 แสดงผลการติดต่อกับกล้องเว็บแคม 28
4.2 แสดงหน้าต่างในส่วนของการบันทึกภาพ 29
4.3 แสดงหน้าต่างในส่วนของการบันทึกภาพและประมวลผลภาพ 29
4.4 แสดงรูปสุดท้ายของการประมวลผลภาพ 30
4.6 แสดงรูปการทดลองการทำงานของโปรแกรมเมื่อมีแสงสว่างเพียงพอ 31
4.7 แสดงรูปการทดลองการทำงานของโปรแกรมเมื่อไม่มีแสงสว่าง 31

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.8 รูปต้นฉบับที่ใช้ในการทดลองที่ 4.2.3 34
4.9 รูปต้นฉบับที่ใช้ในการทดลองที่ 4.2.4 36
4.10 รูปต้นฉบับที่ใช้ในการทดลองที่ 4.2.4 (2) 37
4.11 รูปต้นฉบับที่ใช้ในการทดลองที่ 4.2.5 39

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ในคอมพิวเตอร์ทุกๆเครื่องต้องมีฮาร์ดดิสก์เป็นส่วนประกอบ เพราะฮาร์ดดิสก์เป็นส่วนที่เก็บข้อมูลต่างๆ ไว้ในเครื่องคอมพิวเตอร์ เช่น ไฟล์เอกสารต่างๆ ไฟล์รูปภาพ ไฟล์วีดิโอ ไฟล์เพลง เป็นต้น

โดยหลักการบันทึกข้อมูลบนฮาร์ดดิสก์จะใช้สารบันทึกซึ่งคือสารแม่เหล็ก โดยสารแม่เหล็กนั้นจะสามารถลบหรือเขียนได้ใหม่อยู่ตลอดเวลา โดยสารแม่เหล็กนี้จะเคลือบอยู่ในแผ่นแก้ว หรือแผ่นอลูมิเนียมที่มีความเรียบมากจนเหมือนกระจก โดยส่วนประกอบของฮาร์ดดิสก์จะมีการทำงานดังนี้

- หัวอ่าน (Head) เป็นส่วนหนึ่งของแขนหัวอ่าน ซึ่งเจ้าหัวอ่านตัวนี้สร้างจากขดลวด เพื่อใช้อ่านหรือเขียนข้อมูลลงบนจานแม่เหล็ก โดยการรับคำสั่งจากตัวคอนโทรลเลอร์ ก่อนเกิดความเหนี่ยวนำทางแม่เหล็ก และไปเปลี่ยนแปลงโครงสร้างของสนามแม่เหล็ก และทำให้เกิดการเปลี่ยนแปลงของข้อมูลนั่นเอง
- แขนหัวอ่าน (Actuator Arm) มีลักษณะเป็นแท่งเหล็กยาวๆ ซึ่งสามารถรับคำสั่งจากวงจรให้เลื่อนไปยังตำแหน่งที่ต้องการได้ ไม่ว่าจะเป็นอ่านหรือเขียนข้อมูลลงบนจานแม่เหล็ก โดยต้องทำงานร่วมกับหัวอ่าน
- จานแม่เหล็ก (Platters) มีลักษณะเป็นจานกลมๆ เคลือบด้วยสารแม่เหล็กวางซ้อนกันหลายๆ ชั้นขึ้นอยู่กับความจุ เจ้าสารแม่เหล็กที่เองที่เป็นข้อมูลต่างๆ ของเรา โดยข้อมูลนั้นจะถูกบันทึกในลักษณะของเลข 0 และ 1 จานแม่เหล็กนั้นติดกับมอเตอร์สำหรับหมุน (Spindle Motor) และสามารถเก็บข้อมูลได้ทั้ง 2 ด้าน
- มอเตอร์หมุนจานแม่เหล็ก (Spindle Motor) เป็นตัวควบคุมจานแม่เหล็กให้หมุนไปยังตำแหน่งที่ต้องการเพื่อบันทึก หรือแก้ไขข้อมูล ปกติมักมีความเร็วในการหมุนประมาณ 7200 รอบต่อนาที แต่ด้วยเทคโนโลยีการผลิตที่มีประสิทธิภาพมากกว่าเดิมทำให้ตัวมอเตอร์มาสามารถเพิ่มความเร็วได้ถึง 1 หมื่นรอบต่อนาที
- เคส (Case) หรือตัวกล่องสี่เหลี่ยม ซึ่งเป็นที่บรรจุส่วนต่างๆ ที่ใช้ในการทำงานของฮาร์ดดิสก์

ในที่นี้เราจะสนใจจานแม่เหล็ก (Platters) ที่เป็นส่วนที่ใช้สำหรับบันทึกข้อมูล สำหรับในส่วนของการผลิตฮาร์ดดิสก์ ดังนั้นโครงงานนนี้จึงได้ทำระบบควบคุมการทำงานส่วนการ

บันทึกภาพ โดยการทำงานนั้นจะใช้เซนเซอร์อินฟาเรดในการตรวจจับวัตถุที่เคลื่อนตามสายพาน จากนั้นส่งข้อมูลผ่านไมโครคอนโทรเลอร์ไปควบคุมการถ่ายภาพของ กล้องเว็บแคม โดยภาพนิ่งที่ถ่ายนั้นจะแสดงผลผ่านซอฟท์แวร์ที่พัฒนาโดยโปรแกรม Microsoft Visual Studio 2010 ภาพที่ถ่ายได้ก็จะถูกเก็บบันทึกลงในคอมพิวเตอร์ และนำรูปภาพที่ได้มาประมวลผลภาพ เพื่อตรวจหารอยตำหนิบนจานแม่เหล็ก ซึ่งก่อให้เกิดปัญหาในการบันทึกข้อมูล

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อสร้างอุปกรณ์ที่ใช้ถ่ายรูปอัตโนมัติในระบบสายพานการผลิตฮาร์ดดิสก์

2. เพื่อพัฒนาประสิทธิภาพในการตรวจสอบรอยตำหนิบนจานแม่เหล็กในฮาร์ดดิสก์

3. เพื่อสร้างโปรแกรมประมวลผลภาพจาก Microsoft Visual Studio 2010

4. ศึกษาการใช้งานโปรแกรม Microsoft Visual Studio 2010 การใช้ไมโครคอนโทรเลอร์ Arduino Uno R3 และการเขียนโปรแกรมประมวลภาพด้วยไลบรารี่ Aforge .NET

1.3 ขอบเขตของโครงงาน

1. ศึกษาการทำงานของโปรแกรมที่รับข้อมูลภาพจากกล้องประเภท video camera

2. ศึกษาการทำงานของ Arduino ที่ใช้ร่วมกันกับ เซ็นเซอร์ และ รีเลย์

3. สายพานมีความกว้าง 19 เซนติเมตร ยาว 50 เซนติเมตรและสูง 9 เซนติเมตร กล้องมีความสูงจากตัวสายพาน 20 เซนติเมตรและมีความละเอียดของภาพอยู่ที่ 1.3 ล้าน พิกเซล กล่องควบคุมแสงมีความกว้าง 32 เซนติเมตร ยาว 68 เซนติเมตรและสูง 35 เซนติเมตร และระยะตรวจจับของเซ็นเซอร์ถึงฮาร์ดดิสก์ไม่เกิน 3 เซนติเมตร

4. ออกแบบอัลกอริทึมของตัวเซ็นเซอร์ในการตรวจจับเมื่อมีวัตถุเคลื่อนผ่านและควบคุมกล้องให้ทำการถ่ายภาพนิ่งโดยนำข้อมูลไปเก็บไว้ในคอมพิวเตอร์

5. ตรวจสอบตำหนิบนฮาร์ดดิสก์โดยผ่านการประมวลผลภาพ

6. เป็นการตรวจสอบรอยตำหนิบนจานแม่เหล็กในฮาร์ดดิสก์ ในกระบวนการผลิตฮาร์ดดิสก์ขั้นตอนสุดท้าย

7. ฮาร์ดดิสก์ที่ใช้ตรวจสอบ ยี่ห้อ Maxtor รุ่น DiamondMax Plus 8

1.4 ขั้นตอนและแผนการดำเนินงาน

ตารางที่ 1.1 ขั้นตอนและแผนการดำเนินโครงงาน

รายละเอียด	พ.ศ. 2558						
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค	ม.ค.	ก.พ.
1) รวบรวมข้อมูล							
2) ศึกษาทฤษฎีและโปรแกรม							
3) ออกแบบอัลกอริทึม							
4) ทดสอบและปรับปรุงชิ้นงาน							
5) สรุปผลและจัดทำรูปเล่มปริญญานิพนธ์							

1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน

1.ประหยัดเวลาในการตรวจสอบตำหนิบนแผ่นฮาร์ดดิสก์

2.สามารถนำโปรแกรมไปประยุกต์ใช้ในการตรวจสอบชิ้นงานประเภทอื่นๆได้

3.ได้รับความรู้เกี่ยวกับการออกแบบอัลกอริทึมที่เกี่ยวข้องในการถ่ายภาพนิ่ง

4.ผู้ที่สนใจสามารถนำไปศึกษาและพัฒนาไปใช้ในงานต่างๆ ต่อไปได้

1.6 งบประมาณ

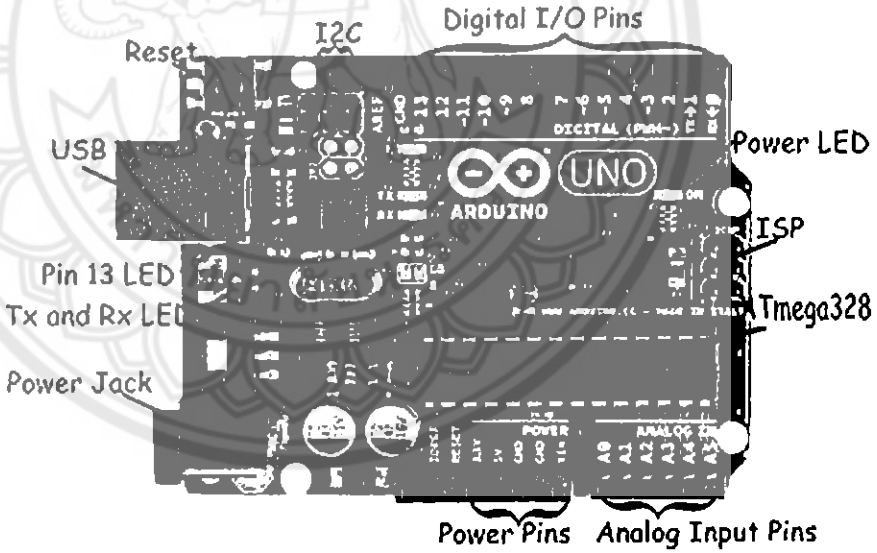
1) ค่าอุปกรณ์อิเล็กทรอนิกส์	2200 บาท
2) ค่าถ่ายเอกสารและเข้าเล่มปริญญานิพนธ์	800 บาท
รวมเป็นเงินทั้งสิ้น (สามพันบาทถ้วน)	3,000 บาท

หมายเหตุ: ถัวเฉลี่ยทุกรายการ

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 แผงวงจร Arduino [1]

ไมโครคอนโทรลเลอร์ที่ใช้ในโครงงานนี้เป็นแผงวงจรArduino รุ่น Uno r3 จัดอยู่ในตระกูลเอวีอาร์(AVR) ขนาด 28 ขา ซึ่งใช้ไมโครคอนโทรลเลอร์หมายเลขATmega328 ดังแสดงในรูปที่ 2.1 โดยไมโครคอนโทรลเลอร์ Arduino เป็นแพลตฟอร์ม (Platform)ของอินพุตและเอาท์พุต (I/O)ขั้นพื้นฐานที่พอเพียงกับการใช้งานและการเรียนรู้ โดยตัวแผงวงจรมีชุดคำสั่งที่ใช้ควบคุมพอร์ตอินพุตและเอาท์พุต รวมถึงพอร์ตดิจิตอล พอร์ตแอนะล็อกพีดับเบิลยูเอ็มและพอร์ตอนุกรมซึ่งแผงวงจร Arduino ทำให้คอมพิวเตอร์สามารถรับสัญญาณจากภายนอกและส่งสัญญาณไปควบคุมอุปกรณ์ภายนอกได้อย่างมีประสิทธิภาพมากกว่าการใช้เครื่องคอมพิวเตอร์ ตัวแผงวงจรออกแบบจากไมโครคอมพิวเตอร์เดี่ยวและมีโปรแกรมพัฒนาสำหรับให้แผงวงจร Arduino สามารถรับสัญญาณจากสวิตช์หรือตัวรับรู้หรืออุปกรณ์อื่นๆ



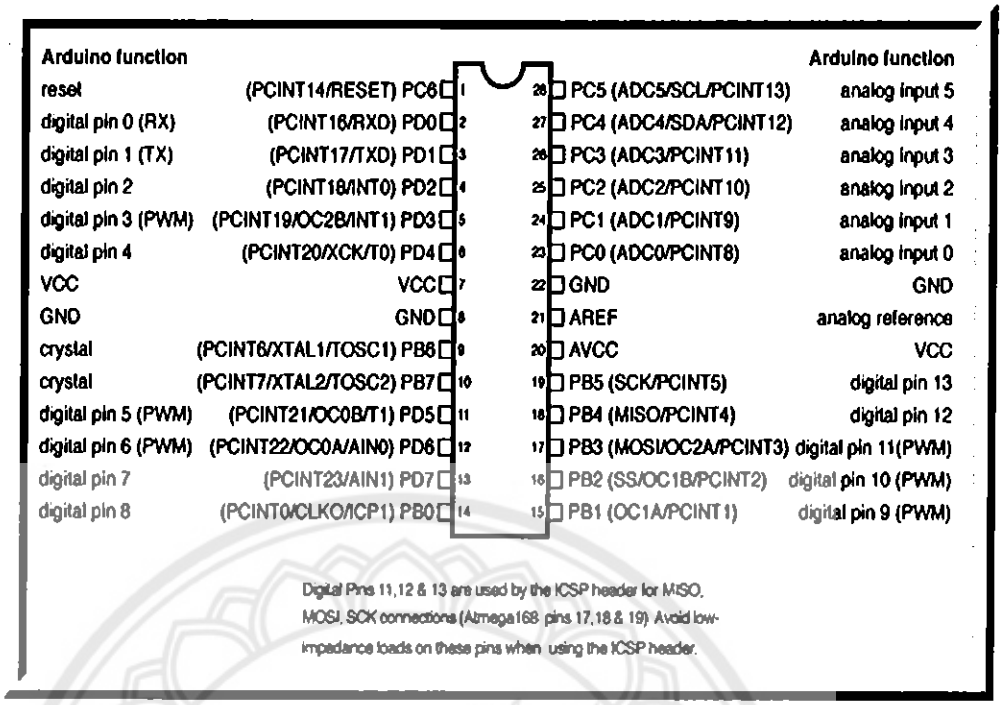
รูปที่ 2.1 รูปของแผงวงจร Arduino รุ่น Uno r3

แผงวงจร Arduino มีจุดเด่นในเรื่องของความง่ายต่อการเรียนรู้และใช้งานเนื่องจากได้มีการออกแบบคำสั่งต่าง ๆ ขึ้นมาสนับสนุนการใช้งานด้วยรูปแบบที่ง่ายไม่ซับซ้อนและมีข้อดีกว่าบอร์ดสำเร็จรูปตัวอื่นคือใช้งานง่ายมีโปรแกรมพัฒนาที่ไม่ซับซ้อนมีโปรแกรมพัฒนา Arduino ใช้งานง่ายสำหรับมือใหม่และมีความสามารถครบตามความต้องการของนักพัฒนามืออาชีพซึ่งแผงวงจร Arduino เป็นไมโครคอนโทรลเลอร์ที่ใช้ตัวประมวลผลตระกูลเอวีอาร์ขนาดเล็กเหมาะ

สำหรับนำไปใช้ในการศึกษาเรียนรู้ระบบไมโครคอนโทรลเลอร์และสามารถนำไปประยุกต์ใช้งานเกี่ยวกับการควบคุมอุปกรณ์อินพุตและเอาท์พุตได้มากมายภาษาในการเขียนโปรแกรมลงบน Arduino ใช้ภาษา C++ ซึ่งเป็นรูปแบบของโปรแกรมภาษาซีประยุกต์แบบหนึ่งมีโครงสร้างของตัวภาษาโดยรวมใกล้เคียงกันกับภาษาซีมาตรฐานเพียงแต่ได้มีการปรับปรุงรูปแบบในการเขียนโปรแกรมบางส่วนที่ผิดเพียนไปจากมาตรฐานเล็กน้อยเพื่อทำให้ลดความยุ่งยากในการเขียนโปรแกรมและให้ผู้เขียนโปรแกรมสามารถเขียนโปรแกรมได้ง่ายและสะดวกมากขึ้นกว่าการเขียนภาษาซีตามแบบมาตรฐานโดยตรง

ตัวแผงวงจร Arduino ที่ใช้ในโครงงานนี้จะกล่าวถึงสถาปัตยกรรมของเอวีอาร์ขนาด 8 บิต โดยเป็นตัวประมวลผลแบบ RISC(Reduced instruction set computer) และมีหน่วยความจำแบบฮาวาร์ด (Harvard) ซึ่งแยกหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลออกจากกัน ดังแสดงในรูปที่ 2.3 โดยใช้หน่วยความจำแบบแฟลต (Flash) เป็นหน่วยความจำโปรแกรมและใช้หน่วยความจำแบบ SRAM สำหรับเป็นหน่วยความจำข้อมูลนอกจากนี้ยังมีหน่วยความจำแบบ EEPROM ซึ่งสามารถเก็บข้อมูลได้โดยไม่จำเป็นต้องมีไฟเลี้ยงซึ่งมีคุณสมบัติเด่นดังนี้

1) หน่วยความจำโปรแกรมแบบ FLASH ขนาด 32 กิโลไบต์
2) หน่วยความจำข้อมูลแบบ SRAM ขนาด 2 กิโลไบต์
3) หน่วยความจำข้อมูลแบบ EEPROM ขนาด 1 กิโลไบต์
4) สนับสนุนการเชื่อมต่อแบบ I2C bus
5) พอร์ตอินและพุตเอาต์พุตจำนวน 23 บิต
6) วงจรแปลงแอนะล็อกเป็นดิจิตอลขนาด 10 บิตในตัวจำนวน 8 ช่อง
7) ทำงานได้ตั้งแต่ย่านแรงดัน 1.8 - 5.5 โวลต์
8) ความถี่ใช้งานสูงสุด 20 เมกกะเฮิรตซ์
9) วงจรสื่อสารอนุกรม
10) ตัวจับเวลาและตัวนับขนาด 8 บิต จำนวน 2 ตัวและU3586 ขนาด 16 บิตจำนวน 1 ตัว
11) สนับสนุนช่องสัญญาณสำหรับสร้างสัญญาณพีดับเบิลยูเอ็ม (PWM) จำนวน 6 ช่อง



รูปที่ 2.2 โครงสร้างไมโครคอนโทรลเลอร์ รุ่นATmega328P-PU

2.2 การประมวลผลภาพ (Image Processing) [2]

2.2.1 การประมวลผลภาพ

การประมวลผลภาพ (Image Processing) หมายถึง การนำภาพมาประมวลผลหรือคิดคำนวณด้วยคอมพิวเตอร์ เพื่อให้ได้ข้อมูลที่เราต้องการทั้งในเชิงคุณภาพและปริมาณ

รูปที่ 2.3 กระบวนการประมวลผลภาพ

โดยมีขั้นตอนต่าง ๆ ที่สำคัญ คือ การทำให้ภาพมีความคมชัดมากขึ้น การกำจัดสัญญาณรบกวนออกจากภาพ การแบ่งส่วนของวัตถุที่เราสนใจออกมาจากภาพ เพื่อนำภาพวัตถุที่ได้ไปวิเคราะห์หาข้อมูลเชิงปริมาณ เช่น ขนาด รูปร่าง และทิศทางการเคลื่อนของวัตถุในภาพ จากนั้นเราสามารถนำข้อมูลเชิงปริมาณเหล่านี้ไปวิเคราะห์ และสร้างเป็นระบบ เพื่อใช้ประโยชน์ในงานด้านต่างๆ เช่น

- ระบบรู้จำลายนิ้วมือเพื่อตรวจสอบว่าภาพลายนิ้วมือที่มีอยู่นั้นเป็นของผู้ใด

- ระบบตรวจสอบคุณภาพของผลิตภัณฑ์ในกระบวนการผลิตของโรงงานอุตสาหกรรม

- ระบบคัดแยกเกรดหรือคุณภาพของพืชผลทางการเกษตร

- ระบบอ่านรหัสไปรษณีย์อัตโนมัติเพื่อคัดแยกปลายทางของจดหมายที่มีจำนวนมากในแต่ละวัน โดยใช้ภาพถ่ายของรหัสไปรษณีย์ที่อยู่บนซอง

- ระบบเก็บข้อมูลรถที่เข้าและออกอาคาร โดยใช้ภาพถ่ายของป้ายทะเบียนรถเพื่อประโยชน์ในด้านความปลอดภัย

- ระบบดูแลและตรวจสอบสภาพการจราจรบนท้องถนนโดยการนับจำนวนรถบนท้องถนนในภาพถ่ายด้วยกล้องวงจรปิดในแต่ละช่วงเวลา

- ระบบรู้จำใบหน้าเพื่อเฝ้าระวังผู้ก่อการร้ายในอาคารสถานที่สำคัญ ๆ หรือในเขตคนเข้าเมือง

จะเห็นได้ว่าระบบเหล่านี้จำเป็นต้องมีการประมวลผลภาพจำนวนมาก และเป็นกระบวนการที่ต้องทำซ้ำ ๆ กันในรูปแบบเดิมเป็นส่วนใหญ่ ซึ่งงานในลักษณะเหล่านี้ หากให้มนุษย์วิเคราะห์เอง มักต้องใช้เวลามากและใช้แรงงานสูง อีกทั้งหากจำเป็นต้องวิเคราะห์ภาพเป็นจำนวนมาก ผู้วิเคราะห์ภาพเองอาจเกิดอาการล้า ส่งผลให้เกิดความผิดพลาดขึ้นได้ ดังนั้นคอมพิวเตอร์จึงมีบทบาทสำคัญในการทำหน้าที่เหล่านี้แทนมนุษย์ อีกทั้ง เป็นที่ทราบโดยทั่วกันว่า คอมพิวเตอร์มีความสามารถในการคำนวณและประมวลผลข้อมูลจำนวนมหาศาลได้ในเวลาอันสั้น จึงมีประโยชน์อย่างมากในการเพิ่มประสิทธิภาพการประมวลผลภาพและวิเคราะห์ข้อมูลที่ได้จากภาพในระบบต่าง ๆ ดังกล่าวข้างต้น

2.2.2 การทำงานของ Image Processing

Image Processing ทำงานโดยการนำภาพที่ ได้จากกล้องมาใส่ฟังก์ชัน และสมการทางคณิตศาสตร์ที่ต้องการเข้าไป เพื่อให้ได้ผลตามที่เรา ต้องการ โดยแบ่งการทำงานออกเป็น 3 ส่วน คือ

2.2.2.1 อุปกรณ์รับภาพ ทำหน้าที่จับภาพที่ ต้องการเข้าสู่คอมพิวเตอร์ เช่น กล้อง Video กล้อง CCD เครื่อง Scanner และรวมไปถึง Capture card สำหรับแปลงสัญญาณจากกล้อง Analog เข้าสู่คอมพิวเตอร์

2.2.2.2 อุปกรณ์ในการประมวลผล ทำหน้าที่ประมวลผลภาพที่รับมาจากอุปกรณ์รับภาพ โดยการคำนวณทางคณิตศาสตร์เพื่อให้ได้ผลลัพธ์ตามที่เราต้องการ อุปกรณ์ที่ใช้ในการประมวลผลอาจอยู่ในรูปคอมพิวเตอร์ ไมโครคอนโทรลเลอร์ หรือเป็นคอนโทรลเลอร์สำเร็จรูปสำหรับแต่ละยี่ห้อโดยจัดเป็นชุดพร้อมใช้งานประกอบด้วย กล้อง Capture Card คอนโทรลเลอร์ และอุปกรณ์เพื่อต่อไปยัง Output

2.2.2.3 โปรแกรมประมวลผลภาพ ถือเป็นหัวใจหลักในกระบวนการประมวลผลภาพ เป็นตัวจัดการวิธีการทำงานกับภาพที่รับมา มีการปรับปรุงคุณภาพของภาพก่อนการประมวลผล การใช้อัลกอริทึม และฟังก์ชันทางคณิตศาสตร์เพื่อให้ได้ Output ตามที่เราต้องการ โปรแกรมที่ใช้ในการประมวลผลภาพมี หลายประเภททั้งแบบที่สามารถ Download มาใช้ กันแบบฟรี ๆ เช่น OpenCV, OpenVIDIA หรือแบบ ที่ต้องเสียค่าใช้จ่าย เช่น IMAQ, Scorpion และ แบบที่เป็น Firmware ติดมากับคอนโทรลเลอร์ เช่น KEYENCE, Omron, Schneider, Siemens

ระบบกระบวนการประมวลผลภาพที่ดีจะต้องประกอบ ด้วยอุปกรณ์รับภาพที่เหมาะสม มีความละเอียดและความเร็วในการจับภาพเพียงพอกับงานที่ใช้ อุปกรณ์ที่ใช้ในการประมวลผลมีความเร็วและความสามารถในการคำนวณ โปรแกรมจะต้องจัดการภาพอย่างอย่างเหมาะสมและเป็นระเบียบ รวมถึง สภาพแวดล้อมต่างๆในการทำงาน เช่น แสงจากภายนอกต้องไม่ไปรบกวนการทำงานของระบบขนาดความยาวโฟกัสของเลนส์กล้องและระบบแสงสว่างที่เหมาะสม

2.3 การเตรียมข้อมูลภาพ (Preprocessing) [3]

การประมวลผลภาพเรียกอีกอย่างหนึ่งว่าการประมวลผลภาพ (Image Processing) [5] หมายถึง การเรียกใช้ขั้นตอนหรือกรรมวิธีใดๆมากระทำกับภาพ โดยมีวัตถุประสงค์เพื่อปรับปรุงคุณภาพของภาพให้ได้ภาพใหม่ที่มีคุณสมบัติตามต้องการ เช่น ความคมชัด หรือการประหยัดพื้นที่ในการเก็บข้อมูลหรือใช้สำหรับการประมวลผลระดับสูง เช่น การจดจำรูปร่างลักษณะให้ได้อย่างแม่นยำ โดยทั่วไปแล้ววัตถุประสงค์ของการประมวลผลภาพ (Image Processing) ก็คือ

การประมวลผลภาพ (Image Processing): Image in - Image out : วิธีนี้จะใช้กระบวนการทาง การประมวลผลภาพดิจิตทัลเพื่อทำให้ได้ภาพออกมา เช่น การตกแต่งภาพด้วยโปรแกรม Photoshop

การวิเคราะห์ภาพ (Image Analysis): Image in - Measurement out : วิธีนี้จะใช้กระบวนการทางการประมวลผลภาพดิจิตทัลเพื่อทำให้ได้ค่าการวัดออกมา เช่น การวัดขนาดในงานอุตสาหกรรม

การเข้าใจและแปลความหมายภาพ (Image Understanding): Image in - High -level Description out : วิธีนี้จะใช้กระบวนการทางการประมวลผลภาพดิจิตทัลเพื่อให้ได้ผลลัพธ์ออกมาเป็นความหมาย ตัวอย่างเช่น การจดจำตัวอักษร

การประมวลผลด้วยคอมพิวเตอร์ สามารถทำได้โดยนำภาพที่ได้มาจากกล้องหรือ Image Source ต่างๆ ซึ่งเป็นสัญญาณ Analog แล้วนำมาแปลงเป็นสัญญาณดิจิตอลที่มีลักษณะ เป็นรหัสเชิง

ตัวเลข 0,1 ที่สามารถใช้รูปแบบทางคณิตศาสตร์เข้ามาช่วยในการคำนวณและการประมวลผลข้อมูลภาพด้วยคอมพิวเตอร์ต่อไป การประมวลผลภาพแบ่งออกได้เป็น 2 ระดับ คือ

1. การประมวลผลภาพระดับต่ำ

เป็นการประมวลผลขั้นแรกสุดก่อนที่จะนำไปสู่การประมวลผลภาพระดับสูงต่อไป นั่นคือหลังจากได้ภาพมา ภาพที่ได้ก็จะประกอบไปด้วยองค์ประกอบต่างๆ มากมาย รวมถึงสิ่งที่เราไม่ต้องการด้วย ในที่นี้เราจะเรียกว่า Noise ซึ่งทำให้ภาพที่ได้มีคุณภาพไม่ดี ยังไม่สามารถที่นำไปใช้ในการประมวลผลได้ ดังนั้น การประมวลผลภาพระดับต่ำจึงจะประกอบไปด้วยการกำจัดสัญญาณรบกวน การทำภาพให้ชัด การหาขอบภาพ การแปลงภาพขาว-ดำ (Binary Image) การแบ่งแยกรูปร่างวัตถุ เพื่อหาค่าตัวแปรต่างๆ มาอธิบายข้อมูลภาพ และมีวัตถุประสงค์ที่จะนำตัวแปรเหล่านี้มาใช้ในการประมวลผลภาพระดับสูงต่อไป

2. การประมวลผลภาพระดับสูง

เป็นการทำให้คอมพิวเตอร์รู้จักและเข้าใจภาพนั้นได้ เช่น การจดจำใบหน้าคน หรืออาจจะเป็นการจดจำตัวอักษร เป็นต้น ความแตกต่างของการประมวลผลภาพระดับต่ำและระดับสูง คือ ข้อมูลที่นำมาใช้ในการประมวลผลโดยการประมวลผลระดับต่ำจะใช้ค่าความสว่างหรือความเข้มของแสงโดยตรง ส่วนกลางประมวลผลระดับสูง ข้อมูลที่นำมาใช้ในการประมวลผลจะถูกแสดงในรูปแบบของสัญลักษณ์ โดยสัญลักษณ์เหล่านี้จะแสดงถึงสิ่งต่างๆ ที่อยู่ในภาพ และการใช้ตัวแปรที่ได้จากการประมวลผลภาพระดับต่ำ มาอธิบายถึงสัญลักษณ์เหล่านี้ การประมวลผลภาพระดับสูงนั้นส่วนใหญ่มักจะใช้ทฤษฎีต่างๆ เข้ามาใช้เป็นตัวช่วยในการทำงาน หรือเป็นหัวใจของโปรแกรม

3. การแทนภาพด้วยข้อมูลแบบดิจิตอล

ข้อมูลภาพแบบดิจิตอลเป็นภาพที่ถูกดัดแปลงมาจากภาพแบบต่อเนื่องหรือ Analog image ให้อยู่ในรูปตัวเลข หรือ Digital Image ด้วยวิธีการ Digitization โดยภาพ Analog จะถูกแบ่งเป็นพื้นที่สี่เหลี่ยมเล็ก ๆ ที่เรียกว่า พิกเซล (Pixels) โดยในแต่ละพิกเซลจะใช้ (x, y) ในการระบุตำแหน่ง การแสดงข้อมูลภาพดิจิตอลสามารถอธิบายได้ด้วยเมตริกซ์ (Matrix) M*N และให้จุดต่างๆ ที่อยู่ในเมตริกซ์เป็นจุดพิกัด (x, y) ใดๆ เป็นส่วนประกอบของภาพ ค่าของพิกเซล หรือฟังก์ชัน (x, y) ณ จุดใดๆ จะแสดงได้ด้วยค่าของความเข้มของแสง ซึ่งอาจแบ่งได้หลายระดับ ถ้ามี 2 ระดับก็จะเป็นแค่ 0 กับ 1 จุดต่างๆ ที่แสดงอยู่ในพิกัดก็ คือ พิกเซล หรือ Picture Element ซึ่งก็คือ ความสว่างหรือค่า Luminance ของภาพ ถ้าภาพนั้นเป็นภาพขาวดำ ขนาด8 บิตจะมีค่า L เท่ากับ หรือเท่ากับ 256 คือ ตั้งแต่ระดับ 0 ถึง 255 บางครั้งค่าความสว่างอาจหมายถึงระดับความละเอียดของภาพ

2.4 โปรแกรม Microsoft Visual Studio [4]

2.4.1 Microsoft Visual Studio คือ

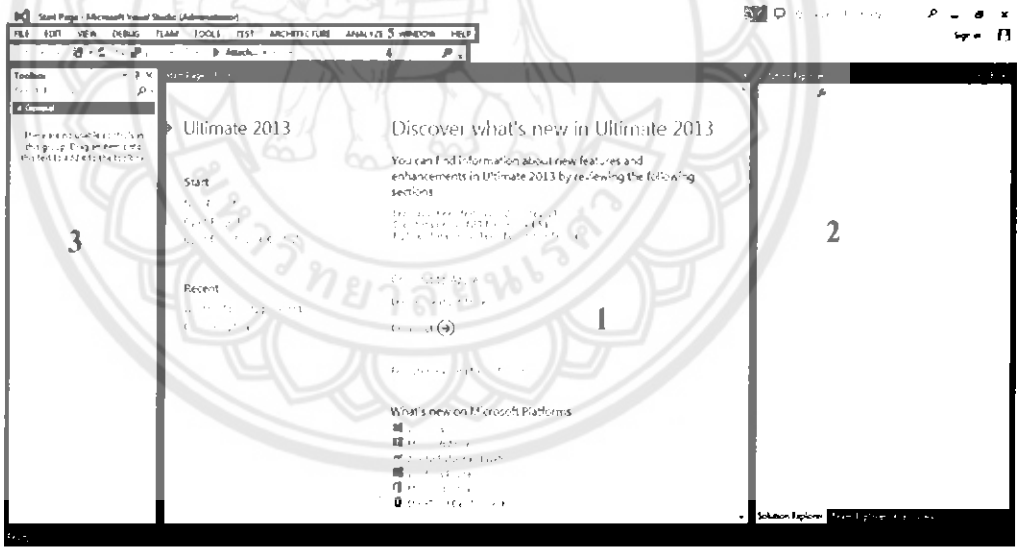
Microsoft Visual Studio คือ ชุดพัฒนาโปรแกรมด้วยภาษา Visual Basic.NET กล่าวคือ เป็นเครื่องมือที่ช่วยให้เราเขียนโปรแกรมด้วยภาษา Visual Basic .NET ได้

Visual Studio เป็นชุดพัฒนาแบบ IDE (Integrated Development Environment) ซึ่งหมายถึง สภาพแวดล้อมที่รวบรวมเครื่องมือและคุณสมบัติทุกอย่างที่จำเป็นสำหรับการพัฒนาโปรแกรมเข้าไว้ด้วยกันในที่เดียวกัน ไม่ว่าจะเป็นการออกแบบหน้าจอ เขียนโค้ด รันเพื่อทดสอบการทำงาน ค้นหาและแก้ไขข้อผิดพลาด เผยแพร่โปรแกรม ฯลฯ

นอกจาก Visual Basic .NET แล้ว Visual Studio ยังมีภาษาอื่นๆ ให้ใช้เขียนโปรแกรมได้อีก เช่น ภาษา C# และภาษา C++ Visual Studio ยังมี ภาษา F#

2.4.2 ส่วนประกอบของ Microsoft Visual Studio

จากรูปที่ 2.30 จะแบ่งส่วนประกอบของหน้าต่างโปรแกรมออกเป็น 5 ส่วน ซึ่งจะแบ่งได้ดังนี้



รูปที่ 2.4 หน้าต่างแสดงส่วนประกอบของ Visual Studio

1. Start Page เป็นหน้าต่างที่แสดงขึ้นมาตลอด ตอนที่เราเปิดโปรแกรมขึ้นมา หน้าต่างนี้จะเป็นเหมือน Intro ของโปรแกรม จะประกอบด้วย

- New Project ไว้สร้างโปรเจคใหม่ในการเขียนโปรแกรม
- Open Project เปิดโปรเจค ที่เราบันทึกเอาไว้ กลับมาแก้ไขใหม่ได้

- Recent Project จะแสดงรายชื่อโปรเจคที่เคยเปิดมาล่าสุด จำนวนหนึ่ง สามารถทำให้เราเปิดโปรเจคได้รวดเร็ว

2. Solution Explorer และ Properties

- Solution Explorer หน้าต่างแสดงโปรเจคของเราว่ามีอะไรบ้าง คือ My Family ประกอบด้วยไอเท็มที่เราสร้างขึ้นมา
- Properties หน้าต่าง Properties นี้ จะแสดงคุณสมบัติของ Object ที่เราเลือกไว้ แก้คุณสมบัติต่างๆของ Object นั้น

3. Toolbox เป็นหน้าต่างที่รวมเครื่องมือต่างๆไว้สร้าง Application ของเรา ไว้ออกแบบหน้าจอได้อย่างง่ายๆ โดยเพียงคลิกลาก คอนโทรล (Control) มาวางบน ฟอร์ม (Form) ก็จะได้ ออบเจ็ค (Object) บนฟอร์มนั้น ซึ่ง มีอยู่หลายกลุ่มคอนโทรลด้วยกัน

- Common Control คอนโทรลพื้นฐานต่างๆ เช่น Checkbox, Label, List Box ฯลฯ
- Containers คอนโทรลที่ไว้จัดกลุ่มให้กลับคอนโทรลอื่นๆ โดยบรรจุคอนโทรลนั้นไว้ภายในตัวเดียวกัน เช่น Group Box, Panel ฯลฯ
- Menus & Toolbars เป็นคอนโทรลที่ช่วยสร้าง เมนู และ ทูลบาร์
- Data คอนโทรลที่ไว้ใช้งานกับฐานข้อมูล
- Components เป็นคอนโทรลที่ไม่แสดงรูปร่างออกมาทางฟอร์ม แต่จัดเตรียมทำงานให้กับโปรแกรม เช่น Timer, Process ฯลฯ
- Printing คอนโทรลที่เกี่ยวกับการพิมพ์เอกสารออกทาง Printer
- Dialogs คอนโทรลที่ใช้แสดงไดอะล็อกซ์พื้นฐานชนิดต่างๆ เช่น ไดอะล็อกซ์สำหรับเปิดไฟล์ และไดอะล็อกซ์สำหรับเลือกสี เป็นต้น

4. ทูลบาร์หลัก (Standard Toolbar)

เป็นแถบเครื่องมือที่รวบรวมปุ่มต่างๆเอาไว้ ซึ่งปุ่มเหล่านี้จะเรียกใช้คำสั่งที่ใช้บ่อย เพื่อความสะดวกในการทำงาน เช่น ปุ่มแรกจะเทียบเท่ากับการสร้างโปรเจ็คใหม่ (New Project) ในเมนูไฟล์เป็นต้น

5. เมนูบาร์ (Menu Bar)

เป็นส่วนที่รวบรวมคำสั่งทุกอย่างในการทำงานของ Visual Studio โดยแบ่งเป็นเมนูคำสั่งตามไอเท็มต่างๆดังนี้

- File คำสั่งที่ใช้สร้างโปรเจ็คใหม่ เปิดโปรเจ็ค ปิดโปรเจ็ค เป็นต้น
- Edit คำสั่งที่ใช้แก้ไข ตัด คัดลอก วาง ย้อนกลับ เป็นต้น
- View คำสั่งที่ใช้แสดงเครื่องมือต่างๆของ Visual Studio
- Properties คำสั่งที่ใช้จัดการเกี่ยวกับโปรเจค แก้ไขคุณสมบัติต่างๆของ ออบเจ็ค
- Build คำสั่งที่ใช้คอมไพล์โปรเจค เป็นไฟล์ *.EXE
- Debug คำสั่งที่ช่วยในการรันและตรวจสอบหาข้อผิดพลาดของโปรแกรม
- Data คำสั่งที่ใช้ติดต่อกับฐานข้อมูล
- Format คำสั่งที่ใช้จัดตำแหน่งของออบเจ็ค
- Tool คำสั่งที่ใช้เรียกเครื่องมือส่วนเสริม

2.5 ภาษา C# [5]

2.5.1 C# คืออะไร

C# คือ ภาษาคอมพิวเตอร์ประเภท object-oriented programming พัฒนาโดย Microsoft โดยมีจุดมุ่งหมายในการรวมความสามารถการคำนวณของ C++ ด้วยการโปรแกรมง่ายกว่าของ Visual Basic โดย C# มีพื้นฐานจาก C++ และเก็บส่วนการทำงานคล้ายกับ Java C# ได้รับการออกแบบให้ทำงานกับ .NET platform ของ Microsoft จุดมุ่งหมายคือ อำนวยความสะดวกในการแลกเปลี่ยนสารสนเทศและบริการผ่านเว็บ และทำให้ผู้พัฒนาสร้างโปรแกรมประยุกต์ในขนาดกะทัดรัด C# ทำให้โปรแกรมง่ายขึ้นผ่านการใช้ Extensible Markup Language (XML) และ Simple Object Access Protocol (SOAP) ซึ่งยอมให้เข้าถึงอ๊อบเจคของโปรแกรมหรือเมธอด โดยปราศจากความต้องการให้ผู้เขียนโปรแกรมเขียนคำสั่งเพิ่มในแต่ละขั้นตอน เนื่องจากผู้เขียนโปรแกรมสามารถสร้างบนคำสั่งที่มีอยู่ แทนที่การคัดลอกซ้ำ C# ภาษา C# ถูกพัฒนาขึ้นโดยเป็นส่วนหนึ่งในการพัฒนาโครงสร้างพื้นฐานของ .NET Framework เป็นการการนำข้อดีของภาษาต่างๆ (เช่นภาษา Delphi , ภาษา C++) มาปรับปรุงเพื่อให้มีความเป็น OOP (โปรแกรมเชิงวัตถุ) มากขึ้น ขณะเดียวกันก็ลดความซับซ้อนในโครงสร้างของภาษาลง (เรียบง่ายกว่าภาษา C++) และมีสิ่งที่เกินความจำเป็นน้อยลง (เมื่อเทียบกับ Java) C# ถูกรับรองจากหน่วยงาน

ECMA (หน่วยงานกำหนดมาตรฐานสากลด้านสารสนเทศ) และ ISO และปัจจุบันไมโครซอฟท์ยังพัฒนาภาษานี้อย่างต่อเนื่อง (ปัจจุบันเป็นเวอร์ชัน 3.0)

2.5.2 เปรียบเทียบภาษา C# กับภาษาอื่นๆ

1. ถ้าพูดถึงความใกล้เคียงกับภาษาอื่นๆ ภาษา C# ใกล้เคียงกับภาษา Java มากที่สุด โดยมีความเหมือนกันถึง 70% ดั้งนั้นนักเขียนโปรแกรมภาษา Java จึงอาจย้ายมาเขียนภาษา C# ได้โดยศึกษาว่ามีสิ่งใดที่แตกต่างกันบ้าง ภาษา C# ยังมีความคล้ายคลึงกับภาษ C++.NET และภาษา VB.NET เป็นอย่างมาก ทำให้นักเขียนโปรแกรมภาษา C# สามารถอ่าน-เขียนโค้ดในภาษากลุ่มนี้ได้เมื่อฝึกฝนเพียงเล็กน้อย

2. C# และภาษา Java ทั้งคู่เป็นแบบสืบจากคลาสหลักได้คลาสเดียว ขณะที่ภาษา C++ สามารถสืบจากคลาสหลักได้มากกว่าหนึ่ง (Multiple inheritance) โดยภาษา C# และภาษา Java ใช้ Interface มาทดแทน Multiple inheritance เหมือนกันทั้งคู่

3. สิ่งที่ภาษา C# และ Java มีร่วมกันคือเรื่อง Garbage Collection แต่ไม่มีใน C++ จึงทำให้ดูเหมือนว่าภาษา Java ต่อยอดมาจากภาษา C++ และ C# ต่อยอดมาจาก Java อีกที ที่เป็นเช่นนั้นเพราะทั้ง Java และ C# มีต้นสายมาจาก C++ ทำให้สองภาษานี้ดูคล้ายกัน แต่ภาษา C# ไม่ใช่ภาษา Java มันมีกลไกที่เป็นเอกลักษณ์หลายอย่าง เช่น พารามิเตอร์แบบ reference และ output การจัดเก็บ object ไว้ใน stack (struck) การทำ Versioning และยังมีสิ่งใหม่ๆ ที่เป็นข้อดี เช่น delegate, properties และ operator overloading ซึ่งจะไม่พบในภาษา Java

2.5.3 จุดเด่นหลักๆ ของภาษา C#

1. Component oriented – เป็นภาษาที่เน้นชิ้นส่วนโดยถูกออกแบบมาเป็นอย่างดีทำให้สามารถนำมาใช้ต่อกันเป็นอะไรก็ได้

2. สิ่งต่าง ๆ ใน C# เป็นออบเจ็กต์ทั้งหมด

3. เป็นภาษา ที่ทนทาน (robust) - ทนต่อความผิดพลาด ไม่ทำให้ระบบแฮงก์หรือระบบทำงานช้า เพราะ C# มีข้อดีคือ garbage collection, exception, type-safety และ versioning

4. ภาษา C# จัดเตรียมกลไกไว้หลายอย่างที่ช่วยให้ผู้เขียนโปรแกรมสามารถนำโค้ดที่เขียนไว้ในโปรเจคหนึ่งไปใช้กับอีกโปรเจคหนึ่งได้ง่าย นอกจากนั้นภาษา C# ยังสามารถเรียกใช้คลาสหลายพันคลาสใน .NET Framework ได้โดยตรง ทำให้ลดเวลาการพัฒนาซอฟท์แวร์ได้มาก

2.6 ภาพระดับสีเทา (Grayscale Image) [6]

ภาพระดับสีเทาเป็นภาพซึ่งค่าในแต่ละจุดภาพคือค่าความเข้มของสีแต่ละตำแหน่งของจุดภาพนั้น ซึ่งค่าที่เป็นไปได้ของภาพระดับสีเทาทั้งหมดขึ้นอยู่กับจำนวนบิตที่ใช้ ตัวอย่าง เช่น ภาพระดับสีเทา 8 บิตที่ระดับสีทั้งหมด 256 ระดับ โดยนิยมระบุในช่วง 0-1 หรือ 0-255 แสดงระดับสีเทาดังในรูปที่ 2.31

รูปที่ 2.5 ภาพแสดงระดับสีเทา

ภาพระดับสีเทาจะมีการไล่ระดับความอ่อนแก่ของสี ซึ่งอยู่ระหว่างสีขาวและสีดำ Halftone Image อย่างต่อเนื่อง โดยที่ค่าแต่ละพิกเซลของภาพจะหมายถึงความเข้มแสงแต่ละตำแหน่งของพิกเซล ที่อยู่ในรูประดับสีเทา การเปลี่ยนภาพจากระบบสีอาร์จีบี (RGB) เป็นระดับสีเทา (Gray Scale) จะใช้สมการดังนี้ในการเปลี่ยน

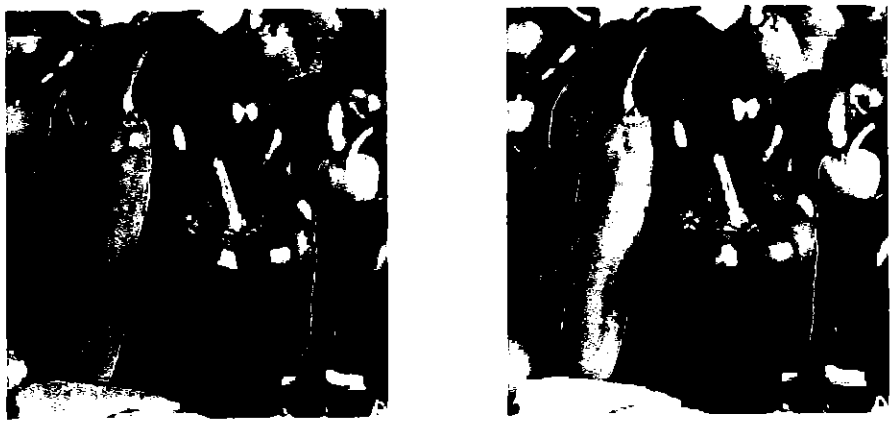
$$Y = 0.3R+0.59G+0.11B \quad (2.1)$$

โดย Y แทน ค่าระดับสีเทา ณ จุดพิกเซล ที่เราต้องการหา โดยมีค่าระหว่าง 0-255

R แทน ค่าสีแดง ณ จุด ที่ต้องการหา โดยมีค่าระหว่าง 0-255

G แทน ค่าสีเขียว ณ จุด ที่ต้องการหา โดยมีค่าระหว่าง 0-255

B แทน ค่าสีน้ำเงิน ณ จุด ที่ต้องการหา โดยมีค่าระหว่าง 0-255

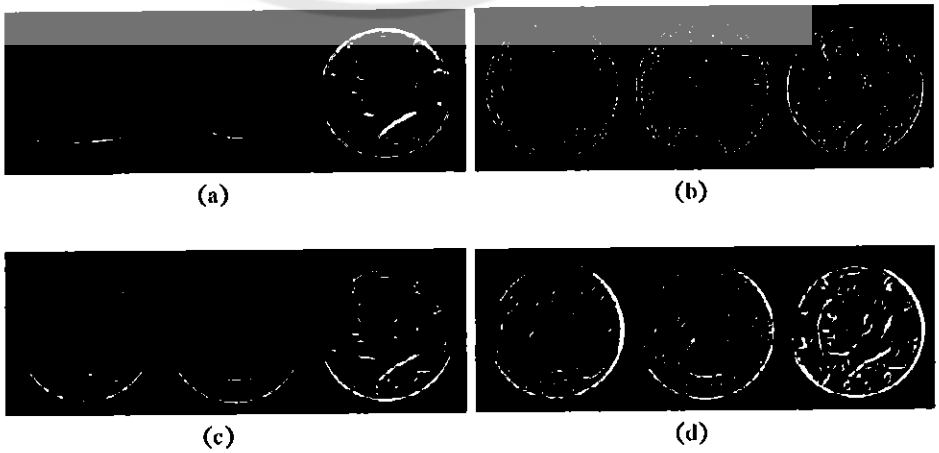


รูปที่ 2.6 การแปลงภาพอาร์จีบีเป็นภาพระดับสีเทา

2.7 การหาขอบภาพ (Edge Detection) [7]

การตรวจหาเส้นขอบของภาพ (Edge Detection) เป็นวิธีการที่สำคัญในการประมวลผลภาพ เพราะหากมีอัลกอริทึมที่ดี ผลลัพธ์ของภาพที่ตัดแบ่งออกมาจะมีประสิทธิภาพ และส่งผลให้ง่ายต่อการทำ งานในขั้นตอนต่อไป ซึ่งขั้นต่อไปคือ กระบวนการการรู้จำ (Recognition) ซึ่งการตรวจหาเส้นขอบของภาพ จะอาศัยค่า ความแตกต่างของค่าสีในพื้นที่ของภาพ ซึ่งเปรียบเทียบโดยค่าเกรย์สเกล (grey scale)

การหาขอบภาพด้วย Mask เป็นการหาวัตถุในภาพ หรือลายเส้นที่มีระดับ ความแตกต่างของความเข้ม แสงมากๆ ซึ่งมีวิธีที่ใช้ในการหาหลายวิธีเช่น Sobel Operator, Pewit Operator และ Laplacian Operator เป็นต้น ซึ่งผลลัพธ์ที่ออกมานั้นมีความคล้ายกัน อย่างมาก ต่างกันเพียงแค่ Mask ของแต่ละวิธีเท่านั้นเอง ดูผลลัพธ์ ของแต่ละวิธีดังรูปที่ 2.33



รูปที่ 2.7 ตัวอย่างภาพผลลัพธ์ที่ได้จากการหาขอบภาพด้วยวิธีต่างๆ

(a) ภาพต้นฉบับ

(b) ภาพการหาขอบด้วยวิธี Laplace Operator

(c) ภาพการหาขอบด้วยวิธี Prewit Operator

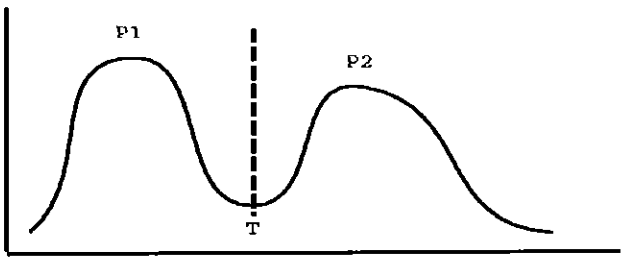
(d) ภาพการหาขอบด้วยวิธี Sobel Operator

2.8 การแปลงภาพสีให้เป็นภาพขาว-ดำ (Thresholding) [8]

การแปลงภาพสีให้เป็นภาพขาว-ดำ (Thresholding) เป็นกระบวนการแปลงภาพสีให้มีการแสดงผลได้แค่ 2 ระดับ คือ ขาว และดำ โดยจะแปลงข้อมูลภาพให้เป็นภาพ binary (Binary Image) มีกระบวนการแปลงภาพที่มีความเข้มหลายระดับ (Multilevel Image) ให้เป็นภาพที่มีความเข้มเพียง 2 ระดับ หรือ 1 บิต (bit) คือ 0 และ 1 โดย 0 แทนด้วยจุดที่มีภาพสีขาว และ 1 แทนด้วยจุดที่มีภาพสีดำ

Thresholding Technique คือการพิจารณาจุดภาพ ในภาพว่าจุดใดควรจะเป็นจุดขาว หรือจุดใดควรจะเป็นจุดที่มีค่าเท่ากับ 1 (จุดดำ) โดยจะทำการเปรียบเทียบค่าของแต่ละจุดภาพ (f(x,y)) กับค่าคงที่ที่เรียกว่า ค่าขีดแบ่ง (Threshold) เทคนิคนี้นิยมใช้กันมากในกรณีที่ความแตกต่างระหว่างวัตถุ (Object) และพื้นหลัง (Background) ค่าจุดภาพในภาพที่มีค่าน้อยกว่าค่าขีดแบ่งที่กำหนดไว้ จะถูกกำหนดเป็น 1 (จุดดำ) และถ้าค่าของจุดภาพใด ๆ ในภาพมีค่ามากกว่าหรือเท่ากับค่า Threshold จะถูกกำหนดให้เป็น 0 (จุดขาว)

สำหรับวิธีการกำหนดค่าขีดแบ่งแบบค่าเชิงเดี่ยว (Single Threshold) ที่จะใช้ในการแปลงภาพสีให้เป็นภาพขาว-ดำ สามารถทำได้โดยการกำหนดค่าขีดแบ่งของผู้ใช้เองหรือใช้วิธีการของ Otsu ก็ได้ โดยหลักการเลือกค่าขอบของ Otsu นั้นคือ จะต้องเป็นค่าที่สามารถทำให้ฮิสโตแกรมทั้งสองกลุ่มมีการกระจายตัวน้อยที่สุด ดังแสดงในภาพที่ 2.6



รูปที่ 2.8 แสดงฮิสโตแกรมที่วัตถุและพื้นหลังมีค่าความเข้มแสงแยกออกจากกัน

ในทางปฏิบัติเราไม่สามารถทำการเปลี่ยนรูปร่างของฮิสโตแกรมทั้งสองยอดได้ แต่เราสามารถเปลี่ยนลักษณะการกระจายตัวของทั้งสองยอดได้ด้วยการใช้ค่าThresholdเป็นตัวแบ่ง นั่นคือ ถ้าเราเพิ่มค่าดังกล่าว เรากำลังทำให้การกระจายตัวของยอดหนึ่งลดลงและการกระจายตัวของอีกยอดหนึ่งเพิ่มขึ้น ซึ่งเป้าหมายของ Otsu คือ การเลือกค่าThreshold ที่ทำให้"การกระจายตัวรวม"ของทั้งสองยอดมี ค่าต่ำที่สุด "การกระจายตัวรวม" ของทั้งสองยอดนั้น สามารถวัดได้โดยความแปรปรวนภายในกลุ่มรวมกัน (Within-class variance, σ^2_{Within}) ซึ่งมีค่าเท่ากับผลรวมของความแปรปรวน (Variance) คูณกับจำนวนพิกเซลของแต่ละกลุ่มและสมการทางคณิตศาสตร์ที่ใช้วัดการกระจายตัวรวมของทั้งสองกลุ่มนั้น แสดงไว้ในสมการที่ 2.2

$$\sigma^2_{Within}(T) = n_D(T)\sigma^2_D(T) + n_B(T)\sigma^2_B(T) \tag{2.2}$$

เมื่อ T คือ ค่า Threshold ที่ใช้แบ่งทั้งสองบริเวณออกจากกัน

$\sigma^2_D(T)$ คือ ความแปรปรวน (Variance) ของบริเวณด้านมืด

$\sigma^2_B(T)$ คือ ความแปรปรวน (Variance) ของบริเวณด้านสว่าง

p(i) คือ จุดพิกเซลที่ i ในพิกัดภาพ

$n_D(T)$ คือ จำนวนพิกเซลทั้งหมดของบริเวณด้านมืด (Dark area) ที่มีค่าความเข้มแสงตั้งแต่ 0 จนถึงค่าความเข้มแสงเท่ากับ T-1 ซึ่งสามารถคำนวณได้จากสมการที่ 2.3

$$n_D(T) = \sum_{I=0}^{T-1} p(i) \tag{2.3}$$

$n_B(T)$ คือ จำนวนพิกเซลทั้งหมดของด้านสว่าง (Bright area) ที่มีค่าความเข้มแสงตั้งแต่ T จนถึงค่าความเข้มแสงเท่ากับค่าสูงสุดคือ 2^B-1 เมื่อ B คือจำนวนบิตของระบบภาพ ซึ่งถ้าเป็นระบบทั่วไปที่ เป็นระบบภาพ 8 บิต พจน์ 2^B-1 จะมีค่าเท่ากับ 255 และจำนวน พิกเซลทั้งหมดของด้านสว่างสามารถคำนวณได้จากสมการที่ 2.4

$$n_B(T) = \sum_{I=T}^{2^B-1} p(i) \tag{2.4}$$

เราสามารถใช้สมการที่ 2 เพื่อหาค่า Threshold ที่เหมาะสมได้ โดยการเลือกค่า Threshold ที่ทำให้พจน์ดังกล่าวมีค่าน้อยที่สุด อย่างไรก็ตาม การคำนวณสมการที่ 2 กับทุกค่า Threshold ที่เป็นไปได้นั้นมีความยุ่งยากมาก เนื่องจากจะต้องคำนวณความ แปรปรวนของแต่ละบริเวณ ทั้งบริเวณที่มืดและสว่างของ Threshold ทุกค่า ซึ่งเราสามารถเลือกค่า Threshold ที่เหมาะสมได้ ด้วยวิธีการที่ง่ายกว่านั้น นั่นคือ ถ้าเรานำค่าความแปรปรวนภายในกลุ่มรวมกันมาลบออกจากค่าความ

แปรปรวนรวม เราจะได้ 2 พจน์ที่ Otsu เรียกว่า ความแปรปรวนระหว่างกลุ่ม (Between-class variance, $\sigma^2_{Between}$) ซึ่งสามารถคำนวณได้จากสมการที่ 2.5

$$\sigma^2_{Between} = \sigma^2 - \sigma^2_{Within} \tag{2.5}$$

หรือ $\sigma^2_{Between} = n_D(T)[u_B(T) - \mu]^2 + n_B(T)[u_B(T) - \mu]^2$ (2.6)

เมื่อ $\sigma^2_{Between}$ คือ ความแปรปรวนรวมของทั้งฮิสโตแกรม

μ คือ ค่าเฉลี่ยรวมของทั้งฮิสโตแกรม

หลังจากที่หาค่าขอบได้แล้ว จึงทำการกำหนดให้จุดภาพที่มีค่ามากกว่าค่าขอบเป็นจุดภาพพื้นหน้าที่มีค่าเท่ากับ 1 ส่วนจุดภาพที่เหลือเป็นพื้นหลังที่มีค่าเท่ากับ 0

ในการทำภาพ Binary โดยการทำ Thresholding ให้ได้ภาพดีและคมชัด ต้องเกิดจากการเลือกค่า Threshold ที่ถูกต้องและเหมาะสม ถ้าเลือกค่า Threshold ไม่เหมาะสม เช่น ค่า Threshold ที่มากหรือน้อยจนเกินไป ภาพที่ได้จะขาดความคมชัดหรืออาจทำให้รายละเอียดของภาพขาดหายไป หรือภาพที่ได้อาจจะมืดเกินไป หรือสว่างเกินไป หรืออาจจะเป็นภาพที่มีสิ่งรบกวน (Noise) เกิดขึ้น ทำให้ภาพผลลัพธ์ที่ได้ไม่ชัดเจน

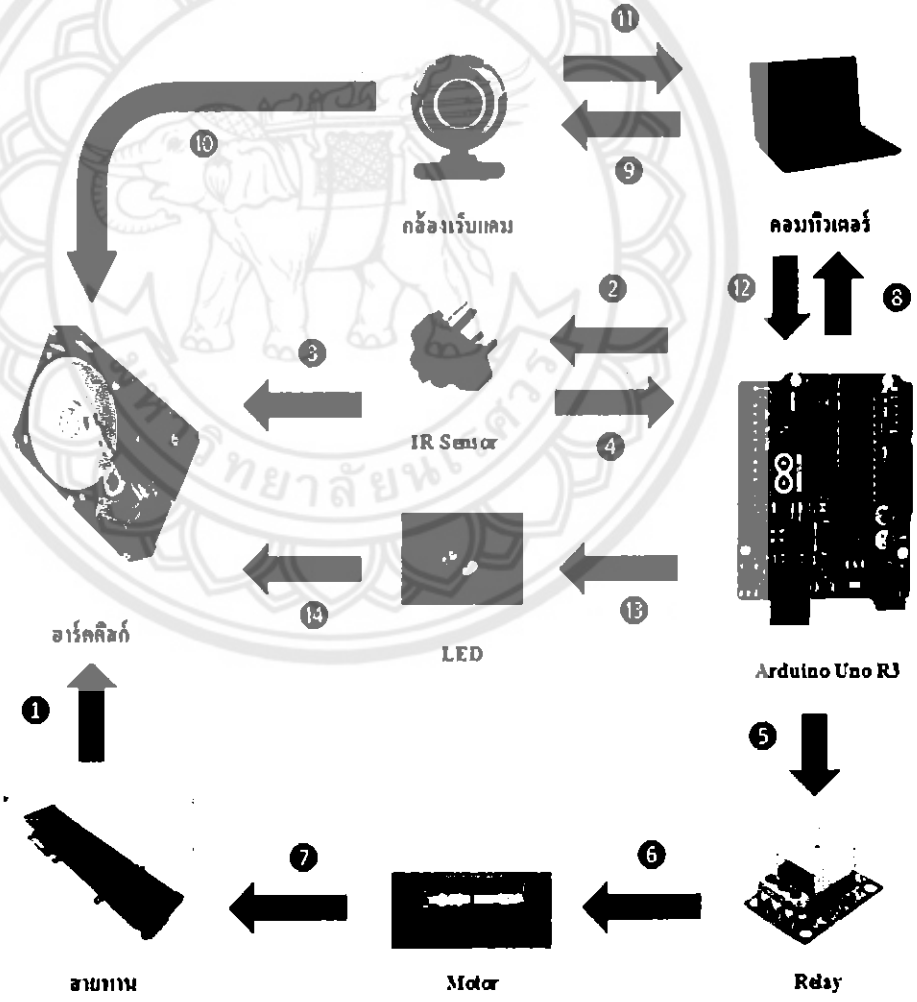
2.9 การเข้าคู่รูปแบบ (Template Matching) [9]

วิธีการเข้าคู่รูปแบบ(Template Matching) คือ การนำรูปภาพที่ต้องการหาค่าไปเปรียบเทียบกับรูปแบบตัวอย่าง (Template) ซึ่งรูปแบบตัวอย่างนี้จะเก็บค่า และกำหนดลักษณะสำคัญต่างๆ ที่สามารถแยกความแตกต่างของตัวอักษรต่างๆ ได้ และเมื่อนำเอาภาพตั้งต้นที่มีมาเปรียบเทียบกับรูปแบบที่มี เพื่อตรวจสอบความคล้ายคลึงของภาพทั้งสอง ซึ่งอาศัยค่าระดับที่ตั้งขึ้นมาเพื่อตรวจสอบ ซึ่งจะมีข้อเสียของวิธีการนี้ คือ มีความอ่อนไหวต่อข้อมูลที่มีการแทรกซ้อนขนาดและความเอียง ซึ่งปัญหาเหล่านี้แก้ไขได้ แต่ต้องอาศัยขั้นตอนการประมวลผลขั้นต้นที่ดี

บทที่ 3
ขั้นตอนการดำเนินงาน

3.1 การออกแบบโครงงาน

จากรูปที่ 3.1 แสดงระบบการทำงานโดยรวมของการโครงงาน โดยเริ่มจากนำฮาร์ดดิสก์วางบนสายพาน เมื่อเซนเซอร์ตรวจเจอฮาร์ดดิสก์ เซนเซอร์ก็จะส่งคำสั่งไปยังไมโครคอนโทรลเลอร์เพื่อสั่งให้มอเตอร์หยุดหมุนสายพาน หลังจากนั้นกล้องเว็บแคมจะทำการบันทึกภาพฮาร์ดดิสก์บนสายพาน แล้วนำไปประมวลผลภาพหาตำหนิบนจานแม่เหล็กและแสดงผลบนโปรแกรมที่ทำการออกแบบด้วย Visual Studio 2010 หลังจากนั้นภาพที่ผ่านการประมวลผลก็จะถูกจัดเก็บลงในคอมพิวเตอร์



รูปที่ 3.1 แสดงระบบการทำงานโดยรวม

อธิบายขั้นตอนการทำงานของระบบ

1. สายพานจะทำหน้าที่ลำเลียงฮาร์ดดิสก์

2. ไมโครคอนโทลเลอร์จ่ายไฟและควบคุมการทำงานของอินฟราเรดเซ็นเซอร์

3. อินฟราเรดเซ็นเซอร์ตรวจจับระยะการสะท้อนของวัตถุ

4. อินฟราเรดเซ็นเซอร์ส่งสัญญาณดิจิตอลไปยังไมโครคอลโทลเลอร์

5. ไมโครคอนโทลเลอร์ส่งสัญญาณดิจิตอลไปยังรีเลย์

6. รีเลย์ควบคุมการจ่ายไฟให้กับมอเตอร์

7. มอเตอร์จะทำหน้าที่ขับเคลื่อนสายพานลำเลียงฮาร์ดดิสก์

8. ไมโครคอนโทลเลอร์ ส่งคำสั่งไปยัง Microsoft Visual Studio 2010

9. Microsoft Visual Studio 2010 สั่งการให้กล้องเว็บแคมถ่ายภาพนิ่ง

10. กล้องเว็บแคมทำการถ่ายภาพฮาร์ดดิสก์บนสายพาน

11. กล้องเว็บแคมส่งภาพที่ได้จากการถ่ายไปยัง Microsoft Visual Studio 2010 เพื่อทำการประมวลผลภาพ

12. Microsoft Visual Studio 2010 ส่งคำสั่งไปยังไมโครคอนโทลเลอร์

13. ไมโครคอนโทลเลอร์ส่งสัญญาณดิจิตอลไปยังหลอดไฟแอลอีดี

14. หลอดไฟแอลอีดีมีหน้าที่ให้แสงสว่างแก่ฮาร์ดดิสก์

3.1.1 ส่วนขยายของระบบการทำงานโดยรวม

รูปที่ 3.2 แสดงแผนผังการทำงานโดยรวมของระบบ

จากรูป 3.2 สามารถอธิบายในส่วนขยายระบบการทำงานโดยรวมได้ดังนี้

1. วางฮาร์ดดิสก์ลงบนสายพานที่ควบคุมโดยมอเตอร์ ซึ่งใช้โปรแกรม Arduino ในการติดต่อ

2. เมื่อฮาร์ดดิสก์เคลื่อนผ่านอินฟราเรดเซ็นเซอร์ เซ็นเซอร์จะทำการรับข้อมูลและส่งข้อมูลไปยังไมโครคอนโทรลเลอร์ เพื่อสั่งให้มอเตอร์หยุดทำงาน 5 วินาที

3. หลังจากนั้นไมโครคอนโทรลเลอร์จะทำการติดต่อกับโปรแกรมที่ออกแบบโดย Visual Studio 2010 เพื่อสั่งให้กล้องบันทึกภาพ โดยใช้ไลบรารี่ AForge.Net ในการติดต่อกับกล้องเว็บแคม

4. กล้องเว็บแคมจะทำการถ่ายภาพและนำภาพมาผ่านกระบวนการประมวลผลภาพ ดังนี้

4.1 ปรับภาพให้เป็นสีเทา (Grayscale)

4.2 ทำการตรวจหาขอบภาพ (Edge Detection)

4.3 ทำการปรับความเข้มของภาพ (Thresholding)

4.4 ทำการสลับสีภาพ (Convert Black <-> White)

4.5 ทำการตรวจหาจานแม่เหล็กบนฮาร์ดดิสก์โดยใช้การตรวจจับวงกลม (Circle Detection)

4.6 หลังจากตรวจเจอวงกลม ก็ทำปรับพิกเซลนอกวงกลมกับตรงจานแกนแม่เหล็ก

4.7 ทำการตัดภาพตามรัศมีวงกลม (Cropping Circle)

4.8 นำภาพที่ได้จากข้อ 4.7 มาเทียบหาความแตกต่างกับภาพต้นฉบับด้วยวิธี Template Matching

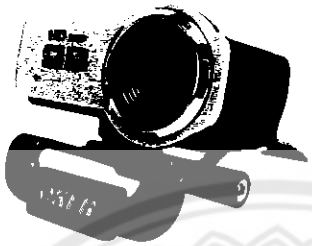
5. ภาพสุดท้ายที่ได้มา จะแสดงพิกเซลที่เหลือจากการลบกันของภาพ

6. นับจำนวนพิกเซลที่เหลือจากการลบ แล้วนำมาตีความแยกแยะว่า จานแม่เหล็กมีความสะอาด (Good) หรือ มีรอย (Defect) ตามเงื่อนไขที่กำหนดเอาไว้

7. สิ้นสุดการทำงาน

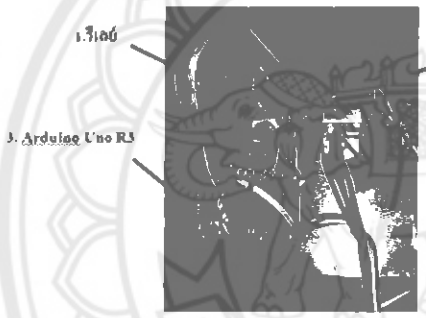
3.2 อุปกรณ์ที่ใช้ทำโครงงาน

อุปกรณ์ที่ใช้จะประกอบไปด้วยดังนี้ กล้องเว็บแคม , คอมพิวเตอร์ , สายพาน , บอร์ด Arduino Uno R3 , รีเลย์ , ตัวแปลงไฟฟ้า , อินฟราเรดเซ็นเซอร์ , สวิตช์ และ หลอดไฟ LED



รูปที่ 3.3 กล้องเว็บแคม

รูปที่ 3.4 สายพานลำเลียง



รูปที่ 3.5 วงจรตัวแปลงไฟฟ้า

รูปที่ 3.6 มอเตอร์ควบคุม

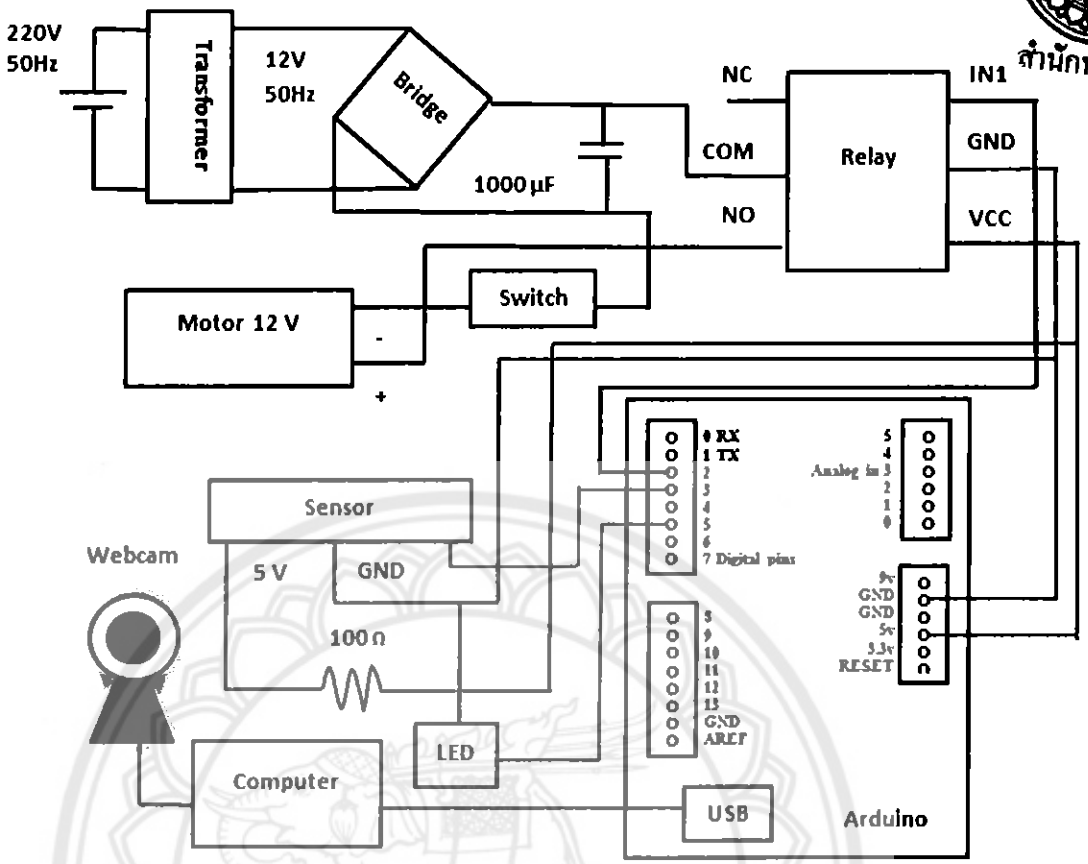
รูปที่ 3.7 อินฟราเรดเซ็นเซอร์

รูปที่ 3.8 คอมพิวเตอร์

รูปที่ 3.9 สวิตซ์

รูปที่ 3.10 หลอดไฟ LED

รูปที่ 3.11 ภาพรวมของโครงงาน



รูปที่ 3.12 วงจรในโครงงาน

จากรูปที่ 3.12 เป็นการแสดงการเชื่อมต่อของอุปกรณ์ไมโครคอนโทรลเลอร์ (Arduino) ซึ่งเป็นบอร์ดที่ใช้ในการรับข้อมูลจากอินฟราเรดเซนเซอร์และส่งคำสั่งไปควบคุมมอเตอร์ หลังจากนั้นข้อมูลที่เก็บไว้ในไมโครคอนโทรลเลอร์จะทำการส่งคำสั่งไปติดต่อกับโปรแกรมที่สร้างขึ้นมาเพื่อควบคุมให้กล้องเว็บแคมบันทึกภาพ และนำภาพที่บันทึกได้มาประมวลผลเพื่อหาสิ่งแปลกปลอมบนฮาร์ดดิสก์ แล้วก็จัดเก็บภาพไว้ลงในคอมพิวเตอร์

ส่วนของการเชื่อมต่ออุปกรณ์ต่างๆกับ Arduino มีดังนี้

1. อินฟราเรดเซนเซอร์จะเลือกใช้สายรับสัญญาณต่อเข้ากับบอร์ดควบคุมขาที่ 3 (Digital Pin)
2. สายรับสัญญาณ (IN1) ของรีเลย์ จะเชื่อมต่อกับ Digital Pin ที่มีให้เลือกทั้งหมด 14 ขาด้วยกัน และจากตัวอย่างเลือกใช้ขาที่ 2
3. สายที่ใช้รับแรงดันไฟฟ้า (VCC) ต่อกับขาที่ 5V ของบอร์ด
4. สายกราวด์ ต่อกับขา GND ของบอร์ด

การเชื่อมต่อไมโครคอนโทรลเลอร์เข้ากับคอมพิวเตอร์นี้จะใช้การเชื่อมต่อโดยผ่านสายยูเอสบี (USB) เช่นกันกับการเชื่อมต่อระหว่างกล้องวีดีโอกับคอมพิวเตอร์

3.3 การสร้างโปรแกรม

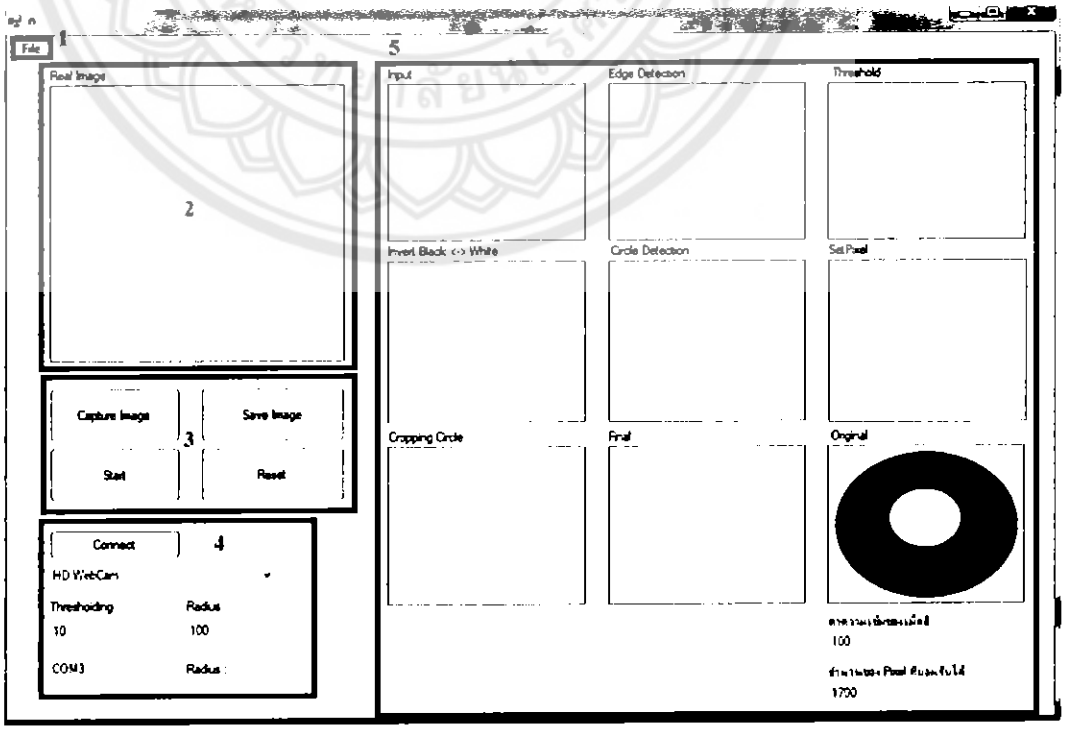
1. ติดตั้งโปรแกรม Microsoft Visual Studio 2010 เพื่อใช้ในการเขียนคำสั่งโปรแกรม
2. ติดตั้งไลบรารี่ AForge.Net เพื่อใช้ในการติดต่อกล้องเว็บแคมและประมวลผลภาพ
3. สร้างโปรแกรมควบคุมมอเตอร์และอินฟราเรดเซ็นเซอร์โดยใช้ Arduino IDE
4. สร้างหน้าต่าง User Interface ในส่วนของการแสดงภาพที่ได้จากกล้องเว็บแคมและภาพที่ได้จากการประมวลผล

3.4 การออกแบบหน้าต่างการใช้งานโปรแกรม

โปรแกรมตรวจสอบรอยขีดข่วนบนฮาร์ดดิสก์โดยใช้กล้องเว็บแคม เขียนด้วยโปรแกรม Microsoft Visual Studio 2010 เนื่องจากเป็นโปรแกรมที่เข้าใจง่าย และมีอ็อปเจ็คช่วยสร้างหน้าต่างของโปรแกรม เพื่อให้ผู้ใช้งานสามารถใช้งานได้ง่าย

สำหรับขั้นตอนการเขียนโปรแกรม หลังจากได้ทำความเข้าใจเกี่ยวกับความต้องการของระบบแล้วขั้นตอนต่อไปก็คือการออกแบบโปรแกรม โดยจะมีหน้าต่างและฟังก์ชั่นการใช้งานดังนี้

3.4.1 หน้าต่างการใช้งานโปรแกรมหลัก



รูปที่ 3.12 หน้าต่างโปรแกรมหลัก

จากรูปที่ 3.12 ในหน้าต่างโปรแกรมหลักนี้ประกอบไปด้วย

1. ส่วนของเมนูจะเป็นการเลือกภาพนิ่งจากโฟลเดอร์จัดเก็บรูป

2. หน้าต่างแสดงภาพที่ถ่ายจากกล้องเว็บแคม

3. ส่วนของปุ่มที่ใช้ควบคุมระบบประมวลผลภาพ

- Capture Image คือ ถ่ายภาพนิ่ง

- Start คือ เริ่มกระบวนการประมวลผลภาพ

- Save Image คือ บันทึกภาพในช่อง Cropping Circle และ Final

- Reset คือ เริ่มกระบวนการประมวลผลภาพใหม่ทั้งหมด

4. ส่วนของการติดต่อกล้องเว็บแคมกับการปรับค่า Threshold และค่า Radius มีหน่วนเป็นพิกเซล

5. ส่วนของหน้าต่างการประมวลผลภาพ

- Input คือ ภาพที่ได้จากการถ่ายภาพนิ่งที่เกิดจากการกดปุ่ม Capture Image

- Edge Detection คือ การหาเส้นรอบรูปที่เกิดจากการเปลี่ยนของสีจากอีกสีหนึ่งไปอีกสีหนึ่งจากภาพของ Input ที่ถูกนำไปทำเป็นภาพระดับสีเทา

- Threshold คือ การปรับความเข้มของเม็ดสี โดยมีค่าของระดับสีเทาเป็นเกณฑ์ในการแบ่งว่า ถ้าหากเม็ดสีนั้นมีค่าน้อยกว่าที่กำหนดจะให้เป็นสีดำหรือศูนย์แต่ถ้ามีค่ามากกว่าที่กำหนดจะให้เป็นสีขาวหรือหนึ่ง

- Invert Black <-> White คือ การนำภาพมาทำการสลับสีของเม็ดสีจากสีขาวเป็นสีดำและสีดำเป็นสีขาว

- Circle Detection คือ การตรวจหาวงกลมภายในภาพ โดยให้ตรวจหาวงกลมที่มีความยาวรัศมี 100 พิกเซล ขึ้นไป

- Set Pixel คือ การกำหนดเม็ดสีตามที่ต้องการ ซึ่งต้องการให้เม็ดสีนอกวงกลมนั้นมีสีเป็นสีขาวทั้งหมด พร้อมกับการทำให้แกนจานแม่เหล็กนั้นมีสีขาวด้วย

- Cropping Circle คือ การตัดภาพให้มีขนานตามที่ต้องการ ในที่นี้นั้นต้องการตัดภาพตามรัศมีของวงกลมที่ตรวจพบ และให้ภาพเหลือขนาด 400x400 พิกเซล

- Final คือ ภาพที่ผ่านกระบวนการเข้าคู่รูปแบบ (Templete Matching) ระหว่างภาพ Original และภาพ Cropping Circle

- Original คือ ภาพต้นฉบับที่ใช้ในกระบวนการ Templete Matching

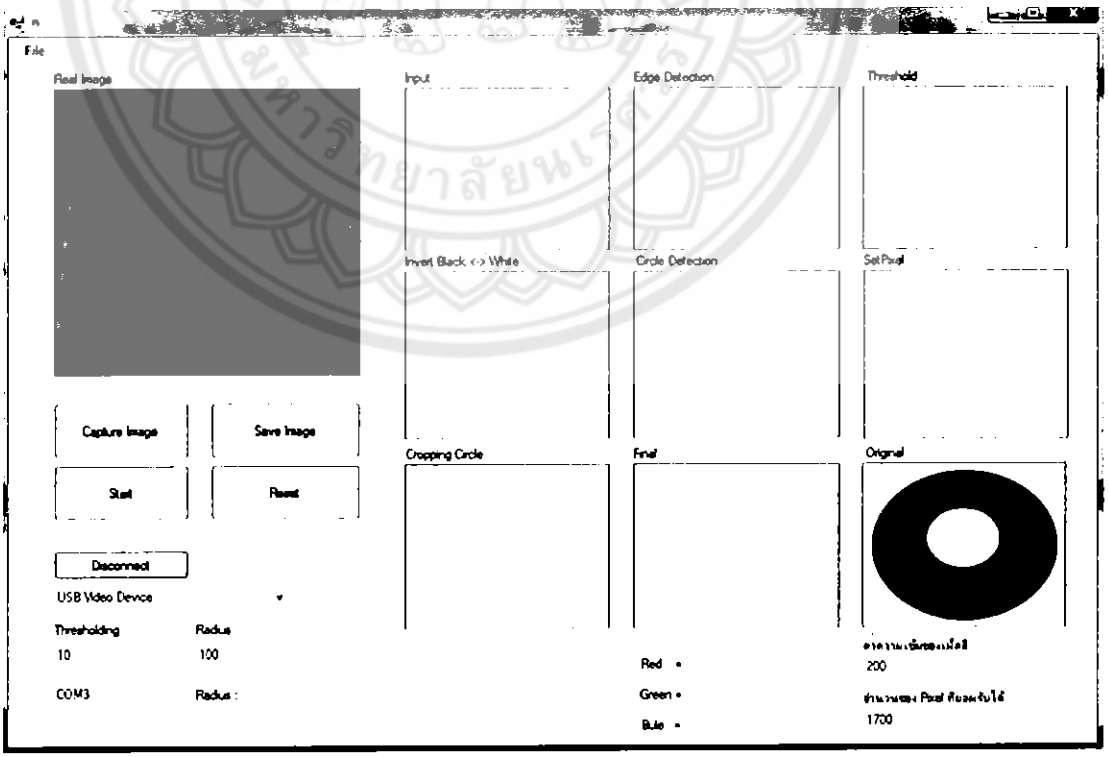
บทที่ 4
ผลการทดลอง

ผลการทดลองนี้เริ่มต้นจากการทดสอบการทำงานของอุปกรณ์ต่างๆของระบบ ทดสอบฟังก์ชั่นการทำงานของโปรแกรมที่เขียนและทดสอบการทำการจริงในสภาวะที่ต่างกันเพื่อจะวิเคราะห์ว่าโปรแกรมบันทึกภาพและประมวลผลภาพเพื่อตรวจสอบตำหนิบนจานแม่เหล็กของฮาร์ดดิสก์ มีขอบเขตความสามารถในการทำงานอย่างไรเพื่อที่จะได้ประสิทธิภาพในการใช้งานสูงสุด

4.1 ขั้นตอนการทดสอบการเชื่อมต่อและทำงานของระบบ

4.1.1 ทดสอบการเชื่อมต่อกับกล้องเว็บแคม

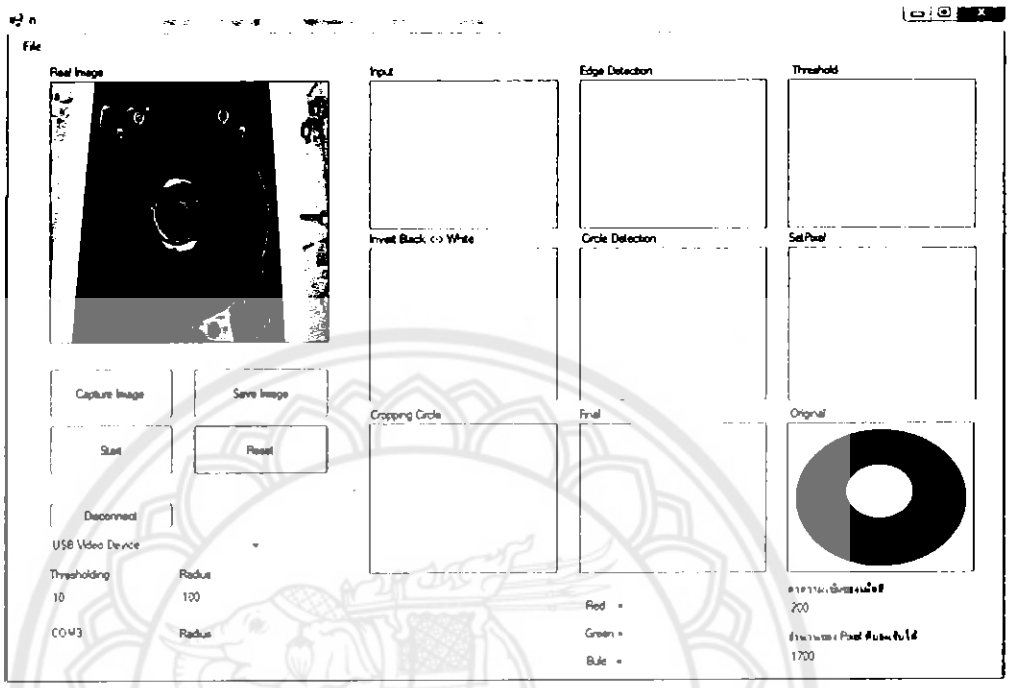
การเชื่อมต่อกล้องเป็นรูปแบบการทำงานของโปรแกรม Visual Studio 2010 โดยใช้ไลบรารี่ AForge.Net ในการติดต่อกล้องกับโปรแกรม ทำให้เราสามารถรับภาพจากกล้องมาแสดงผลในหน้าจอคอมพิวเตอร์ได้ตามรูปที่ 4.1



รูปที่ 4.1 แสดงผลการติดต่อกับกล้องเว็บแคม

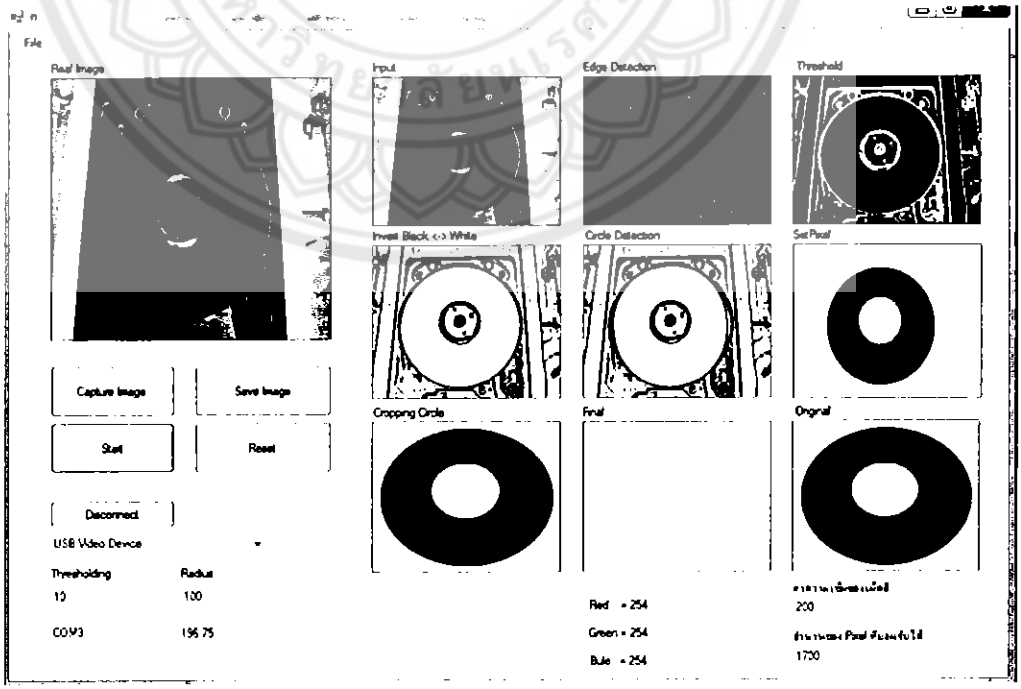
4.1.2 ทดสอบฟังก์ชั่นการทำงานของโปรแกรม

1. ภาพที่บันทึกได้จากกล้องเว็บแคมจะถูกแสดงในช่อง Real Image ตามรูปที่ 4.2



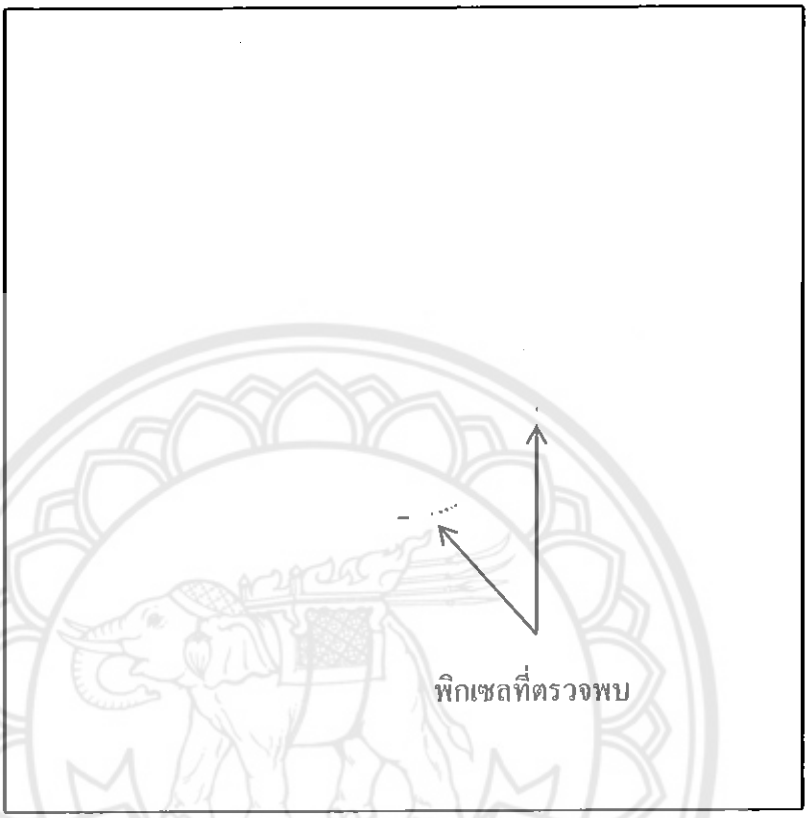
รูปที่ 4.2 แสดงหน้าต่างในส่วนของการบันทึกภาพ

2. หลังจากนั้นรูปจะถูกประมวลผลภาพและแสดงออกในกรอบรูป 8 กรอบทางขวา ตามรูปที่ 4.4



รูปที่ 4.3 แสดงหน้าต่างในส่วนของการบันทึกภาพและประมวลผลภาพ

3. ในส่วนของการจัดเก็บภาพ ภาพที่ถูกจัดเก็บจะเป็นภาพในช่อง Final ซึ่งเป็นภาพสุดท้ายของขั้นตอนการประมวลผลภาพ ส่วนที่จัดเก็บสามารถจัดเก็บได้ตามที่เราต้องการ



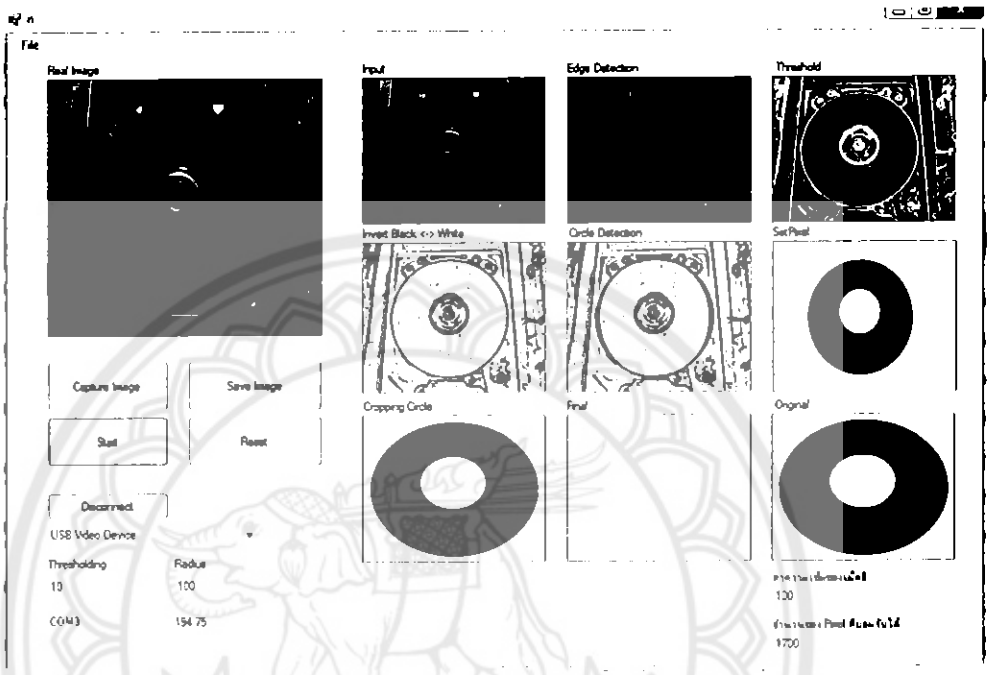
รูปที่ 4.4 แสดงรูปสุดท้ายของการประมวลผลภาพ

4.2 ทดสอบขอบเขตของการใช้งาน

4.2.1 ทดสอบการทำงานของโปรแกรมโดยมีการควบคุมแสงสว่าง

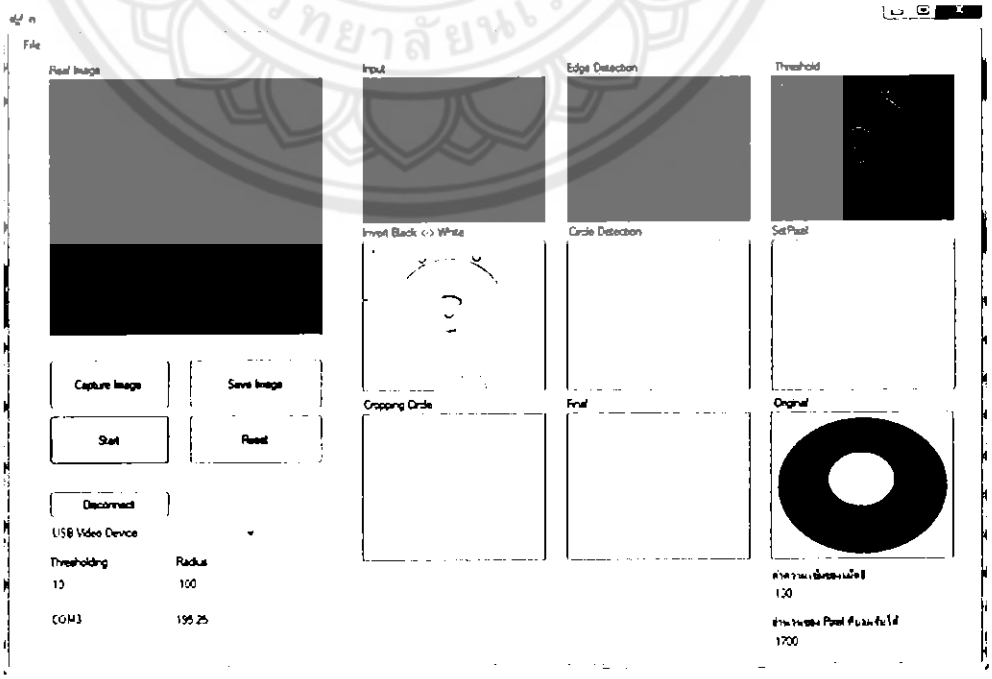
การทดสอบนี้จะเป็นการทดสอบการทำงานของโปรแกรมในสภาวะที่แสงสว่างต่างกัน

4.2.1.1 กรณีทดลองในพื้นที่มีแสงสว่าง



รูปที่ 4.6 แสดงรูปการทดลองการทำงานของโปรแกรมเมื่อมีแสงสว่าง**เพียงพอ**

4.2.1.1 กรณีทดลองในพื้นที่ไม่มีแสงสว่าง



รูปที่ 4.7 แสดงรูปการทดลองการทำงานของโปรแกรมเมื่อไม่มีแสงสว่าง

จากกรณีที่ไม่มีแสงสว่าง พบว่าโปรแกรมไม่สามารถตรวจเจอวงกลมได้ ทางผู้ทดลองจึงได้ปรับค่า Threshold ด้วยค่าต่างๆ เพื่อเปลี่ยนความเข้มสีของภาพ จึงสรุปเป็นตารางได้ดังนี้

ตารางที่ 4.1 ตารางผลการทดลองการปรับค่าระดับสีเทา (Threshold) ที่สภาวะแสงต่างกัน

ค่า Threshold ในการแบ่งสีขาว-ดำ	จำนวนครั้งที่ตรวจพบวงกลมจากการทดลองทั้งหมด 15 ครั้ง (มีแสง)	จำนวนครั้งที่ตรวจพบวงกลมจากการทดลองทั้งหมด 15 ครั้ง (ไม่มีแสง)
2	0 ครั้ง	0 ครั้ง
4	11 ครั้ง	0 ครั้ง
6	12 ครั้ง	0 ครั้ง
8	13 ครั้ง	0 ครั้ง
10	15 ครั้ง	0 ครั้ง
12	14 ครั้ง	0 ครั้ง
14	10 ครั้ง	0 ครั้ง
16	0 ครั้ง	0 ครั้ง

จากผลการทดลองในตารางที่ 4.1 พบว่า การปรับค่า Threshold มีผลต่อการตรวจเจอวงกลม โดยที่ค่า Threshold=10 ให้ผลออกมาที่เหมาะสมที่สุดจากการทดลองทั้งหมด 15 ครั้ง ซึ่งค่า Threshold ดังกล่าวจะถูกใช้เป็นค่าหลักเพื่อทำการทดลองถัดไป การทดลองนี้จึงสรุปได้ว่า แสงมีผลต่อการทดลองของโปรแกรมเป็นอย่างมาก

4.2.2 การทดลองหาจำนวนพิกเซลทั้งหมดที่ยอมรับได้

การทดลองนี้จะเป็นการจำนวนพิกเซลทั้งหมดที่ได้จากวิธีการ Template Matching เพื่อนำไปใช้เป็นตัวชี้วัดว่าจานแม่เหล็กที่ตรวจสอบมีคุณสมบัติแบบไหน โดยจะนำจานแม่เหล็กที่มีสภาพสะอาดมาทำการทดลอง

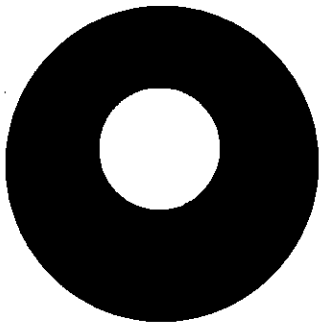
ตารางที่ 4.2 ตารางการทดลองหาจำนวนพิกเซลที่ผ่านจากกระบวนการ Template Matching

ครั้งที่ในการทดสอบ (ครั้ง)	ผลลัพธ์ที่ผ่านกระบวนการ Template Matching (ภาพอินพุต – ภาพต้นฉบับ)
1	7 พิกเซล
2	8 พิกเซล
3	6 พิกเซล
4	0 พิกเซล
5	2 พิกเซล
6	1 พิกเซล
7	5 พิกเซล
8	10 พิกเซล
9	9 พิกเซล
10	5 พิกเซล
มากกว่า 10	น้อยกว่าหรือเท่ากับ 10 พิกเซล

จากตารางที่ 4.2 พบว่าจำนวนพิกเซลทั้งหมดที่ตรวจพบมีค่ามากสุดอยู่ที่ 10 พิกเซล ซึ่งค่าๆ นี้จะนำไปกำหนดเป็นตัวชี้วัดคุณสมบัติของจานแม่เหล็กว่า "ดี" ถ้ามีจำนวนพิกเซลไม่เกิน 10 พิกเซล ถ้าเกิน 10 พิกเซลจะชี้วัดคุณสมบัติของจานแม่เหล็กว่า "ผิดพลาด" และจะนำไปใช้ในการทดลองหัวข้อถัดไป

4.2.3 การทดลองความแม่นยำของการทำงานโปรแกรม

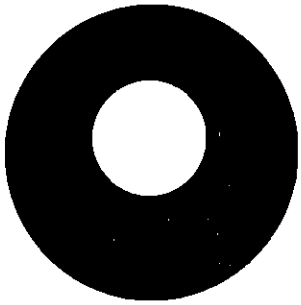
การทดลองนี้จะเป็นการทดลองความแม่นยำของโปรแกรมในการตรวจหาพิกเซลที่ได้จากการลบภาพด้วยวิธี Template Matching โดยมีรูปต้นฉบับที่ใช้เทียบดังรูปที่ 4.8

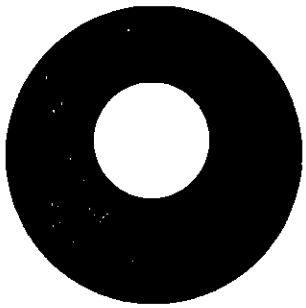


รูปที่ 4.8 รูปต้นฉบับที่ใช้ในการทดลองที่ 4.2.3

ตารางที่ 4.3 จะเป็นการทดลองหารูปภาพที่ผ่านการลบกัน เพื่อนำมาเป็นข้อมูลในการหาความแม่นยำของโปรแกรม ซึ่งจะอยู่ในส่วนของตารางถัดไป

ตารางที่ 4.3 ตารางผลการทดลองของรูปที่ผ่านการลบกัน

การทดลองที่	รูปที่ทำการทดลอง	รูปที่ผ่านการลบกัน
1		
2		

3		
4		

จากตารางที่ 4.3 พบว่าได้มีการเพิ่มตำหนิเข้าไปบนจานแม่เหล็กของฮาร์ดดิสก์ และได้เปลี่ยนตำแหน่งของตำแหน่งด้วยการหมุนจานแม่เหล็ก เพื่อทดสอบความแม่นยำของการนับพิกเซล ซึ่งได้ผลดังตารางที่ 4.4

ตารางที่ 4.4 ตารางบันทึกค่าพิกเซลที่นับได้แต่ละการทดลอง

การทดลองที่	ค่าพิกเซลที่นับได้ (Pixel)	ความคลาดเคลื่อนของพิกเซลเทียบกับการทดลองที่ 1 (Pixel)
1	172	0
2	264	มากกว่า 92 (+34.85%)
3	255	มากกว่า 83 (+32.55%)
4	173	มากกว่า 1 (+0.58%)

จากตารางที่ 4.4 จะสามารถสรุปได้ว่า การหมุนตำแหน่งของจานแม่เหล็กโดยที่รอยตำหนิยังเหมือนเดิม ทำให้ค่าพิกเซลที่นับได้มีความคลาดเคลื่อน ซึ่งอาจจะเกิดจากแสงที่กระทบกับตำหนิบนจานแม่เหล็กซึ่งทำให้เกิดเงาขึ้นมา จึงทำให้การตีความพิกเซลมีความคลาดเคลื่อนบ้าง

หมายเหตุ : การทดลองนี้ได้ทำการนับค่าความเข้มเม็ดสีที่มีค่ามากกว่า 100 ตามภาพระดับสีเทา (สีดำ=0 , สีขาว=255) และจำนวนของ Pixel สีดำที่ยอมรับได้ที่ 10 พิกเซล (น้อยกว่า=good , มากกว่า=defect)

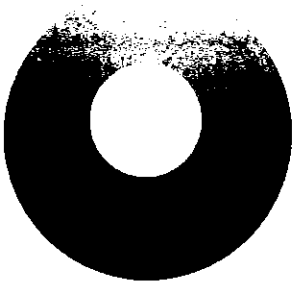
4.2.4 การทดลองหาข้อจำกัดของขนาดตำหนิที่ตรวจพบ

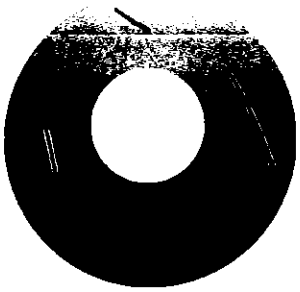
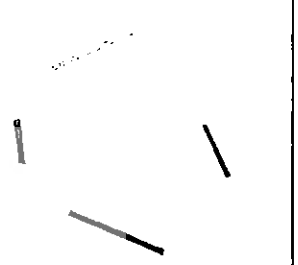
การทดลองนี้จะเป็นการทดลองหาข้อจำกัดของขนาดตำหนิ โดยมีรูปต้นฉบับที่ใช้เทียบดังรูปที่ 4.9

รูปที่ 4.9 รูปต้นฉบับที่ใช้ในการทดลองที่ 4.2.4

ตารางที่ 4.5 จะเป็นการทดลองหารูปภาพที่ผ่านการลบกันด้วยวิธีการ Template Matching ซึ่งการทดลองแต่ละครั้งจะกำหนดขนาดของตำหนิที่แตกแตกต่างกัน

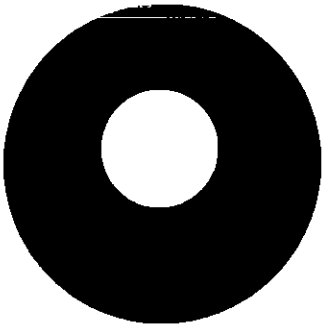
ตารางที่ 4.5 ตารางผลการทดลองของรูปที่ผ่านการลบกัน โดยที่กำหนดรอยตำหนิขนาดต่างๆ กัน

การทดลองที่	ขนาดของตำหนิ	รูปที่ทำการทดลอง	รูปที่ผ่านการลบกัน
1	น้อยกว่า 1 มม.		

2	1 มม.		
3	2 มม.		
4	3 มม.		

จากตารางที่ 4.5 จะสามารถสรุปได้ว่า ขนาดของตำหนิที่สามารถตรวจพบได้ ควรมีขนาดตั้งแต่ 1 มม. ขึ้นไป ถ้าหากตำหนิมีขนาดน้อยกว่า 1 มม. โปรแกรมจะไม่สามารถตรวจเจอรอยได้เนื่องจากข้อจำกัดความละเอียดของกล้อง

ในกรณีที่เป็นตำหนิประเภทรอยขีดข่วน โดยมีรูปต้นฉบับที่ใช้เทียบดังรูปที่ 4.10 ซึ่งจะได้ผลการทดลองดังตารางที่ 4.6



รูปที่ 4.10 รูปต้นฉบับที่ใช้ในการทดลองที่ 4.2.4 (2)

ตารางที่ 4.6 ตารางผลการทดลองของรูปที่ผ่านการลบในกรณีที่ตำหนิเป็นรอยขีดข่วน

การทดลองที่	รูปที่ทำการทดลอง	รูปที่ผ่านการลบกัน
1		
2		
3		

จากตารางที่ 4.6 จะสามารถสรุปได้ว่า โปรแกรมไม่สามารถตรวจเจอรอยขีดข่วนได้ เนื่องจากสีของรอยขีดข่วนใกล้เคียงกับจานแม่เหล็ก ดังนั้นตำหนิใช้ตรวจสอบควรมีสีที่มีเข้มกว่าจานแม่เหล็กจึงจะตรวจพบ

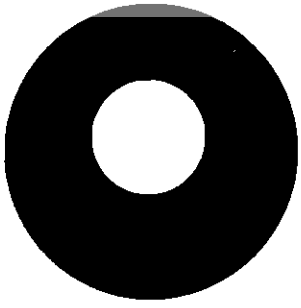
4.2.5 การทดลองการทำงานของโปรแกรมทั้งหมด

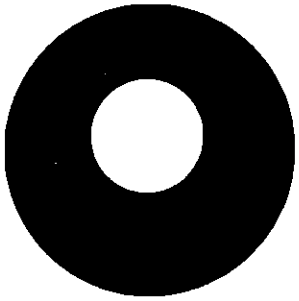
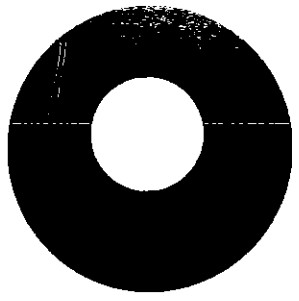
การทดลองนี้จะเป็นการทดสอบการทำงานของโปรแกรมของโปรแกรมทั้งหมด ตั้งแต่การนำภาพมาลบกันด้วยวิธีการ Template Matching จนถึงการระบุว่าการทดลองแต่ละครั้งมีการตรวจพบตำหนิกี่พิกเซล และสามารถตีความได้ว่า มีตำหนิหรือไม่ โดยมีรูปต้นฉบับที่ใช้เทียบดังรูปที่ 4.10

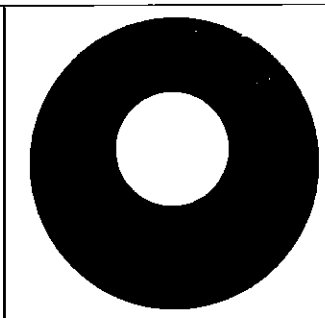
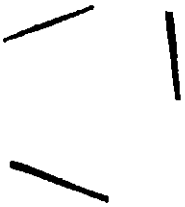
รูปที่ 4.11 รูปต้นฉบับที่ใช้ในการทดลองที่ 4.2.5

ตารางที่ 4.7 จะเป็นการทดลองหารูปภาพที่ผ่านการลบกันด้วยวิธีการ Template Matching เพื่อนำมาเป็นข้อมูลในการตีความ ซึ่งจะอยู่ในส่วนของตารางถัดไป

ตารางที่ 4.7 ตารางผลการทดลองของรูปที่ผ่านวิธีการ Template Matching

การทดลองที่	รูปที่ทำการทดลอง	รูปที่ผ่านการลบกัน
1		

2		
3		
4		
5		
6		

7		
8		

จากตารางที่ 4.7 พบว่าได้มีการเพิ่มตำหนิเข้าไปบนจานแม่เหล็กของฮาร์ดดิสก์ และได้เพิ่มปริมาณของตำหนิ เพื่อทดสอบเงื่อนไขการตีความการนับพิกเซลซึ่งได้ผลดังตารางที่ 4.8

ตารางที่ 4.8 ตารางบันทึกค่าพิกเซลที่นับได้แต่ละการทดลอง

การทดลองที่	ค่าพิกเซล ที่นับได้ (Pixel)	การตีความผลการทดลอง ที่ได้ (Good/Defect)
1	8	Good
2	151	Defect
3	293	Defect
4	527	Defect
5	21	Defect
6	1359	Defect
7	2800	Defect
8	4932	Defect

จากตารางที่ 4.8 จะสามารถสรุปได้ว่า จำนวนพิกเซลรวมที่ยอมรับได้คือ 10 พิกเซล ซึ่งถ้ามีค่าน้อยกว่า 10 จะทำให้โปรแกรมตีความการทดลองนี้ว่า "Good" หรือไม่พบตำหนิ และถ้ามีค่ามากกว่า 10 พิกเซล จะทำให้ถูกตีความว่า "Defect" หรือพบตำหนิ

หมายเหตุ : การทดลองนี้ได้ทำการนับค่าความเข้มเม็ดสีที่มีค่ามากกว่า 100 ตามภาพระดับสีเทา (สีดำ=0 , สีขาว=255) และจำนวนของ Pixel สีดำที่ยอมรับได้ที่ 10 พิกเซล (น้อยกว่า=good , มากกว่า=defect)

บทที่ 5
สรุปผลและข้อเสนอแนะ

โครงงานนี้ทำการศึกษาและพัฒนาการประมวลผลภาพกับกล้องเว็บแคมในการตรวจรอยบนจานแม่เหล็กของฮาร์ดดิสก์ แล้วนำภาพที่ได้จากวิธีการลบกัน (Template Matching) ซึ่งจะทำให้เกิดพิกเซลที่เหลือจากการลบกัน พิกเซลเหล่านี้จะถูกนำมานับและตีความว่าสภาพจานแม่เหล็กของฮาร์ดดิสก์นั้นมีคุณสมบัติที่ "ดี (Good)" หรือ " ผิดพลาด (Defect)" โดยผ่านเงื่อนไขที่ได้กำหนดไว้ ทั้งนี้ได้ทำการทดลองปรับแสงสว่าง ทดสอบความแม่นยำในการนับพิกเซล และทดสอบการตีความพิกเซลที่นับได้

เมื่อทำการทดลองแล้วทำให้ทราบถึงปัญหาบางประการ ทั้งในระหว่างการพัฒนาและการทดลองใช้งานโดยสามารถสรุปได้ดังต่อไปนี้

5.1 สรุปผลการทดลอง

จากผลการทดลอง การตรวจรอยบนจานแม่เหล็กของฮาร์ดดิสก์พบว่าจำนวนพิกเซลที่ยอมรับได้ว่าคุณสมบัติบนจานแม่เหล็กที่มีสภาพดีได้ไม่เกิน 10 พิกเซลจากทั้งหมด 160,000 พิกเซล แต่ถ้ามากกว่าจำนวนนี้จะตีความคุณสมบัติว่า "ผิดพลาด"

ในการทดลอง แสงสว่างมีผลต่อการตรวจหาวงกลม เพื่อนำวงกลมนั้นมาประมวลผลภาพในขั้นต่อไป แต่ถ้าแสงสว่างไม่เพียงพอก็ทำให้ไม่สามารถตรวจหาวงกลมได้ นอกจากนี้ยังมีปัจจัยอื่นๆ เช่น ค่า Threshold , ค่าความเข้มเม็ดสี เป็นต้น ทำให้ผลจากการทดลองมีความผิดพลาดพอสมควร

ปัญหาข้างต้นนี้สามารถแก้ไขได้โดยการสร้างกล่องขึ้นมาและติดตั้ง LED ภายในกล่องเพื่อควบคุมแสงสว่างให้คงที่ ส่วนค่า Threshold และความเข้มของเม็ดสี ได้ทำการเลือกค่าที่เหมาะสมและคงที่ไว้ วิธีการแก้ปัญหาดังกล่าวจะช่วยลดปัจจัยที่มีผลต่อระบบประมวลผลภาพ

5.2 ข้อเสนอแนะ

5.2.1 จากการทดลอง ได้ใช้กล้องเว็บแคมความละเอียดต่ำ จึงมีผลต่อการประมวลผลภาพพอสมควร สิ่งที่ต้องพัฒนาเพิ่มเติมคือ ใช้กล้องความละเอียดสูงในการทดลอง

5.2.2 ในการทดลองจำเป็นต้องใช้กับสถานที่ที่เหมาะสม คือ มีแสงสว่างมากเพียงพอ

5.2.3 มุมของกล้องและเงาสะท้อนมีผลต่อการประมวลผลภาพ จึงต้องมีการปรับมุมของกล้องให้เหมาะสมกับแสงสว่างและเงา

5.2.4 ในส่วนของโปรแกรมที่ออกแบบมา ได้ใช้โปรแกรม Microsoft Visual Studio 2010 และใช้ภาษา C# ในการเขียนโปรแกรม ซึ่งโปรแกรมนี้เหมาะสมสำหรับออกแบบแอพลิเคชั่นต่างๆ แต่เนื่องจากผู้จัดทำมีความรู้และความเข้าใจเกี่ยวกับโปรแกรมไม่มาก จึงทำให้การสร้างโปรแกรมมีปัญหา ดังนั้นจึงใช้เวลาในการทำความเข้าใจและแก้ปัญหาพอสมควร

5.2.5 เนื่องจากโครงงานนี้มีการใช้ Image Processing เข้ามาด้วย เราจึงใช้ไลบรารี่ของ Aforge.net เข้ามาช่วยในการทำงาน ดังนั้นผู้จัดทำจึงต้องศึกษาหาข้อมูลเพิ่มเติม

5.2.6 ควรใช้กล้องในการถ่ายรูปหลายมุม เพื่อจะได้รายละเอียดของภาพที่ดีขึ้น และนำภาแต่ละมุมมาประมวลผลภาพเพื่อความแม่นยำของการตรวจสอบ

เอกสารอ้างอิง

[1] **Arduino UnoBoard "Overview arduinouno R3"**, สืบค้นเมื่อ 12 ธันวาคม 2558 จาก http://arduino.cc/en/Main/arduinoBoardUno

[2] **เทคโนโลยีประมวลผลภาพ (Image Processing)**, สืบค้นเมื่อ 12 ธันวาคม 2558 จาก http://jaratcyberu.blogspot.com/2009/10/image-processing.html

[3] **กิตติศักดิ์ ผาภู, ปิยะวัฒน์ ภูมิศาสตร์.โครงการกล้องหมุนตามการเคลื่อนที่,** คณะวิศวกรรมศาสตร์สาขา วิศวกรรมไฟฟ้า, มหาวิทยาลัยขอนแก่น, 2550. สืบค้นเมื่อ 12 ธันวาคม 2558, จาก https://app.enit.kku.ac.th/mis/administrator/doc.../20100308170015.pdf

[4] **รู้จักกับ Microsoft Visual Studio**, สืบค้นเมื่อ 12 ธันวาคม 2558 จาก http://visualgramming.blogspot.com/

[5] **C# คืออะไร**, สืบค้นเมื่อ 12 ธันวาคม 2558 8 จาก http://www.mindphp.com/คู่มือ/73-คืออะไร/2184-c-ชาร์ป-คืออะไร.html

[6] **ภาพระดับสีเทา**, สืบค้นเมื่อ 12 ธันวาคม 2558 จาก http://www.research-system.siam.edu/images/thesistec/AUTOMATIC_VEHICLE_MODEL_DETECTION_SYSTEM/10_บทท_3.pdf

[7] วิโรจน์ องอาจ, **การประมวลผลภาพวัดขนาดเส้นผ่านศูนย์กลางเส้นใยโพลิเมอร์จากการปั่นด้วยไฟฟ้าสถิต**, สืบค้นเมื่อ 12 ธันวาคม 2558 จาก http://www.gits.kmutnb.ac.th/ethesis/data/4610183065.pdf

[8] **การแปลงภาพสีให้เป็นภาพขาว-ดำ (Thresholding)**, สืบค้นเมื่อ 5 กุมภาพันธ์ 2559 จาก http://www.eng.buu.ac.th/~chboonyu/Mit5/TsisMIT4/paper.doc

[9] **โปรแกรมการอ่านค่าตัวเลขการใช้น้ำหรือไฟฟ้าจากมิเตอร์**, สืบค้นเมื่อ 5 กุมภาพันธ์ 2559 จาก https: / / app. enit. kku. ac. th/ mis/ administrator/ doc_upload/ 20120312103907. pdf

ภาคผนวก

ภาคผนวก ก.

โปรแกรมในส่วนของ Microsoft Visual Studio 2010

```
//ส่วนของ NameSpace ที่เรียกไปใช้ในโปรแกรมนี้
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Diagnostics;
using System.Drawing.Imaging;
using System.Threading;
using System.Windows.Forms;

// NameSpeace ที่เรียกใช้จาก AForge
using AForge.Video.DirectShow;
using AForge.Video;
using AForge.Imaging.Filters;
using AForge.Imaging;
using AForge;
using AForge.Math.Geometry;

namespace FinalProject
{
public partial class Form1 : Form
 {
  //เป็นการประกาศตัวแปรชนิด Global
  //ตัวแปรเหล่านี้เป็นการประกาศเพื่อเรียกใช้งานทั้งโปรแกรม
public int CenterX;
public int CenterY;
public int r;
public int reprocess =0;
public Form1()
    {
       InitializeComponent();
    }

  //เป็นการเรียก class จาก AForge เพื่อนำมาใช้กับตัวแปรที่กำหนด
private FilterInfoCollection CaptureDevice;
private VideoCaptureDevice FinalFrame;

  //เป็นเหตุการณ์ที่จะเกิดขึ้นเมื่อทำการเปิดโปรแกรมขึ้นมา
private void Form1_Load(object sender, EventArgs e)
 {
```

```
//การติดต่อกับกล้องและอุปกรณ์ที่เชื่อมต่อ
 CaptureDevice =newFilterInfoCollection(FilterCategory.VideoInputDevice);

foreach(FilterInfo Device in CaptureDevice)
  {
          //ให้ขึ้นรายชื่อของกล้องที่ติดต่อกับคอมพิวเตอร์
    comboBox1.Items.Add(Device.Name);
          }
          comboBox1.SelectedIndex =0;
          FinalFrame =newVideoCaptureDevice();
   //การเรียกข้อมูลจาก Arduino
          serialPort1.PortName =textBox3.Text.ToString();
        serialPort1.Open();
    }

//เมื่อเรียกใช้งานกล้องจะให้ทำงานที่ pictureBox1
void FinalFrame_NewFrame(object sender, NewFrameEventArgs eventArgs)
 {
   pictureBox1.Image =(Bitmap)eventArgs.Frame.Clone();
  }

//คำสั่งเมื่อทำการกดปุ่ม Save Image
private void button3_Click(object sender, EventArgs e)
 {
     SaveRealImage();
     SaveInputImage();
     SaveEdgeImage();
     SaveThresholdImage();
     SaveInvertImage();
     SaveCircleImage();
     SaveSetPixelImage();
  SaveCropImage();
     SaveFinalImage();
  }

//คำสั่ง SaveRealImage
private void SaveRealImage()
  {
  if(pictureBox1.Image != null)
  {
  Bitmap RealImage =new Bitmap(pictureBox1.Image);
  string NameSave ="Real"+DateTime.Now.Day.ToString()+"-"+DateTime.Now.Month.ToString()+
"-"+DateTime.Now.Year.ToString()
+""+DateTime.Now.Hour.ToString()+"-"+DateTime.Now.Minute.ToString()+"-"+
DateTime.Now.Second.ToString()+".jpg";
  string SaveStr =@"C:\Users\jomanjo\Pictures\ผลการทดลอง5\"+NameSave;
  Debug.Print(SaveStr);
  RealImage.Save(SaveStr, ImageFormat.Jpeg);
     RealImage.Dispose();
     RealImage =null;
  }
   }
```

//คำสั่ง SaveInputImage

```
private void SaveInputImage()
 {
 if(pictureBox2.Image != null)
   {
 Bitmap InputImage = new Bitmap(pictureBox2.Image);
 string NameSave = "Input" + DateTime.Now.Day.ToString() + "-" + DateTime.Now.Month.ToString()
+ "-" + DateTime.Now.Year.ToString() + " " + DateTime.Now.Hour.ToString() + "-" +
DateTime.Now.Minute.ToString() + "-" + DateTime.Now.Second.ToString() + ".jpg";
 string SaveStr = @"C:\Users\jomanjo\Pictures\ผลการทดลอง5\" + NameSave;
 Debug.Print(SaveStr);
    InputImage.Save(SaveStr, ImageFormat.Jpeg);
    InputImage.Dispose();
 InputImage = null;
 }
}
```

//คำสั่ง SaveEdgeImage

```
private void SaveEdgeImage()
  {
 if(pictureBox3.Image != null)
 {
 Bitmap EdgeImage = new Bitmap(pictureBox3.Image);
 string NameSave = "EdgeImage" + DateTime.Now.Day.ToString() + "-" +
DateTime.Now.Month.ToString() + "-" + DateTime.Now.Year.ToString() + " " +
DateTime.Now.Hour.ToString() + "-" + DateTime.Now.Minute.ToString() + "-" +
DateTime.Now.Second.ToString() + ".jpg";
 string SaveStr = @"C:\Users\jomanjo\Pictures\ผลการทดลอง5\" + NameSave;
 Debug.Print(SaveStr);
    EdgeImage.Save(SaveStr, ImageFormat.Jpeg);
    EdgeImage.Dispose();
    EdgeImage = null;
 }
 }
```

//คำสั่ง SaveThresholdImage

```
private void SaveThresholdImage()
  {
 if(pictureBox4.Image != null)
    {
 Bitmap ThresholdImage = new Bitmap(pictureBox4.Image);
 string NameSave = "ThresholdImage" + DateTime.Now.Day.ToString() + "-" +
DateTime.Now.Month.ToString() + "-" + DateTime.Now.Year.ToString() + " " +
DateTime.Now.Hour.ToString() + "-" + DateTime.Now.Minute.ToString() + "-" +
DateTime.Now.Second.ToString() + ".jpg";
 string SaveStr = @"C:\Users\jomanjo\Pictures\ผลการทดลอง5\" + NameSave;
 Debug.Print(SaveStr);
 ThresholdImage.Save(SaveStr, ImageFormat.Jpeg);
     ThresholdImage.Dispose();
    ThresholdImage = null;
     }
  }
```

//คำสั่ง SaveInvertImage

```
private void SaveInvertImage()
  {
  if(pictureBox5.Image != null)
     {
  Bitmap InvertImage = new Bitmap(pictureBox5.Image);
  string NameSave = "InvertImage" + DateTime.Now.Day.ToString() + "-" +
DateTime.Now.Month.ToString() + "-" + DateTime.Now.Year.ToString() + " " +
DateTime.Now.Hour.ToString() + "-" + DateTime.Now.Minute.ToString() + "-" +
DateTime.Now.Second.ToString() + ".jpg";
  string SaveStr = @"C:\Users\jomanjo\Pictures\ผลการทดลอง5\" + NameSave;
  Debug.Print(SaveStr);
      InvertImage.Save(SaveStr, ImageFormat.Jpeg);
       InvertImage.Dispose();
       InvertImage = null;
      }
    }
```

//คำสั่ง SaveCircleImage

```
private void SaveCircleImage()
  {
  if(pictureBox6.Image != null)
     {
  Bitmap CircleImage = new Bitmap(pictureBox6.Image);
  string NameSave = "CircleImage" + DateTime.Now.Day.ToString() + "-" +
DateTime.Now.Month.ToString() + "-" + DateTime.Now.Year.ToString() + " " +
DateTime.Now.Hour.ToString() + "-" + DateTime.Now.Minute.ToString() + "-" +
DateTime.Now.Second.ToString() + ".jpg";
  string SaveStr = @"C:\Users\jomanjo\Pictures\ผลการทดลอง5\" + NameSave;
  Debug.Print(SaveStr);
      CircleImage.Save(SaveStr, ImageFormat.Jpeg);
   CircleImage.Dispose();
  CircleImage = null;
   }
 }
```

//คำสั่ง SaveSetPixelImage

```
private void SaveSetPixelImage()
  {
  if(pictureBox7.Image != null)
     {
  Bitmap SetPixelImage = new Bitmap(pictureBox7.Image);
  string NameSave = "SetPixelImage" + DateTime.Now.Day.ToString() + "-" +
DateTime.Now.Month.ToString() + "-" + DateTime.Now.Year.ToString() + " " +
DateTime.Now.Hour.ToString() + "-" + DateTime.Now.Minute.ToString() + "-" +
DateTime.Now.Second.ToString() + ".jpg";
  string SaveStr = @"C:\Users\jomanjo\Pictures\ผลการทดลอง5\" + NameSave;
  Debug.Print(SaveStr);
     SetPixelImage.Save(SaveStr, ImageFormat.Jpeg);
     SetPixelImage.Dispose();
  SetPixelImage = null;
  }
 }
```

//คำสั่ง SaveCropImage

```
private void SaveCropImage()
  {
  if(pictureBox8.Image != null)
     {
  Bitmap CropImage = new Bitmap(pictureBox8.Image);
  string NameSave = "Crop" + DateTime.Now.Day.ToString() + "-" + DateTime.Now.Month.ToString() +
"-" + DateTime.Now.Year.ToString() + " " + DateTime.Now.Hour.ToString() + "-" +
DateTime.Now.Minute.ToString() + "-" + DateTime.Now.Second.ToString() + ".jpg";
  string SaveStr = @"C:\Users\jomanjo\Pictures\ผลการทดลอง5\" + NameSave;
  Debug.Print(SaveStr);
      CropImage.Save(SaveStr, ImageFormat.Jpeg);
      CropImage.Dispose();
  CropImage = null;
   }
 }
```

//คำสั่ง SaveFinalImage

```
private void SaveFinalImage()
 {
  if(pictureBox9.Image != null)
     {
  Bitmap FinalImage = new Bitmap(pictureBox9.Image);
  string NameSave = "Final" + DateTime.Now.Day.ToString() + "-" + DateTime.Now.Month.ToString()
+ "-" + DateTime.Now.Year.ToString() + " " + DateTime.Now.Hour.ToString() + "-" +
DateTime.Now.Minute.ToString() + "-" + DateTime.Now.Second.ToString() + ".jpg";
  string SaveStr = @"C:\Users\jomanjo\Pictures\ผลการทดลอง5\" + NameSave;
  Debug.Print(SaveStr);
      FinalImage.Save(SaveStr, ImageFormat.Jpeg);
      FinalImage.Dispose();
      FinalImage = null;
     }
   }
```

//คำสั่งเมื่อทำการกดปุ่ม Connect และ Disconnect

```
private void button1_Click(object sender, EventArgs e)
    {
  // เมื่อ pictureBox1 ยังไม่แสดง
  if(FinalFrame == null)
       {
        FinalFrame = new VideoCaptureDevice(CaptureDevice
                  [comboBox1.SelectedIndex].MonikerString);
        FinalFrame.NewFrame += new NewFrameEventHandler(FinalFrame_NewFrame);
        FinalFrame.Start();// เมื่อกล้องเริ่มเปิดใช้งาน
        button1.Text = "Disconnect";
             }
    else
             {
                 FinalFrame.Stop();// เมื่อกล้องเริ่มหยุดใช้งาน
     FinalFrame = null;
                 button1.Text = "Connect";
             }
```

```
        }
```

//ส่วนของการถ่ายภาพแบบอัตโนมัติ

```
public void Capture()
    {
  try
  {
        pictureBox2.Image = pictureBox1.Image;
  }
  catch { }
 }
```

//คำสั่งเมื่อทำการกดปุ่ม Capture Image

```
private void button2_Click(object sender, EventArgs e)
   {
      Capture();
   }
```

//คำสั่งเมื่อทำการกดปุ่ม Capture Image แล้วให้รูปไปแสดงในภาพ Input

```
private void button2_Click_1(object sender, EventArgs e)
  {
     pictureBox2.Image = (Bitmap)pictureBox1.Image.Clone();
 }
```

//คำสั่งในส่วนของการเปิดไฟล์

```
private void openToolStripMenuItem_Click(object sender, EventArgs e)
  {
 if(openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)
    {
 pictureBox1.Image = (Bitmap)System.Drawing.Image.FromFile
                     (openFileDialog1.FileName);
 }
 }
```

//คำสั่งเมื่อทำการกดปุ่ม exit

```
private void exitToolStripMenuItem_Click(object sender, EventArgs e)
  {
 this.Close();
 }
```

//คำสั่งเมื่อทำการกดปุ่ม Reset

```
private void button4_Click(object sender, EventArgs e)
  {
     pictureBox2.Image = null;
  pictureBox3.Image = null;
   pictureBox4.Image = null;
     pictureBox5.Image = null;
     pictureBox6.Image = null;
     pictureBox7.Image = null;
     pictureBox8.Image = null;
     pictureBox9.Image = null;
```

```
  }

//คำสั่ง Start แบบอัตโนมัติ
public void Start()
  {
     button5.Click += new EventHandler(button5_Click);
 }

//คำสั่งเมื่อทำการกดปุ่ม Start
private void button5_Click(object sender, EventArgs e)
 {
 int test=0;
 ////// Edge Detection  ///////
 Bitmap tempBitmapColor = (Bitmap)pictureBox2.Image;
 DifferenceEdgeDetector filter = new DifferenceEdgeDetector();
 Grayscale gray = new Grayscale(0.2125, 0.7154, 0.0721);
 Bitmap grayImage = gray.Apply((Bitmap)pictureBox2.Image);
    filter.ApplyInPlace(grayImage);
    pictureBox3.Image = grayImage;

 ////// Threshold  ///////
 int t = int.Parse(textBox1.Text.ToString());
 Threshold filter1 = new Threshold(t);
 Bitmap Threshold = filter1.Apply((Bitmap)pictureBox3.Image);
 pictureBox4.Image = Threshold;

 ////// Inverter /////
 Invert filter2 = new Invert();
 Bitmap Inverter = filter2.Apply((Bitmap)pictureBox4.Image);
    pictureBox5.Image = Inverter;

 /////// Circle Detection  /////
 Bitmap bitmap = (Bitmap)pictureBox5.Image;
 Bitmap tempBitmap = new Bitmap(bitmap.Width, bitmap.Height);
 Graphics g = Graphics.FromImage(tempBitmap);
    g.DrawImage(bitmap, 0, 0);
 BlobCounter blobCounter = new BlobCounter();
    blobCounter.ProcessImage(tempBitmap);
    bitmap = (Bitmap)pictureBox5.Image;
 int tt = int.Parse(textBox2.Text.ToString());
 Blob[] blobs = blobCounter.GetObjectsInformation();
 Pen redPen = new Pen(Color.Red, 5);
 for(int i = 0, n = blobs.Length; i < n; i++)
    {
 List<IntPoint> edgePoints;
 try
     {
       edgePoints = blobCounter.GetBlobsEdgePoints(blobs[i]);
  SimpleShapeChecker shapeChecker = new SimpleShapeChecker();
       AForge.Point center;
  float radius;
      //การตรวจหาวงกลม
  if(shapeChecker.IsCircle(edgePoints, out center, out radius))
       {
```

//เมื่อค่าของ radius มากกว่าค่าที่กำหนดไว้

```
if(radius > tt)
        {
          g.DrawEllipse(redPen,
(int)(center.X - radius),
(int)(center.Y - radius),
(int)(radius * 2),
(int)(radius * 2));

          label1.Text = radius.ToString();
          g = Graphics.FromImage(tempBitmap);
          pictureBox6.Image = tempBitmap;
```

//การตรวจหาวงกลมวงใหญ่

```
this.r = (int)radius;
this.CenterX = (int)centerX;
this.CenterY = (int)centerY;
for(int j = 0; j < tempBitmap.Height; j++)//Y
        {
 for(int k = 0; k < tempBitmap.Width; k++)//X
 {
  int A = (int)Math.Pow(k - CenterX, 2.0);
  int B = (int)Math.Pow(j - CenterY, 2.0);
  int C = A + B;
  int D = (int)Math.Sqrt(C);
  if(D > radius - 10)
             {
  //ทำการ SetPixel ตามเงื่อนไขให้เป็นสีขาว
  tempBitmapColor.SetPixel(k, j, Color.White);
               }
             }
 }
 //การตรวจวงกลมวงเล็ก
          CenterX = 3;
 CenterY = 18;
 for(int j = 0; j < tempBitmap.Height; j++)//Y
          {
  for(int k = 0; k < tempBitmap.Width; k++)//X
            {
   int A = (int)Math.Pow(k - CenterX, 2.0);
   int B = (int)Math.Pow(j - CenterY, 2.0);
   int C = A + B;
   int D = (int)Math.Sqrt(C);
   if(D < 72)
              {
   //ทำการ SetPixel ตามเงื่อนไขให้เป็นสีขาว
               tempBitmapColor.SetPixel(k, j, Color.White);
              }
            }
          }
          pictureBox7.Image = tempBitmapColor;
```

```
//การตัดขอบภาพตามวงกลม
            radius = 200;
   Crop filter4 = new Crop(new Rectangle((int)(centerX - radius),
   (int)(centerY - radius),
   (int)(radius * 2),
   (int)(radius * 2)));
            tempBitmapColor = filter4.Apply(tempBitmapColor);
            pictureBox8.Image = tempBitmapColor;
  }

  //เมื่อไม่เจอวงกลม
  else
       {
         test = 1;
  //ให้ Visual Studio ส่งข้อมูลดิจิตอลเป็น A ไปยังไมโครคอนโทรเลอร์
         serialPort1.Write("A");
         }
 }
 else//เมื่อไม่เจอวงกลม
    {
      test = 2;
  //ให้ Visual Studio ส่งข้อมูลดิจิตอลเป็น A ไปยังไมโครคอนโทรเลอร์
  serialPort1.Write("A");
    }
   }
catch(Exception)
  {
 break;
}
}


if(test == 1)
 {
    check = true;
    Capture();
 }
if(test == 2)
 {
    check = true;
    Capture();
 }
try
 {
//คำสั่งเมื่อทำการ Template Matching
Bitmap pic1 = gray.Apply((Bitmap)pictureBox8.Image);
Bitmap pic2 = gray.Apply((Bitmap)pictureBox10.Image);
Bitmap pic3 = new Bitmap(pic1.Width, pic1.Height);
int count = 0;
for(int x = 0; x < pic1.Width; x++)
 {
for(int y = 0; y < pic1.Height; y++)
```

```
    {
  //ประกาศตัวแปร num
  int num = pic2.GetPixel(x, y).R - pic1.GetPixel(x, y).R;

  if(num > int.Parse(textBox8.Text.ToString()))
     {
        count++;
        pic3.SetPixel(x, y, Color.Black);
  }
  else
     {
   pic3.SetPixel(x, y, Color.White);
     }
    }
   }
   pic3 = gray.Apply((Bitmap)pic3);
   pictureBox9.Image = (pic3);
   reprocess += 1;
 MessageBox.Show(count + "");

 //เมื่อนับจำนวน Pixel แล้วจะทำการตีความตามเงื่อนไข
 if (count > int.Parse(textBox9.Text.ToString()))
   {
  MessageBox.Show("Defect");
   }
 else
   {
  MessageBox.Show("Good");
   }
//ให้ Visual Studio ส่งข้อมูลดิจิตอลเป็น B ไปยังไมโครคอนโทรเลอร์
serialPort1.Write("B");
}
catch
{
}
redPen.Dispose();
g.Dispose();
}
public bool check2 = false;
public bool check = false;

//เมื่อรับข้อมูลดิจิตอลจาก Arduino
private void serialPort1_DataReceived(object sender,
System.IO.Ports.SerialDataReceivedEventArgs e)
{
  reprocess = 0;
  Capture();
  check = true;
}

private void timer1_Tick(object sender, EventArgs e)
   {
  if(check == true)
     {
       check = false;
```

```
  if(reprocess ==1)
       {
           reprocess =2;
  }
  if(reprocess ==0)
       {
    button5.PerformClick();
       }
 }
 if(check2 ==true)
      {
         check2 =false;
      }
 }
}
}
```

ภาคผนวก ข.

โปรแกรมในส่วนของ Arduino ควบคุมมอเตอร์

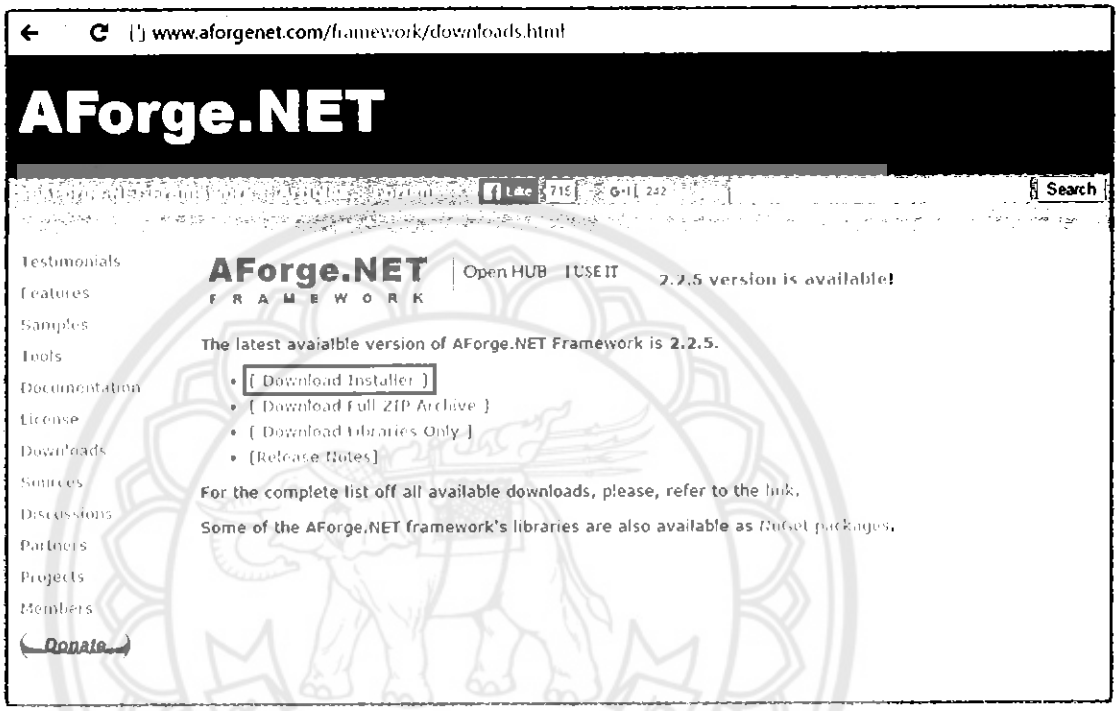
```
const int Motor  = 2;   //ประกาศตัวแปร Motor ให้ใช้ DigitalPin 2
const int Sensor = 3;   //ประกาศตัวแปร Sensor ให้ใช้ DigitalPin 3
int LED = 5;  //ประกาศตัวแปร LED ให้ใช้ DigitalPin 5
void setup()
{
 Serial.begin(9600);
 pinMode(Motor,OUTPUT);
 pinMode(Sensor,INPUT_PULLUP);
 digitalWrite(Motor,LOW);  //ให้มอเตอร์ทำงานเมื่อมีการจ่ายไฟ
 pinMode(LED,OUTPUT);
 digitalWrite(LED,LOW);  // ให้ LED ดับเมื่อไมโครคอนโทรเลอร์ทำงาน
}
int DelayMotor = 10; //ประกาศให้ตัวแปร DelayMotor เก็บค่าจำนวนเต็มเท่ากับ 10
boolean InfraredSensor = false;  //ประกาศให้ตัวแปร InfraredSensor เก็บค่า false
void loop()
{
//คำสั่งการทำงานของ Sensor
 if(digitalRead(Sensor)==LOW)  //เมื่อ Sensor ทำงาน
 {
    if(InfraredSensor==false)  //เมื่อตัวแปร InfraredSensor มีค่าเท่ากับ false
    {
      InfraredSensor = true;  //ให้เปลี่ยนตัวแปร InfraredSensor มาเก็บค่า true
      DelayMotor = 0;   //ให้เปลี่ยนตัวแปร DelayMotor มีค่าเท่ากับ 0
      Serial.print("Detect");    //ให้หน้า Serial monitor แสดงคำว่า Detect ออกมา
    } }
else
 {
    InfraredSensor = false; //เมื่อตัวแปร InfraredSensor มีค่าเท่ากับ false
```

```
 }
//คำสั่งการทำงานของ Motor
 if(DelayMotor<10)  //ถ้า DelayMotor มีค่าน้อยกว่า 10
 {
    DelayMotor++;  //ให้ค่า DelayMotor ทำการเพิ่มค่าขึ้น 1 ค่า
    digitalWrite(Motor,HIGH);  //สั่งให้มอเตอร์หยุดทำงาน
 }
 else  //ถ้าค่าของ DelayMotor มีค่ามากกว่าหรือเท่ากับ 10 ให้มอเตอร์ทำงาน
 {
  digitalWrite(Motor,LOW);  //สั่งให้มอเตอร์ทำงาน
 }
 delay(500);//สั่งให้มอเตอร์ทำงานเป็น เวลา 5 วินาที
}
//คำสั่งเมื่อ ไมโครคอนโทรเลอร์ รับข้อมูลดิจิตอล จาก Visual Studio 2010 แล้วไปควบคุมการ
ทำงานของ LED
void serialEvent()
{
 while (Serial.available())
 {
  char inChar = (char)Serial.read();
  if (inChar == 'A') //ถ้ารับข้อมูลดิจิตอลเป็น 'A' จาก Visual Studio 2010
  {
   digitalWrite(LED,HIGH); // ให้ LED ทำงาน
  }
  else //ถ้ารับข้อมูลดิจิตอลที่ไม่ใช่ 'A' จาก Visual Studio 2010
  {
   digitalWrite(LED,LOW); // ให้ LED ดับ
  } }}
```

ภาคผนวก ค.

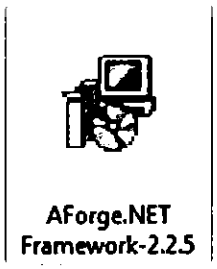
วิธีการติดตั้ง Aforge.net

1. ดาวน์โหลด Aforge.net ได้ที่เว็บ http://www.aforgenet.com/framework/downloads.html แล้วเลือกการดาวน์โหลดแบบ Download Installer



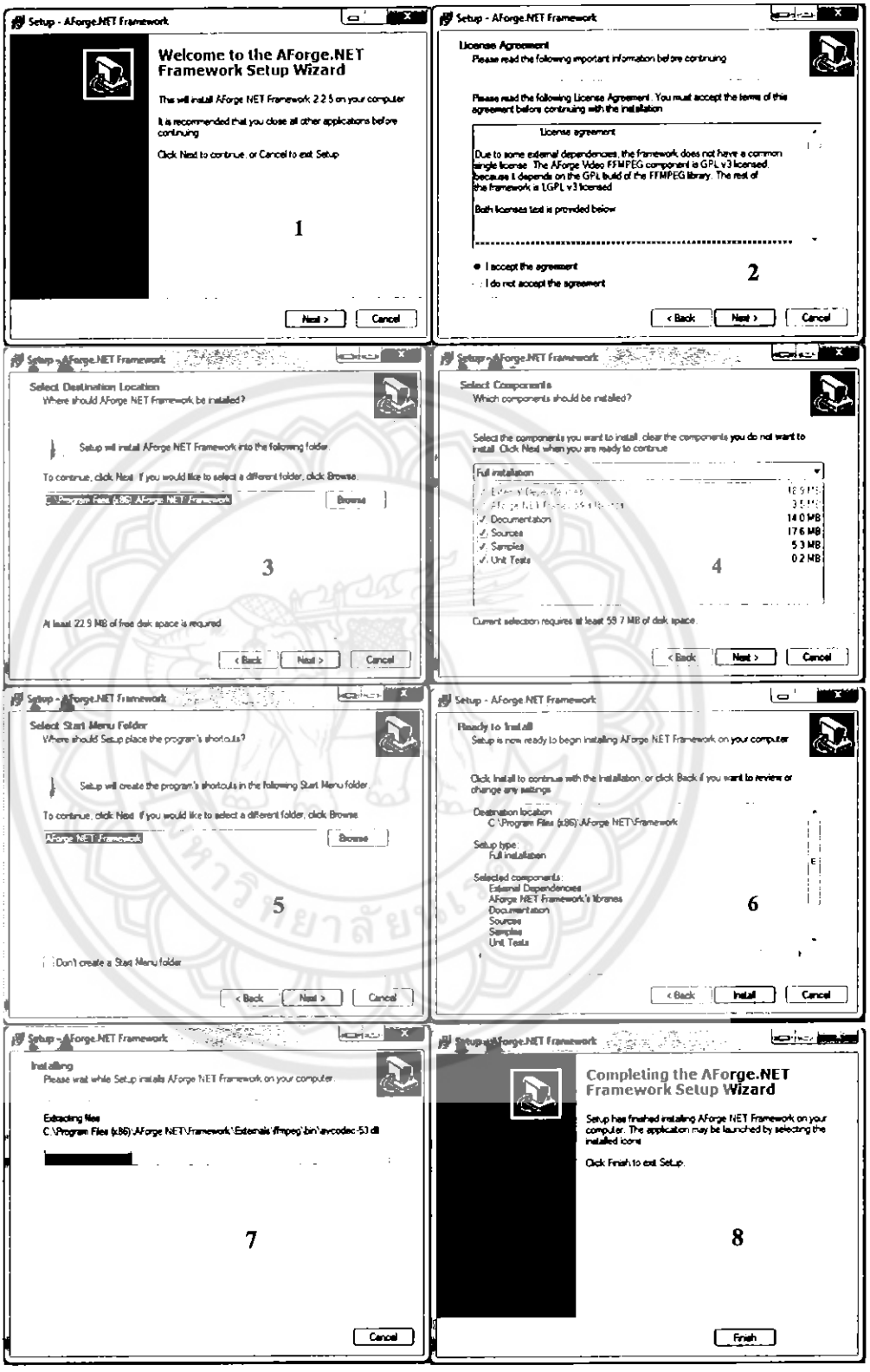
รูปที่ 1.1 แสดงส่วนการดาวน์โหลดของ Aforge.net

2. เมื่อดาวน์โหลดเสร็จแล้ว จะได้ไฟล์ติดตั้งชื่อ AForge.NET Framework-2.2.5 หลังจากนั้นทำการดับเบิ้ลคลิกที่ไฟล์



รูปที่ 1.2 ตัวติดตั้ง AForge.net

3. ทำการติดตั้งโปรแกรมตามลำดับดังนี้



รูปที่ 1.2 ขั้นตอนการติดตั้ง Aforge.net