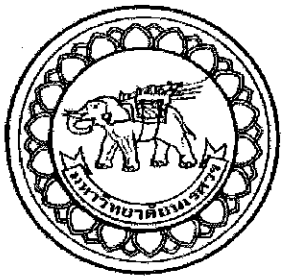


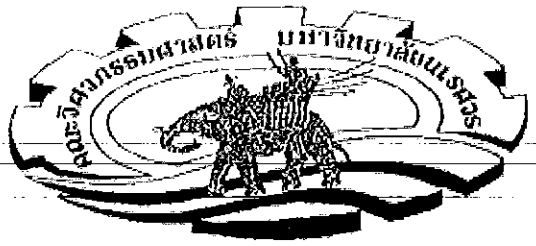
# ระบบแยกแยะสัญญาณไฟและป้ายจราจรโดยการประมวลผลภาพ สำหรับรถไร้คนขับ

**Traffic Light Sign Classification via Image Processing for Unman Vehicle**


| | | | |
|---|---|---|---|
| นายกฤษณรงค์ | อุทธิยัง | รหัส | 50360487 |
| นายภูวดล | ยั้งระหัด | รหัส | 50365239 |


ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2553


ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 26 / พ.ค. / 55
เลขทะเบียน 1674862
เลขเรียกหนังสือ วิ.
มหาวิทยาลัยนเรศวร ก281 ร
2553



# ใบรับรองปริญญานิพนธ์

| | |
|---|---|
| **หัวข้อโครงงาน** | ระบบแยกแยะสัญญาณไฟและป้ายจราจรโดยการประมวลผลภาพสำหรับรถไร้คนขับ |
| **ผู้ดำเนินโครงงาน** | 1. นายกฤษณรงค์ อุทธิยัง รหัส 50360487 |
| | 2 นายภูวดล ยั้งระหัด รหัส 50365239 |
| **อาจารย์ที่ปรึกษา** | อาจารย์เศรษฐา ตั้งค้าวานิช |
| **สาขาวิชา** | วิศวกรรมคอมพิวเตอร์ |
| **ภาควิชา** | วิศวกรรมไฟฟ้าและคอมพิวเตอร์ |
| **ปีการศึกษา** | 2553 |

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพจน์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตร์บัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

เศรษฐา ตั้งค้าวานิช .....ที่ปรึกษาโครงการ

( อาจารย์เศรษฐา ตั้งค้าวานิช )

.....กรรมการ

( ผศ.ดร.พนมขวัญ ริยะมงคล )

สิรภพ ศรัตน์ .....กรรมการ

( อาจารย์สิรภพ ศรัตน์ )

| | |
|---|---|
| หัวข้อโครงงาน | ระบบแยกแยะสัญญาณไฟและป้ายจราจรโดยการประมวลผลภาพสำหรับรถไร้คนขับ |
| ผู้ดำเนินโครงงาน | 1. นายกฤษณรงค์ อุทธิยัง รหัส 50360487 |
| | 2 นายภูวดล ยั้งระหัด รหัส 50365239 |
| อาจารย์ที่ปรึกษา | นายเศรษฐา ตั้งค้าวานิช |
| สาขาวิชา | วิศวกรรมคอมพิวเตอร์ |
| ภาควิชา | วิศวกรรมไฟฟ้าและคอมพิวเตอร์ |
| ปีการศึกษา | 2553 |

## บทคัดย่อ

ปัจจุบันยานพาหนะบนท้องถนนมีมากขึ้น การขับรถตามกฎจราจรเป็นสิ่งสำคัญ ผู้ขับต้องใช้สมาธิสังเกตสัญญาณจราจรต่างๆ เพื่อให้การขับขี่ปลอดภัย แต่ในความเป็นจริงนั้น ผู้ขับอาจไม่ทันสังเกต หรือตีความหมายสัญญาณจราจรผิดไป ซึ่งเป็นสาเหตุให้เกิดอุบัติเหตุได้

ระบบการตรวจจับและตีความป้ายสัญญาณจราจร จะช่วยให้ผู้ขับขี่สะดวกในการสังเกตป้ายสัญญาณจราจรต่างๆ ทำให้การขับขี่มีความปลอดภัยมากขึ้น ระบบที่จะพัฒนานี้ เป็นการตรวจจับป้ายสัญญาณจราจรจากวีดีโอ โดยใช้เทคนิคการหาลักษณะเด่นของภาพ ที่ชื่อว่า Haar like-feature เพื่อระบุตำแหน่ง และแปลความหมายของป้ายสัญญาณจราจรภายในภาพวีดีโอนั้น ซึ่งระบบนี้เป็นการจำลองการทำงานเบื้องต้น ซึ่งสามารถนำไปพัฒนาต่อ สำหรับตรวจจับเครื่องหมายจราจรบนท้องถนนจริงต่อไปได้

| | | |
|---|---|---|
| Project title | Traffic Light Sign Classification via Image Processing for Unman Vehicle. | |
| Name | 1. Kridnarong Auttiyang | ID. 50360487 |
| | 2. Phuvadon Yungrahud | ID. 50365239 |
| Project advisor | Settha Thangkawanit. | |
| Major | Computer Engineering. | |
| Department | Electrical and Computer Engineering. | |
| Academic year | 2010. | |


## Abstract

The number of vehicles on the road is increasing. Traffic rules are important for safety driving that all driver needs to pay good attention to traffic signs. However in reality, some drivers may not notice or interpret traffic signs in time, which might cause accidents.

Traffic sign detection and recognition from video will help the driver easy to observe different traffic signal label. Driving would be more safety. The system detects a sign from the video by using the technique called 'Haar like-feature' to identify the location and interprets the meaning of traffic signs. This system simulates the actual work, which can be developed for a Real-time detection to actual use on the road.

# กิตติกรรมประกาศ

การศึกษาและทำโครงงาน ในครั้งนี้ สำเร็จลุล่วงไปได้ด้วยดี เพราะบุคคลหลายท่านได้ให้คำแนะนำ และกรุณาสละความรู้ ความสามารถ ให้คำแนะนำและข้อคิดที่มีประโยชน์อย่างยิ่งต่อผู้จัดทำโครงงาน ผู้จัดทำโครงงานขอกราบขอบพระคุณ อาจารย์เศรษฐา ตั้งเค้าวานิช อาจารย์ที่ปรึกษาโครงงาน ที่คอยแนะนำ ชี้แนะแนวทางในการปฏิบัติงาน ร่วมถึงขัดเกล่าโครงงานจนสำเร็จ รวมถึง อาจารย์สิรภพ คชรัตน์ และ ผู้ช่วยศาสตราจารย์ ดร.พนมขวัญ ริยะมงคล กรรมการโครงงานที่คอยช่วยเหลือ ให้คำปรึกษาและสนับสนุนผู้จัดทำโครงงานมาโดยตลอด และ ขอบคุณ เพื่อนๆทุกๆคน ที่คอยให้กำลังใจตลอดมา

สุดท้ายนี้ ขอกราบขอบพระคุณ บิดา และมารดา อันเป็นที่รักของข้าพเจ้า ที่ไม่เคยทอดทิ้ง และเป็นกำลังใจให้ มาโดยตลอด

นายกฤษณรงค์ อุทธิยัง

นายภูวดล ยั้งระหัด

# สารบัญ


| | หน้า |
|---|---|
| บทคัดย่อ | ก |
| Abstract | ข |
| กิตติกรรมประกาศ | ค |
| สารบัญ | ง |
| สารบัญตาราง | จ |
| สารบัญรูป | ช |
| บทที่ 1 บทนำ | |
| 1.1 ที่มาของโครงงาน | 1 |
| 1.2 วัตถุประสงค์ของโครงงาน | 2 |
| 1.3 ลักษณะและขอบเขตของโครงงาน | 2 |
| 1.4 เครื่องมือที่ใช้ในการพัฒนาโครงงาน | 3 |
| 1.5 ขั้นตอนของการดำเนินงาน | 3 |
| 1.6 แผนผังการดำเนินโครงงาน | 4 |
| 1.7 ผลที่คาดว่าจะได้รับ | 7 |
| 1.8 งบประมาณที่ใช้ | 7 |
| บทที่ 2 ทฤษฎีพื้นฐานที่เกี่ยวข้อง | |
| 2.1 ภาพและความหมายของ pixel | 8 |
| 2.2 ระบบมาตรฐานสี(Colors Standard) | 9 |
| 2.3 การแยกภาพออกเป็นส่วนๆ | 12 |
| 2.4 การสร้างภาพไบนารี่ | 14 |
| 2.5 รูปร่างและโครงสร้างของภาพ | 16 |
| 2.6 การหาขอบภาพ | 21 |
| 2.7 Pattern Recognition | 23 |
| 2.8 กล้องและการเชื่อมต่อกล้อง web Camera | 29 |
| 2.9 ระบบวีดีโอ | 31 |
| 2.10 Opencv (Open Source Computer Vision Library) | 32 |

# สารบัญ(ต่อ)


หน้า

บทที่ 3 การวิเคราะห์ ออกแบบ และพัฒนาระบบ

3.1 การออกแบบระบบโดยรวม ..... 34

3.2 ขั้นตอนในการตรวจหาป้าย และสัญญาณไฟจราจร ..... 37

3.3 ขั้นตอนการ Training data ..... 39

3.4 อุปกรณ์และระบบ ..... 45

บทที่ 4 ผลการทดลอง

4.1 ผลการทดสอบการติดต่อกล้อง ..... 46

4.2 การทดสอบประสิทธิภาพของระบบจากกล้องวีดีโอ ..... 47

4.3 สรุปผลการทดสอบ ประสิทธิ์ภาพของระบบ ..... 55

บทที่ 5 บทสรุป

5.1 ข้อสรุป ..... 56

5.2 ปัญหาและอุปสรรค์ ..... 57

5.3 ข้อเสนอแนะและแนวทางการพัฒนา ..... 57

เอกสารอ้างอิง ..... 58

ภาคผนวก ..... 60

- การติดตั้ง โปรแกรมและ Opencv 2.1 ..... 61

# สารบัญตาราง


| ตารางที่ | หน้า |
|---|---|
| 1.1 แผนผังการดำเนินโครงงาน | 4 |
| 4.1 ตารางแสดงผลประสิทธิ์ภาพ การตรวจจับสัญญาณไฟจราจร ไฟแดงและไฟเขียว | 47 |
| 4.2 ตารางแสดงผลประสิทธิ์ภาพ การตรวจจับป้ายห้ามหยุด ป้ายห้ามจอด ป้ายเลี้ยวขวา ป้ายเลี้ยวซ้าย | 48 |
| 4.3 ตารางแสดงผลประสิทธิ์ภาพ การตรวจจับ ป้ายเลี้ยวขวา ที่มีขนาด จำนวน positive และ negative ที่แตกต่างกัน | 49 |

# สารบัญรูป


| รูปที่ | | หน้า |
|---|---|---|
| 2.1 | แสดงค่าสีในแต่ละ pixel | 8 |
| 2.2 | เมตริกซ์ pixel ของภาพ | 9 |
| 2.3 | ระบบสี RGB | 10 |
| 2.4 | ระบบสี HSV | 10 |
| 2.5 | แสดง Amplitude Segmentation Method ด้วยวิธีแบบวิธีการ Intensity thresholding | 12 |
| 2.6 | แสดงผลของการทำงานของวิธี Region Segmentation | 13 |
| 2.7 | แสดงผลของการทำงานของวิธี Edge Segmentation | 13 |
| 2.8 | ตัวอย่างภาพแบบ Binary หรือ ภาพขาว-ดำ | 14 |
| 2.9 | ภาพแบบ Grayscale หรือ ภาพระดับเทา | 15 |
| 2.10 | การขยายข้อมูลแบบ Histogram Equalization | 16 |
| 2.11 | ตัวประกอบโครงสร้างรูปสีเหลี่ยมขนมเปียกปูน (diamond) ขนาด 3 หน่วย | 16 |
| 2.12 | แสดงกระบวนการขยายภาพ (Dilation Operation) | 18 |
| 2.13 | แสดงภาพการขยายวัตถุภายในภาพ โดยใช้ Structure Element ที่มีรูปทรงเป็นเส้นตรงในแนวตั้ง | 18 |
| 2.14 | แสดงกระบวนการกร่อนภาพ (Erosion Operation) | 19 |
| 2.15 | แสดงภาพการกร่อนวัตถุในภาพด้วย structure element ที่มีรูปทรงจาน | 19 |
| 2.16 | ภาพการทำ Opening | 20 |
| 2.17 | รูปการทำ Closing | 20 |
| 2.18 | ตัวอย่างการหาขอบภาพ | 21 |
| 2.19 | (ก) แสดงถึงความแตกต่างของระดับความเข้มของสี (ข) กราฟแสดงการหาขอบด้วยวิธี Gradient methods (ค) และ Laplacian method | 22 |
| 2.20 | ตัวอย่างการหาขอบภาพโดยใช้ Edge detector แบบต่างๆ | 22 |
| 2.21 | รูปแบบของ Features สำหรับการตรวจจับลักษณะแบบต่างๆ | 26 |
| 2.22 | ตัวอย่างการใช้ Feature ตรวจจับลักษณะต่างๆ | 26 |
| 2.23 | การคำนวณแบบ Integral image | 26 |

# สารบัญรูป(ต่อ)


| รูปที่ | | หน้า |
|---|---|---|
| 2.24 | การทำงานของ Adaboost | 27 |
| 2.25 | ผลลัพธ์จากการทำกระวนการ AdaBoost | 27 |
| 2.26 | การทำงานของ Haar Cascade Classifier | 28 |
| 2.27 | ลักษณะการทำงานของ Cascade Classifier | 28 |
| 2.28 | ภาพแสดงกล้อง Web Cam | 29 |
| 3.1 | การทำงานของโปรแกรมตรวจจับและแปลความหมายป้ายและสัญญาณไฟ | 35 |
| 3.2 | แผนภาพวิธีการตรวจจับ และแปลความ | 35 |
| 3.3 | แผนผังการทำงานของโปรแกรม | 36 |
| 3.4 | ภาพ Positive image | 39 |
| 3.5 | ภาพ Negative image | 40 |
| 3.6 | ตัวอย่างการสร้างไฟล์ในการ Train ของภาพ Negatives | 41 |
| 3.7 | ตัวอย่างการสร้างไฟล์ในการ Train ของภาพ Positive | 41 |
| 3.8 | ตัวอย่างของ log ในการ Training | 43 |
| 3.9 | การแยกภาพจากกล้องวีดีโอ | 44 |
| 3.10 | การแสดงผลการตรวจจับป้ายจราจร | 44 |
| 4.1 | แสดงผลการติดต่อกล้อง | 46 |
| 4.2 | แสดงผลการตรวจจับป้ายหยุดและป้ายเลี้ยวซ้าย | 50 |
| 4.3 | แสดงผลการตรวจจับสัญญาณไฟจราจรสีเขียว | 51 |
| 4.4 | แสดงผลการตรวจจับสัญญาณไฟจราจรสีแดง | 52 |
| 4.5 | แสดงผลการตรวจจับป้ายหยุดและป้ายเลี้ยวขวา | 54 |

# บทที่ 1

# บทนำและความสำคัญของปัญหา

ในบทนี้จะกล่าวถึงความเป็นมา ประเด็นปัญหาและจุดประสงค์ของโครงงาน ซึ่งมีรายละเอียดดังนี้

## 1.1 ที่มาของโครงงาน

ในปัจจุบันยานพาหนะบนท้องถนนมีจำนวนที่เพิ่มมากขึ้น และผู้ขับต้องใช้ชีวิตบนท้องถนนเป็นเวลานาน เพราะฉะนั้นความปลอดภัยบนท้องถนนจึงเป็นสิ่งสำคัญเป็นอย่างยิ่งที่จะต้องคำนึงถึงเป็นอันดับแรก ในระบบช่วยเหลือในการขับขี่นี้ จึงมีบทบาทสำคัญในการที่จะอำนวยความสะดวกแก่ผู้ขับขี่ ทั้งยังช่วยในการตัดสินใจ และรับมือต่อสถานการณ์ต่างๆรอบตัว ช่วยให้ผู้ขับมีสมาธิในการขับขี่ ส่งผลให้ความปลอดภัยบนท้องถนนมีมากขึ้น

ป้ายสัญญาณจราจรเป็นส่วนสำคัญ ที่ช่วยแนะนำและเตือนผู้ขับขี่ให้ปฏิบัติตามกฎจราจร ปัญหาการมองป้ายสัญญาณจราจรไม่ชัดเจนหรือแปลความไม่ถูกต้อง เป็นสาเหตุทำให้ผิดกฎจราจรและก่อให้เกิดอุบัติเหตุได้ ด้วยเทคโนโลยีการประมวลผลภาพในปัจจุบัน ทำให้เราสามารถสร้างระบบช่วยเหลือการอ่านป้ายจราจร ด้วยการตรวจจับและแปลความป้ายสัญญาณจราจร เพื่อให้ผู้ขับขี่สามารถรับรู้และเข้าใจความหมายของป้ายสัญญาณจราจรได้อย่างชัดเจนและถูกต้อง ส่งผลให้ผู้ขับขี่สามารถปฏิบัติตามกฎจราจร รวมทั้งช่วยลดการเกิดอุบัติเหตุบนท้องถนนได้

เทคโนโลยีในปัจจุบันมีความเจริญก้าวหน้าไปอย่างมาก จากความเจริญก้าวหน้าในปัจจุบันได้มีการนำระบบตรวจหา (Detection) ซึ่งเป็นงานวิจัยที่สำคัญแขนงหนึ่งของระบบปัญหาปัญญาประดิษฐ์ (Artificial Intelligence) นำมาใช้ประโยชน์กับระบบตรวจหาป้ายจราจร (Traffic Detection System) โดยใช้คุณลักษณะเฉพาะของป้ายจราจรจากภาพถ่ายดิจิตอลหรือภาพจากวีดีโอมาประมวลผล ขั้นตอนสำหรับการตรวจหาป้ายจราจรนั้นจะประกอบไปด้วย การแบ่งแยกภาพโดยใช้สี การหาขอบของภาพและทำการเปรียบเทียบเพื่อวัดความคล้ายระหว่างป้ายจราจรต้นแบบกับป้ายจราจรที่หาได้ ว่าใช่ป้ายจราจรหรือไม่ แล้วสัญลักษณ์ของป้ายที่พบนั้นคืออะไร

## 1.2 วัตถุประสงค์ของโครงงาน

วัตถุประสงค์ของโครงการ เพื่อจัดทำระบบจำลองการตรวจจับ และตีความหมาย ป้ายสัญญาณจราจรและสัญญาณไฟจราจรบางส่วนที่ใช้บ่อยในท้องถนนจากกล้อง ที่ถ่ายจริงบนท้องถนน ได้แก่ ป้ายสัญญาณจราจรประเภทป้ายบังคับและประเภทป้ายเตือน ได้แก่ ป้ายหยุด, ป้ายเลี้ยวซ้าย-ขวา, ป้ายห้ามจอด และสัญญาณไฟจราจร สีเขียว กับ สีแดง โดยโครงการนี้ จะช่วยให้ผู้ขับขี่ยานยนต์สะดวกในการสังเกต และตีความหมายของป้ายสัญญาณจราจร ในสถานการณ์ต่างๆได้ดียิ่งขึ้น รวมถึงส่งผลให้การขับขี่ยานพาหนะ บนท้องถนนมีความปลอดภัยยิ่งขึ้น

## 1.3 ลักษณะและขอบเขตของโครงงาน

การศึกษาวิเคราะห์การจัดทำระบบตรวจจับสัญญาณจราจร และแปลความหมายป้ายจราจรและสัญญาณไฟจราจรที่พบจากกล้องที่ติดบนตัวรถ ครั้งนี้ได้จำกัดขอบเขตของงานไว้ดังนี้

1.3.1 ด้านการตรวจจับภาพ และ แปลความหมาย

- สามารถตรวจจับภาพและแปลความของป้ายจราจร ประเภทป้ายบังคับได้แก่ ป้ายหยุด, ป้ายเลี้ยว ซ้าย-ขวา ,ป้ายห้ามจอด ในช่วงเวลา 12.00 น -17.00 น ได้อย่างถูกต้อง
- สามารถตรวจจับภาพและแปลความของสัญญาณไฟจราจร ได้แก่ ไฟแดง(หยุดรถ), ไฟเขียว(รถไปได้) ในช่วงเวลา 12.00 น -17.00 น ได้อย่างถูกต้อง

1.3.2 ด้านการแสดงผล

- สามารถแสดงผลการตรวจจับ โดยการใส่กรอบล้อมรอบป้ายสัญญาณจราจรนั้นๆ
- สามารถแสดงผลของการแปลความ โดยบอกได้ว่า ภาพสัญญาณจราจรหรือป้ายจราจรที่พบ หมายถึงป้ายจราจรชนิดใด หรือถ้าเป็นสัญญาณไฟจราจรก็สามารถบอกได้ว่าเป็นช่วงที่รถหยุด(ไฟแดง) หรือสามารถเคลื่อนที่ได้(ไฟเขียว)
- ระบบไม่สามารถทำงานได้ในภาวะที่มีแสงน้อย

## 1.4 เครื่องมือที่ใช้ในการพัฒนาโครงงาน

1) ฮาร์ดแวร์

- เครื่องคอมพิวเตอร์
- เว็บแคมความละเอียดระดับ VGA

2) ซอฟท์แวร์

- โปรแกรม Visual Studio 2008 C++ สำหรับพัฒนาระบบด้วยภาษา C ++
- OpenCV version 2.1 ไลบรารีสำหรับพัฒนาระบบด้านรับข้อมูล
- Microsoft Windows 7 Ultimate 64bit เป็นระบบปฏิบัติการ
- Microsoft Office 2007 เพื่อช่วยสนับสนุนการพัฒนาระบบ

## 1.5 ขั้นตอนของการดำเนินงาน

1. รวบรวมข้อมูลที่เกี่ยวข้องกับโครงงาน
2. ศึกษาทำความเข้าใจกับเนื้อหาที่เลือกมา
3. ออกแบบและเขียนโปรแกรมในส่วนการรับภาพและประมวลผลภาพ
4. ออกแบบและเขียนโปรแกรมในส่วนที่เชื่อมต่อกับกล้องที่เชื่อมต่อกับตัวรถ
5. ออกแบบวิธีการทำให้โปรแกรมส่วนต่างๆ ทำงานร่วมกัน
6. แก้ไขข้อบกพร่องของโปรแกรม
7. สรุปและจัดทำเอกสารรายงาน

## 1.6 แผนผังการดำเนินโครงงาน

**ตารางที่ 1.1 แผนผังการดำเนินโครงงาน**

| ตารางแผนการดำเนินงานและระยะเวลาในการพัฒนาโครงงาน | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| No | รายละเอียดการดำเนินงาน | ดัชนีชี้วัด | พ.ศ. 2553 | | | | | | | | พ.ศ. 2554 | |
| | | | ภาคเรียนที่ 1 ปีการศึกษา 2553 | | | | ภาคเรียนที่ 2 ปีการศึกษา 2553 | | | | | |
| | | | กรกฎาคม | สิงหาคม | กันยายน | ตุลาคม | พฤศจิกายน | ธันวาคม | มกราคม | กุมภาพันธ์ | มีนาคม | |
| | **1. ศึกษาและวิเคราะห์ระบบงาน** | | | | | | | | | | | |
| 1.) | ศึกษา openCV, Visual Studio C++ | 1. เข้าใจการทำการของฟังก์ชั่น และสามารถเขียนโปรแกรมได้ | | | | | | | | | | |
| 2.) | ศึกษาทฤษฎีที่เกี่ยวข้อง | 1 เข้าใจทฤษฎีที่เกี่ยวกับระบบ | | | | | | | | | | |
| 3.) | ศึกษาอัลกอริทึ่ม | 1.เข้าใจอัลกอริทึม และสามารถสรุปข้อดีข้อเสียได้ | | | | | | | | | | |
| | **2. รวบรวมสถิติและจัดเก็บตัวอย่าง** | | | | | | | | | | | |
| 4.) | รวบรวมสถิติของป้ายสัญญาณจราจร | 1. ระบุได้ว่า ป้ายประเภทใดที่พบเจอได้บ่อย | | | | | | | | | | |
| 5.) | เก็บตัวอย่างและแบ่งประเภทตัวอย่าง | แบ่งประเภทของตัวอย่างตามลักษณะเฉพาะได้ | | | | | | | | | | |

**ตารางที่ 1.1 แผนผังการดำเนินโครงงาน(ต่อ)**

| No | รายละเอียดการดำเนินงาน | ดัชนีชี้วัด | พ.ศ. 2553 | | | | | | | พ.ศ. 2554 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | ภาคเรียนที่ 1 ปีการศึกษา 2553 | | | ภาคเรียนที่ 2 ปีการศึกษา 2553 | | | | | |
| | | | กรกฎาคม | สิงหาคม | กันยายน | ตุลาคม | พฤศจิกายน | ธันวาคม | มกราคม | กุมภาพันธ์ | มีนาคม |
| **3. ออกแบบระบบ** | | | | | | | | | | | |
| 6.) | ออกแบบการรับข้อมูลและแสดงผล | Graphic User Interface Model | | | | | | | | | |
| 7.) | ออกแบบระบบโดยรวม | Flow Chart | | | | | | | | | |
| **4. พัฒนาระบบ** | | | | | | | | | | | |
| 8.) | ระบบการตรวจจับป้ายสัญญาณจราจร | สามารถตรวจจับได้อย่างถูกต้อง | | | | | | | | | |
| 9.) | ระบบการแปลความป้ายสัญญาณจราจร | สามารถแปลความได้อย่างถูกต้อง | | | | | | | | | |
| 10.) | Graphic User Interface | โปรแกรมสามารถทำงานได้ | | | | | | | | | |

**ตารางที่ 1.1** แผนผังการดำเนินโครงงาน (ต่อ)

| No | รายละเอียดการดำเนินงาน | ดัชนีชี้วัด | พ.ศ. 2553 | | | | | | พ.ศ. 2554 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | ภาคเรียนที่ 1 ปีการศึกษา 2553 | | | ภาคเรียนที่ 2 ปีการศึกษา 2553 | | | | | |
| | | | กรกฎาคม | สิงหาคม | กันยายน | ตุลาคม | พฤศจิกายน | ธันวาคม | มกราคม | กุมภาพันธ์ | มีนาคม |
| **2. เก็บตัวอย่างจริง และทดสอบการใช้งาน** | | | | | | | | | | | |
| 11.) | ทดสอบการใช้งาน | ผลการพัฒนาระบบ | | | | | | | | | |
| 12.) | แก้ไขข้อผิดพลาดของระบบ | สามารถแก้ไขข้อผิดพลาดของระบบได้ | | | | | | | | | |
| **1. ประเมินผลการทำงานของระบบ** | | | | | | | | | | | |
| 13.) | ตรวจสอบประสิทธิภาพของระบบ | ผลการประเมินผลการทำงาน | | | | | | | | | |
| **2. จัดทำคู่มือเอกสาร** | | | | | | | | | | | |
| 14.) | ทำเอกสารประกอบการใช้งานระบบ | เอกสารประกอบการใช้งาน | | | | | | | | | |

## 1.7 ผลที่คาดว่าจะได้รับ

1. เข้าใจหลักการประมวลผลของภาพและนำไปประยุกต์ใช้งานจริง
2. เพิ่มความสะดวกสบายในการขับขี่ยานพาหนะ
3. นำระบบที่สร้างขึ้นไปใช้เป็นระบบเตือนภัยให้กับผู้ขับขี่ยานพาหนะ เพื่อเพิ่มความปลอดภัยบนท้องถนน
4. ช่วยให้ผู้ขับขี่ยานพาหนะมีความเข้าใจที่ถูกต้องในความหมายของป้ายสัญญาณจราจร
5. สามารถนำไปประยุกต์ใช้งานกับงานที่ใกล้เคียงได้
6. เข้าใจการสร้างการติดต่อกับกล้องและการเลือกกล้องแต่ละประเภทให้เหมาะสมกับงาน
7. สามารถนำความรู้ที่ได้ไปประยุกต์กับงานอื่นที่เกี่ยวกับการประมวลผลภาพได้

## 1.8 งบประมาณที่ใช้

| | | | |
|---|---|---|---|
| 1.8.1 ค่าถ่ายเอกสารและค่าเข้าเล่มรายงานฉบับสมบูรณ์ | เป็นเงิน | 1,500 | บาท |
| 1.8.2 ค่ากล้องเว็บแคม 1 ตัว | เป็นเงิน | 500 | บาท |
| | รวมเป็นเงินทั้งสิ้น | **2,000** | บาท |
| | (สองพันบาทถ้วน) | | |

**หมายเหตุ** ถัวเฉลี่ยทุกรายการ

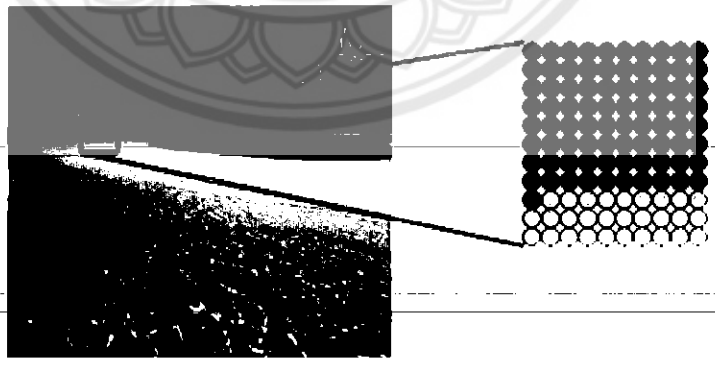
# บทที่ 2

# ทฤษฎีพื้นฐานที่เกี่ยวข้อง

ระบบที่จะพัฒนาขึ้นนั้นได้อาศัยเทคโนโลยีการประมวลผลภาพเข้ามาเกี่ยวข้อง ซึ่งมีทฤษฎีต่างๆมากมาย ล้วนแล้วแต่เป็นองค์ประกอบที่สำคัญ สำหรับการพัฒนาระบบการทำงาน โดยมีรายละเอียดดังนี้

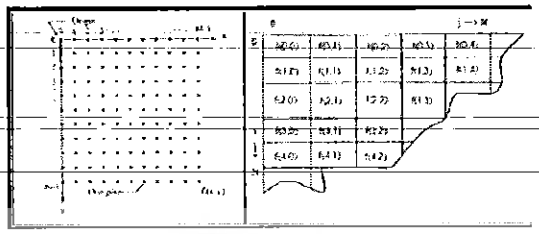
## 2.1 ภาพและความหมายของ pixel

ในด้านงานกราฟิกที่ใช้กับคอมพิวเตอร์นั้น pixel[1] ถือเป็นหน่วยย่อยที่เล็กที่สุดของภาพ ซึ่งจะมีลักษณะเป็นจุดเล็กๆ โดยในแต่ละภาพนั้นประกอบไปด้วย pixel จำนวนมากมายที่รวมกัน และในแต่ละภาพนั้นจะมีความหนาแน่นของ pixel ที่แตกต่างกันออกไป ซึ่งความหนาแน่นของ pixel นี้จะเป็นตัวบ่งบอกถึงความละเอียดของภาพมีหน่วยเป็น ppi (Pixel Per Inch) โดยปกติทั่วไปถือว่าภาพที่มีความละเอียดสูงหรือมีคุณภาพที่ดีนั้นจะต้องมีความละเอียด 300x300 ppi ขึ้นไป ยิ่งค่า ppi ยิ่งมีค่าสูงขึ้นเท่าใด ภาพที่ได้ก็จะมีความละเอียดและมีความคมชัดมากยิ่งขึ้นเท่านั้น โดยในแต่ละจุดจะเป็นตัวสร้างสีประกอบกันเป็นภาพรวมซึ่งอาจจะมีขนาดความเข้มและมีสีที่แตกต่างกันได้ จึงทำให้เกิดเป็นภาพที่มีสีสันต่างๆ ได้ ดังรูป 2.1



**รูป 2.1** แสดงค่าสีในแต่ละ pixel [2]

ในส่วนของภาพๆ หนึ่งนั้น สามารถที่จะอธิบายถึง Pixel ในรูปแบบเมตริกซ์ ขนาด N x M ได้ ดังรูปที่ 2.2 โดยจะใช้คู่อันดับ f (i, j) แทนค่าของแต่ละ pixel และสามารถที่จะบ่งชี้ความเข้มแสงที่ pixel นั้นๆ ของภาพได้ โดยที่ N = จำนวน pixel ที่มากที่สุดในแกน Y, M = จำนวน pixel ที่มากที่สุดในแกน X



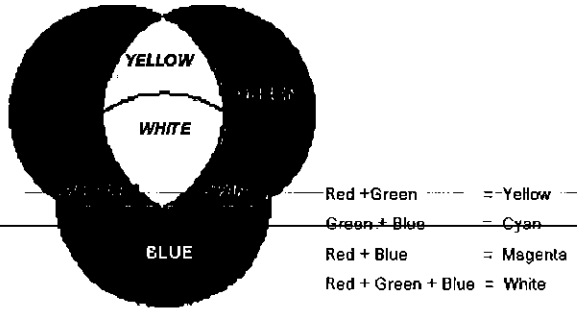
รูปที่ 2.2 เมตริกซ์ pixel ของภาพ

## 2.2 ระบบมาตรฐานของสี (Colors Standard)

มาตรฐานของสีที่มีใช้ในปัจจุบันมีอยู่มากมาย สำหรับในการนำไปใช้งานก็ขึ้นอยู่กับความเหมาะสมกับความต้องการของงานในแต่ละงาน แต่โดยทั่วไปแล้วทุกมาตรฐานของสีนั้นจะมีแนวคิดเดียวกันก็คือ การแทนจุดสีด้วยจุดที่อยู่ภายในสเปส 3 มิติ โดยจะมีแกนอ้างอิงสำหรับจุดสีนั้นในสเปส ซึ่งในแต่ละแกนจะมีความอิสระต่อกัน อย่างเช่น ในระบบ RGB จะมีแกนสี คือ แกนสีแดง(red) แกนสีเขียว(green) และแกนสีน้ำเงิน(blue) ในระบบ HSV ก็จะมี แกนค่าสี(hue) ความบริสุทธิ์ของสี (saturation) และ ค่าความสว่างของสี (value) เป็นต้น แต่ในงานปริญญานิพนธ์นี้เราจะเน้นการใช้มาตรฐานของสีแบบ HSV (Hue Saturation Value) เพราะในงานปริญญานิพนธ์นี้ไม่ต้องการมีตัวแปรเรื่องของแสงเข้ามาเกี่ยวข้องด้วยโดยจะเสนอรายละเอียดในหัวข้อของบทถัดไป

### 2.2.1 ระบบสี RGB (Red/Green/Blue) [3]

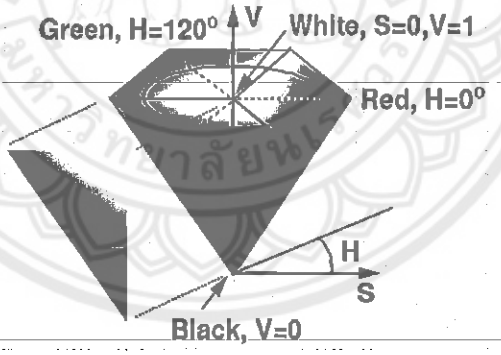
ระบบสี RGB เป็นระบบที่เกิดจากการรวมกันของแสงสีแดง สีเขียว และสีน้ำเงิน โดยมีการรวมกันแบบ Additive Color Mixing เพื่อให้ได้สีต่างๆ ขึ้นมา สำหรับในการผสมสีแบบบวกนี้ จะอยู่ในรูปแบบของการผสมของแสง ไม่ใช่การผสมของ วัตถุที่มีสีบนกระดาษ เนื่องจากแสงสีขาวประกอบไปด้วยลำแสงที่มีสีต่างๆ ตามความยาวของคลื่นแสงพื้นฐานได้แก่ สีแดง สีเขียว และสีน้ำเงิน เมื่อคลื่นแสงเหล่านี้มีการซ้อนทับกันก็จะก่อให้เกิด การบวกและรวมตัวกันของความยาวคลื่นแสง สีแบบบวกของแสงหรือแม่สีสามสีนี้เป็นสีขั้นต้น เมื่อผสมเข้าด้วยกันเป็นคู่หรือการผสมแบบบวก(Additive Mixing) จะได้สีขั้นที่สองดังรูป 2.3 โดยแต่ละสีจะมีค่าตั้งแต่ 0-255 โดยปกติแล้วระบบสี RGB นี้จะนำไปใช้กับระบบเครื่องใช้ไฟฟ้า เช่น โทรทัศน์ เป็นต้น




**รูปที่ 2.3** ระบบสี RGB [4]

**2.2.2 ระบบสี HSV (Hue Saturation Value)** [5]

ระบบสี HSV เป็นการพิจารณาสีโดยใช้ Hue Saturation และ Value ซึ่ง Hue คือ ค่าสีของสีหลัก ได้แก่ สีแดง สีเขียวและสีน้ำเงิน ในทางปฏิบัติจะอยู่ระหว่าง 0 – 255 ถ้าค่า Hue มีค่าเท่ากับ 0 จะแทนสีแดง และเมื่อค่า Hue มีค่าเพิ่มขึ้นเรื่อยๆ สีที่ได้ก็จะเปลี่ยนแปลงไปตามสเปกตรัมของสีจนถึง 256 จึงจะกลับมาเป็นสีแดงอีกครั้ง สามารถแทนให้อยู่ในรูปขององศาได้ ดังนี้คือ สีแดง = 0 องศา สีเขียว = 120 องศา และสีน้ำเงิน=240 องศา ค่า Hue สามารถแบ่งได้จากระบบสี RGB ได้รูป 2.4 ดังนี้




**รูปที่ 2.4** ระบบสี HSV

จากลักษณะโมเดลของระบบ Hue พบว่าจะมีค่าอย่างน้อยหนึ่งค่าที่จะเท่ากับ 0 แต่ถ้ามีสองค่าที่เท่ากับ 0 แล้วค่า Hue จะเป็นมุมของสี (ค่าสี) มีค่าเป็นไปตามสีที่สามและถ้าทั้งสามสีมีค่าเท่ากับ 0 แล้วจะทำให้ไม่มีค่าของ Hue แสดงว่าสีที่ได้จะเป็นสีขาวนั่นเอง ตัวอย่างเช่น จอภาพขาว – ดำ ถ้าเกิดมีสีได้สีหนึ่งมีค่าเท่ากับ 0 จะทำให้ค่าสีที่ได้เป็นไปตามสีที่เหลือ การให้น้ำหนักในการพิจารณาเมื่อสีแดงมีค่าเท่ากับ 0 สำหรับการแปลงสีจากมาตรฐาน RGB ไปสู่มาตรฐานสีแบบ HSV สามารถที่จะแปลงโดยใช้สมการที่ 1 ดังต่อไปนี้

$$H = \begin{cases} undefined\ , & if\ MAX = MIN \\ 60 \times \dfrac{G-B}{MAX + MIN} + 0, & if\ MAX = R\ and\ G \geq B \\ 60 \times \dfrac{G-B}{MAX + MIN} + 360, & if\ MAX = R\ and\ G < B \\ 60 \times \dfrac{G-B}{MAX + MIN} + 120, & if\ MAX = G \\ 60 \times \dfrac{G-B}{MAX + MIN} + 240, & if\ MAX = B \end{cases} \quad (1)$$

ให้ R G B แทนค่าของสีใน RGB Model ที่มีค่าระหว่าง 0.0 - 1.0

MAX = ค่าสูงสุดใน (R, G, B), MIN = ค่าต่ำสุดใน (R, G, B)

Saturation คือ ความบริสุทธิ์ของสีซึ่งถ้า Saturation มีค่าเท่ากับ 0 แล้วสีที่ได้จะไม่มี Hue ซึ่งจะเป็นสีขาวล้วนแต่ถ้า Saturation มีค่าเท่ากับ 255 แสดงว่าจะไม่มีแสงสีขาวผสมอยู่เลย สามารถคำนวณได้ตามสมการที่ 2 ดังนี้

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

Value คือ ค่าความสว่างของสี โดยที่ค่า Value ต่ำสุดนั้นจะหมายถึง สีดำ ไม่ว่าจะมีค่า Hue หรือ ค่า Saturation เท่าใดก็ตาม และในทางกลับกันจะเป็นสีขาวเมื่อมีค่า Value สูงสุด ซึ่งเป็นสีที่มีความสว่างที่สุดของค่า Hue และค่า Saturation นั้นๆ เช่น ที่ค่าHue ใดๆ มีค่า Saturation เท่ากับ 0 เมื่อ Value สูงสุดก็จะได้สีขาวในทางกลับกันถ้ามีค่าValue ต่ำสุดก็จะเป็นสีดำหรือ ที่ค่า Hue เป็นสีเหลือง Saturation มีค่าเท่ากับ 100 เมื่อ Value สูงสุดจะได้สีเหลืองและ Value ต่ำสุดก็จะสีดำซึ่งสามารถวัดได้โดยค่าความเข้มของความสว่างของแต่ละสีที่ประกอบกันสามารถคำนวณได้จากสมการที่ 3

$$Value = \max(red, green, blue) \quad (3)$$

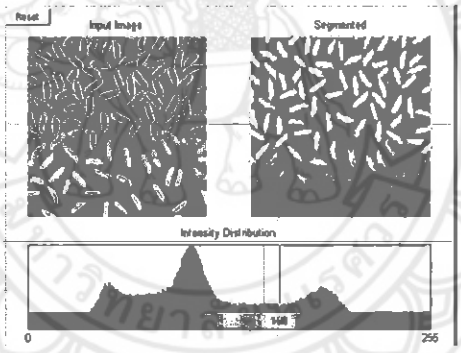
จะพบว่าในมาตรฐานสี HSV นั้นจะมีความคงทนต่อการเปลี่ยนแปลงของแสงมากกว่าการใช้ระบบสีแบบ RGB เนื่องจากว่าในการหาภาพที่มีความสว่างมาก ค่า Hue และ ค่า Saturation จะไม่มีการเปลี่ยนแปลง ใดๆเลย จะมีเพียงแต่ค่า Value เท่านั้นที่จะมีการเปลี่ยนแปลง เพราะฉะนั้นในมาตรฐานสีแบบ HSV ได้มีการแยกองค์ประกอบของแสงออกมาด้วยค่าของ Value ทำให้เมื่อพิจารณาภาพที่มีความสว่างของแสงหรือต้องการที่จะปรับค่าความสว่างของแสงจะทำได้อย่างอิสระโดยจะไม่มีผลกระทบกระทบต่อเฉดสีของภาพนั้นๆ เนื่องจากว่าระบบสีแบบ HSV ได้มีการแยกเฉดสีออกมาโดยจะขึ้นอยู่กับค่า Hue

## 2.3 การแยกภาพออกเป็นส่วน ๆ (Image Segmentation) [5]

การแยกภาพออกเป็นส่วนๆ นั้น จะเป็นการช่วยให้การทำงานเกี่ยวกับการวิเคราะห์ได้ง่ายและตรงจุดมากขึ้น ในการแยกภาพออกเป็นส่วนๆ นี้จะทำให้เราสามารถแยกภาพในส่วนที่เราไม่สนใจหรือส่วนของภาพที่เราไม่ต้องการออกไป โดยทั่วไปสามารถแบ่งวิธีการพื้นฐานสำหรับการแยกภาพออกเป็นส่วนๆ นี้ออกเป็น 3 กลุ่ม ดังนี้

### 2.3.1 Amplitude Segmentation Method

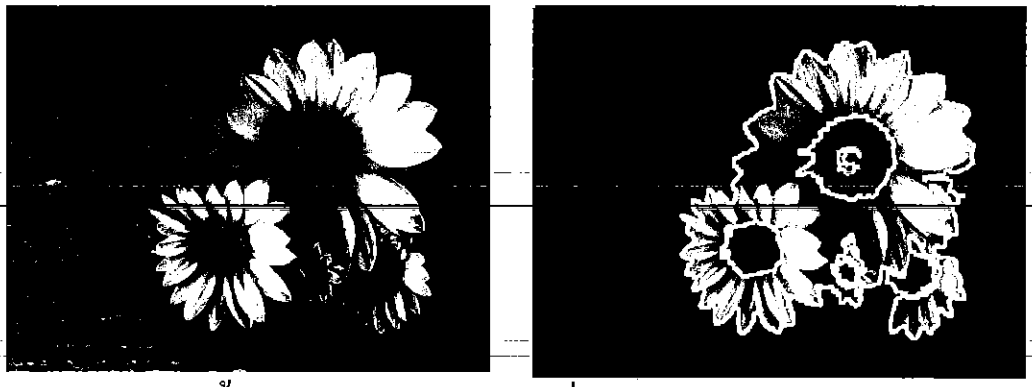
เป็นวิธีการแยกองค์ประกอบของภาพโดยดูจากความเหมือนกันของคุณสมบัติของ Pixel ภายในพื้นที่เพียงอย่างเดียว เช่น วิธีการ Intensity thresholding เป็นการแยกแยะวัตถุออกจากฉากหลังโดยดูจาก Intensity ของ pixel เป็นหลักจะแสดงให้เห็นดังรูป 2.5 ข้อดีของวิธีนี้ก็คือจะมีขั้นตอนในการทำงานที่ง่ายไม่ซับซ้อนและทำงานได้อย่างรวดเร็ว แต่วิธีนี้ก็มีข้อเสียอยู่เช่นกันนั่นก็คือจะไม่สามารถใช้กับภาพที่มีสัญญาณรบกวนมากๆ หรือภาพที่มีความสว่างที่ไม่สม่ำเสมอ




**รูปที่ 2.5** แสดง Amplitude Segmentation Method ด้วยวิธีแบบวิธีการ Intensity thresholding

### 2.3.2 Region Segmentation Method

เป็นวิธีการแยกองค์ประกอบของภาพโดยดูจากตำแหน่งของ pixel และความเหมือนกันของคุณสมบัติของ pixel ภายในพื้นที่เป็นหลัก โดยถ้า pixel ที่อยู่ติดกันและมีคุณสมบัติเหมือนกันจะถูกจัดให้เข้าอยู่ในกลุ่มเดียวกันดังรูปที่ 2.6 สำหรับข้อดีของวิธีนี้ก็คือ จะทำให้ได้พื้นที่ที่ต่อเนื่องกัน แต่ก็มียังมีปัญหาอยู่สำหรับในวิธีนี้นั่นก็คือ ในการที่จะกำหนดกฎเกณฑ์ที่เหมาะสมเพื่อใช้ในการตรวจสอบ pixel ที่มีคุณสมบัติเหมือนกัน เพื่อใช้ในการรวมกลุ่มเข้าด้วยกันนั้นทำได้ยาก และมีความซับซ้อนซึ่งต้องอาศัยอัลกอริทึมและโครงสร้างข้อมูลที่มีความซับซ้อนมากๆ จากความซับซ้อนนี้จึงทำให้ต้องใช้เวลานานในการประมวลผลแล้ว

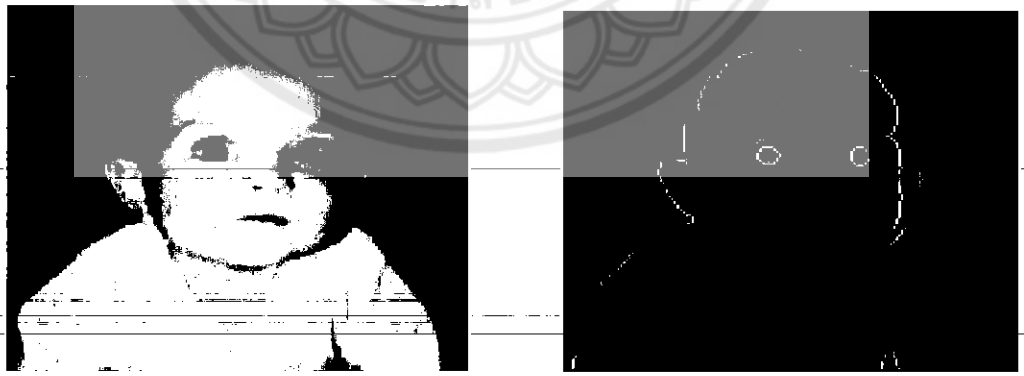


ต้นแบบ ผ่านกระบวนการ Region Segmentation

**รูปที่ 2.6** แสดงผลของการทำงานของวิธี Region Segmentation

### 2.3.3 Edge Segmentation Method

เป็นวิธีการแยกองค์ประกอบของภาพ โดยอาศัยความไม่ต่อเนื่องของคุณสมบัติของ Pixel ที่บริเวณขอบของวัตถุ ดังนั้นวิธีการนี้จึงมุ่งเน้นที่จะตรวจหาขอบของวัตถุ ดังรูปที่ 2.7 ข้อดีสำหรับวิธีนี้ก็คือ จะมีความรวดเร็วในการประมวลผล เนื่องจากว่าวิธีนี้จะใช้ข้อมูลบริเวณขอบของวัตถุเท่านั้น แต่ก็มีข้อเสียสำหรับวิธีนี้อยู่ก็คือ ผลลัพธ์ที่ได้ นั้นจะอยู่ในรูปของขอบของวัตถุเท่านั้น ซึ่งอาจจะต้องผ่านกระบวนการอีกต่อหนึ่งเพื่อที่จะนำผลลัพธ์มาใช้กับงานได้ และขอบวัตถุที่ได้นี้อาจจะมีความไม่ต่อเนื่องเกิดขึ้นได้สำหรับในวัตถุที่มีสีที่ไม่สม่ำเสมอ



ภาพต้นฉบับก่อนนำไปหาขอบภาพ ภาพที่ได้จากการหาขอบภาพ

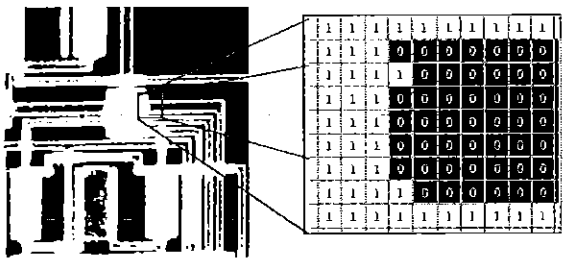
**รูปที่ 2.7** แสดงผลของการทำงานของวิธี Edge Segmentation [6]

## 2.4 การสร้างภาพไบนารี่ (binary image processing) [7-9]

การสร้างภาพไบนารี่สามารถทำได้โดยใช้เทคนิคการทำเทรชโฮล (Thresholding Technique) โดยพิจารณาว่าพิกเซลใดเป็นสีขาวหรือสีดำ จะกระทำโดยการเปรียบเทียบระหว่างพิกเซลของภาพเริ่มต้นกับค่าคงที่ค่าหนึ่งที่เรียกว่า "ค่าเทรชโฮล" (Threshold Value) เทคนิคนี้ใช้กันมากในกรณีที่ข้อมูลภาพมีลักษณะที่ต่างกันระหว่างวัตถุ (Object) และพื้นหลัง (Background) โดยค่าของพิกเซลของภาพใด ๆ ที่มีค่ามากกว่าหรือเท่ากับค่าเทรชโฮลจะถูกเปลี่ยนเป็น 0 (จุดดำ) ดังรูป 2.5 ในการสร้างภาพไบนารี่โดยใช้เทคนิคเทรชโฮลเพื่อให้ได้ผลลัพธ์ที่เหมาะสมและคมชัด สิ่งที่สำคัญที่สุดคือ ค่าเทรชโฮล เนื่องจากถ้าเลือกค่าเทรชโฮลที่ไม่เหมาะสม เช่น ค่าเทรชโฮลที่มีค่าน้อยเกินไปหรือมากเกินไป ภาพที่ได้อาจจะมืดเกินไป หรือสว่างมากเกินไป หรือภาพที่ได้มีสิ่งรบกวน (Noise) เกิดขึ้น อันเป็นผลทำให้ภาพที่ได้ไม่สวยงามเท่าที่ควร ดังนั้นปัญหาของการสร้างภาพไบนารี่ คือวิธีการกำหนดค่าเทรชโฮลที่เหมาะสมสำหรับแต่ละภาพที่จะนำมาทำการสร้างภาพไบนารี่ ซึ่งมีวิธีคำนวณหาค่าเทรชโฮลหลายวิธี โดยแต่ละวิธีเหมาะสมกับลักษณะการทำงานที่แตกต่างกันไป เช่น การหาค่าเทรชโฮลโดยกำหนดค่าล่วงหน้า (Pre-assigned Threshold Value) การหาค่าเทรชโฮลจากค่ากลาง (Mid-range Threshold Value) ซึ่งแต่ละวิธีอธิบายได้ดังนี้

2.4.1 การหาค่าเทรชโฮลโดยการกำหนดค่าล่วงหน้า เป็นการกำหนดค่าเทรชโฮลโดยการกำหนดเองจากผู้ใช้ ซึ่งการกำหนดนี้จะขึ้นอยู่กับประสบการณ์ของผู้ใช้คนนั้น ๆ โดยการเลือกค่าคงที่ค่าหนึ่ง ซึ่งเรียกว่า ค่าเทรชโฮล โดยค่าที่เลือกมานี้จะเป็นค่าที่อยู่ระหว่างค่าต่ำสุดและค่าสูงสุดของระดับความเข้มแสงของภาพ เช่น ภาพอินพุทมีระดับความเข้มแสง 256 ระดับ จะมีค่าได้ตั้งแต่ 0-255 เมื่อเลือกค่าเทรชโฮลได้แล้วสามารถสร้างภาพไบนารี่ได้

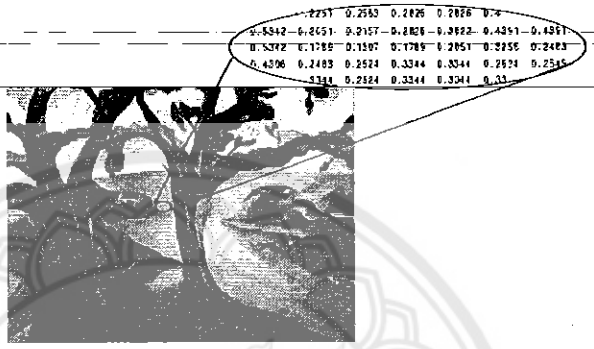
2.4.2 การหาค่าเทรชโฮลจากค่ากลาง เป็นการหาเทรชโฮลที่แตกต่างจากการหาค่าเทรชโฮลวิธีแรก สำหรับวิธีนี้จะเป็นการคำนวณหาค่าเทรชโฮลโดยอัตโนมัติโดยไม่ต้องให้ผู้ใช้เป็นผู้กำหนด โดยการหาค่าเทรชโฮลนี้ใช้วิธีทางสถิติในเรื่องการหาค่ากลางหรือค่าเฉลี่ย (Mean) ค่าเทรชโฮลที่คำนวณได้จากค่ากึ่งกลางที่อยู่ระหว่างค่าระดับความเข้มสูงสุด (Maximum Level) และระดับความเข้มต่ำสุด (Minimum Level) ของภาพ เมื่อทำการคำนวณค่าเทรชโฮลได้แล้วก็สามารถสร้างภาพไบนารี่ได้โดยนำค่าเทรชโฮลที่ได้มาใช้



รูปที่ 2.8 ตัวอย่างภาพแบบ Binary หรือ ภาพขาว-ดำ [10]

**Grayscale Image** เป็นรูปที่เก็บโดยใช้รูปแบบของอาร์เรย์ 2 มิติ โดยค่าที่เก็บจะมีค่าอยู่ในช่วงๆหนึ่ง ซึ่งระดับของสีขึ้นอยู่กับขนาดของบิตที่ใช้เก็บค่าสี โดยจะมีระดับความเข้มของสีเท่ากัน คือ 0 – 255 (8 bit) ภาพเกรย์สเกลเกิดจากการเปลี่ยนแปลงภาพสี RGB มาเป็นภาพ Grayscale โดยการใช้สูตรทางคณิตศาสตร์ ดังนี้

**Gray = 0.299 x Red + 0.587 x Green +0.114 x Blue**



รูปที่ 2.9 ภาพแบบ Grayscale หรือ ภาพระดับเทา

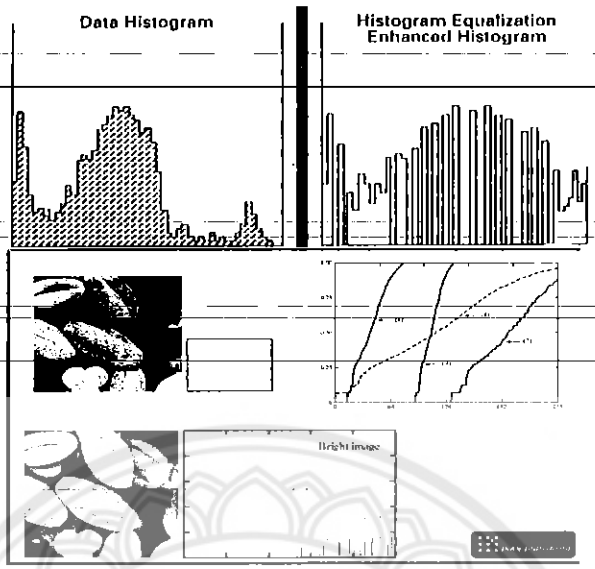
**Histogram Equalization [11]**

วัตถุประสงค์ของการใช้วิธีนี้ เป็นการสร้างภาพที่มีจำนวนจุดภาพใกล้เคียงกัน หรือ ข้อมูลที่แปลงแล้วจะมีการกระจายอย่างสม่ำเสมอ วิธีการนี้ใช้การกำหนดจำนวนจุดภาพที่เหมาะสมให้กับแต่ละค่า

จำนวนจุดภาพที่เหมาะสม = จำนวนจุดภาพทั้งหมด / จำนวนค่าในข้อมูล

การคำนวณจะเริ่มจากค่าที่น้อยที่สุด โดยเริ่มบวกจำนวนจุดภาพเข้าด้วยกัน จนกระทั่งเกินจำนวนจุดภาพที่เหมาะสมที่คำนวณได้ ก็ให้ค่าจุดภาพเหล่านั้นเป็นค่าค่าแรก และใช้ค่าถัดไปเป็นค่าใหม่ที่สองจุดภาพที่มีจำนวนเกินที่คำนวณได้ก็จะคงจำนวนเดิมไว้ แต่ถ้าเกินมากกว่า 1 เท่า ก็ยังคงค่าเดิมไว้ แต่จำนวนจุดภาพของค่าความเข้มของแสงค่าถัดไปจะไม่มี

ผลกราฟที่ได้หลังจากการขยาย จะคล้ายกราฟแท่งที่แบนราบกว่าเดิม ตามรูปที่ 2.10 จะเห็นได้ว่า บริเวณภาพที่มีค่า หรือมีจุดภาพที่มีค่าใกล้เคียงกัน จะถูกขยายออกให้มีความแตกต่างของค่าเพิ่มขึ้น (บริเวณกราฟที่มีความถี่สูงจะถูกขยายออก) ในขณะที่ส่วนน้อยของภาพที่มีค่าแตกต่างกัน (ส่วนหางของกราฟ) จะมีการต่างกันของค่าลดลง หรือมีช่วงห่างของความเข้มของแสงลดลง

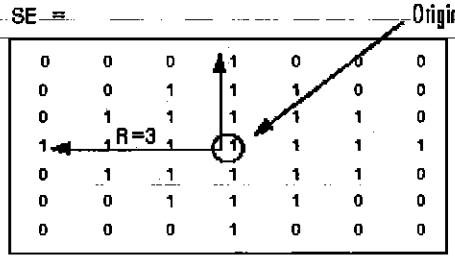



รูปที่ 2.10 การขยายข้อมูลแบบ Histogram Equalization

## 2.5 รูปร่างและโครงสร้างของภาพ (Morphological Operation) [12]

การประมวลผลภาพกับรูปร่างและโครงสร้างของภาพเป็นการประมวลผลภาพโดยการเปลี่ยนแปลงลักษณะรูปร่างหรือโครงสร้างของภาพเหมาะสำหรับภาพชนิด ขาว - ดำ โดยการประยุกต์ตัวประกอบโครงสร้าง (Structure Element) มากระทำกับภาพนำเข้า และได้ภาพผลลัพธ์ที่มีขนาดเท่ากับภาพนำเข้า

ตัวประกอบโครงสร้าง (Structure Element) คือ เมตริกซ์ที่มีขนาดเล็กกว่าภาพที่กระทำด้วย ซึ่งสามารถสร้างขึ้นเป็นรูปแบบต่างๆ ได้ เช่น จานสี่เหลี่ยมขนมเปียกปูน ดังรูป 2.11 โดยจุดกำเนิดจะกำหนดที่จุดภาพไหนก็ได้ แต่โดยปกติแล้วมักจะกำหนดจุดที่จุดศูนย์กลางของส่วนย่อยโครงสร้าง




**รูปที่ 2.11** ตัวประกอบโครงสร้างรูปสี่เหลี่ยมขนมเปียกปูน (diamond) ขนาด 3 หน่วย

เนื่องจาก Morphological Operation ได้มีการพัฒนามาจากพื้นฐานในเรื่องของ เซต (set) ดังนั้นจึงอธิบายเกี่ยวกับพื้นฐานของ เซต ที่เกี่ยวข้องได้แก่ ตัวดำเนินการทางตรรกะ นั่นก็คือ Union (OR) และ Intersection (AND)

กำหนดให้ ข้อมูลเมตริกซ์ A และ B ได้ดังนี้

$$A = \begin{matrix} 1 & 1 & * & 1 & 1 \\ * & 1 & 1 & 1 & 1 \\ 1 & * & 1 & * & 1 \end{matrix} \qquad B = \begin{matrix} 1 & * & * & * & 1 \\ 1 & * & * & * & 1 \\ 1 & * & * & * & 1 \end{matrix}$$

เมื่อนำเมตริกซ์ทั้งสองมา Union (OR) และ Intersection (AND) จะได้ดังนี้

$$A \text{ OR } B = \begin{matrix} 1 & 1 & * & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & * & 1 & * & 1 \end{matrix} \qquad A \text{ AND } B = \begin{matrix} 1 & * & * & * & 1 \\ * & * & * & * & 1 \\ 1 & * & * & * & 1 \end{matrix}$$

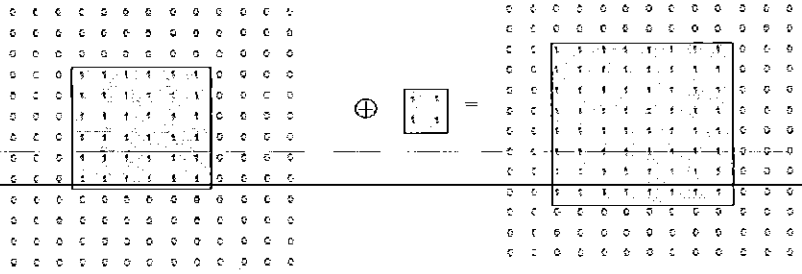
เลื่อนขนาน (Translation) คือ การเลื่อนส่วนย่อยของโครงสร้างไปที่บริเวณใดๆ ของภาพ โดยให้จุดศูนย์กลางของส่วนย่อยโครงสร้างตรงกับจุดภาพที่เริ่มจากมุมซ้ายบนของภาพไปทางขวาจนสุด แล้วเริ่มแถวสองด้านซ้ายสุดอีกทีเลื่อนเช่นนี้ไปเรื่อยๆ จนถึงแถวสุดท้ายของภาพ

### 2.5.1 การขยายภาพ (Dilation)

การขยายขนาดเป็นการขยายขนาดของวัตถุในภาพ การขยาย Pixel ที่สว่างให้ใหญ่ขึ้น สำหรับรูปภาพขาวดำ หรือแบบ Gray Scale เพื่อจุดประสงค์บางประการ เช่น เพื่อปิดรูเล็กๆ ในวัตถุ หรือใช้เพื่อช่วยให้วัตถุ 2 วัตถุที่ไม่มีสมาชิกร่วมกันแต่อยู่ใกล้กันสามารถเชื่อมต่อกันได้ เป็นต้น การขยายขนาดทำได้โดยวางสมาชิกโครงสร้าง(Structure Element) ลงบนภาพแล้วเลื่อนสมาชิกโครงสร้าง มีขั้นตอนดังต่อไปนี้

1. ถ้าจุดศูนย์กลางของสมาชิกโครงสร้างตรงกับค่า '0' ในภาพไม่ต้องดำเนินการใดๆ แต่ให้เลื่อนสมาชิกโครงสร้างไปยังจุดภาพถัดไป

2. ถ้าจุดศูนย์กลางของสมาชิกโครงสร้างตรงกับค่า '1' ในภาพให้ดำเนินการด้วยตัวดำเนินการทางตรรกะออร์ (OR) ระหว่างภาพกับสมาชิกโครงสร้าง กระบวนการดังกล่าวแสดงได้ดังรูปที่ 2.12 และตัวอย่างของการขยายภาพแสดงให้เห็นในรูปที่ 2.13 การขยายขนาดของภาพ A ด้วยตัวประกอบโครงสร้าง B สามารถแสดงได้ดังสมการ $A \oplus B$



**รูปที่ 2.12** แสดงกระบวนการขยายภาพ (Dilation Operation)

Original Dilation

**รูปที่ 2.13** แสดงภาพการขยายวัตถุภายในภาพ โดยใช้ Structure Element ที่มีรูปทรงเป็นเส้นตรงในแนวตั้ง

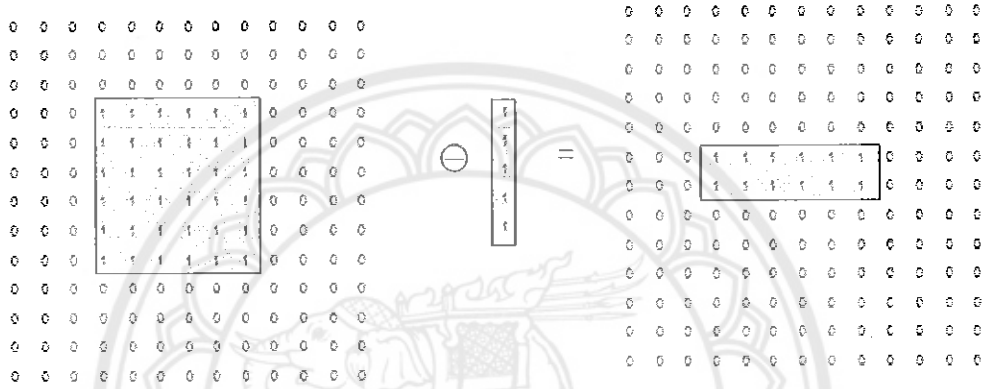
**2.5.2 การกร่อนภาพ (Erosion)**

การกร่อนภาพ (Erosion) เป็นกระบวนการย่อเนื้อที่ของบริเวณสีขาว หรือวัตถุในภาพให้เล็กลง โดยมีขั้นตอนการประมวลผลเช่นเดียวกับการขยายภาพ แต่มีวิธีการดำเนินการที่ต่างกัน โดยมีหลักการคือ ในสำหรับทุกตำแหน่งที่เลื่อนตัวประกอบโครงสร้างบนภาพและจะมีการเปรียบเทียบกับข้อมูลภาพ โดยกำหนดเงื่อนไขไว้ดังนี้

1. ถ้าจุดศูนย์กลางของตัวประกอบโครงสร้างตรงกับค่า '0' ในภาพไม่ต้องดำเนินการใดๆ แต่ให้เลื่อนตัวประกอบโครงสร้างไปยังจุดภาพต่อไป
2. ถ้าจุดศูนย์กลางของตัวประกอบโครงสร้างตรงกับค่า '1' ในภาพ ให้พิจารณาดังนี้
    a. ถ้าค่าในตัวประกอบโครงสร้างเหมือนกันกับค่าในภาพทุกตำแหน่ง จะทำการกำหนดค่าข้อมูลภาพที่ตำแหน่งที่ตรงกับจุดศูนย์กลางของตัวประกอบโครงสร้างให้มีค่าเท่ากับ '1'

b. ถ้ามีค่าในข้อมูลภาพตำแหน่งใดไม่ตรงกับค่าในตัวประกอบโครงสร้าง จะทำการกำหนดค่าข้อมูลภาพในตำแหน่งที่ตรงกับจุดศูนย์กลางของตัวประกอบโครงสร้างให้มีค่าเท่ากับ '0' ดังแสดงในรูป 2.14

ดังนั้นจะเห็นได้ว่าการทำการกร่อนภาพจะทำให้วัตถุในภาพต้นฉบับมีขนาดเล็กลงตามลักษณะของตัวประกอบโครงสร้าง ดังรูปที่ 2.14 การขยายขนาดของภาพ A ด้วย ตัวประกอบโครงสร้าง B สามารถแสดงได้ด้วยสมการ $A \ominus B$



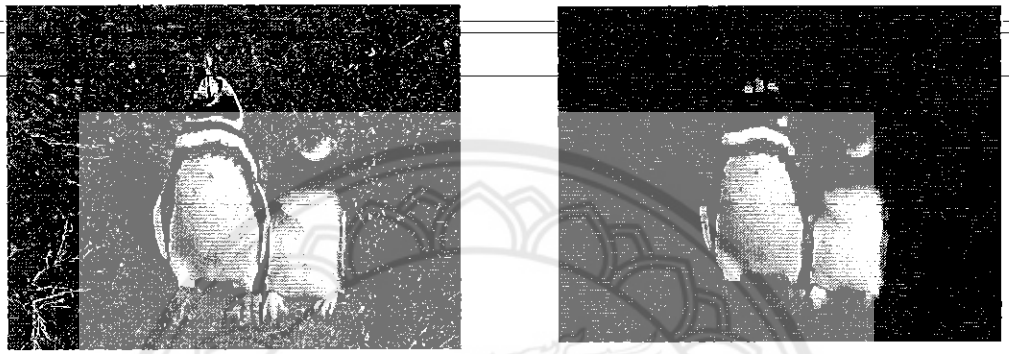
**รูปที่ 2.14** แสดงกระบวนการกร่อนภาพ (Erosion Operation)

**รูปที่ 2.15** แสดงภาพการกร่อนวัตถุในภาพด้วย structure element ที่มีรูปทรงจาน

### 2.5.3 การเปิดพื้นที่ว่างภายในภาพ (Opening)

Opening เป็นวิธีการเพื่อกำจัดรายละเอียดขนาดเล็กๆ ของภาพ ในการทำ Opening นั้นจะทำให้ pixel ของภาพถูกทำให้เปิดกว้างมากขึ้น ดังรูปที่ 2.16 วิธีในการของ Opening นี้ก็คือ การทำ Erosion ก่อนแล้วหลังจากนั้นก็จะไปทำ Dilation ตามลำดับได้ดังสมการ

$A \circ B = (A \ominus B) \oplus B$ ; B = Structuring Element (ตัวประกอบโครงสร้าง)

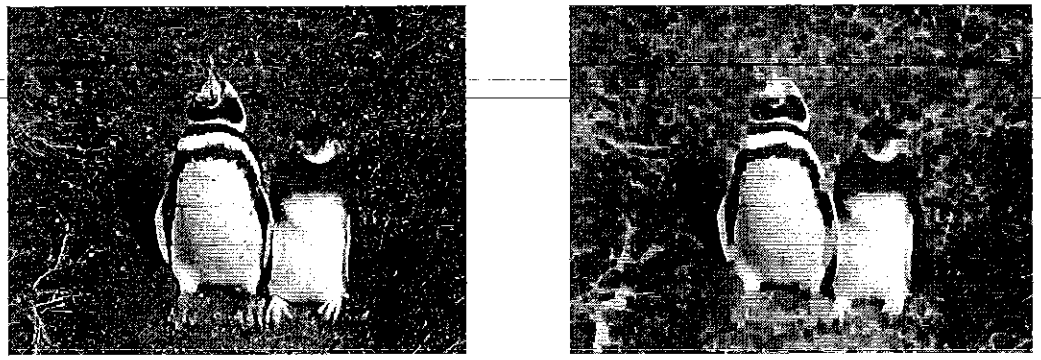


ก ข

**รูปที่ 2.16** ภาพการทำ Opening (ก) ภาพต้นฉบับ (ข) ผลลัพธ์จากการทำ Opening

### 2.5.4 การปิดพื้นที่ว่างภายในภาพ (Closing)

Closing เป็นวิธีการที่ตรงกันข้ามกับวิธีการ Opening โดยวิธีนี้จะเป็นการทำให้ภาพมีการเชื่อมต่อกันมากขึ้น สำหรับการ Closing นี้จะทำให้ pixel ของภาพถูกปิดเชื่อมต่อกันของเส้นมากขึ้น ดังรูป 2.17 วิธีการทำแบบ Closing นั้นมาจากการทำ Dilation ก่อน จากนั้นก็มาทำ Erosion ต่ออีกทีตามลำดับ ดังสมการ $A \bullet B = (A \oplus B) \ominus B$; B = Structuring Element (ตัวประกอบโครงสร้าง)



(ก) (ข)

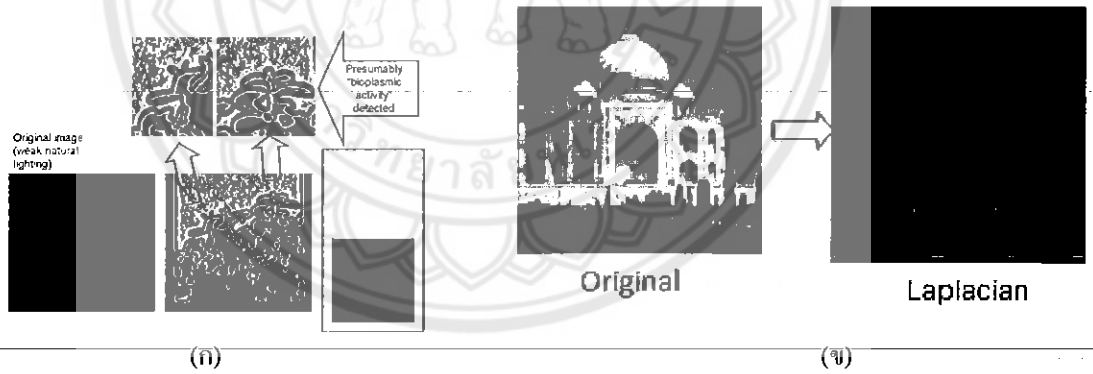
**รูปที่ 2.17** รูปการทำ Closing (ก) รูปต้นฉบับ (ข) ผลลัพธ์จากการทำ Closing

## 2.6 การหาขอบภาพ (Edge Detection) [13]

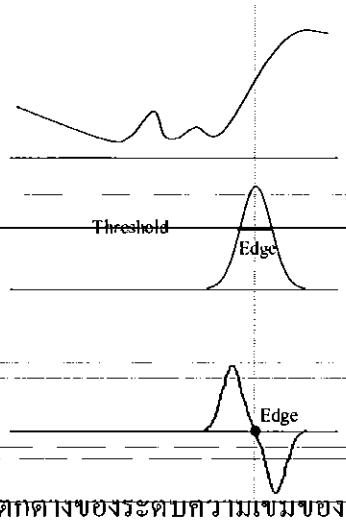
การหาขอบของภาพ เป็นกระบวนการเพื่อใช้ตรวจสอบว่าเส้นและขอบได้มีการลากผ่านหรือใกล้เคียงกับจุดใด ในภาพ โดยหลักวิธีการวัดนั้นจะวัดจากการเปลี่ยนแปลงของความเข้มในตำแหน่งที่ใกล้เคียงที่จุดอ้างอิง วิธีการหาขอบนั้นมีด้วยกันหลายวิธี สามารถแบ่งออกเป็นกลุ่มหลักๆ ได้ 2 กลุ่ม คือ Gradient Method และ Laplacian Method โดยในแต่ละวิธีมีรายละเอียดดังต่อไปนี้

**2.6.1 Gradient Method** เป็นวิธีที่หาขอบภาพโดยมีหลักการคือ จะหาขอบของภาพโดยการหาจุดต่ำสุดและจุดสูงสุดในรูปของอนุพันธ์อันดับหนึ่ง โดยจุดที่เป็นขอบจะอยู่ในส่วนที่มีค่าเหนือกว่าค่าของ Threshold (รูป 2.18 ข) จึงอาจจะทำให้เส้นขอบที่ได้มีลักษณะเส้นที่หนา ดังรูปที่ 2.17 (ก) ตัวอย่างวิธีการหาขอบในกลุ่มนี้ ตัวอย่างเช่น Sobel, Canny, Roberts และ Prewitt

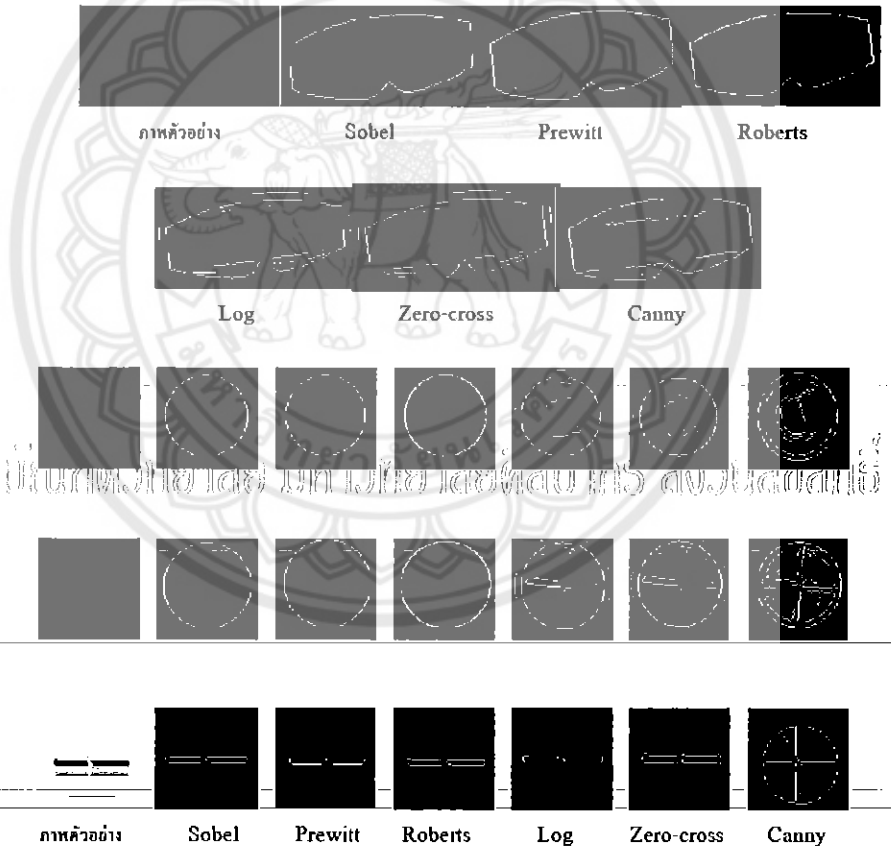
**2.6.2 Laplacian Method** เป็นวิธีหาขอบภาพโดยมีหลักการคือจะหาขอบโดยใช้อนุพันธ์อันดับ 2 โดยใช้จุดที่ค่า y เป็น 0 (Zerocrossing) (รูป 2.19 ค) ซึ่งวิธีนี้จะใช้เวลาในการคำนวณมากกว่า Gradient method ดังรูปที่ 2.18 (ก) ตัวอย่างวิธีการหาขอบของกลุ่มนี้ เช่น Laplacian of Gaussian และ Marrs-Hildreth เป็นต้น




**รูป 2.18** ตัวอย่างการหาขอบภาพ (ก) Gradient Method (ข) Laplacian Method




**รูปที่ 2.19** (ก) แสดงถึงความแตกต่างของระดับความเข้มของจุด (ข) กราฟแสดงการหาขอบด้วยวิธี Gradient method (ค) และ Laplacian method




**รูปที่ 2.20** ตัวอย่างการหาขอบภาพโดยใช้ Edge detector แบบต่างๆ

## 2.7 Pattern Recognition[14]

pattern recognition เป็นการให้คอมพิวเตอร์แยกแยะสิ่งต่างๆ เช่น แยกแยะได้ว่าสิ่งไหนคือวงกลม สิ่งไหนคือสี่เหลี่ยม สี่เหลี่ยมจัตุรัส หรือสี่เหลี่ยมผืนผ้า จากตัวอย่างข้างต้น จะเห็นได้ว่าสิ่งต่างๆ ที่จะนำมาแยกแยะนั้น ต้องมีลักษณะเฉพาะที่บ่งบอกความเป็นตัวมันเอง และอาจจะมีลักษณะปลีกย่อยต่างกันได้ วิธีการในการทำ pattern recognition จะพยายามสอนให้คอมพิวเตอร์รู้ว่าลักษณะสำคัญนั้นคืออะไร เมื่อมีข้อมูลใหม่เข้ามาก็จะพิจารณาจากสิ่งที่เรียนรู้มา แล้วสามารถที่จะตัดสินใจได้ หลังจากสอนแล้วถ้าคอมพิวเตอร์จะสามารถที่จะตัดสินใจกับข้อมูลใหม่ๆที่เข้ามาได้ จะถือได้ว่า computer มี intelligence ดังนั้น ในขั้นตอนการทำ pattern recognition จึงจัดอยู่ในแขนงหนึ่งในวิชา Artificial Intelligence

### 2.7.1 ความสำคัญในการทำ pattern recognition และปัญหา

สิ่งที่ทำอยู่ในปัจจุบัน ส่วนใหญ่เป็นสิ่งที่คนมีความสามารถ Recognize ได้ดีกว่าคอมพิวเตอร์มากๆ เช่น การ recognize ลายมือ, คำพูด, ภาพ 2 และ 3 มิติ ของสิ่งของต่างๆ แต่ในงานที่มีความซ้ำซากบางประเภทการใช้คอมพิวเตอร์เข้ามาช่วยก็จะลดภาระงานและเวลาของคนได้เป็นอย่างมาก แต่มักจะพบปัญหาก็คือ ความแม่นยำในการ recognize ยังไม่สูงพอหรือว่ายังขาดความแม่นยำนั่นเอง ซึ่งในงานบางประเภทต้องการความแม่นยำสูงมากๆ เช่น Speaker Identification เป็นต้น

สาเหตุหลักที่ทำให้ขาดความแม่นยำ คือ คอมพิวเตอร์ตัดสินใจสิ่งต่างเป็นแบบ Logic 0 กับ 1 หรือว่า true กับ false เท่านั้น แต่ในสิ่งต่างๆ ที่ต้องการให้คอมพิวเตอร์ตัดสินใจนั้น ลักษณะของมันไม่ได้เหมือนกันทุกประการ อย่างเช่น สี่เหลี่ยมผืนผ้า สี่เหลี่ยมขนมเปียกปูน ซึ่งก็นับเป็นสี่เหลี่ยมเหมือนกัน เพราะฉะนั้น จึงต้องสอนให้คอมพิวเตอร์รู้จักก่อนว่า มีลักษณะที่เหมือนกันแค่ไหน แนวทางที่เรียกว่าสอนคอมพิวเตอร์ที่เป็นอยู่ในตอนนี้ จึงเป็นการหาสมการที่เหมาะสมสำหรับ pattern ต่างๆ ซึ่งก็ยังเป็นเรื่องที่ยากและซับซ้อนมาก

### 2.7.2 General Steps

Pattern Recognition ทุกชนิดแบ่งขั้นตอนการทำงานได้เป็น 2 ส่วน คือ

- ขั้นการสอน (train) เพื่อสร้างสมการ (model) ทำได้โดยใช้ training data มาสร้าง model สำหรับ pattern ทุก pattern ใน domain
- การนำ model ไปใช้ เป็นขั้นตอนที่คอมพิวเตอร์จะตัดสินใจข้อมูลใหม่ๆที่เข้ามาโดยใช้ model ที่ถูก train มาในการตัดสินใจ

ทั้งสองส่วนมีลำดับการทำงานเหมือนกัน คือ Pre-processing, recognition และ post-processing แต่ต่างกันในขั้นตอนของการ recognize

**2.7.2.1 Pre-processing**

เป็นการเตรียมข้อมูลเข้าให้อยู่ในรูปแบบที่เหมาะสมกับการ recognize ซึ่งรูปแบบจะเป็นอะไรก็ขึ้นอยู่กับ recognizer ที่ใช้ ขั้นตอนย่อยตามปกติที่มักจะทำในขั้นนี้มีดังนี้

- กำจัดข้อมูลรบกวนต่างๆ
- normalize ข้อมูลให้อยู่ใน scale มาตรฐาน
- หาลักษณะเด่นของข้อมูลที่จะใช้ในการ recognize (Feature Extraction) เป็นขั้นตอนที่สำคัญที่สุดของการทำ recognition โดย feature ที่ใช้จะส่งผลอย่างมากต่อ % ความถูกต้องที่ได้

**2.7.2.2 Recognition**

สำหรับการ Train model ขั้นนี้จะเป็นการนำ feature ของ training data ต่างๆที่ได้จากขั้นแรกมาสร้างเป็น model ไว้สำหรับเปรียบเทียบกับข้อมูลจริงส่วนการ recognize จะเป็นการนำข้อมูลมาเปรียบเทียบกับ model ที่มีอยู่ แต่ขั้นตอนปลีกย่อยจะแตกต่างกันไปสำหรับ recognizer แต่ละตัว

**2.7.2.3 Post-processing**

ขั้นตอนนี้ไม่จำเป็นต้องมีหาก Recognizer ทำงานได้ดีเพียงพอ ปกติแล้วในขั้นนี้จะเป็นการปรับปรุงผลที่ได้จาก recognizer โดยใช้ความรู้อื่นๆที่เกี่ยวข้องกับงานนั้นๆมาช่วย

**2.7.3 การค้นหารูปภาพ**

ในกระบวนการวิธีที่จะค้นหารูปภาพนั้นมีหลากหลายวิธี แต่มีวิธีพื้นฐานที่ใช้กันในช่วงแรกๆ คือจะใช้วิธีการค้นหาจากชื่อดัชนี (Index) ของรูปภาพ แต่วิธีการนี้จะต้องมีการกำหนดชื่อไว้ก่อนล่วงหน้าและจะต้องกำหนดไว้ในทุกๆ ภาพ เมื่อต้องการค้นหาก็จะต้องใส่ชื่อที่ต้องการและมีความถูกต้องในการค้นหา วิธีการนี้จะทำการค้นหาไปตามชื่อดัชนีของภาพตั้งแต่ภาพแรกไปเรื่อยๆ จนเจอภาพที่ต้องการเมื่อมีดัชนีตรงกับที่ต้องการค้นหา โดยชื่อดัชนีนี้จะบอกถึงรายละเอียดของภาพนั้นไว้ เช่น ภาพคอมพิวเตอร์ อาจจะมีชื่อดัชนีต่อไปนี้ประกอบ เช่น หน้าจอ คีย์บอร์ด เมาท์ เคส เป็นต้น หากค้นหาคำว่า “หน้าจอ” ก็จะสามารถค้นภาพเจอได้ เนื่องจากการทำชื่อดัชนีของภาพนั้นทำให้เสียเวลามากเพราะจะต้องทำในทุกๆ ภาพที่มีอยู่ และประสิทธิภาพของการค้นหาจะขึ้นอยู่กับว่าใช้คำค้นได้ถูกต้องกับชื่อดัชนีของภาพหรือไม่ ดังนั้นการค้นด้วยวิธีใช้ชื่อดัชนีนี้จึงยังไม่สามารถสร้างความพึงพอใจได้ จึงเกิดเทคนิคต่างๆของการค้นหารูปภาพ ได้แก่ การค้นหาโดยสี การค้นหาโดยลาย และการค้นหาโดยรูปร่าง เป็นต้น

ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร


เทคนิคการค้นหาโดยสี เป็นวิธีหนึ่งที่นิยมมากเช่นเดียวกัน เหมาะสำหรับการค้นหารูปภาพที่เป็นรูปถ่ายที่มีสีต่างๆ ประกอบกันอยู่ในภาพ โดยจะดูจากสีและตำแหน่งที่อยู่ของสี เช่น ถ้าต้องการค้นหารูปป้ายจราจร จะสามารถที่ค้นหารูปป้ายจราจรได้จากวัตถุที่มีสีเหลืองและมีรูปร่างเป็นสีเหลี่ยมได้เลย เป็นต้น

เทคนิคการค้นหาโดยลาย (Texture) ของรูปภาพ นิยมใช้กับภาพที่มีลายประกอบ เช่น ภาพของลายผ้า ซึ่งการค้นหาจะทำโดยการหาค่าพลังงาน (Energy) ของลายต้นแบบก่อน แล้วจึงไปหาค่าพลังงานของภาพต่างๆ แล้วนำค่าที่ได้มาเปรียบเทียบกัน เนื่องจากถ้ามีลายเดียวกัน ค่าพลังงานก็จะมีค่าเท่ากันหรือใกล้เคียงกัน

เทคนิคการค้นหาโดยดูจากรูปร่างของวัตถุในรูปภาพ มักจะใช้กับภาพที่ประกอบด้วยวัตถุบางอย่างที่มีรูปทรง เช่น ภาพกล่อง ภาพลูกบอล เป็นต้น ซึ่งการค้นหาจะต้องพิจารณาโครงร่างของภาพวัตถุต้นแบบไปเปรียบเทียบกับวัตถุปลายทาง ถ้าพบว่าวัตถุต้นแบบกับวัตถุปลายทางมีรูปร่างคล้ายคลึงกันก็จะสามารถสรุปได้ว่ามีวัตถุที่ต้องการอยู่ในรูปภาพนั้น ซึ่งเทคนิควิธีที่นิยมใช้กันก็คือเทคนิคการแปลงฮัฟ (Hough Transform) ซึ่งเป็นวิธีการใช้ในการจดจำวัตถุในรูปแบบต่างๆ เช่น เส้นตรง วงกลม วงรี โดยสมการที่ใช้ในการหาเส้นตรง (Straight Line Hough Transform) และสมการที่ใช้ในการหาเส้นโค้ง (Circular Hough Transform) แสดงดังสมการที่ (3) และ (4) ตามลำดับ

$$X \cos \theta + y \sin \theta = r \tag{3}$$

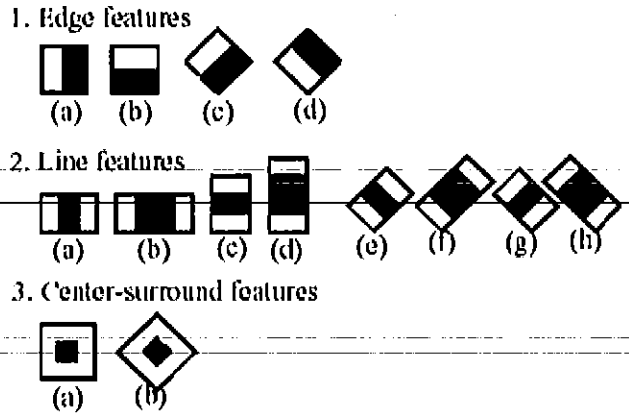
$$(x - c_x)^2 + (y - c_y)^2 = r^2 \tag{4}$$

เมื่อ $r$ คือระยะจากจุดศูนย์กลางถึงเส้นตรง และ $\theta$ คือมุมระหว่าง $r$ กับแนวแกน $x$ เมื่อกำหนดจุดของค่า $r$ และ $\theta$ สำหรับทุกๆ ค่า x และ y ที่เป็นไปได้ แล้วจะได้เป็นภาพในรูปของฮัฟสเปซ (Hough Space) โดยในฮัฟสเปซจะสามารถระบุค่าที่เป็นค่าสูงสุด (Local Maxima) ที่มีอยู่ได้ ซึ่งจุดนั้นจะเป็นจุดที่เส้นตรงอยู่นั่นเอง

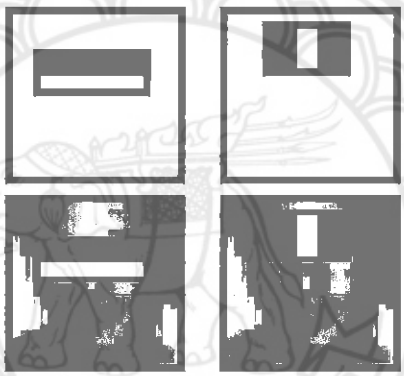
### 2.7.4 Haar like-feature

Haar like-Features ตามวิธีของ Viola และ Jones เป็นวิธีการตรวจจับและตีความวัตถุภายในภาพ ด้วยหลักการของ Haar Wavelet สำหรับสร้างรูปที่เหลี่ยม (Feature) โดยที่ภาพนี้แสดงถึงผลต่างระหว่างพื้นที่ส่วนสีขาว และส่วนที่เป็นสีดำ ซึ่ง Feature สามารถเปลี่ยนแปลงขนาด และตำแหน่งได้ ใช้สำหรับการตรวจจับลักษณะบนภาพแบบต่าง เช่น เส้นตรง, วงกลม เป็นต้น

เ574386ร
วร.
ก2815
2553

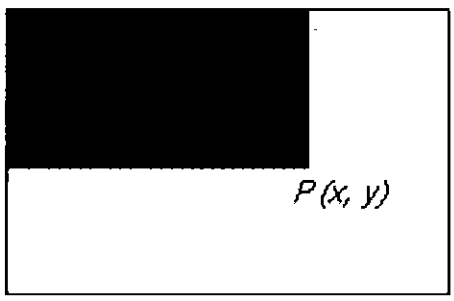



**รูปที่ 2.21** รูปแบบของ Features สำหรับการตรวจจับลักษณะแบบต่างๆ [29]



**รูปที่ 2.22** ตัวอย่างการใช้ Feature ตรวจจับลักษณะต่างๆ

การคำนวณค่าของ Feature นั้น ใช้หลักการคำนวณแบบ Integral image ซึ่ง Integral image คือผลรวมของค่าในทุกๆ พิกเซล ที่ตำแหน่ง (x, y) ใดๆ ซึ่งมีเวลาการทำงานเป็น O (1) ทำให้การคำนวณ Feature นั้นทำได้เร็วมาก



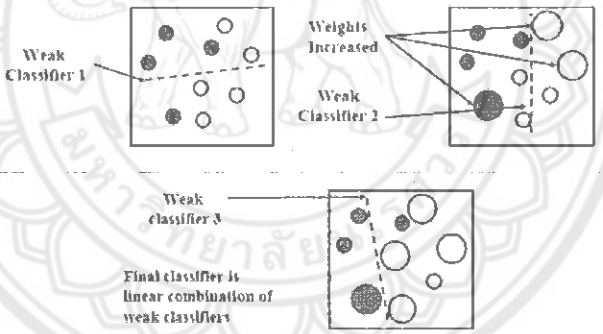

$$P(x, y) = \sum_{x' \le x, y' \le y} i(x', y')$$

**รูปที่ 2.23** การคำนวณแบบ Integral image

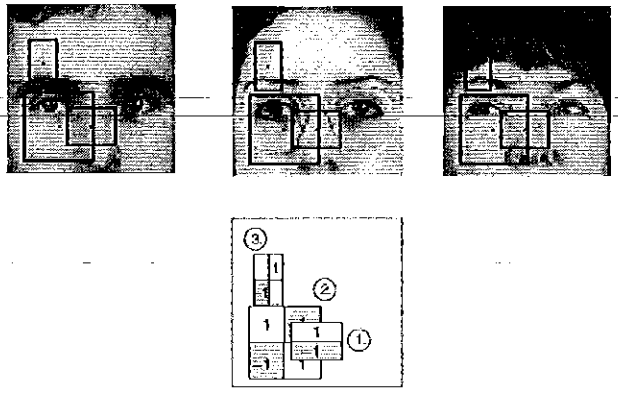
ในการทำ Haar like-Feature นั้น จำเป็นต้องมีภาพตัวอย่างจำนวนมาก ซึ่งใช้ในการคัดเลือกลักษณะของรูปที่ต้องการตรวจจับและตีความหมาย ซึ่งมีสองลักษณะคือ Positive Image หรือรูปที่มี Object นั้นๆประกอบอยู่ภายในภาพ และ Negative Image หรือภาพใดๆที่ไม่มี Object ที่เราต้องการอยู่ภายในภาพ

Haar like-Feature ใช้หลักการของ **AdaBoost (Adaptive Boost)** ซึ่งเป็นกระบวนการหา Feature ที่มีลักษณะใกล้เคียง และแตกต่างกับภาพนำเข้า สำหรับการจัดประเภทของภาพ โดยการถ่วงน้ำหนักให้ส่วนต่างๆภายในภาพ บนภาพ Positive และภาพ Negative เพื่อใช้หาลักษณะของ Object ที่ "ใช่" และ "ไม่ใช่" ในลักษณะต่างๆ มีกระบวนการดังนี้

- เริ่มแรกกำหนด ค่าน้ำหนักให้กับ Feature ที่วิ่งหาภายในภาพตัวอย่าง
- หาบริเวณที่ประกอบด้วย ส่วนที่เราต้องการ
- เพิ่มค่า น้ำหนักให้กับส่วนที่เหลือ เฉพาะลักษณะที่เราต้องการ ที่ยังไม่ได้ แบ่งลักษณะไว้
- ทำวนเช่นนี้ซ้ำไปเรื่อยๆ จนสุดท้าย นำบริเวณที่ได้ทั้งหมดมารวมกัน จะได้บริเวณของ Object ที่เราต้องการหา และลักษณะในส่วนต่างๆภายใน Object นั้น

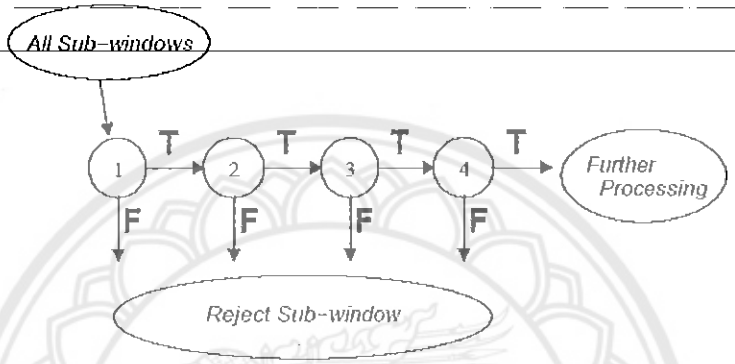



**รูปที่ 2.24** การทำงานของ Adaboost [15]

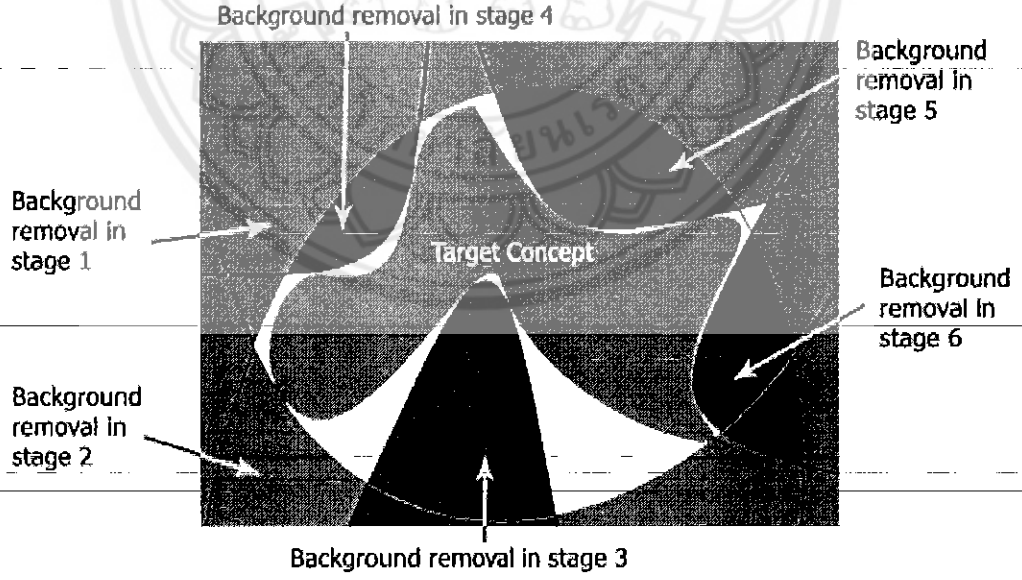



**รูปที่ 2.25** ผลลัพธ์จากการทำกระบวนการ AdaBoost

**Cascade Classifiers** เป็นกระบวนการตีความหมายของภาพ โดยการแบ่งประเภทของภาพตามลักษณะภายในภาพ โดยเริ่มต้นจากการตัดส่วนของ Sub window ที่เป็น Negative ออกไปก่อน แล้วจากนั้นค่อยใช้ ส่วนที่เป็น Positive วิ่งวนภายในภาพ หากไม่เจอลักษณะที่ตรงกัน ก็จะเปลี่ยนลักษณะการตรวจจับภายใน Sub window หากเจอลักษณะที่ตรงกัน ก็จะเปลี่ยนลักษณะในการตรวจจับ ทำเช่นนี้จนครบ จะได้รูปที่สามารถบอกได้ว่าภาพดังกล่าวเป็นภาพอะไรจากลักษณะต่างๆ ภายในภาพ



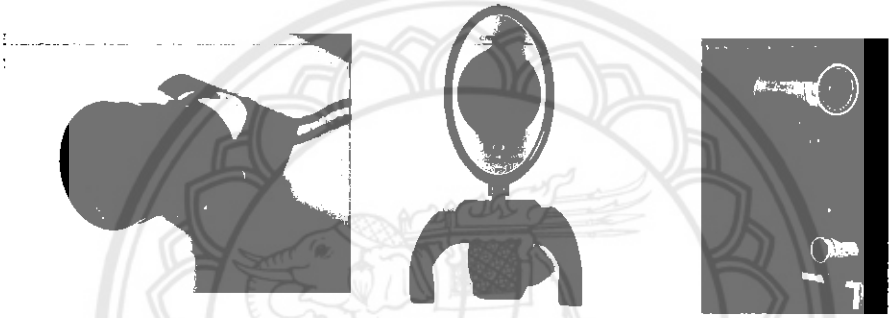

รูปที่ 2.26 การทำงานของ Haar Cascade Classifier




**รูปที่ 2.27** ลักษณะการทำงานของ Cascade Classifier

## 2.8 กล้อง และ การเชื่อมต่อกล้อง Web Camera

เว็บแคม (Webcam) หรือ ชื่อเรียกเต็มๆว่า Web Camera แต่ในบ้างครั้งก็มีคนเรียกว่า Video Camera หรือ Video Conference เป็นอุปกรณ์อินพุตที่สามารถจับภาพเคลื่อนไหวและสามารถส่งผ่านระบบเครือข่าย เว็บไซต์ โปรแกรม ถือเป็นอุปกรณ์นำข้อมูล (อินพุต) ที่สามารถจับภาพเคลื่อนไหวให้ไปปรากฏในจอภาพ และสามารถส่งภาพเคลื่อนไหวหรือภาพนิ่งนี้ไปให้คนอีกฟากหนึ่งเห็นตัวเราเคลื่อนไหวได้เหมือนอยู่ต่อหน้า ปัจจุบันมีทั้งแบบที่เชื่อมต่อกับคอมพิวเตอร์ผ่านสายยูเอสบี และเชื่อมต่อแบบไร้สาย โดยตัวอย่างของกล้อง Web Camera แสดงในรูปที่ 2.28



รูปที่ 2.28 ภาพแสดงกล้อง Web Cam

### 2.8.1 ประเภทของเว็บแคม [16]

อุปกรณ์อย่างกล้องเว็บแคมไม่ใช่ว่าจะเหมือนกันหมดทุกตัวแต่ละรุ่นแต่ละยี่ห้อจะมีลักษณะและคุณสมบัติที่แตกต่างกันไปตามแต่ผู้ผลิตจะคิดค้นและออกแบบมาให้เหมาะกับการใช้งานอย่างไรซึ่งสามารถแบบประเภทของเว็บแคมได้ดังนี้

#### 2.8.1.1 แบ่งตามรูปทรงของกล้อง

โดยปกติกล้องเว็บแคมส่วนใหญ่จะเป็นทรงกลม เนื่องจากเป็นรูปทรงต้นแบบที่ทำกันมานานและก็ทำให้รู้ได้ทันทีว่านี้คืออุปกรณ์ เว็บแคม แต่ไม่จำเป็นที่กล้องเว็บแคมต้องเป็นทรงกลมเสมอไปเพราะบางครั้งกล้องเว็บแคมก็จำเป็นต้องมีรูปทรงอื่นๆ เพื่อให้เข้ากับการใช้งานในบางลักษณะดังนั้น การเลือกรูปทรงให้เหมาะสมนั้นก็จะขึ้นอยู่กับลักษณะการใช้งานของเรามากกว่า

**2.8.1.2 แบ่งตามประเภทของขาตั้งกล้อง**

โดยส่วนใหญ่ลักษณะของฐานตั้งกล้องจะเป็นแบบตั้งพื้นเสียส่วนใหญ่ โดยแบบแรก คือแบบมีขาสำหรับวางบนพื้น อาจจะมีขา 3 ขา หรือ 4 ขา ก็แล้วแต่การออกแบบ แต่ฐานแบบ3ขาจะมีปัญหาตรงที่วางแล้วยังไม่มั่นคงดีนักละไม่สามารถหมุนตัวกล้องได้สะดวกนัก ดังนั้นถ้าต้องการเว็บแคมที่มีฐานมั่นคงและสามารถหมุนได้ง่ายๆ ก็ต้องเลือกแบบฐานทรงกลมขนาดใหญ่ซึ่งแบบนี้จะมีข้อดีตรงที่วางได้มั่นคงและยังสามารถหมุนแกนของตัวกล้องได้

**2.8.1.3 แบ่งตามชนิดของเซ็นเซอร์**

สำหรับเซ็นเซอร์ที่กล้องเว็บแคมใช้นั้นจะมีหลักๆอยู่ 2 ชนิด คือ CCD และ CMOS แต่ที่นิยมใช้กันมากที่สุดในตอนนี้ก็คือ CMOS เนื่องจากเหตุผลหลายๆประการและตัวเซ็นเซอร์ แบบ CMOS เองก็สามารถแบบออกได้ถึง2ชนิดด้วยกันคือ CLF Color CMOS Censor ที่มีความละเอียดของพิกเซลแค่110,000พิกเซล(367x291)เท่านั้นในขณะที่ VGA Color CMOS Censor ให้ความละเอียดที่สูงกว่าที่350,000พิกเซล(655x493)ดังนั้น เวลาเลือกซื้อกล้องเว็บแคมก็ดูได้ทั้งความละเอียดที่ระบุไว้หรือชนิดของCMOSก็ได้ครับสำหรับเซ็นเซอร์แบบ จะเป็นเซ็นเซอร์ที่นิยมใช้ในกล้องดิจิตอลเพราะให้ความละเอียดที่สูงกว่าและก็มี noiseไม่มากเท่าแบบCMOS

**1.8.1.4 แบ่งตามรูปแบบการเชื่อมต่อ**

สำหรับการเชื่อมต่อของกล้องเว็บแคมในปัจจุบันส่วนใหญ่ จะเป็นอินเทอร์เฟซแบบ USB แทบทั้งสิ้นโดย USBที่ใช้ก็จะเป็นเวอร์ชัน1.1เสียส่วนมากแต่ก็จะมีเวอร์ชัน 2.0ในบางรุ่นกล้องเว็บแคมไร้สายจะใช้การเชื่อมต่อในแบบ WiFi หรือ Wireless LAN นั่นเองทำให้สามารถเคลื่อนย้ายไปได้ทุกที่ แต่เว็บแคมที่เป็น Wireless ตอนนี้ค่อนข้างหายาก ยังมีราคาแพงอยู่

**2.8.2 การเลือกซื้อกล้องเว็บแคม**

ขั้นตอนแรกเราต้องรู้ว่าจะนำกล้องเว็บแคมมาใช้งานกับเครื่องคอมพิวเตอร์ประเภทใดถ้าเป็นโน้ตบุ๊กก็ต้องเป็นกล้องเว็บแคมขนาดเล็กกะทัดรัด และสามารถติดตั้งบนจอแอลซีดีของโน้ตบุ๊กได้แต่ถ้าใช้กับเครื่องคอมพิวเตอร์เดสก์ทอปก็ แนะนำรุ่นที่มีขาตั้งที่มั่นคงสามารถวางบนจอมอนิเตอร์ เมื่อเลือกรูปแบบของกล้องได้แล้ว ก็มาเลือกตามคุณสมบัติภายในของกล้องเว็บแคมโดยเลือกจากชนิดของเซ็นเซอร์ที่ใช้กับภาพ โดยจะมีให้เลือกเป็น CMOS ในแบบ CIF และ VGA ซึ่งแนะนำว่าเป็นแบบVGAจะให้ความละเอียดที่สูงกว่า หรือถ้าต้องการความละเอียดที่มากกว่านี้ ก็เลือกเซ็นเซอร์แบบ CCD จะดีกว่าแต่ทั้งนี้ราคาก็จะเพิ่มสูงขึ้น ตามชนิดของเซ็นเซอร์และความละเอียดของตัวกล้องเว็บแคม

## 2.9 ระบบวิดีโอ

**2.9.1 วิดีโอ** คือการเรียงต่อกันของภาพเป็นเรื่องราว แบ่งออกได้เป็น 2 ประเภทใหญ่ ๆ ได้แก่ วิดีโอแบบอนาล็อก (Analog Video) และแบบดิจิตอล (Digital Video) ซึ่งวิดีโอแบบอนาล็อกนี้จะเก็บข้อมูลภาพและเสียงในรูปแบบของสัญญาณไฟฟ้า มีลักษณะเป็นคลื่นและเก็บข้อมูลลงในม้วนเทป วิดีโอในระบบอนาล็อกจะมีความไวต่อการผิดเพี้ยนสูง เมื่อมีการบันทึกต่อกันหลาย ๆ ครั้ง จะทำให้คุณภาพของภาพและเสียงลดลง สำหรับวิดีโอแบบดิจิตอลนั้นจะเก็บอยู่ในรูปของไฟล์คอมพิวเตอร์ที่สามารถนำกลับมาใช้ใหม่ได้ ทำสำเนาได้ และสามารถปรับแต่งแก้ไขได้เนื้อหาภายในวิดีโอ แสดงดังภาพที่ 2-1 มีการแบ่งออกเป็นส่วนต่าง ๆ ดังนี้

2.9.1.1 เฟรม คือภาพนิ่งแต่ละภาพที่เรียงต่อกันจนเป็นเรื่องราว

2.9.1.2 ช็อต คือกลุ่มของเฟรมที่เรียงกันอย่างต่อเนื่อง โดยที่มีเนื้อหาเดียวกันอย่างต่อเนื่อง

2.9.1.3 ฉาก คือช็อตหลาย ๆ ช็อตที่นำมาเรียงต่อเนื่องกัน

ดังนั้นในวิดีโอหนึ่งสามารถประกอบไปด้วยฉากหลาย ๆ ฉาก และในแต่ละฉากก็จะประกอบไปด้วยช็อตอย่างน้อย 1 ช็อต

**2.9.2 มาตรฐานการแพร่ภาพของวิดีโอ** แบ่งออกเป็น 3 รูปแบบดังนี้

2.9.2.1 ระบบ PAL (Phase Alternate Line) เป็นรูปแบบวิดีโอที่มีความละเอียดในแนวนอนเท่ากับ 625 เส้น ใช้อัตราการรีเฟรช (Frame Rate) เท่ากับ 25 เฟรม ต่อ 1วินาที วิดีโอระบบนี้จะใช้ในประเทศออสเตรเลีย เบลเยียม จีน ฟินแลนด์ เยอรมัน ฮ่องกง อิตาลี สิงคโปร์ มาเลเซีย และไทย เป็นต้น

2.9.2.2 ระบบ NTSC (National Television System Committee) เป็นรูปแบบวิดีโอที่มีความละเอียดในแนวนอนเท่ากับ 525 เส้น ใช้อัตราการรีเฟรช เท่ากับ 29.97 เฟรม ต่อ 1 วินาที วิดีโอระบบนี้จะใช้ในประเทศสหรัฐอเมริกา ญี่ปุ่น แคนาดา โคลัมเบีย เม็กซิโก ไต้หวัน ฟิลิปปินส์ เป็นต้น

2.9.2.3 ระบบ SECAM (Sequential Color and Memory) เป็นรูปแบบวิดีโอที่มีความละเอียดในแนวนอนเท่ากับ 816 เส้น ใช้อัตรารีเฟรช เท่ากับ 25 เฟรม ต่อ 1 วินาที วิดีโอระบบนี้จะใช้ในประเทศบัลแกเรีย ฝรั่งเศส อิรัก อิหร่าน รัสเซีย โปแลนด์ เป็นต้น

2.9.2.4 ระบบ HDTV (High Definition Television) เป็นเทคโนโลยีของการแพร่ภาพที่มีความละเอียดสูง คือ 1280 x 720 พิกเซล

**2.9.3 คุณภาพของวิดีโอ** การวัดคุณภาพของวิดีโอสามารถวัดได้จาก อัตราเฟรม และความละเอียดของภาพ

2.9.3.1 อัตราเฟรม คืออัตราความถี่ในการแสดงภาพจาก Timeline ออกทางหน้าจอภาพ อัตราถูกแสดงในวิดีโอมีหน่วยเป็นเฟรมต่อวินาที (fps: Frame per second) เป็นหน่วยวัดปริมาณข้อมูลที่ใช้ในการเก็บบันทึกและแสดงวิดีโอ) เช่น อัตราเฟรมของภาพยนตร์เท่ากับ 24 เฟรม 6 ต่อ 1 วินาที อัตราเฟรมระบบ PAL เท่ากับ 25 เฟรม ต่อ 1 วินาที และอัตราเฟรมของ NTSC เท่ากับ 30 เฟรมต่อ 1 วินาที

2.9.3.2 ความละเอียด หมายถึง ความชัดของภาพที่แสดงผลออกมาทางจอภาพ ความละเอียดของจอภาพขึ้นอยู่กับจำนวนพิกเซลทั้งหมดที่เกิดบนจอภาพ เช่น ความละเอียดของจอภาพ 640 x 480 พิกเซล หมายถึง มีจำนวนพิกเซลแสดงผลเรียงกันอยู่บนจอภาพในแนวนอน 640 พิกเซล และแนวตั้ง 480 พิกเซล

**2.9.4 AVI (Audio Video Interleave)** เป็นแฟ้มที่ประกอบด้วยวิดีโอ และเสียง ในรูปแบบของมัลติมีเดีย เริ่มต้นโดยบริษัทไมโครซอฟต์ในเดือนพฤศจิกายน ปี 1992 เป็นเทคโนโลยีสำหรับระบบปฏิบัติการของวินโดว์ (Windows) แฟ้ม AVI เป็นรูปแบบของวิดีโอและเสียงที่ต่อเนื่องกัน และแฟ้ม AVI พัฒนาโดย Matrox และ OpenDML ในเดือนกุมภาพันธ์ ปี 1996 และได้รับการสนับสนุนโดยบริษัทไมโครซอฟต์ ถูกเรียกว่า "AVI 2.0"

## 2.10 OpenCV (Open Source Computer Vision Library)

OpenCV เป็นไลบรารีสำหรับใช้งานเรื่องการประมวลผลภาพ (Image Processing) และคอมพิวเตอร์วิทัศน์ (Computer vision) ซึ่งความสามารถของ OpenCV ได้แก่ การทำภาพเบลอ, การหา threshold, การหา Histogram ของภาพ เป็นต้น แต่ความสามารถโดยส่วนใหญ่ มักใช้ค้นหาขอบของภาพ การตรวจสอบการเคลื่อนไหว และการแบ่งภาพออกเป็นส่วน (Image segmentation)

นอกจากนี้ OpenCV สามารถจัดการกับข้อมูลแบบวิดีโอได้ด้วย เนื่องจาก OpenCV เป็นชุดคำสั่งที่ไม่ได้เป็นตัวโปรแกรม เมื่อต้องการเรียกใช้งานจึงต้องเขียนโปรแกรมเพื่อเรียกชุดคำสั่งเหล่านั้น ซึ่งภาษาที่นิยมใช้ ได้แก่ ภาษา C, ภาษา C++ และภาษา Python ซึ่ง OpenCV จะประกอบด้วยสองส่วน คือ ส่วนโครงสร้างข้อมูล (Data Structure) ที่ใช้ในการเก็บข้อมูลต่าง ๆ อาทิ เช่น รูปภาพ เมตริกซ์และพิกัด อีกส่วนหนึ่ง คือขั้นตอนวิธี (Algorithm) ซึ่งจะใช้ในการประมวลผลต่าง ๆ โดยเฉพาะการประมวลทางรูปภาพ สำหรับใน OpenCV จะประกอบด้วยไลบรารีอยู่ 4 ส่วน ได้แก่

1. CXCORE

เป็นฟังก์ชันเบื้องต้นที่ใช้จัดการเกี่ยวกับจุด ขนาด อาร์เรย์ หน่วยความจำคำสั่งในการวาดภาพการประกาศตัวแปรภาพเป็นต้น ตัวอย่างคำสั่งในการประกาศรูปภาพ คือ IplImage, CvMat, CvMatND

2. CV

ใช้ในการประมวลผลและการวิเคราะห์รูปภาพ ฟังก์ชันส่วนใหญ่จะทำงานกับจุดภาพที่เป็นอาร์เรย์สองมิติหรือที่เรียกว่าภาพนั้นเอง เช่น การหาขอบหรือมุม การทำฮิสโตแกรม (Histogram) เป็นต้น

3. Machine Learning

เป็นไลบรารีที่รวมคลาสและฟังก์ชันทางสถิติ (Statistical) การแยกคลาสและการแบ่งกลุ่มข้อมูล (Clustering)

4. HighGUI

เป็นไลบรารีที่ใช้ในการดึงภาพ การบันทึกภาพ การติดต่อกับกล้องวีดีโอ (VDO) การสร้างหน้าต่างเพื่อแสดงภาพและทำลายภาพ การเปลี่ยนขนาดและเคลื่อนย้ายหน้าต่าง รวมไปถึงการตรวจสอบเมาส์(Mouse) และแป้นพิมพ์

จากบทที่ 2 ทำให้เราทราบว่าระบบสีตามธรรมชาติ มี 3แกน R G B ที่ทำให้เกิดสี่ต่างๆ ยังมีอัลกอริทึมที่เกี่ยวกับ การหาขอบของภาพ histogram ยังทราบถึง การทำงาน ของ Haar like - Features ซึ่งเป็นวิธีการตรวจจับและตีความวัตถุภายในภาพ ด้วยหลักการของ Haar Wavelet ซึ่งเนื้อหาเหล่านี้เป็นองค์ประกอบที่ใช้ ในการการวิเคราะห์ ออกแบบ และการพัฒนาระบบ ในบทที่ 3 ต่อไป

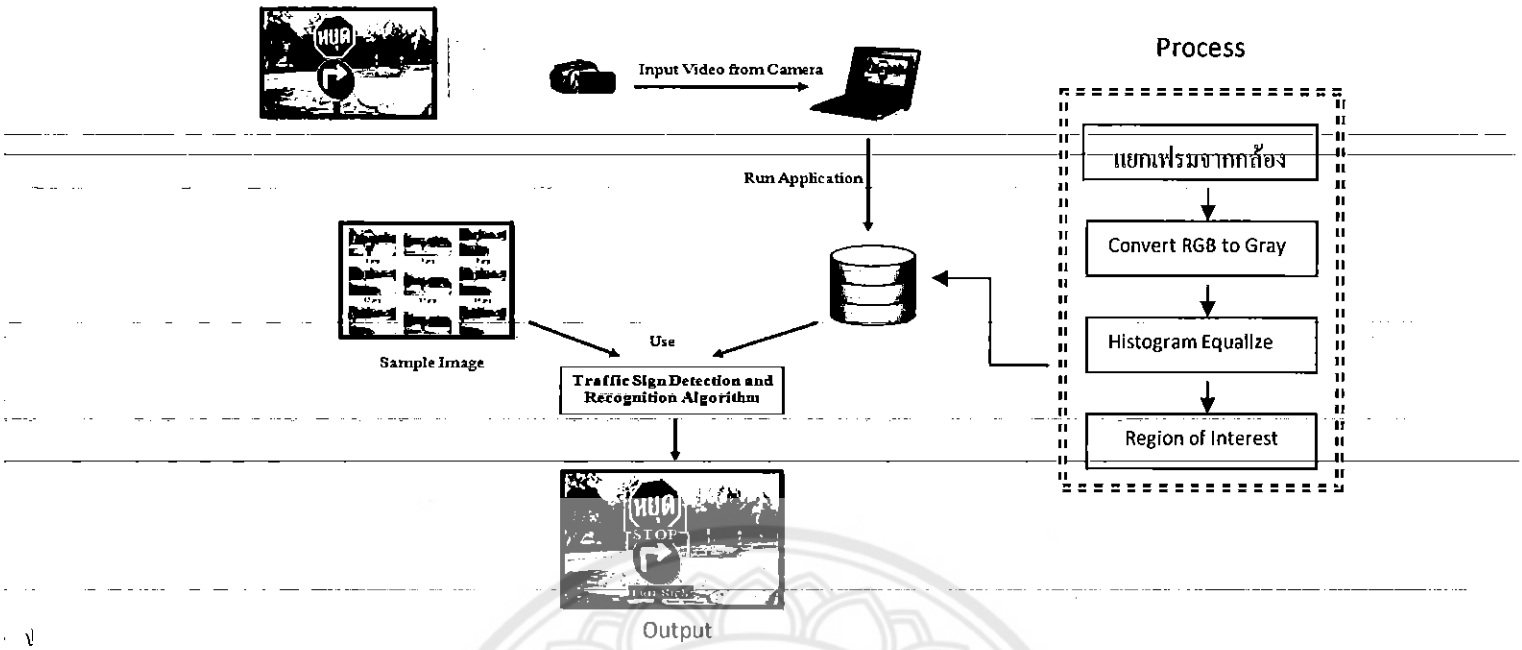
# บทที่ 3

# การวิเคราะห์ ออกแบบ และการพัฒนาระบบ

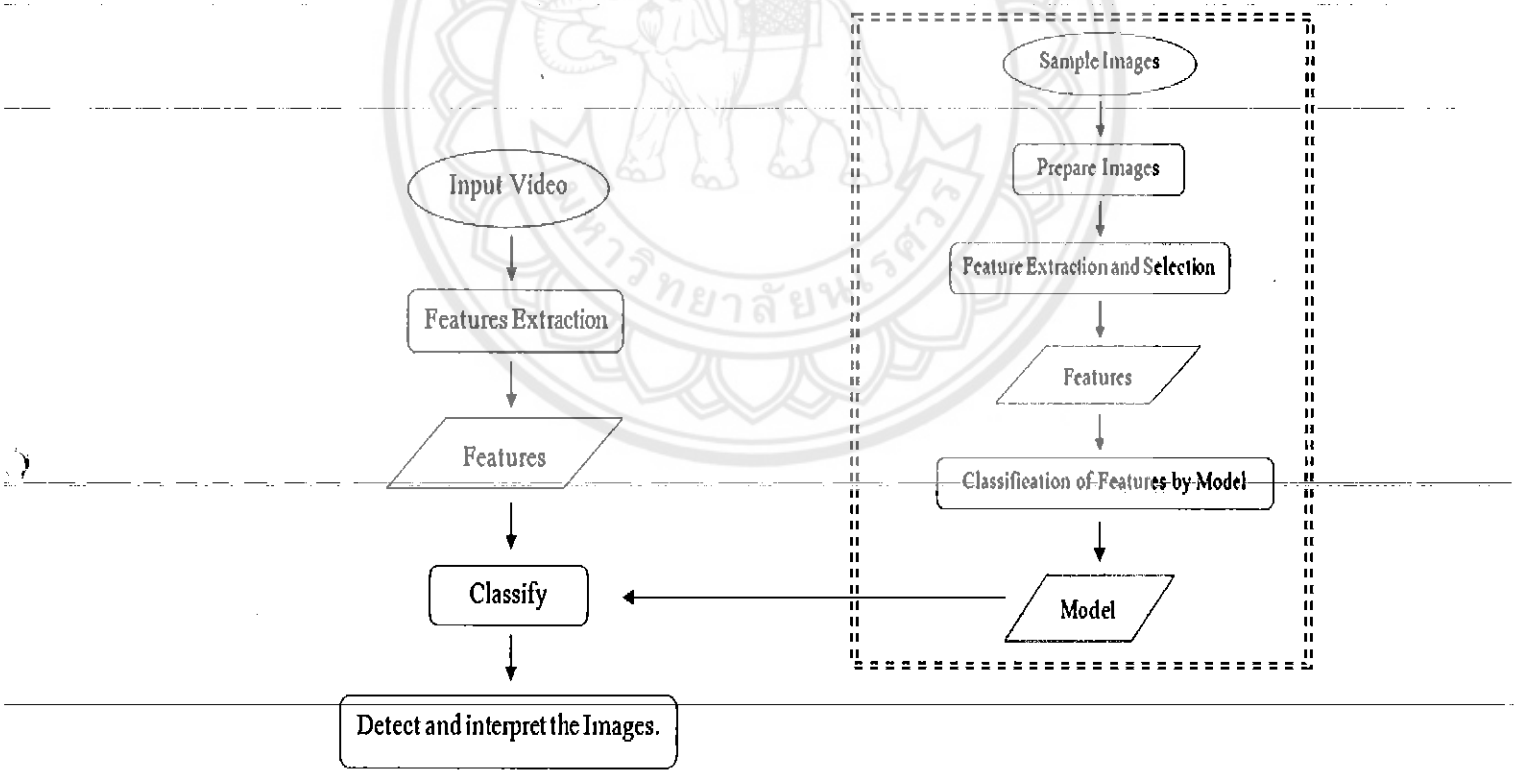
จากการศึกษาค้นคว้าข้อมูลและทฤษฎีข้างต้น ทำให้สามารถเข้าใจถึงหลักการและกระบวนวิธีการในการดำเนินงานนี้ ในบทนี้จะกล่าวถึงโครงสร้างของการตรวจจับสัญญาณจราจรและป้ายจราจรโดยนำความรู้ที่ได้จากทฤษฎีข้างต้นมาประยุกต์ใช้ในการพัฒนาระบบความคุมรถอัจฉริยะไร้คนขับ การออกแบบระบบโดยรวมนั้น ใช้การดึงภาพจากวีดีโอ มาใช้เป็นภาพนำเข้าสู่ระบบ โดยใช้วิธี Haar like-feature ในการตรวจจับ และตีความหมายของภาพ ซึ่งมีรายละเอียดการออกแบบดังต่อไปนี้

## 3.1 การออกแบบระบบ

การออกแบบระบบ เริ่มจากการเก็บภาพตัวอย่างของป้ายจราจรประเภทบังคับและสัญญาณไฟจราจรที่จะใช้เป็นภาพตัวอย่างในการทดสอบโปรแกรมเบื้องต้น ซึ่งก็คือป้ายห้ามจอด ป้ายหยุด ป้ายเลี้ยวซ้าย ป้ายเลี้ยวขวา สัญญาณไฟแดงและสัญญาณไฟเขียว โดยกล้องที่ใช้ในการถ่ายภาพจะสามารถจับป้ายได้อย่างชัดเจนที่ระยะประมาณ 5 - 20 เมตร แล้วนำภาพวิดีโอที่ถ่ายได้มาทำการประมวลผลเพื่อหาตำแหน่ง แล้วแปลความหมายของป้ายสัญญาณจราจรและสัญญาณไฟจราจร โดยมีการออกแบบการทำงานของโปรแกรมตรวจจับและแปลความหมายป้ายและสัญญาณไฟไว้ดังรูปที่ 3.1

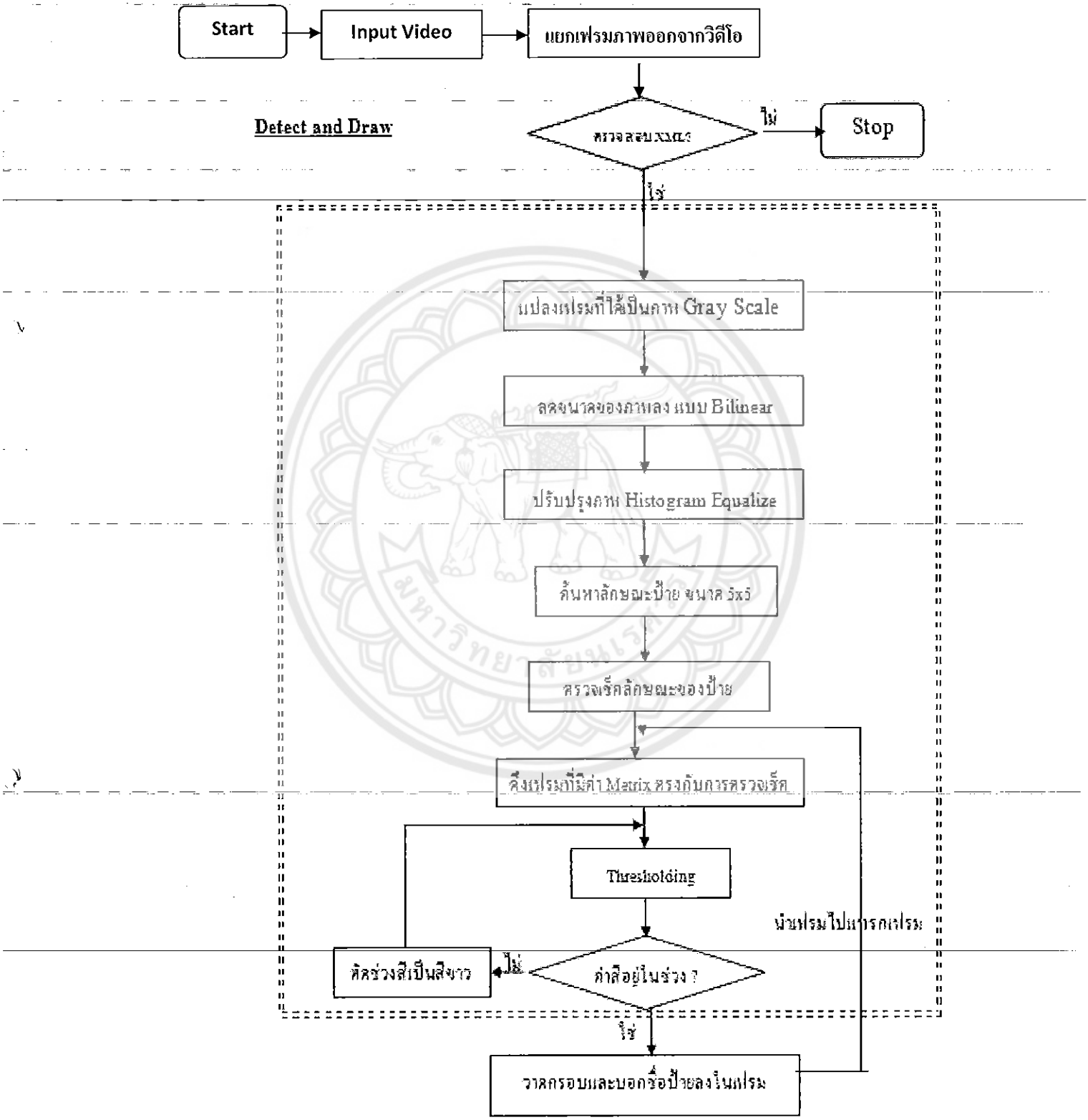



**รูปที่ 3.1** การทำงานของโปรแกรมตรวจจับและแปลความหมายป้ายและสัญญาณไฟ




**รูปที่ 3.2** แผนภาพวิธีการตรวจจับ และแปลความ

จากรูปที่ 3.1 และ3.2 สามารถนำมาวิเคราะห์การทำงานออกมาเป็นแผนผังการทำงานอย่างละเอียด ตามขั้นตอนการประมวลผลภาพ ได้ดังรูป 3.3




**รูปที่ 3.3** แผนผังการทำงานของโปรแกรม

## 3.2 ขั้นตอนในการตรวจหาป้าย และสัญญาณไฟจราจร

จากแผนผังในรูปที่ 3.3 ขั้นตอนวิธี ที่ใช้ในการตรวจหาป้าย และสัญญาณไฟจราจร ทำการทดลองบนโปรแกรมภาษา C/C++ และใช้คำสั่งในการประมวลผลภาพจาก OpenCV library โดยมีขั้นตอนการทำงานดังนี้

**3.2.1 ขั้นตอนวิธีการตรวจหาป้าย และสัญญาณไฟจราจร** ด้วย Haar Detection การตรวจหาป้าย และสัญญาณไฟจราจร ด้วย Haar-like Feature เป็นวิธีทางสถิติ สามารถแบ่งแยกป้ายและสัญญาณไฟจราจรได้ โดยการเปรียบเทียบลักษณะของ Haar-like Feature จากฐานข้อมูล ที่ผ่านการฝึกให้จดจำลักษณะ Haar-like Feature ของป้ายสัญญาณไฟจราจร และจดจำลักษณะ Haar-like Feature ของสิ่งที่ไม่ใช่ป้ายหรือสัญญาณไฟจราจรไว้ โดยมีขั้นตอนการทำงานดังนี้

3.2.1.1 ภาพที่รับเข้ามาเป็นได้ทั้งภาพระดับสีเทาและภาพสี แล้วแปลงภาพที่รับเข้ามา เป็นภาพระดับสีเทา

3.2.1.2 นำภาพสีเทา ปรับปรุงภาพด้วยวิธี Histogram Equalization เพื่อกระจายให้แต่ละระดับสีเทามีจำนวน Pixel ใกล้เคียงกัน ทำให้เกิดการกระจายของฮิสโตแกรม ทำให้ภาพที่ได้มีความคมชัดที่มากขึ้น ทำให้ระดับความสว่างไม่ค่อยมีผลต่อระบบการตรวจจับ

3.2.1.3 นำภาพที่ปรับปรุงแล้ว ตรวจหาป้ายและสัญญาณไฟจราจร โดยทำการเลือกตีกรอบของพื้นที่ในภาพ หาลักษณะ Haar-like Feature เปรียบเทียบกับลักษณะ Haar-like Feature จากฐานข้อมูลทั้งตัวอย่างที่ถูกและตัวอย่างที่ผิด ถ้าลักษณะ Haar-like Featue มีลักษณะของตัวอย่างที่ผิดให้ตัดสินว่าไม่มีป้าย และสัญญาณไฟจราจรอยู่ในกรอบที่ถูกก็พิจารณาทันที โดยทำการเปรียบเทียบกับฐานข้อมูลไปเรื่อยๆ จนหมดฐานข้อมูล แล้วจึงตัดสินว่ากรอบที่พิจารณาอยู่นั้นมีป้ายหรือสัญญาณไฟจราจรที่สนใจอยู่หรือไม่ กรอบที่ใช้ในการพิจารณานี้สามารถที่จะเปลี่ยนขนาดในการตรวจหาเพื่อให้ตรวจหาป้ายและสัญญาณไฟจราจรได้ในหลายขนาดตามขนาดของกรอบขณะเปลี่ยนกรอบตรวจหานี้มีการทำงานพร้อมกันแบบขนาน เพื่อความรวดเร็วในการตรวจหาผลลัพธ์ที่ได้จาก Haar Detection คือ ตำแหน่งจุดศูนย์กลางของวัตถุที่ถูกพิจารณาว่าเป็นป้ายหรือสัญญาณไฟจราจรที่ต้องการหา ถึงขนาดของกรอบของพื้นที่ในการตรวจหาแล้วจึงตีกรอบล้อมรอบตำแหน่งที่พบ บนภาพที่รับเข้ามา ด้วยเส้นผ่านศูนย์กลางที่เท่ากับขนาดของกรอบพื้นที่ของวัตถุที่ตรวจพบ

3.2.2 ขั้นตอนวิธีการตรวจหาสีร่วมกับ Haar Detection เพื่อตรวจหาป้ายและสัญญาณไฟจราจร

การตรวจหาป้าย และสัญญาณไฟจราจรโดยใช้การตรวจหาสีป้ายร่วมกับ Haar Detection ทำโดยการนำผลลัพธ์ที่ได้จากการตรวจหาป้าย และสัญญาณไฟจราจรด้วย Haar Detection ซึ่งก็คือภาพที่รับเข้ามาโดยมีการทำกรอบสี่เหลี่ยมล้อมบริเวณต่างๆ ที่ได้รับการพิจารณาจาก Haar Detection แล้วว่าเป็นป้าย และสัญญาณไฟจราจรอะไร โดยในขั้นตอนวิธีนี้จะทำการพิจารณากรอบสี่เหลี่ยมที่ตรวจหาได้ แต่ละกรอบสี่เหลี่ยม โดยนำมาพิจารณาร่วมกับภาพสีของป้ายและสัญญาณไฟจราจร ซึ่งได้จากการนำภาพที่รับเข้ามาเลือกช่วงค่าสีของป้ายหรือสัญญาณไฟจราจร โดยค่าสามารถเลือกได้จากการทดลอง ซึ่งค่าจะเป็นโมเดลสี RGB ประกอบด้วย 3 องค์ประกอบ คือ สีแดง (Red) สีเขียว (Green) และสีน้ำเงิน (Blue) การหาช่วงของสีป้ายและสัญญาณไฟจราจรนั้น จึงต้องใช้ทั้งสามองค์ประกอบ โดยทำการทดลองเพื่อหาช่วงที่เหมาะสม โดยมีขั้นตอนรายละเอียดดังต่อไปนี้

3.2.2.1 นำภาพที่รับเข้ามา (เป็นภาพสีเท่านั้น) ที่ได้ทำการตรวจหาป้ายจราจรและสัญญาณจราจร ด้วย Haar Detection ตามข้อ 3.2.1 ซึ่งเป็นภาพที่ทำกรอบสี่เหลี่ยมล้อมรอบวัตถุที่พิจารณาแล้วว่ามีลักษณะเหมือนป้าย และสัญญาณไฟจราจร

3.2.2.2 นำภาพที่รับมาเลือกช่วงค่าสีที่ได้จากโมเดลสี RGB ดังที่กล่าวมาแล้วข้างต้น

3.2.2.3 พิจารณาทีละกรอบที่ได้จากข้อ 3.2.2.1 และทำการเติมสีในพื้นที่ตรวจจับที่ได้จาก Haar detection แสดงด้วยพื้นที่สีขาว มีค่าเป็น 1 ส่วนพื้นที่สีดำที่อยู่ภายนอกกรอบการตรวจจับให้มีค่าเป็น 0

3.2.2.4 นำภาพที่ได้ทีละกรอบสี่เหลี่ยมจากข้อ 3.2.2.3 มาทำตรรกะ AND กับภาพที่ได้จากข้อ 3.2.2.2 จะเหลือภาพสีเฉพาะตำแหน่งที่ตรงกับสี่เหลี่ยมสีขาว แสดงถึงพื้นที่สีเฉพาะส่วนที่ตรวจจับได้จาก Haar Detection

3.2.2.5 คำนวณพื้นที่ภาพจากข้อ 3.2.2.4 เทียบกับพื้นที่สีขาวในข้อ 3.2.2.3 ที่เรียกว่าค่า PERCENT OF AREA (PA) พิจารณาค่า PA ที่ได้ถ้ามากกว่าค่าที่กำหนดไว้ ให้ตัดสินว่ากรอบสี่เหลี่ยมที่พิจารณาอยู่นี้มีป้าย และสัญญาณไฟจราจรที่สนใจอยู่หรือไม่ โดยค่าที่กำหนดนี้จะพิจารณาจากป้าย และสัญญาณไฟจราจรจากฐานข้อมูลภาพสีที่ใช้ในการทดลอง

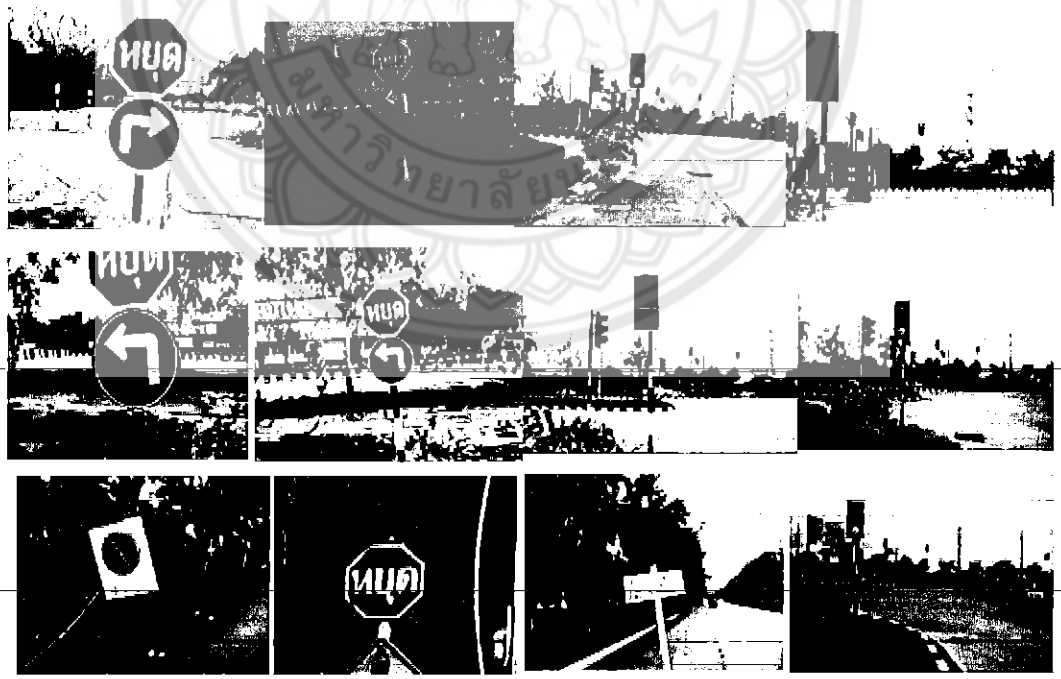
## 3.3 ขั้นตอนการ Training data

จากแผนภาพในรูปที่ 3.3ได้แบ่งการทำงาน ทั้งหมดเป็น 4 ส่วน ได้แก่

1. การเก็บและเตรียมภาพตัวอย่าง
2. Training Data
3. การตรวจจับและแปลความ
4. การแสดงผล

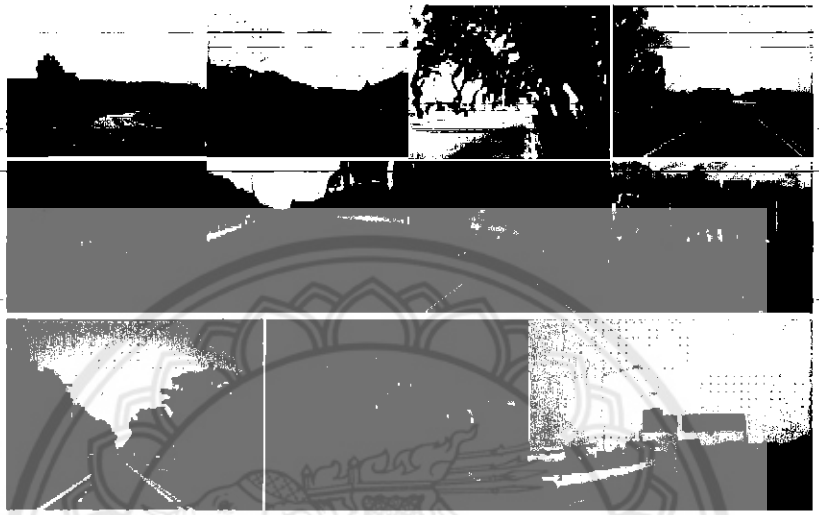
### 3.3.1 การเก็บและเตรียมภาพตัวอย่าง

เราจำเป็นต้องเก็บภาพตัวอย่างทั้งภาพ Positive และภาพ Negative ซึ่งมีรายละเอียดดังนี้ **ภาพ Positive:** เป็นภาพของวัตถุที่เราสนใจจะตรวจจับ โดยเก็บภาพในมุมมองต่างๆ และ ควรใช้รูปที่มีจำนวนมากในระดับหนึ่ง เพื่อให้สามารถตรวจจับวัตถุที่ต้องการหาได้ง่าย ถูกต้อง และแม่นยำ




**รูปที่ 3.4** ภาพ Positive image

**ภาพ Negative**: เป็นภาพสิ่งแวดล้อมอื่นๆ โดยต้องไม่มีสิ่งที่เราสนใจอยู่ในภาพ เช่น ภาพสิ่งของต่างๆ ภาพวิวทิวทัศน์ ภาพอาคารหรือสถานที่ต่างๆ เป็นต้น ภาพ Negative นี้จำเป็นต้องใช้ภาพจำนวนมาก ในการพัฒนาโครงงานนี้ใช้ภาพ Negative จำนวน 10109 ภาพ ใช้เป็นภาพสำหรับ Training 10109 ภาพ



รูปที่ 3.5 ภาพ Negative image

**ขั้นตอนการเตรียมภาพ Positive และภาพ Negative**

ก. ถ่ายภาพ Negative ด้วยกล้องดิจิทัล หรืออาจหาภาพจากอินเทอร์เน็ต โดยเน้นภาพที่มีลักษณะหลากหลาย และ มีขนาดของภาพใกล้เคียงกับขนาดของวีดีโอในที่นี้คือ 640x480

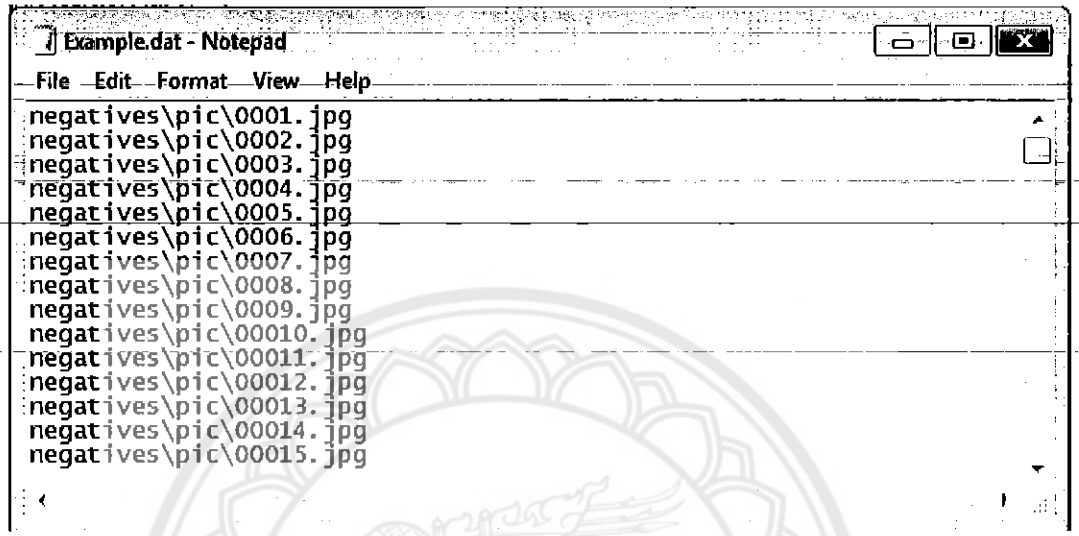
ข. เก็บภาพ Positive จากกล้องวีดีโอหรือจากกล้องดิจิทัล แต่ควรมีอัตราส่วนของภาพจากกล้องวีดีโอมากกว่า เนื่องจากว่าจะทำให้ได้ภาพที่มีความละเอียดใกล้เคียงกัน

ค. ทำการคัดภาพ Positive และภาพ Negative อีกครั้ง แปลงไฟล์ภาพให้เป็น .jpg ทั้งหมด แล้วปรับขนาดให้มีขนาดเท่ากับไฟล์วีดีโอ คือ 640x480 พิกเซล อาจมีขนาดใหญ่หรือเล็กกว่านี้ก็ได้ตามความเหมาะสม และตัดบางส่วนของรูปที่ไม่ต้องการทิ้งไปได้ สำหรับภาพ Negative

ง. ปรับค่าความสว่างหรือสีทั้งภาพ Positive และ Negative ได้ตามต้องการตามความเหมาะสมกับสภาพแวดล้อมต่างๆ ที่ต้องการตรวจจับ

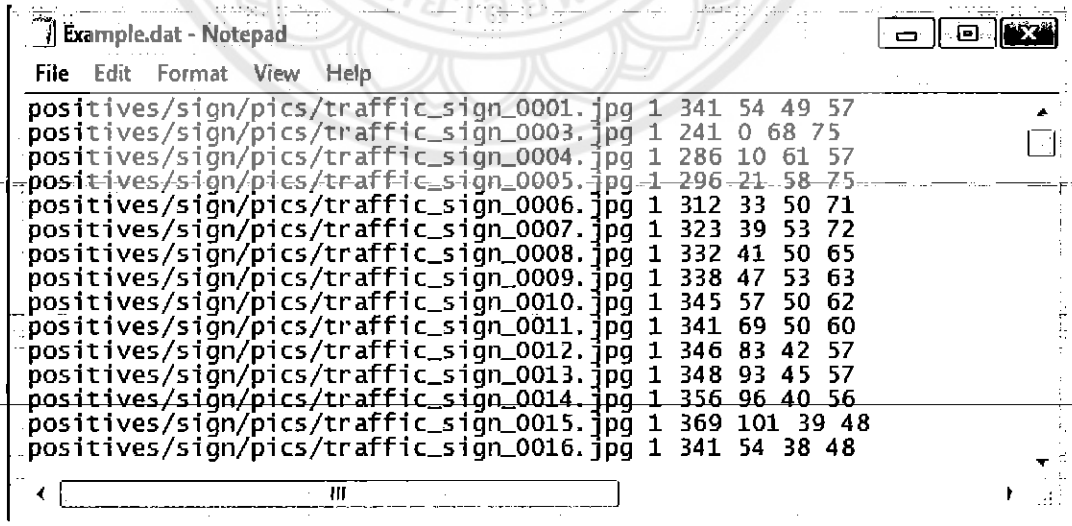
### 3.3.2 Training Data[17]

ขั้นตอนการเตรียมข้อมูลก่อนการ Training โดยการเก็บที่อยู่ของภาพ Negative ที่ใช้สำหรับ Training ไฟล์ .txt หรือ .DAT ดังรูปที่ 3.6




รูปที่ 3.6 ตัวอย่างการสร้างไฟล์ในการ Train ของภาพ Negative

จากนั้นก็เก็บที่อยู่ของภาพ Positive โดยใส่ขนาดข้างหลัง เพื่อระบุตำแหน่งและขนาดของรูปที่ต้องการตรวจจับในภาพนั้นๆดังตัวอย่างในรูปที่ 3.7




**รูปที่ 3.7** ตัวอย่างการสร้างไฟล์ในการ Train ของภาพ Positive

หลังจากนั้นนำไฟล์ที่ได้มาสร้างเป็นภาพเวกเตอร์โดยใช้โปรแกรม createsamples.exe ที่มาพร้อมกับ OpenCV โดยใช้คำสั่งดังนี้

```
createsamples -info nameSign.dat -vec nameSign.vec -w 20 -h 20
```

โดยที่ : - info คือ ไฟล์ที่เก็บที่อยู่และตำแหน่งของภาพ Positive ที่ได้จากข้อ 3
- vec คือ ชื่อไฟล์ .VEC ที่ต้องการเก็บ
- w คือ ความกว้างของภาพ Vector
- h คือ ความสูงของภาพ Vector

ในการ Training นี้ สามารถที่จะนำไฟล์ Vector ที่ได้มาทำการ Training กับภาพ Negative ที่เก็บลงไฟล์ไว้ และใช้โปรแกรม Haar-training ที่มาพร้องกับ OpenCV สร้างไฟล์ XML ขึ้นมา โดยจะเก็บอยู่ใน tree เพื่อใช้สำหรับการตรวจจับภาพ โดยมีรูปแบบตัวอย่างคำสั่งดังนี้

```
haartraining -data name_cascade -vec name.vec -bg negatives/train.dat -nstages 20 -
npos 1000 -nneg 9650 -w 20 -h 20 -nonsym -mem 1024 -mode ALL
```

โดยที่: -data คือ ตำแหน่งของไฟล์ที่เก็บข้อมูลที่ได้จากการ Training
- vec คือ ชื่อตำแหน่งของไฟล์ .VEC ที่จะใช้ Train
- bg คือ ตำแหน่งของไฟล์ที่เก็บข้อมูลของภาพ Negative ไว้
- nstage คือ จำนวน Stage ที่จำ Train
- npos คือ จำนวนของภาพ Positive
- nneg คือ จำนวนของภาพ Negatives
- w คือ ความกว้างของภาพ Vector ของไฟล์ Vector ที่ใช้
- h คือ ความสูงของภาพ Vector ของไฟล์ Vector ที่ใช้
- mem คือ ปริมาณ Memory ที่ใช้ในการ Train
- mode คือ Fโหมดในการ Train
- nonsym ให้กำหนดไว้หากภาพ Positive ที่ Train มีลักษณะไม่สมมาตร หากมีลักษณะสมมาตรไม่จำเป็นต้องใส่ค่านี้

กำหนดค่าโดย ให้มีจำนวน Stages เป็น 20 มีภาพ Positive และภาพ Negative กำหนดขนาดกว้างและ ยาวเป็นขนาดเดียวกับที่สร้างภาพ Vector เนื่องจากภาพที่ต้องการตรวจจับมีรูปร่างไม่สมมาตร จึงต้องกำหนดให้เป็น nonsym ไว้ด้วย และอนุญาตให้ใช้ memory ของเครื่องได้ 1 GB และ Training ในรูปแบบ ALL คือ รูปแบบที่ดีที่สุด ที่สามารถทำให้ตรวจจับภาพ ที่เอียงได้จากปกติถึง 45 องศา แต่อาจทำให้เสียเวลาในการ Training มาก

วิธีดูค่า False alarm นั้นสามารถดู จากผลที่แสดงระหว่างการ Train ในแต่ละ Node โดยค่า False alarm จะอยู่หลัง ค่าจำนวนภาพ Negative ที่ใช้ใน Stage นั้นๆ ตามตัวอย่างนี้ มีค่า False alarm เท่ากับ $1.02595ex10^{-6}$

```
Parent node: 9
*** 1 cluster ***
    POS: 80 80 1.000000
    NEG: 4500 1.02595e-006
    BACKGROUND PROCESSING TIME: 45241.51
    Precalculation time: 58.28
```

รูปที่ 3.8 ตัวอย่างของ log ในการ Training

ในการ Training นี้ ผู้พัฒนาได้ลองทดสอบประสิทธิภาพในการตรวจจับ ของการ Training ในหลายๆแบบ โดยเปลี่ยนแปลงค่าต่างๆ เช่น จำนวนภาพ Positive จำนวนภาพ Negative และการกำหนดจำนวน Stage ซึ่งมีผลกับค่า Minimum false alarm โดยคำนวณได้จาก

$$\text{Required leaf false alarm rate} = \text{Max false alarm rate}^{\text{nStage}}$$

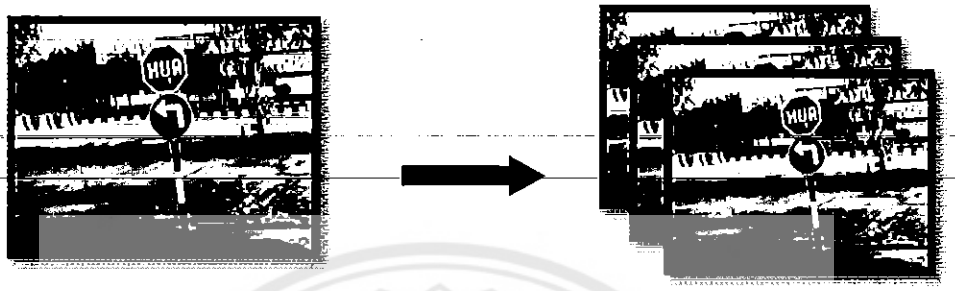
โดยที่: Required leaf false alarm rate คือค่า False Alarm ที่น้อยที่สุดที่เป็นไปได้ในการ Train แต่ละครั้ง

nStage คือ จำนวน Stage ที่ Train

Max false alarm rate มีค่ามาตรฐานที่ 0.5

### 3.3.3 การตรวจจับและแปลความ

ในการตรวจจับนั้น เมื่อรับข้อมูลวีดีโอนำเข้ามาแล้วจำเป็นต้องแยกวีดีโอดังกล่าวออกเป็นเฟรม เพื่อนำภาพในแต่ละเฟรมนั้นมาตรวจจับหาป้ายสัญญาณจราจรที่ต้องการ ซึ่งขึ้นอยู่กับ Frame Rate ที่ใช้ในการบันทึก



Input Capture Camera Frame

รูปที่ 3.9 การแยกภาพจากกล้องวีดีโอ

นำไฟล์ XML ที่ได้ จากกระบวนการ Training มาใช้ตรวจจับหาป้ายสัญญาณจราจร ภายในภาพนำเข้า โดยใช้หลักการของ Sub window mask วิ่งหาลักษณะที่เหมือนกัน ตามหลักการทำ Haar Classifier เนื่องจากภาพป้ายสัญญาณจราจรนั้นจะมีขนาดตั้งแต่ประมาณ 5 พิกเซล ถึง 75 พิกเซล จึงกำหนดให้ขนาดของ Window mask เล็กสุดเป็น 5x5 พิกเซล และเพิ่มขนาดของ Window mask ทีละ 5% ในการวิ่งวนภาพแต่ละรอบ เมื่อตรวจจับได้แล้ว จะทราบตำแหน่ง ขนาด และชนิดของรูปที่ตรวจจับได้

### 3.3.4 การแสดงผล

เมื่อทราบตำแหน่ง ขนาด และ ชนิดของภาพแล้ว จึงตีกรอบล้อมรอบป้ายดังกล่าว และแสดงข้อความที่เป็นความหมายของป้ายสัญญาณจราจรดังกล่าวใต้ภาพสัญญาณจราจรนั้น จากนั้นจึงนำภาพในแต่ละ Frame ที่ได้แล้วแสดงผลไปในความเร็วที่เท่ากับ Frame Rate เดิมของวีดีโอนำเข้า

รูปที่ 3.10 การแสดงผลการตรวจจับป้ายจราจร

## 3.4 อุปกรณ์และระบบ

### 3.4.1 อุปกรณ์ฮาร์ดแวร์

- เครื่องคอมพิวเตอร์ โน้ตบุ๊ก HP รุ่น G42-398TX: Inter Core i3 2.53GB, RAM DDR III 2 GB, Graphic card ATI HD 5470 DRR III 1GB , Hard disk 500GB ใช้สำหรับพัฒนาโปรแกรมและประมวลผล
- กล้องวีดีโอ Canon Power Shot A3000 IS: ใช้สำหรับบันทึกภาพวีดีโอบนท้องถนน
- กล้องดิจิทัล Canon Power Shot A3000 IS : ใช้สำหรับถ่ายภาพตัวอย่าง

### 3.4.2 ซอฟต์แวร์

- ซอฟต์แวร์นี้ทำงานบนระบบปฏิบัติการWindow XP
- Visual Studio C++ 2008 สำหรับพัฒนาระบบ
- .NET เฟรมwork 3.5 เพื่อเป็นแพลตฟอร์มสำหรับการพัฒนาโปรแกรม
- OpenCV 2.1 เพื่อเป็นคลังโปรแกรมสำหรับการพัฒนาระบบ
- Video Mach มาใช้ในการตัดเฟรมจากไฟล์วีดีโอ AVI เป็นภาพ JPEG และ BMP
- Microsoft Office Picture Manager เพื่อเปลี่ยนชื่อ ปรับค่าแสงและสีของรูปภาพ

จากเนื้อหา ในบทที่ 3 ดังกล่าว เราได้เลือก อัลกอลิทึม ของ Haar like-feature ในการตรวจจับ และตีความหมายของภาพ ซึ่งในบทนี้ ประกอบด้วย ส่วนของการออกแบบระบบ ขั้นตอนการเตรียมภาพ Positive และภาพ Negative การ Training และส่วนของการแสดงผล ซึ่งเราจะเห็นผลการ train แต่ละแบบ ว่า สามารถทำงานได้ดี ได้นั้น ขึ้นอยู่กับองค์ประกอบใดบ้าง ซึ่งสามารถติดตาม ในส่วนของบทที่ 4 ต่อไป

# บทที่ 4

# ผลการทดลอง

จากที่ได้ทำการศึกษากระบวนการทำงานของระบบควบคุมรถอัจฉริยะไร้คนขับ โดยการประมวลผลภาพ ซึ่งบทนี้จะกล่าวถึงผลการประมวลผลภาพของแต่ละขั้นตอนที่ได้ออกแบบไว้ในบทที่ 3 ซึ่งสามารถสรุปผลได้ดังต่อไปนี้

## 4.1 ผลการทดสอบการติดต่อกล้อง

การติดต่อกล้องจะทำการติดต่อผ่าน ไลบรารี ของ OpenCV โดยใช้ภาษา C++ โดยภาพที่ได้จะเป็นรูปแบบของไฟล์ภาพวีดีโอ แล้วทำการ Capture ภาพออกมาเป็นภาพนิ่งทีละภาพเพื่อใช้มาทำการประมวลผล

**รูปที่ 4.1** แสดงผลการติดต่อกล้อง

## 4.2 การทดสอบประสิทธิภาพของระบบจากกล้องวีดีโอ

ตอนที่ 1

สำหรับการทดสอบประสิทธิภาพจากวีดีโอ จำนวนป้ายที่ต้องการตรวจจับทั้งหมด 24 ป้าย ประกอบด้วย ป้ายห้ามจอด 4 ป้าย ป้ายเลี้ยวซ้าย 4 ป้าย ป้ายเลี้ยวขวา 4 ป้าย และป้ายหยุด 4 ป้าย ส่วน สัญญาณไฟจราจร ได้แก่ ไฟจราจรสีเขียว 4 เสา ไฟจราจร สีแดง 4 เสา ซึ่งแต่ละป้ายจะใช้ป้ายละ 250 รูป มารวมกัน4 ป้าย เป็น 1000 frame หรือ 250(frame/ป้าย)*4 (จำนวนป้าย) = 1000 frame/ป้ายชนิดเดียวกัน

โดยทดสอบกับตัว Classifier จากการ Train คือ ใช้จำนวนภาพ positive ได้แก่ 1000 500 300 ตามลำดับ โดยใช้ จำนวน negatives ค่าเดียวกันคือ 10109 โดย กำหนดให้ Trained Stage = 20 Stages ทำเช่นนี้ ทุกการ Train แต่ละป้าย แล้วบันทึกผลที่ได้ลงในตาราง 4.1 และ 4.2

ได้ผลการทดสอบดังนี้

**ตารางที่** 4.1 ตารางแสดงผลประสิทธิ์ภาพ การตรวจจับสัญญาณไฟจราจร ไฟแดงและไฟเขียว

| ป้าย | จำนวน Frame | Train | | Hit Rate (%) | False Rate (%) | Miss Rate (%) |
|---|---|---|---|---|---|---|
| | | positive | Negative | | | |
| | | 1000 | 10109 | 71.3 | 29.7 | 11.3 |
| สัญญาณไฟจราจรสีเขียว | 1000 | 500 | 10109 | 62.9 | 37.1 | 16.1 |
| | | 300 | 10109 | 55.3 | 54.7 | 21.0 |
| สัญญาณไฟจราจรสีแดง | 1000 | 1000 | 10109 | 73.3 | 26.7 | 9.8 |
| | | 500 | 10109 | 66.5 | 33.5 | 13.9 |
| | | 300 | 10109 | 59.7 | 40.3 | 19.4 |

**ตารางที่ 4.2** ตารางแสดงผลประสิทธิ์ภาพ การตรวจจับป้ายห้ามหยุด ป้ายห้ามจอด ป้ายเลียวขวา ป้ายเลียวซ้าย

| ป้าย | จำนวน Frame | Train | | Hit Rate (%) | False Rate (%) | Miss Rate (%) |
|---|---|---|---|---|---|---|
| | | positive | Negative | | | |
| | | 1000 | 10109 | 73.1 | 26.9 | 9.1 |
| ป้ายหยุด | 1000 | 500 | 10109 | 66.8 | 33.2 | 13.3 |
| | | 300 | 10109 | 53.5 | 46.5 | 19.3 |
| ป้ายห้ามจอด | 1000 | 1000 | 10109 | 70.9 | 29.1 | 13.9 |
| | | 500 | 10109 | 63.6 | 36.4 | 17.0 |
| | | 300 | 10109 | 56.9 | 43.1 | 23.2 |
| ป้ายเลียวซ้าย | 1000 | 1000 | 10109 | 75.3 | 24.7 | 8.6 |
| | | 500 | 10109 | 70.1 | 29.9 | 11.8 |
| | | 300 | 10109 | 63.7 | 36.3 | 17.5 |
| | | 1000 | 10109 | 73.1 | 26.9 | 8.8 |
| ป้ายเลียวขวา | 1000 | 500 | 10109 | 68.6 | 31.4 | 14.2 |
| | | 300 | 10109 | 59.8 | 40.2 | 19.8 |

**ตอนที่ 2**

ทดสอบโดยการนำ frame ที่เกิดจาก ป้ายเลี้ยวขวา ทั้งหมด 4 ป้าย ป้ายละ 250 รูป มารวมกัน เป็น 1000 frame มาทดสอบกับ การ Train ที่ใช้ จำนวน positive 300 รูป กับ negative 6000 รูป และ จำนวน positive 300 รูป กับ negative 3000 รูป และสุดท้าย เปรียบเทียบ จำนวน positive 300 รูป กับ negative 1500 รูป กำหนดให้ Trained Stage = 20 Stages แล้วบันทึกผลที่ได้ลงในตารางที่ 4.3

**ตารางที่ 4.3** ตารางแสดงผลประสิทธิ์ภาพ การตรวจจับ ป้ายเลี้ยวขวา ที่มีขนาด จำนวน positive และ negative ที่แตกต่างกัน

| ป้าย | จำนวน Frame | Train | | Hit Rate (%) | False Rate (%) | Miss Rate (%) |
|---|---|---|---|---|---|---|
| | | positive | Negative | | | |
| ป้ายเลี้ยวขวา | 1000 | 300 | 6000 | 53.7 | 46.3 | 31.7 |
| | | 300 | 3000 | 41.9 | 59.1 | 58.6 |
| | | 300 | 1500 | 33.4 | 66.6 | 67.1 |

โดยที่ Hit Rate คือ ร้อยละของจำนวนภาพที่ตรวจจับเจอต่อจำนวนFrameในวีดีโอ

False Rate คือ ร้อยละของจำนวนภาพที่ตรวจจับผิดพลาดต่อจำนวน Frameในวีดีโอ

Miss Rate คือ ร้อยละของจำนวนป้ายที่ตรวจจับไม่เจอ ต่อจำนวนป้ายทั้งหมด\

Positive คือ ภาพที่เราสนใจ

Negative คือ ภาพที่เราไม่สนใจ

**ตัวอย่าง** ผลลัพธ์ที่ได้จากการทดสอบด้วยภาพวีดีโอ

**รูปที่ 4.2 ก**

**รูปที่ 4.2 ข**

**รูปที่ 4.2 ค**

**รูปที่ 4.2 ง**

**รูปที่ 4.2 จ**

**รูปที่ 4.2 ฉ**

**รูปที่ 4.2 ก** แสดงผล การตรวจจับป้ายหยุดและป้ายเลี้ยวซ้าย ไม่สามารถตีความหมายได้ เนื่องจากลักษณะป้ายมีองค์ประกอบของป้ายไม่สมบูรณ์

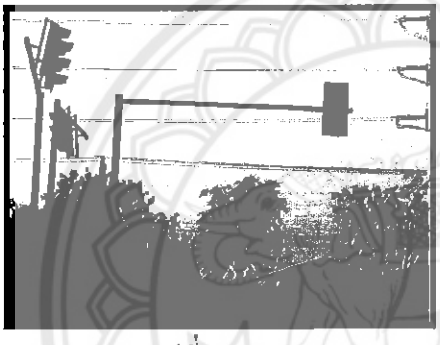
**รูปที่ 4.2 ข** แสดงผล การตรวจจับป้ายหยุดและป้ายเลี้ยวซ้าย ไม่สามารถตีความหมายได้ เนื่องจาก มีระยะห่าง 30 เมตร

**รูปที่ 4.2 ค** แสดงผล การตรวจจับป้ายหยุดและป้ายเลี้ยวซ้าย สามารถตีความหมายได้ เนื่องจาก ป้ายมีองค์ประกอบครบ และมีระยะทางห่าง 3 เมตร

**รูปที่ 4.2 ง** แสดงผล การตรวจจับป้ายหยุดและป้ายเลี้ยวซ้าย สามารถตีความหมายได้ เนื่องจาก มีองค์ประกอบของป้ายครบ และป้ายยังหมุนไปทางซ้าย 7 องศา และมีระยะทางห่าง 7 เมตร

**รูปที่ 4.2 จ** แสดงผล การตรวจจับป้ายหยุดและป้ายเลี้ยวซ้าย สามารถตีความหมายของป้ายเลี้ยวซ้ายได้ เนื่องจาก มีองค์ประกอบของป้ายครบ และป้ายยังหมุนไปทางขวา 20 องศา และมีระยะทางห่าง 1 เมตร แต่ไม่สามารตีความหมายของป้ายหยุดได้เพราะป้ายหยุดมีลักษณะภาพไม่สมบูรณ์

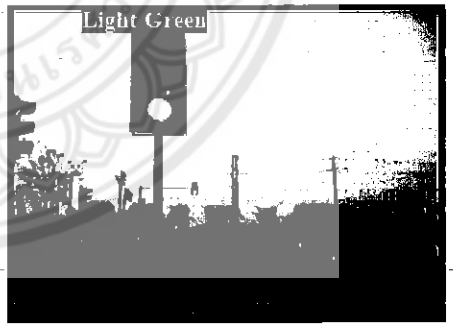
**รูปที่ 4.2 ฉ** แสดงผล การตรวจจับป้ายหยุดและป้ายเลี้ยวซ้าย สามารถตีความหมายได้ เนื่องจาก ป้ายมีองค์ประกอบครบ และป้ายยังหมุนไปทางขวา 15 องศา และมีระยะทางห่าง 3 เมตร



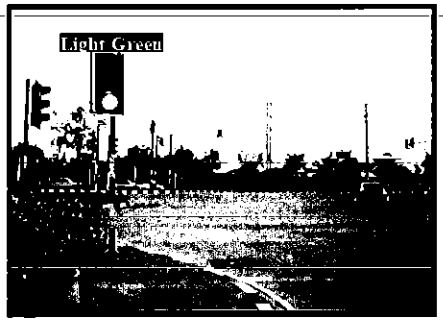
รูปที่ 4.3 ก

รูปที่ 4.3 ข

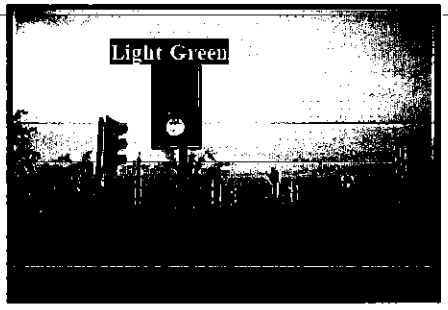
รูปที่ 4.3 ค




รูปที่ 4.3 ง




รูปที่ 4.3 จ




รูปที่ 4.3 ฉ

**รูปที่ 4.3 ก** แสดงผล การตรวจจับสัญญาณไฟจราจรสีเขียว ไม่สามารถตีความหมายได้ เนื่องจาก ดวงไฟสีเขียวไม่ชัดเจน และมีระยะทางห่าง 30 เมตร

**รูปที่ 4.3 ข** แสดงผล การตรวจจับสัญญาณไฟจราจรสีเขียว ไม่สามารถตีความหมายได้ เนื่องจาก มีระยะทางห่าง 45 เมตร

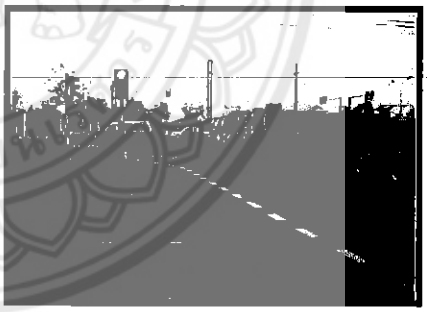
**รูปที่ 4.3 ค** แสดงผล การตรวจจับสัญญาณไฟจราจรสีเขียว สามารถตีความหมายได้ เนื่องจาก มีระยะทางห่าง 4 เมตร

**รูปที่ 4.3 ง** แสดงผล การตรวจจับสัญญาณไฟจราจรสีเขียว สามารถตีความหมายได้ เนื่องจาก มีระยะทางห่าง 3.5 เมตร

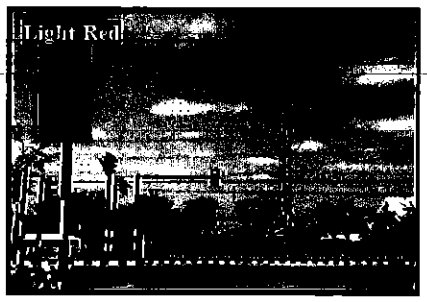
**รูปที่ 4.3 จ** แสดงผล การตรวจจับสัญญาณไฟจราจรสีเขียว สามารถตีความหมายได้ เนื่องจาก มีดวงไฟสีเขียวที่เต็มดวง สัญญาณไฟยังหมุนไปทางขวา 20 องศา และมีระยะทางห่าง 2 เมตร

**รูปที่ 4.3 ฉ** แสดงผล การตรวจจับสัญญาณไฟจราจรสีเขียว สามารถตีความหมายได้ เนื่องจาก มีดวงไฟสีเขียวที่เต็มดวง สัญญาณไฟยังหมุนไปทางขวา 10 องศา และมีระยะทางห่าง 3 เมตร

**รูปที่ 4.4 ก**

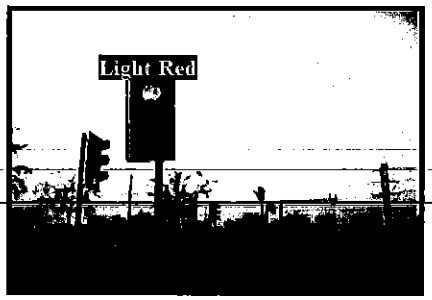


**รูปที่ 4.4 ข**




**รูปที่ 4.4 ค**

**รูปที่ 4.4 ง**




**รูปที่ 4.4 จ** **รูปที่ 4.4 ฉ**

**รูปที่ 4.4 ก** แสดงผล การตรวจจับสัญญาณไฟจราจรสีแดง ไม่สามารถตีความหมายได้ เนื่องจาก ดวงไฟสีแดงไม่เต็ม ดวง และมีระยะทางห่าง 30 เมตร

**รูปที่ 4.4 ข** แสดงผล การตรวจจับสัญญาณไฟจราจรสีแดง ไม่สามารถตีความหมายได้ เนื่องจาก ดวงไฟสีแดงไม่ชัดเจน ดวง และมีระยะทางห่าง 45 เมตร

**รูปที่ 4.4 ค** แสดงผล การตรวจจับสัญญาณไฟจราจรสีแดง สามารถตีความหมายได้ เนื่องจาก มีดวงไฟสีแดงที่เต็มดวง และมีระยะทางห่าง 3 เมตร

**รูปที่ 4.4 ง** แสดงผล การตรวจจับสัญญาณไฟจราจรสีแดง สามารถตีความหมายได้ เนื่องจาก มีดวงไฟสีแดงที่เต็มดวง และมีระยะทางห่าง 10 เมตร

**รูปที่ 4.4 จ** แสดงผล การตรวจจับสัญญาณไฟจราจรสีแดง สามารถตีความหมายได้ เนื่องจาก มีดวงไฟสีแดงที่เต็มดวง สัญญาณไฟยังหมุนไปทางขวา 15 องศา และมีระยะทางห่าง 3 เมตร

**รูปที่ 4.4 ฉ** แสดงผล การตรวจจับสัญญาณไฟจราจรสีแดง สามารถตีความหมายได้ เนื่องจาก มีดวงไฟสีแดงที่เต็มดวง สัญญาณไฟยังหมุนไปทางซ้าย 15 องศา และมีระยะทางห่าง 3 เมตร

รูปที่ 4.5 ก

รูปที่ 4.5 ข

รูปที่ 4.5 ค

รูปที่ 4.5 ง

รูปที่ 4.5 จ

รูปที่ 4.5 ฉ

**รูปที่ 4.5 ก** แสดงผล การตรวจจับป้ายหยุดและป้ายเลี้ยวขวา ไม่สามารถตีความหมายได้ เนื่องจาก ลักษณะป้ายองค์ประกอบของป้ายไม่ครบ

**รูปที่ 4.5 ข** แสดงผล การตรวจจับป้ายหยุดและป้ายเลี้ยวขวา ไม่สามารถตีความหมายได้ เนื่องจาก มีระยะห่าง 25 เมตร

**รูปที่ 4.5 ค** แสดงผล การตรวจจับป้ายหยุดและป้ายเลี้ยวขวา สามารถตีความหมายได้ เนื่องจาก ป้ายมีองค์ประกอบครบ และมีระยะทางห่าง 3 เมตร

**รูปที่ 4.5 ง** แสดงผล การตรวจจับป้ายหยุดและป้ายเลี้ยวขวา สามารถตีความหมายได้ เนื่องจาก มีองค์ประกอบของป้ายครบ และป้ายยังหมุนไปทางขวา 10 องศา และมีระยะทางห่าง 3 เมตร

**รูปที่ 4.5 จ** แสดงผล การตรวจจับป้ายหยุดและป้ายเลี้ยวขวา สามารถตีความหมายได้ เนื่องจาก มีองค์ประกอบของป้ายครบ และป้ายยังหมุนไปทางขวา 20 องศา และมีระยะทางห่าง 2 เมตร

**รูปที่ 4.5 ฉ** แสดงผล การตรวจจับป้ายหยุดและป้ายเลี้ยวขวา สามารถตีความหมายได้ เนื่องจาก ป้ายมีองค์ประกอบครบ และมีระยะทางห่าง 2 เมตร

## 4.3 สรุปผลการทดสอบประสิทธิภาพของระบบ

ในการทดสอบระบบนั้น พบว่าการตรวจจับภาพวีดีโอ ด้วยการ train ที่มีจำนวน positive จำนวนที่แตกต่างกัน ส่งผลให้ค่า hit rate เปลี่ยนแปลงไปด้วย ยิ่งจำนวน positive มาก ส่งผลให้ ค่าความแม่นยำ(hit rate) ยิ่งมาก ตรงกันข้าม ถ้าจำนวน positive น้อย ส่งผลให้ อัตราค่าความแม่นยำ (hit rate) ยิ่งมีค่าน้อยตามไปด้วย รวมถึง การ train ที่ใช้ ภาพ positive คงที่ แต่จำนวน negative แตกต่างกัน ส่งผลให้ อัตราความแม่นยำ(hit rate) มีค่าที่แตกต่างกันด้วย ยิ่งค่า negative ยิ่งมาก ส่งผลให้ การเกิด อัตราค่าความแม่นยำ(hit rate) มากตามไปด้วย ตรงกันข้าม ถ้า ค่า negative ยิ่งน้อย ส่งผลให้ การเกิด ค่าอัตราความแม่นยำ(hit rate) ก็ยิ่งน้อยตามไปด้วย

จากเนื้อหาในบทที่ 4 ทำให้เราทราบว่า จำนวน positive และ negative มีผลต่อระบบโดยตรง และยังมีอีกหลายองค์ประกอบที่เป็นสาเหตุของการ เกิดความผิดพลาดเหล่านั้น ซึ่งสามารถติดตาม ในส่วนของเนื้อหาใน บทที่ 5 ต่อไป

# บทที่ 5 บทสรุป

ในการพัฒนาระบบตรวจจับและตีความป้ายสัญญาณจราจรกล้องวีดีโอนั้น ผู้พัฒนาได้เลือกใช้วิธี Haar like-features ซึ่งเป็นวิธีที่นิยมใช้ในการตรวจจับวัตถุต่างๆ ที่เราสนใจได้ เช่น การตรวจจับใบหน้า การตรวจลายนิ้วมือ เนื่องจากมีความรวดเร็วในการตรวจจับ และสามารถแปลความหมายได้เป็นอย่างดี ในระหว่างการพัฒนาระบบนั้น ผู้พัฒนาได้พบเจอปัญหาและอุปสรรคต่างๆรวมไปถึงได้เล็งเห็นช่องทางในการพัฒนาระบบให้มีความสมบูรณ์เพิ่มมากขึ้น ซึ่งมีรายละเอียดดังนี้

## 5.1 ข้อสรุป

ระบบตรวจจับและตีความป้ายสัญญาณจราจรจากกล้องวีดีโอนั้น มีวิธีการคร่าวๆคือ ระบบจะดึงภาพออกจากไฟล์วีดีโอมาที่ละภาพแล้วนำมาวิเคราะห์และตรวจสอบหาบริเวณที่มีลักษณะใกล้เคียงกับป้ายสัญญาณจราจรที่เราสนใจ แล้วจึงตีกรอบ และแสดงข้อความลงไปภายในภาพดังกล่าว แล้วส่งผลลัพธ์ที่ได้ทั้งหมดออกมาเป็นภาพวีดีโอเช่นเดิม ซึ่งพัฒนาโดยอาศัยหลักการของ Haar like-features ที่จะตรวจจับหาลักษณะต่างๆที่เราต้องการมาเปรียบเทียบกับรูปที่ใช้ทดสอบเพื่อหาลักษณะที่ต้องการ จากนั้นรวมไฟล์รูปที่ได้หลังจากการตรวจจับแล้ว เข้าด้วยกันแสดงผลลัพธ์เป็นภาพวีดีโอเช่นเดิม

กระบวนการ Haar like-Features นั้นเป็นกระบวนการที่มีประสิทธิภาพดีพอสมควร สามารถตรวจจับและตีความได้ค่อนข้างรวดเร็ว แต่ความแม่นยำจะขึ้นอยู่กับจำนวนรูป Positive และ Negative ที่จำเป็นต้องใช้ในปริมาณมาก และต้องมีสภาพแวดล้อมที่หลากหลายและครอบคลุมในกรณีต่างๆที่จะใช้พิจารณา จะเห็นได้จากผลการทดสอบในการ Train โดยใช้ภาพตัวอย่างปริมาณต่างๆกัน แต่ข้อเสียของวิธีการนี้คือ เสียเวลาในการ Training มาก แต่หากต้องการความแม่นยำสูงยิ่งต้องใช้รูปปริมาณมาก และยิ่งเสียเวลา Train นานยิ่งขึ้น แม้ว่าวิธีการนี้ยังมีการตรวจจับที่ผิดบ้างและยังไม่สามารถตรวจจับเจอได้ในทุกๆ Frame แต่หากใช้ร่วมกับวิธีตรวจจับแบบอื่นแล้ว น่าจะทำให้เป็นวิธีที่มีประสิทธิภาพมากยิ่งขึ้น

## 5.2 ปัญหาและอุปสรรค

- ปัญหาเรื่องประสิทธิ์ภาพของระบบ สืบเนื่องจากปัญหาในการ Train ข้อมูลนั้น ยังใช้รูปจำนวนน้อยเกินไป ทำให้การตรวจจับภาพยังมีประสิทธิภาพที่ยังต่ำอยู่

- ระบบที่พัฒนาขึ้นนั้นพัฒนาใน Window Platform ซึ่งจะไม่สามารถรันบนระบบปฏิบัติการอื่นได้

- ปัญหาในการหาป้ายสัญญาณจราจรที่มีลักษณะตามต้องการ เช่น ความไม่สมบูรณ์ของป้าย ซึ่งมีสีที่ไม่ชัดเจน ป้ายสัญญาณจราจรส่วนมากแล้วมักมีรอยขีดเขียน สีซีด หรือมีป้ายประกาศโฆษณาแปะอยู่ จึงเป็นปัญหาในการหา และเก็บภาพป้ายสัญญาณจราจรตามที่ต้องการได้

- ปัญหาการไม่สามารถเก็บภาพในมุมมองที่ต้องการได้ในบางพื้นที่ เนื่องจากปัญหาด้านความปลอดภัยของตัวผู้พัฒนาเอง

- ปัญหาทางด้านอุปกรณ์มีไม่เพียงพอ เนื่องจากในการ Train แต่ละครั้ง ใช้เวลาค่อนข้างนาน ผู้พัฒนาไม่สามารถ Train ไปพร้อมกับการพัฒนาโปรแกรมได้ จึงเป็นเหตุให้เสียเวลาในการพัฒนาโครงงานค่อนข้างมาก

## 5.3 ข้อเสนอแนะและแนวทางการพัฒนา

ระบบตรวจจับและตีความป้ายสัญญาณจราจรจากกล้องวีดีโอที่ได้พัฒนาขึ้นนี้ เป็นเพียงระบบจำลองเบื้องต้นเท่านั้น ยังสามารถนำไปพัฒนาต่อได้ในอีกหลายๆด้าน รวมไปถึงการพัฒนาด้านฮาร์ดแวร์เพื่อให้สามารถนำไปติดตั้งบนยานพาหนะสำหรับใช้งานจริง นอกจากนี้อาจนำไปพัฒนาเพิ่มเติม เพื่อให้มีความแม่นยำในการตรวจจับมากขึ้น เพิ่มการตรวจจับป้ายในแบบอื่นๆ หรืออาจเพิ่มความสามารถในการตรวจจับในหลายๆสภาพอากาศ และสถานที่ เพื่อเพิ่มความสามารถในการตรวจจับให้กับระบบ โดยการ Train ด้วยรูปที่มีจำนวนมากขึ้น และมีหลากหลายสถานะมากขึ้น

นอกจากนี้ยังสามารถนำประยุกต์เพิ่มเติมในด้านอื่นๆอีก เช่น เพิ่มระบบสมองกลอัจฉริยะสำหรับใช้เป็นระบบหนึ่งในระบบขับขี่ยานยนต์อัตโนมัติ หรือ การทำระบบเตือนภัยรูปแบบต่างๆบนท้องถนน เพื่อเพิ่มความปลอดภัยให้กับผู้ขับขี่

# เอกสารอ้างอิง


[1] เจเนซิส มีเดียคอม จำกัด. รวมคลิปอาร์ท (Universal Cliparts) Vol.2. สืบค้นเมื่อ 16 มกราคม 2554, จาก http://www.yupparaj.ac.th/CAI/graphic/graphic.html.

[2] พูนศักดิ์ สักกทัตติยกุล.ความรู้ทั่วไปเกี่ยวกับกราฟิก.สืบค้นเมื่อ 16 มกราคม 2554, จาก http://202.44.68.33/node/19303.

[3] ภาพระบบสี RGB (Red/Green/Blue).สืบค้นเมื่อ 16 มกราคม 2554, จาก http://www.thaigoodview.com/files/u31720/sks8656ce661c12450d1e5275.gif.

[4] ภาพมาตรฐานสี. สืบค้นเมื่อ 16 มกราคม 2554, จาก http://www.klongdigital.com/Editor/assets/RGB0.jpg.

[5] แนะนำสู่การประมวลผลภาพดิจิตอล(1 กรกฎาคม พ.ศ. 2552). สืบค้นเมื่อ 16 มกราคม 2554, จาก http://gutenbug.blogspot.com/2009_07_01_archive.html.

[6] แสดงผลของการทำงานของวิธี Edge Segmentation สืบค้นเมื่อ 16 มกราคม 2554, จาก http://www.cis.upenn.edu/~jshi/software/demo2.html.

[7] Demo step by step: Image Segmentation with Normalized Cuts. Retrieved January 2 , 2009 , from http://www.rtsd.mi.th/New%202008/1251/image_ processing/Image%20Processing4.pdf.

[8] นิมิต เสรีวัตนะ, พีระศักดิ์ สมชัย และ สู้เดช วัฒนอุดมโรจน์ "การปรับภาพเพื่อใช้ในการวิเคราะห์ Binary Image Analysis,"ระบบการเข้าออกรถยนต์อัตโนมัติ.สืบค้นเมื่อ 16 มกราคม 2554, จาก http://web.en.rmutt.ac.th/cp/Project/p_27/index.html.

[9] สันติ สถิตวรรธนะ. (2548). การตรวจสอบลายและสิ่งสกปรกผิวและในเนื้อย่างแผ่น. วิทยานิพนธ์ วศ.บ., มหาวิทยาลัยสงขลานครินทร์, สงขลานครินทร์. สืบค้นเมื่อ 16 มกราคม 2554, จาก http://doc.clib.psu.ac.th/public7/thesis7/full/250935/250935.htm

[10] ฐตวัฒน์ เตชจรัสชีวิน , ณพวีณา ฤกษ์ปรีดาพงศ์. ระบบติดตามลักษณะเด่นบนใบหน้าโดยใช้กล้องเพียงหนึ่งตัว. 2551.วิทยานิพนธ์ วศ.บ. , จุฬาลงกรณ์มหาวิทยาลัย, กรุงเทพมหานคร. สืบค้นเมื่อ 16 มกราคม 2554, จาก

http://161.200.126.55/~achatcha/SeniorProject2552/Comp/Facial%20feature%20tracking%20system%20using%20single%20camera.pdf.

[11] จิปรานัท ศรีพรม. (24 มกราคม 2553).การปรับภาพเชิงพื้นที่.สืบค้นเมื่อ 16 มกราคม 2554, จาก http://jittranutsri.blogspot.com/2010/01/blog-post_5289.html.

[12] ปิยมาส จินตนายากานนท์, นางสาวอรณิช บุญศรี. (2552). ระบบตรวจสอบไบโอฟิล์มของเชื้อ Listeria monocytogenes บนพื้นผิวสายพานลำเลียงในโรงงานอุตสาหกรรมไก่ในประเทศไทย .วิทยานิพนธ์ วศ.บ. , จุฬาลงกรณ์มหาวิทยาลัย ,กรุงเทพมหานคร.สืบค้นเมื่อ 16 มกราคม 2554 , จากhttp://161.200.126.55/~achatcha/SeniorProject2552/Comp/Detection%20system%20for%20listeria%20monocytogenes%20biofilm%20formation%20on%20belt%20conveyers%20used%20in%20Thai%20local%20chicken%20plant.pdf .

[13] สารานิติ .Image Processing การหาขอบภาพ (Edge detection). 10 สิงหาคม 2553. สืบค้นเมื่อ 16 มีนาคม 2554, จาก http://www.boarddev.com/forum/index.php?topic=4081.

[14] Paul Viola and Michael J. Jones. "Rapid Object Detection using a Boosted Cascade of Simple Features". Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2001.

[15] Qing Chen, Nicolas D.Hand Gesture Recognition Using Haar-Like Features and a Stochastic Context-Free Gamma. Georganas, Fellow, IEEE, and Emil M. Petriu, Fellow, IEEE .Retrieved January 16, 2011, from http://www.site.uottawa.ca/~petriu/TrIM08-HandGestRecognHaarFeatures.pdf.

[16] ชารานนิ ชานแซ้ม. เว็บแคม .สืบค้นเมื่อ 16 มกราคม 2554, จาก http://www.webcam2home.com/webcam-knowledge-th.htm.

[17] Naotoshi Seo. Tutorial: OpenCV haartraining (Rapid Object Detection With A Cascade of Boosted Classifiers Based on Haar-like Features). Retrieved January 16, 2009, from http://note.sonots.com/SciSoftware/haartraining.html#l7a173e3.

# ภาคผนวก

# การติดตั้ง โปรแกรมและ Opencv 2.1

โปรแกรมที่ใช้ในการทำงานคือ OpenCV2.1 และ Visual Basic 2008/2010 มีขั้นตอนทั้งหมดดังนี้ต่อไปนี้

**การติดตั้งโปรแกรม OpenCV 2.1**

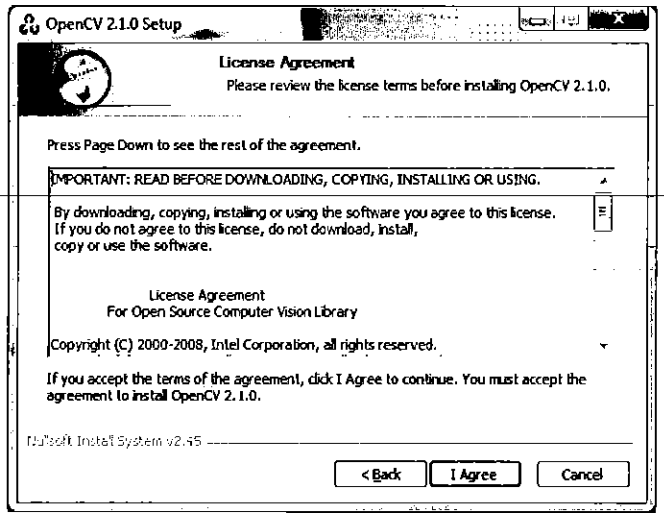
• Download โปรแกรมที่ http://sourceforge.net/projects/opencvlibrary/files/

• เลือก Download ตาม OS ของเรา แล้วเลือก Version ที่เราต้องการ ในที่นี้เลือก windows โปรแกรม Version 2.1

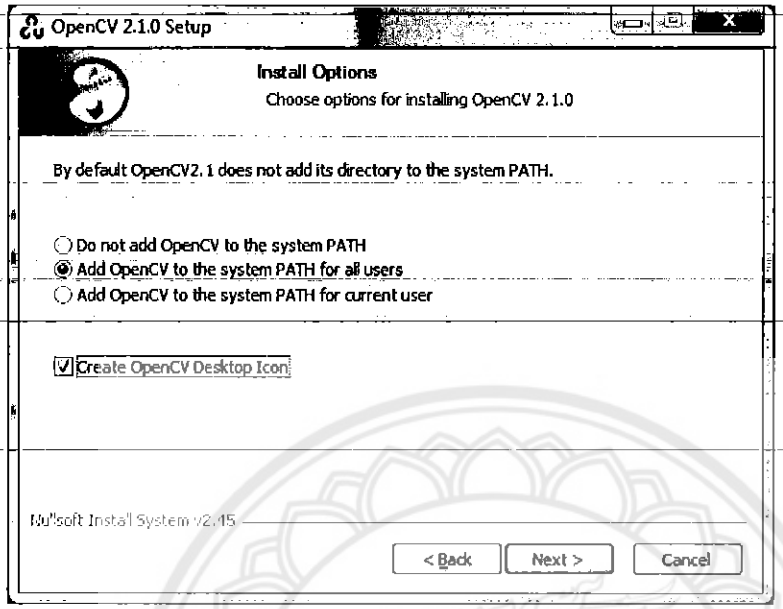
• ดับเบิ้ลคลิกที่ตัวโปรแกรมที่เรา Download มา

• หน้านี้จะเป็นคำอธิบายของโปรแกรม อ่านเสร็จแล้วกด Next >

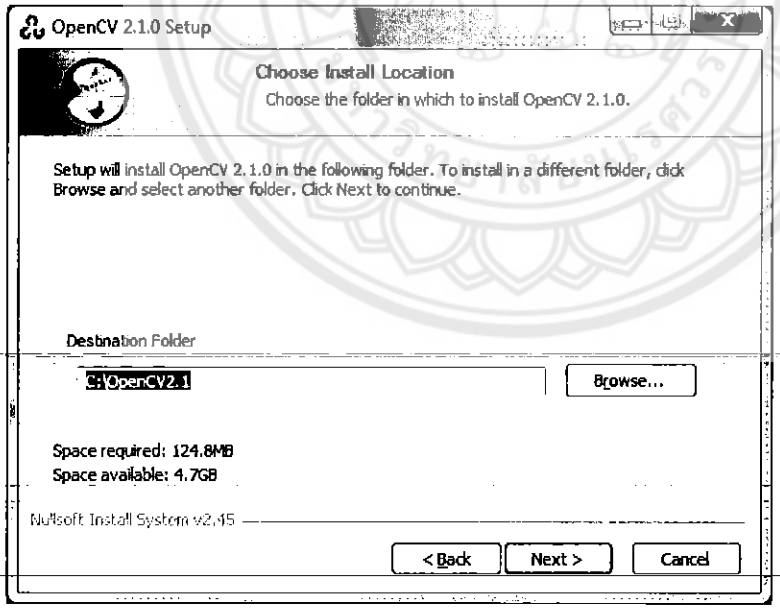
• อ่านข้อตกลงของโปรแกรม แล้วกด I Agree



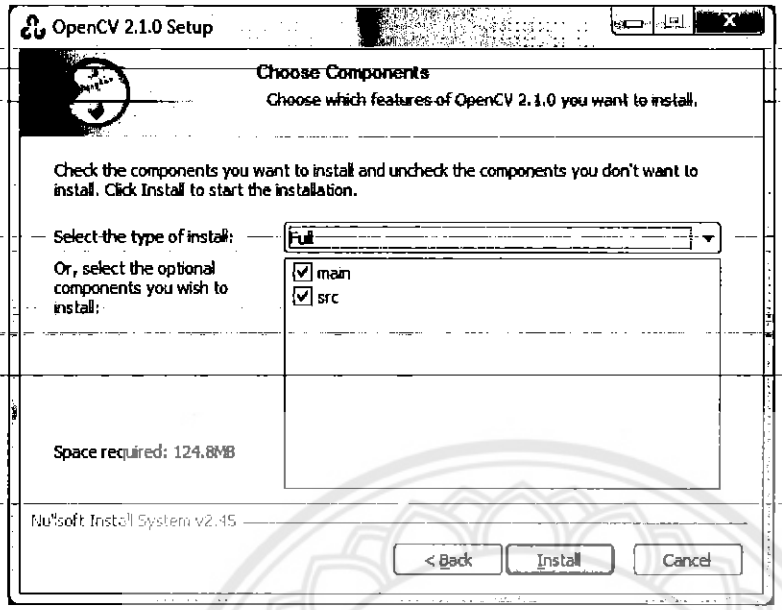
• เลือกตัวเลือก Add OpenCV to the system PATH for all users เพื่อให้ทุกคนใช้งานได้ เสร็จแล้ว กด Next >



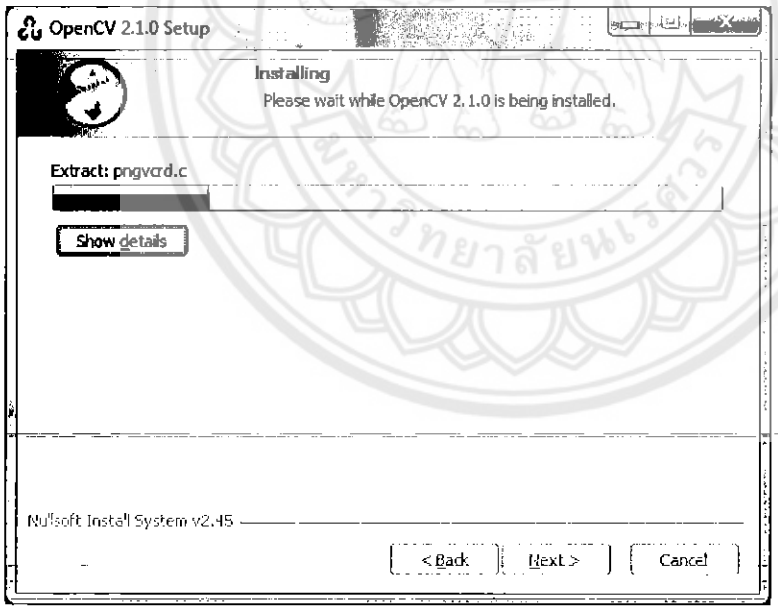

• เลือก Directory ที่ต้องการติดตั้ง แล้วกด Next >



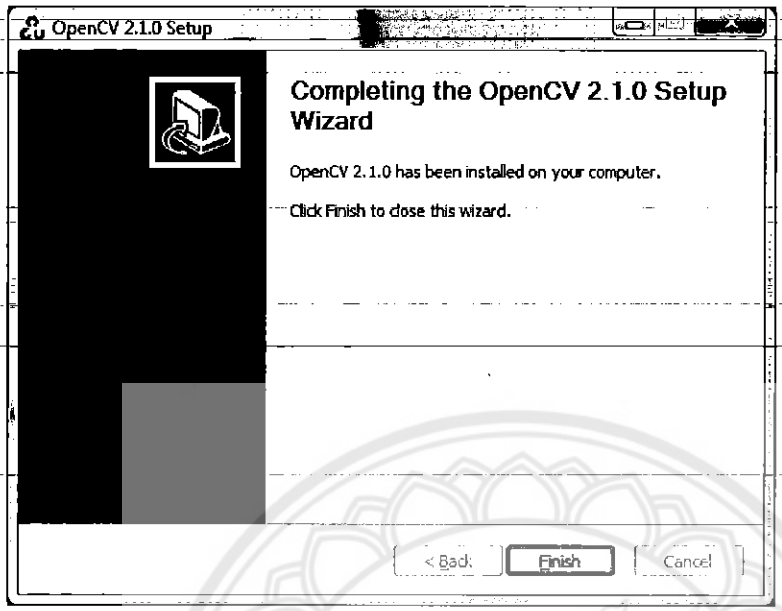
• เลือกว่าต้องการติดตั้งส่วนประกอบเสริมอะไรบ้าง ในที่นี้เลือก Full เพื่อติดตั้งทั้งหมดแล้วกด Install เพื่อติดตั้ง




• รอจนกว่าจะเสร็จสมบูรณ์

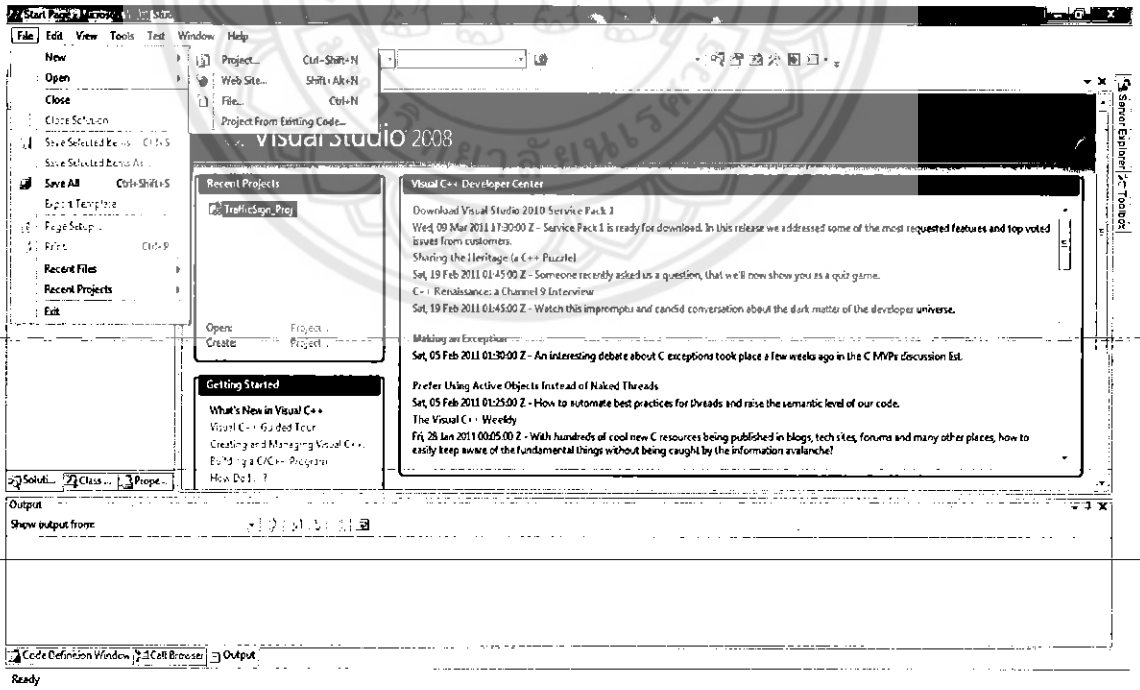


• การติดตั้งเสร็จสมบูรณ์ กด Finish เพื่อออกจากการติดตั้ง

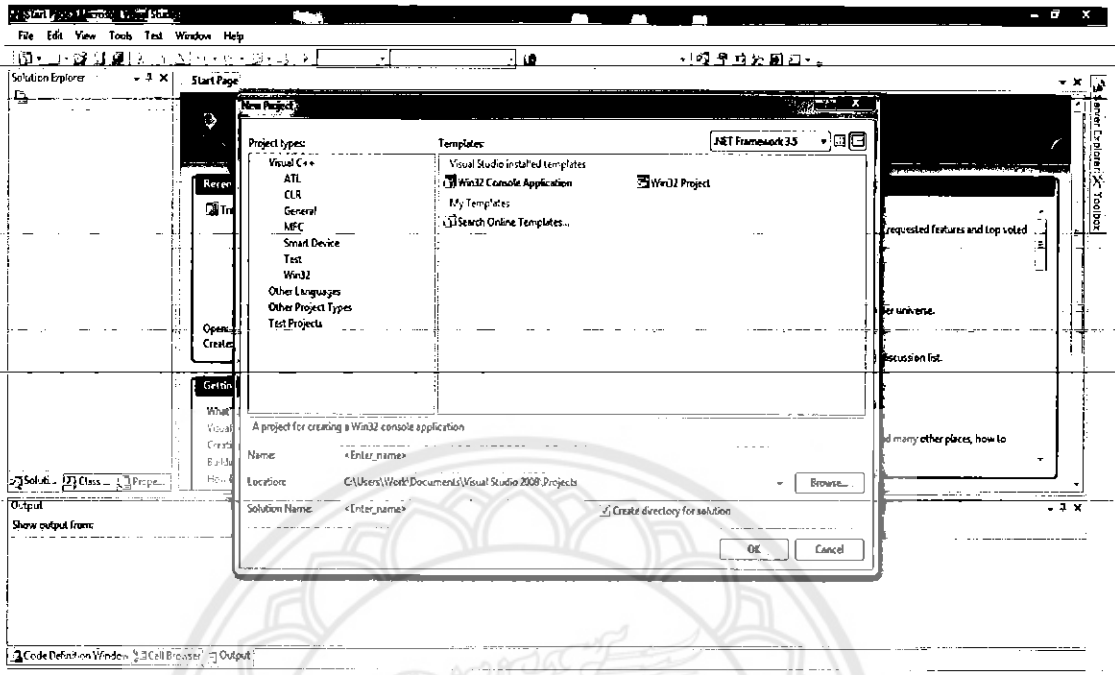



**วิธีการเซ็ตค่าในโปรแกรม Visual Studio ให้เชื่อมต่อกับโปรแกรม OpenCV**

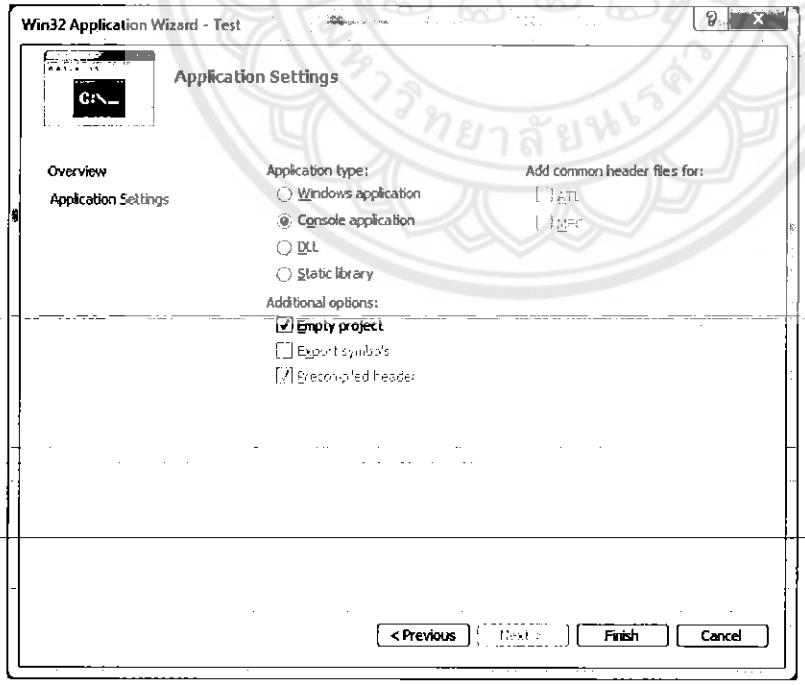
• คลิกที่ File -> New -> Project



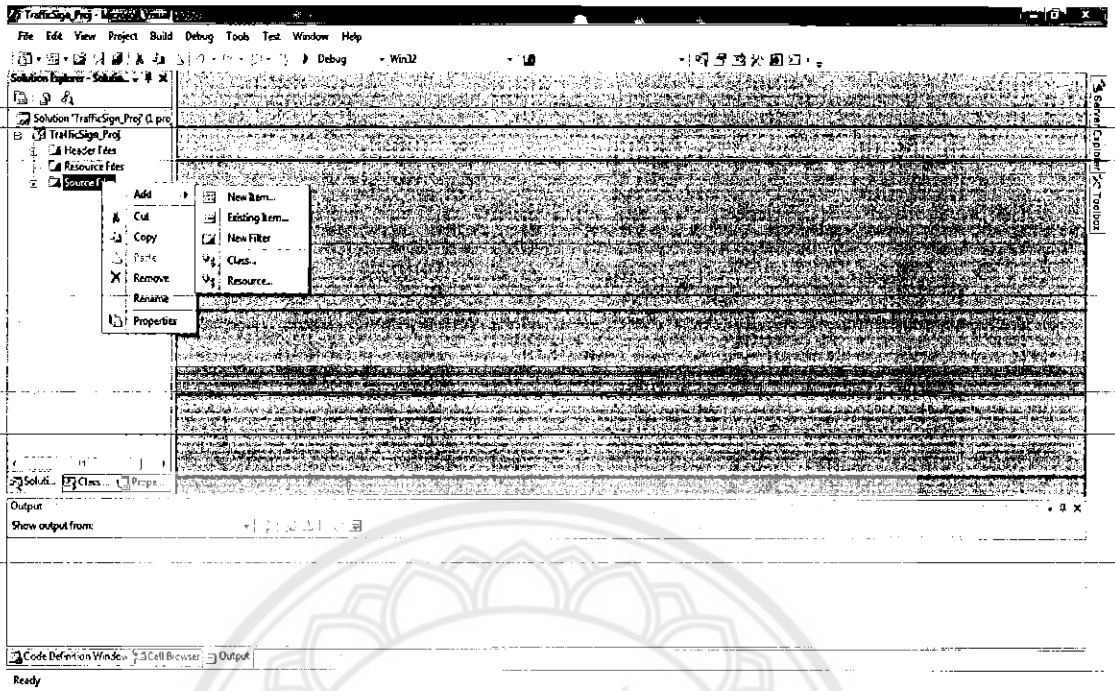
• เลือกที่หัวข้อ Other Languages -> Visual C++ ด้านขวาให้เลือก Win32 Console Application แล้วตั้งชื่อ Project ด้านล่าง ในช่อง Name



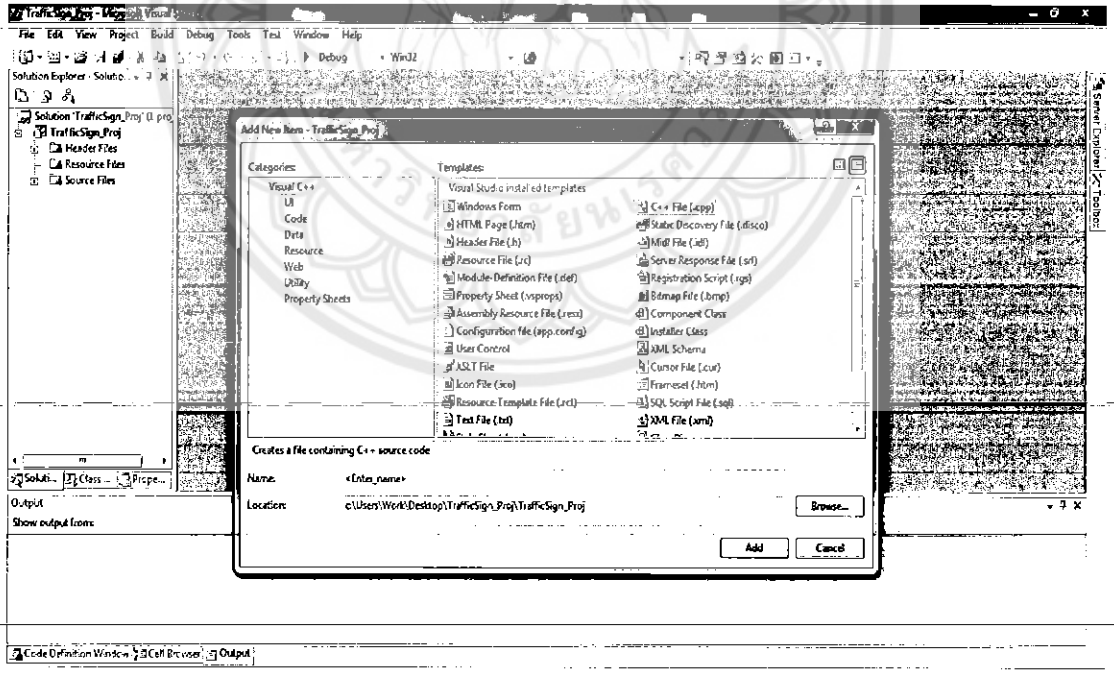

• Application type ให้เลือก Console application และAdditional options ให้เลือก Empty project



• คลิกขวาที่ Source File เมื่อต้องการเพิ่มโค้ดโปรแกรม-> Add -> New Item

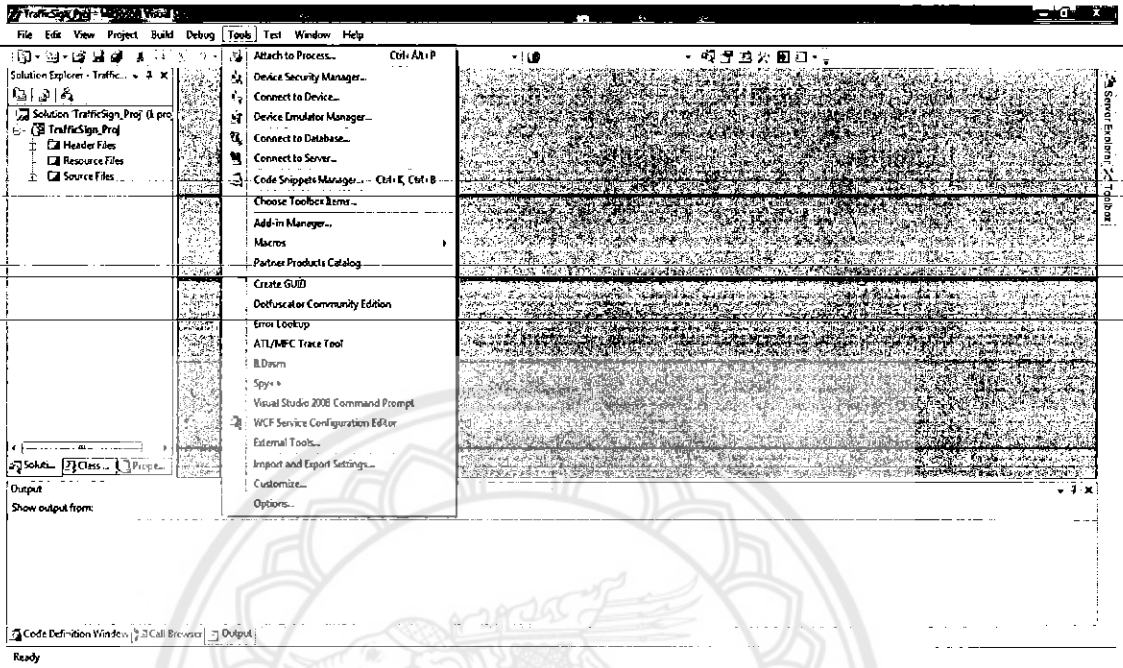



• คลิก Visual C++ เลือก C++ File (.cpp) ใส่ชื่อในช่อง Name กด Add

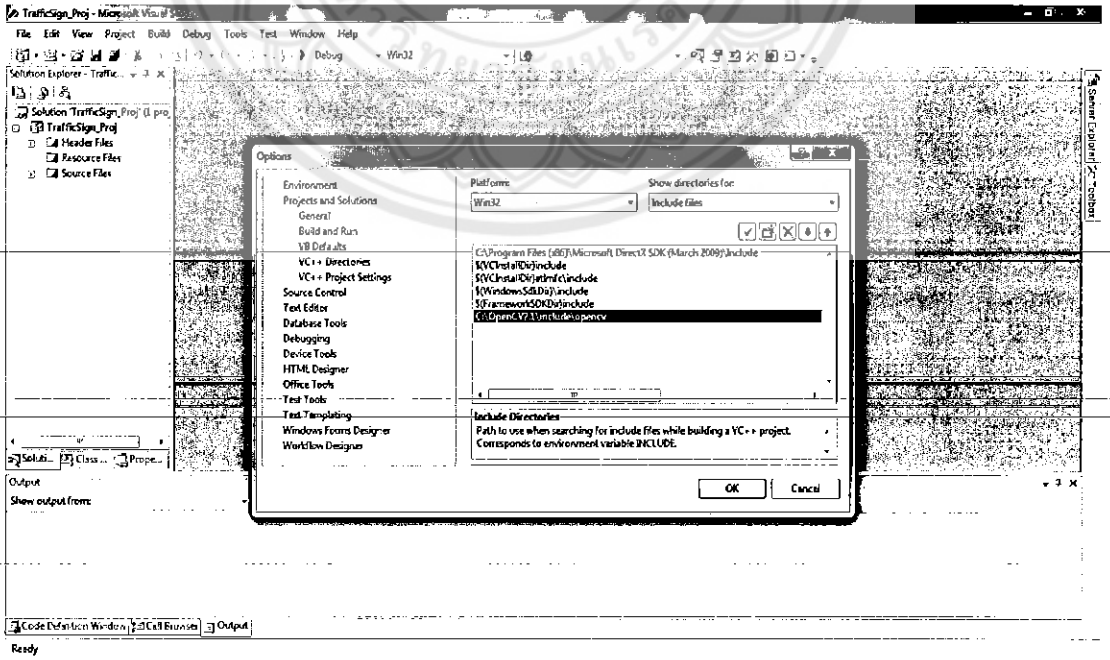


**ขั้นตอนการตั้งค่าเชื่อมต่อระหว่าง Visual Studio 2008 กับ openCV 2.1**

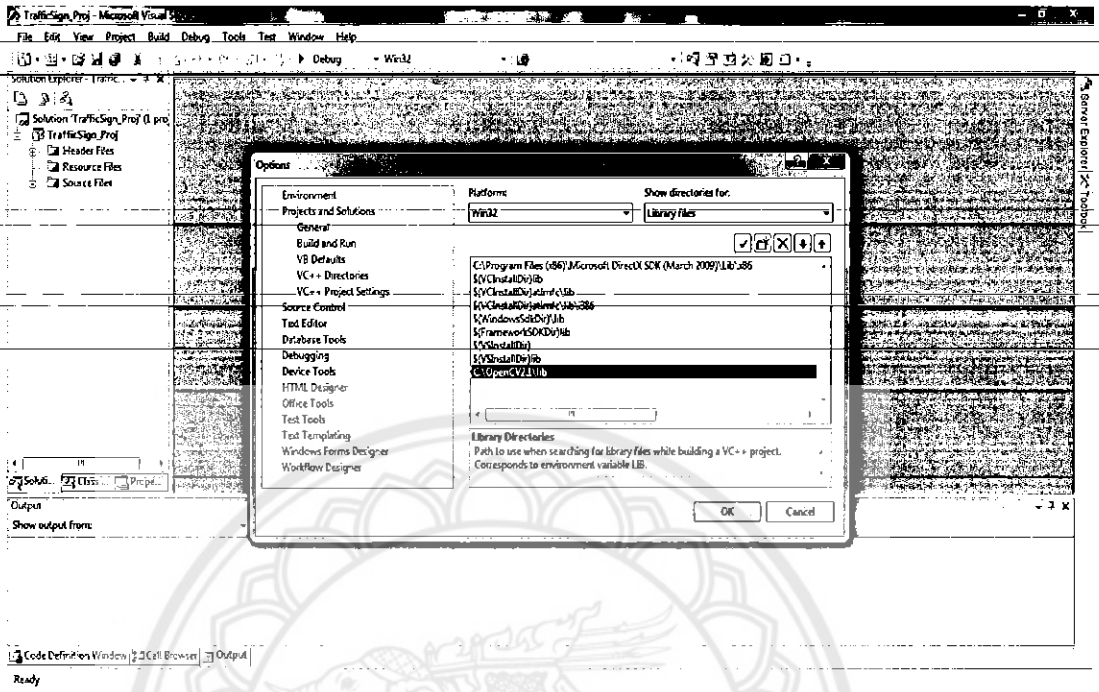
1. หลังจากสร้าง Project แล้วขั้นแรกในการตั้งค่า library ก็คือเลือก Tools ->Options…



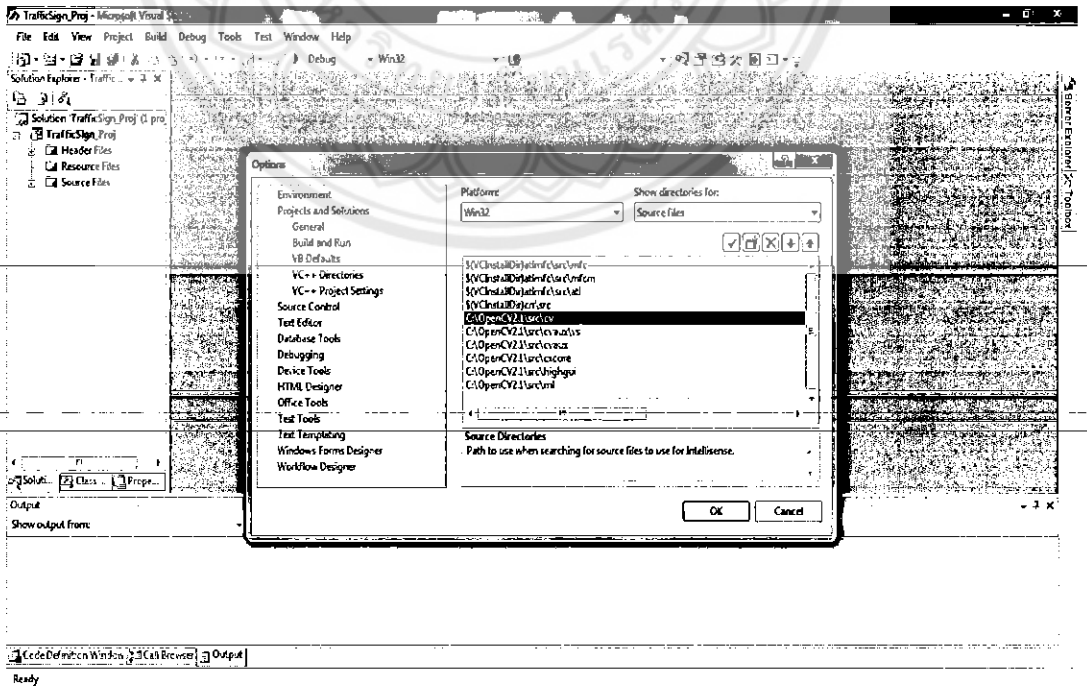

2. จากนั้นเข้าไปที่ VC++ Directory แล้วเลือก Include Files ที่ช่อง Show Directories for:: และเพิ่ม path file ที่อยู่ของ opencv2.1 include ดังรูป



3. เลือก library Files ที่ช่อง Show Directories for:: และเพิ่ม path file ที่อยู่ของ opencv2.1 lib ดังรูป

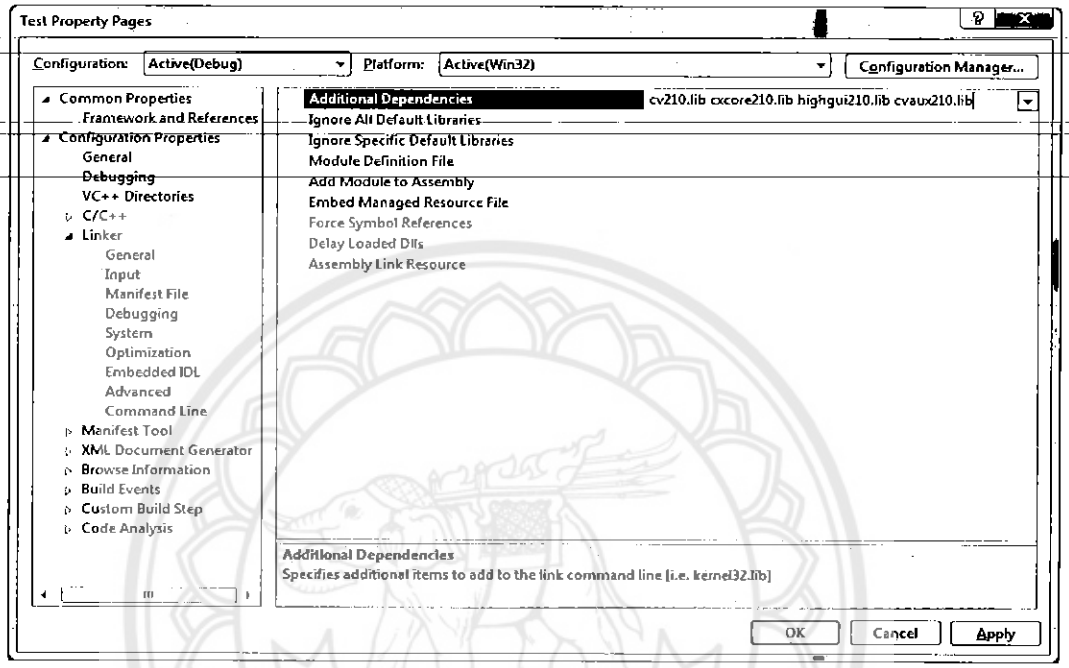



4. เลือก Source Files ที่ช่อง Show Directories for:: และเพิ่ม path file ที่อยู่ของ opencv2.1src ดังรูป



5. คลิกแถบ Project -> ชื่อไฟล์ Properties… โดยขั้นตอนนี้สำคัญมาจะต้องทำทุกครั้งเมื่อมี

• คลิกคำว่า Configuration Properties -> Linker -> Input ด้านขวาในช่อง Additional Dependencies ใส่ค่าในช่องด้านขวาว่า cv210.lib cxcore210.lib highgui210.lib cvaux210.lib (ตาม version)




• เสร็จสมบูรณ์

# ประวัติผู้ดำเนินโครงงาน

ชื่อ นายกฤษณรงค์ อุทธิยัง

ภูมิลำเนา 14 หมู่ 16 ต. ปง อ. ปง จ.พะเยา

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียน ปงรัชดาภิเษก
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4
  สาขาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
  คณะวิศวกรรมศาสตร์
  มหาวิทยาลัยมหาวิทยาลัยนเรศวร

E-mail: **kridnarong@hotmail.com**

ชื่อ นายภูวดล ยั้งระหัด

ภูมิลำเนา 111 หมู่ 1 ต. บ้านยาง อ. เกษตรสมบูรณ์ จ.ชัยภูมิ

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียน ภูเขียว
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4
  สาขาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
  คณะวิศวกรรมศาสตร์
  มหาวิทยาลัยมหาวิทยาลัยนเรศวร

E-mail: **phuvadonbom2@hotmail.com**