การใช้การมองเห็นของเครื่องจักรเพื่อวิเคราะห์สภาพการจราจร

นางสาวฐานิดา ศรีชัยเพชร

มหาวิทยาลัยเทคโนโลยีสุรนารี

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมเมคคาทรอนิกส์
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2563

ANALYSIS OF TRAFFIC CONDITION BY USING MACHINE VISION

Thanida Srichaipetch

A Thesis Submitted in Partial Fulfillment of the Requirement for the

Degree of Master of Engineering in Mechatronic Engineering

Suranaree University of Technology

Academic Year 2020

การใช้การมองเห็นของเครื่องจักรเพื่อวิเคราะห์สภาพการจราจร

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

(ผศ. ดร.โศรฎา แข็งการ)
ประธานกรรมการ

(รศ. ร.อ. ดร.กนต์ธร ชำนิประศาสน์)
กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)

(ศ. น.ท. ดร.สราวุฒิ สุจิตจร)
กรรมการ

(อ. ดร.พิจิตรา เอื้องไพโรจน์)
กรรมการ

(รศ. ร.อ. ดร.กนต์ธร ชำนิประศาสน์)
รองอธิการบดีฝ่ายวิชาการและพัฒนาความเป็นสากล

(รศ. ดร.พรศิริ จงกล)
คณบดีสำนักวิชาวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีสุรนารี

ฐานิดา ศรีชัยเพชร : การใช้การมองเห็นของเครื่องจักรเพื่อวิเคราะห์สภาพการจราจร
(ANALYSIS OF TRAFFIC CONDITION BY USING MACHINE VISION)
อาจารย์ที่ปรึกษา : รองศาสตราจารย์ เรืออากาศเอก ดร.กนต์ธร ชำนิประศาสน์, 50 หน้า.

ปัจจุบันในมหาวิทยาลัยมีนักศึกษาและบุคลากรเป็นจำนวนมาก ซึ่งการเดินทางต่าง ๆ จำเป็นต้องใช้ยานพาหนะทำให้มีการจราจรอย่างคับคั่งและยังมีการเพิ่มจำนวนยานพาหนะมากขึ้นเรื่อย ๆ ซึ่งทำให้เกิดสภาพการจราจรติดขัดอยู่บ่อยครั้งและอาจมีพื้นที่ในการรองรับยานพาหนะไม่เพียงพอ ซึ่งในงานวิจัยนี้ได้ทำการติดตั้งกล้องวงจรปิดบริเวณแยกทางเข้าอาคารเรียนรวม 1 มหาวิทยาลัยเทคโนโลยีสุรนารี และได้นำวิดีโอที่ถูกบันทึกไว้มาใช้ร่วมกับหลักการมองเห็นของเครื่องจักร เพื่อแยกภาพพื้นหลังออกจากวัตถุที่สนใจและทำการตรวจสอบขนาดของวัตถุนั้น ๆ เพื่อตรวจนับปริมาณยานพาหนะและแยกประเภทยานพาหนะ โดยแบ่งประเภทยานพาหนะออกเป็น 3 ประเภท ได้แก่ รถขนาดเล็ก รถขนาดกลาง และรถขนาดใหญ่ ผลที่ได้จากการทดลองโปรแกรมนับจำนวนยานพาหนะและแยกประเภทยานพาหนะเป็นจำนวน 50 วิดีโอ ได้ค่าความแม่นยำเฉลี่ยของรถขนาดเล็กทั้งหมดคิดเป็นร้อยละ 87.51 และค่าส่วนเบี่ยงเบนมาตรฐานของค่าความแม่นยำเท่ากับ 7.16 ค่าความแม่นยำเฉลี่ยของรถขนาดกลางทั้งหมดคิดเป็นร้อยละ 90.91 และค่าส่วนเบี่ยงเบนมาตรฐานของค่าความแม่นยำเท่ากับ 4.41 และค่าความแม่นยำเฉลี่ยของรถขนาดใหญ่ทั้งหมดคิดเป็นร้อยละ 93.72 และค่าส่วนเบี่ยงเบนมาตรฐานของค่าความแม่นยำเท่ากับ 5.71

มหาวิทยาลัยเทคโนโลยีสุรนารี

สาขาวิชา วิศวกรรมเมคคาทรอนิกส์ ลายมือชื่อนักศึกษา ฐานิดา ศรีชัยเพชร

ปีการศึกษา 2563 ลายมือชื่ออาจารย์ที่ปรึกษา กนต์ธร ชำนิประศาสน์

THANIDA SRICHAIPETCH : ANALYSIS OF TRAFFIC CONDITION BY USING MACHINE VISION. THESIS ADVISOR : ASSOC. PROF. FLT. LT. KONTORN CHAMNIPRASART, Ph.D., 50 PP.

MACHINE VISION/TRAFFIC/ANALYSIS

Nowadays, there is a large number of students and staffs at Suranaree University of Technology. Since vehicles are required for daily transportation therefore, the number of students and staffs is having a remarkable impact on the number of vehicles, traffic congestion, and parking availability within the university area. This research focuses on the analysis of traffic condition using machine vision in classifying and counting vehicle. Traffic videos were recorded by using an IP camera installed at the junction area (entrance) of the general inspection building (Building 1). Three types of vehicles were considered in this research: small, mid-size, and large vehicles. The collected videos were used in machine vision process to isolate the interested vehicle from the background and to detect its dimensions. The experimental results of vehicle counting and classification analyzed from 50 traffic videos show that the average accuracy (%) and standard deviation of small vehicles, mid-size vehicles, and large vehicles are 87.51 ± 7.61, 90.01 ± 4.41, and 93.72 ± 5.71, respectively.

School of Mechatronics Engineering

Academic year 2019

Student's Signature Thanida Srichaipetch

Advisor's Signature Kont Chyt

กิตติกรรมประกาศ

วิทยานิพนธ์เรื่องนี้สำเร็จลุล่วงไปได้ด้วยดีด้วยความกรุณาและความช่วยเหลืออย่างดียิ่งจาก รองศาสตราจารย์ เรืออากาศเอก ดร.กนต์ธร ชำนิประศาสน์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ได้ประสิทธิ์ประสาทความรู้ทั้งในด้านการดำเนินงานวิจัยและด้านการดำรงชีวิต ตลอดทั้งช่วยตรวจแก้ไขข้อบกพร่องต่าง ๆ ด้วยความเอาใจใส่และเมตตาเป็นอย่างสูง ตลอดเวลาที่ได้ดำเนินงานวิจัยและให้กำลังใจแก่ผู้วิจัยตลอดมา ผู้วิจัยรู้สึกซาบซึ้งในพระคุณของอาจารย์เป็นอย่างยิ่ง

ขอกราบขอบพระคุณ คณะกรรมการสอบป้องกันวิทยานิพนธ์ทุกท่านได้เสียสละเวลาอันมีค่าเพื่อให้คำแนะนำและชี้แนวทางในการแก้ไข พร้อมให้ข้อเสนอแนะอันเป็นประโยชน์ในการปรับปรุงวิทยานิพนธ์จนวิทยานิพนธ์มีความสมบูรณ์ถูกต้องยิ่งขึ้น

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.ปรเมศวร์ ห่อแก้ว และอาจารย์ ดร.วรวัฒน์ ลวนนท์ ที่ให้ความรู้ในด้านวิชาการที่เป็นประโยชน์ในการดำเนินงานวิจัย ซึ่งผู้วิจัยนำมาประยุกต์ใช้จนทำให้งานวิจัยนี้สำเร็จได้

ขอขอบพระคุณ เจ้าหน้าที่ประจำสำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ที่ได้อำนวยความสะดวกในด้านเอกสารต่าง ๆ

สุดท้ายนี้ ผู้วิจัยขอกราบขอบพระคุณบิดา มารดา และสมาชิกครอบครัวศรีชัยเพชร สำหรับแรงบันดาลใจในการทำวิทยานิพนธ์และให้การอุปการะเลี้ยงดูด้วยความรักความอบอุ่นให้การสนับสนุนและเป็นกำลังใจให้กับผู้วิจัย จนสามารถจัดทำวิทยานิพนธ์สำเร็จลุล่วงได้ด้วยดี

ฐานิดา ศรีชัยเพชร

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย) ..... ก
บทคัดย่อ (ภาษาอังกฤษ) ..... ข
กิตติกรรมประกาศ ..... ค
สารบัญ ..... ง
สารบัญตาราง ..... ฉ
สารบัญรูป ..... ช

บทที่

1 บทนำ 1
1.1 ที่มาและความสำคัญของปัญหาการวิจัย 1
1.2 วัตถุประสงค์ของการวิจัย 1
1.3 ขอบเขตของการวิจัย 1
1.4 ระเบียบวิธีวิจัย 2
1.5 ประโยชน์ที่คาดว่าจะได้รับ 2

2 ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง 3
2.1 กล่าวนำ 3
2.2 ปริทัศน์วรรณกรรม 3
2.3 ค่าขอบเขตการมองเห็น (Field of View, FOV) 4
2.4 การมองเห็นด้วยเครื่องจักร (Machine Vision) 4
2.4.1 การจัดสภาพแวดล้อม (Scene Constraint) 4
2.4.2 การดึงข้อมูลภาพ (Image Acquisition) 5
2.4.3 การประมวลผลภาพ (Image Processing) 7
2.4.4 การแยกบริเวณ (Segmentation) 8
2.4.5 การคำนวณหาคุณสมบัติของวัตถุ (Feature Extraction) 11

สารบัญ (ต่อ)

หน้า

2.4.6 การจำแนกวัตถุและการแปลความหมาย (Classification and Interpretation) 11
2.4.7 กลไกการเคลื่อนไหว (Actuation) 11
2.5 ค่าเริ่มเปลี่ยน (Threshold) 12
2.6 การลบพื้นหลัง (Background Subtraction) 18
3 วิธีดำเนินการวิจัย 19
3.1 กล่าวนำ 19
3.2 การเก็บข้อมูลบริเวณแยกการจราจรทางเข้าอาคารเรียนรวม 1 21
3.3 การออกแบบระบบแยกวัตถุออกจากพื้นหลัง 21
3.4 ออกแบบระบบแยกประเภทยานพาหนะ 23
3.5 การออกแบบระบบนับจำนวนยานพาหนะ 27
4 ผลการดำเนินการวิจัยและการอภิปรายผล 30
4.1 กล่าวนำ 30
4.2 ผลการตรวจสอบความถูกต้องของโปรแกรม 30
5 สรุปผลการวิจัย 40
5.1 กล่าวนำ 40
5.2 สรุปผลการวิจัย 40
รายการอ้างอิง 41
ภาคผนวก
ภาคผนวก ก. การเขียนโปรแกรมควบคุมการทำงานของระบบ 42
ประวัติผู้เขียน 50

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญตาราง

ตารางที่ **หน้า**

4.1 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดเล็ก วันที่ 1.....30
4.2 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดเล็ก วันที่ 2.....31
4.3 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดเล็ก วันที่ 3.....31
4.4 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดเล็ก วันที่ 4.....32
4.5 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดเล็ก วันที่ 5.....32
4.6 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดกลาง วันที่ 1.....33
4.7 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดกลาง วันที่ 2.....34
4.8 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดกลาง วันที่ 3.....34
4.9 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดกลาง วันที่ 4.....35
4.10 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดกลาง วันที่ 5.....35
4.11 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดใหญ่ วันที่ 1.....36
4.12 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดใหญ่ วันที่ 2.....37
4.13 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดใหญ่ วันที่ 3.....37
4.14 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดใหญ่ วันที่ 4.....38
4.15 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดใหญ่ วันที่ 5.....38

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญรูป

รูปที่ **หน้า**

2.1 ความเข้มแสงเทียบกับค่าที่ได้จากเซลล์รับภาพ 5
2.2 สมการของอาร์เรย์ 2 มิติ 6
2.3 ผลของการแยกบริเวณออกเป็นส่วน ๆ 7
2.4 ผลการทำงานกระบวนการ Connected Components Labeling 10
2.5 ค่าของพิกเซล 12
2.6 ภาพหลังผ่านการทำ Gray Scale 13
2.7 ภาพหลังการคำนวน Threshold 13
2.8 ฟังก์ชัน Threshold ของแต่ละแบบ 14
2.9 ภาพเริ่มต้นการทำงานฟังก์ชัน Threshold 14
2.10 ภาพ Gray Scale 14
2.11 รูปหลังจากใช้ Threshold Binary ในการคำนวน 15
2.12 รูปหลังจากใช้ Threshold Binary, Inverted ในการคำนวน 16
2.13 รูปหลังจากใช้ Truncate ในการคำนวน 16
2.14 รูปหลังจากใช้ Threshold to Zero ในการคำนวน 17
2.15 รูปหลังจากใช้ Threshold to Zero, Inverted ในการคำนวน 17
3.1 แผนผังขั้นตอนการทำการวิจัย 19
3.2 บริเวณที่เก็บข้อมูล 21
3.3 บริเวณที่เก็บข้อมูลขณะที่มียานพาหนะเคลื่อนที่ผ่าน 21
3.4 ภาพแบบ Gray Scale 22
3.5 ภาพการใช้ฟังก์ชัน BackgroundSubtractorMOG2() 22
3.6 ภาพที่ทำการลบเงาและลบสัญญาณรบกวนบนภาพ 23
3.7 ภาพที่ทำการลบเงาและลบสัญญาณรบกวนบนภาพของรถขนาดเล็ก 24
3.8 ภาพที่ทำการลบเงาและลบสัญญาณรบกวนบนภาพของรถขนาดกลาง 24
3.9 ภาพที่ทำการลบเงาและลบสัญญาณรบกวนบนภาพของรถขนาดใหญ่ 25

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.10 การวาดเส้นบนเฟรมของภาพ....25
3.11 ภาพที่ได้จากการขอบเขตของพาหนะ....26
3.12 ค่าขนาดของวัตถุที่สนใจ....27
3.13 การตีกรอบและติดตามยานพาหนะ....28
3.14 ค่าที่ได้จากการนับยานพาหนะ....29
3.15 ยานพาหนะที่สนใจ....29

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหาการวิจัย

ในมหาวิทยาลัยนั้นเป็นสถานที่ที่มีปริมาณยานพาหนะจำนวนมาก ซึ่งอาคารแต่ละอาคารนั้นมีพื้นที่จำกัดในการรองรับปริมาณยานพาหนะ และการขับขี่ยานพาหนะนั้นอาจทำให้เกิดปัญหาการจราจรต่าง ๆ ซึ่งมาจากพฤติกรรมการขับขี่และปริมาณยานพาหนะที่เพิ่มขึ้น

จากที่กล่าวมาข้างต้น ผู้วิจัยจึงเล็งเห็นว่าการนำหลักการการมองเห็นด้วยเครื่องจักร (Machine vision) มาใช้ในการนับจำนวนยานพาหนะและแยกประเภทยานพาหนะทำให้ทราบถึงปริมาณของยานพานหนะ และทำให้สามารถคาดการณ์ในช่วงเวลาที่มีการจราจรหนาแน่นว่าจะมียานพานหนะผ่านเข้าสู่แยกในปริมาณมากน้อยเพียงใด ซึ่งงานวิจัยนี้อาจจะสามารถพัฒนาไปใช้ร่วมกับสัญญาณไฟจราจรเพื่อที่จะจัดการกับสภาพการจราจรให้ดีขึ้น

งานวิจัยนี้เป็นการศึกษาการทำงานของระบบการมองเห็นของเครื่องจักร (Machine vision) โดยการวิเคราะห์ภาพเคลื่อนไหวในทุก ๆ ช่วงเวลา และนำมาหาค่าเฉลี่ยของการเปลี่ยนแปลงวัตถุกับภาพพื้นหลังที่ผ่านเข้ามา โดยค่าเฉลี่ยที่ได้จะเป็นตัวบ่งบอกความแตกต่างของวัตถุกับภาพพื้นหลัง และยังสามารถนำค่าเฉลี่ยที่ได้มาวิเคราะห์ประเภทของยานพาหนะ และนับจำนวนยานพาหนะได้

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 นับจำนวนยานพาหนะที่วิ่งเข้าสู่แยกจราจรในแต่ละทิศทางในช่วงเวลานั้น ๆ โดยใช้หลักการมองเห็นด้วยเครื่องจักร

1.2.2 จำแนกประเภทยานพาหนะ รถขนาดเล็ก รถขนาดกลาง และรถขนาดใหญ่

1.3 ขอบเขตของการวิจัย

1.3.1 นับจำนวนยานพาหนะที่วิ่งเข้าสู่แยกเข้าอาคารเรียนรวม 1 ในทัศนวิสัยที่ปกติ ช่วงเวลา 08.00 น. ถึง 17.00 น. โดยใช้หลักการมองเห็นด้วยเครื่องจักร

1.3.2 แยกประเภทยานพาหนะว่าเป็น รถขนาดเล็ก รถขนาดกลาง และรถขนาดใหญ่

มหาวิทยาลัยเทคโนโลยีสุรนารี

1.4 ระเบียบวิธีวิจัย

1.4.1 รวบรวมข้อมูลงานวิจัยที่เกี่ยวข้อง

1.4.2 เก็บข้อมูลสภาวะการจราจรบริเวณแยกเข้าอาคารเรียนรวม 1

1.4.3 วิเคราะห์และสรุปข้อมูลการจราจร

1.4.4 ออกแบบและทดสอบการทำงานของโปรแกรมแยกวัตถุกับพื้นหลัง

1.4.5 ออกแบบและทดสอบการทำงานของโปรแกรมแยกประเภทยานพาหนะ

1.4.6 ออกแบบและทดสอบการทำงานของโปรแกรมตรวจนับจำนวนยานพาหนะ

1.4.7 รวบรวมข้อมูลและสรุปผลการทดลอง

1.4.8 จัดทำวิทยานิพนธ์

1.4.9 สอบวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 โปรแกรมที่ใช้การมองเห็นด้วยเครื่องจักรมานับจำนวนยานพาหนะที่วิ่งเข้าสู่แยกการจราจรในช่วงเวลานั้น ๆ

1.5.2 ต้นแบบระบบที่สามารถแยกประเภทยานพาหนะและนับจำนวนยานพาหนะได้

1.5.3 ลดค่าใช้จ่ายทางด้านซอฟแวร์

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 2
ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

2.1 กล่าวนำ

งานวิจัยนี้มุ่งเน้นศึกษาการทำงานของระบบการมองเห็นด้วยเครื่องจักรที่วิเคราะห์ภาพเคลื่อนไหวในทุก ๆ ช่วงเวลา เพื่อให้ได้ค่าความแตกต่างระหว่างพื้นหลังกับวัตถุ และนำค่ามาประมวลผลแยกประเภทยานพาหนะและนับจำนวนยานพาหนะที่วิ่งเข้าสู่แยกการจราจร เพื่อให้ทราบจำนวนประชากรที่เข้าสู่อาคารเรียนรวมในแต่ละช่วงเวลา

2.2 ปริทัศน์วรรณกรรม

สนั่น งานวิวัฒน์ถาวร (2551) ได้ศึกษาระบบ Motion Detection ที่ไว้สำหรับตรวจสอบการเคลื่อนไหวของวัตถุที่สามารถแบ่งเป็น 2 ประเภท คือ Background Subtraction และ Optical Flow และศึกษาเปรียบเทียบข้อดีและข้อด้อยของแต่ละวิธี รวมถึงทดสอบการทำงานของระบบร่วมกับกล้อง IP Camera รวมไปถึงศึกษาการใช้งาน Library OpenCV

Sheeraz Memon (2018) ศึกษานี้ระบบการนับและจำแนกยานพาหนะ โดยที่ระบบเกี่ยวข้องกับการจับเฟรมภาพจากวิดีโอและทำการลบพื้นหลังเพื่อตรวจจับและนับยานพาหนะ โดยใช้การลบพื้นหลังด้วยวิธี Gaussian Mixture Model (GMM) จากนั้นจะจำแนกยานพาหนะโดยการเปรียบเทียบพื้นที่รูปร่างกับค่าที่สันนิษฐาน โดยการเปรียบเทียบวิธีการจำแนกสองวิธี คือ การจำแนกประเภทโดยใช้ Contour Comparison (CC) เช่นเดียวกับ Bag of Features (BoF) และ Support Vector Machine (SVM) method

ไตรวิทย์ อินทจักร (2555) ได้เสนอระบบตรวจจับและคัดแยกประเภทรถและพัฒนากระบวนการหลากหลายรูปแบบมีทั้งการลบภาพพื้นหลังที่มีการวิเคราะห์ปริมาณแสง การกรองเงาของวัตถุ การตรวจจับขอบของวัตถุ และการคัดแยกขอบของวัตถุเป็นยานพาหนะประเภทต่าง ๆ โดยพัฒนาโปรแกรมและทดสอบการทำงานบนซอฟต์แวร์ MATLAB

K.V. Arya (2016) ได้เสนอวิธีตรวจตรวจสอบยานพาหนะ โดยการจับการรวมกลุ่มของพิกเซลบนรูปภาพ ซึ่งอาจเป็นไปได้ว่านั่นคือยานพาหนะ และส่วนที่เหลือจากการเกาะกลุ่มของพิกเซลนั้นถือว่าเป็นภาพพื้นหลัง และได้ทำการเปรียบเทียบการทำงานของวิธีมาตรฐานทั่วไป

2.3 ค่าขอบเขตการมองเห็น (Field of View, FOV)

ค่าขอบเขตการมองเห็นเป็นสมบัติที่แสดงถึงพื้นที่การมองเห็นวัตถุของกล้องถ่ายภาพความร้อน (Thermal Image Camera, Infrared Thermography) แสดงค่าตามมุมที่กล้องมองเห็น หากระยะทางการวัดเปลี่ยนจะส่งผลต่อการเปลี่ยนแปลงพื้นที่การรับรังสีอินฟราเรด (Infrared) หรือบริเวณการมองเห็นด้วย โดยถ้ากล้องถ่ายภาพความร้อนกับวัตถุมีระยะห่างที่อยู่ใกล้กัน กล้องถ่ายภาพความร้อนจะสามารถรับรายละเอียดของพื้นที่หรือบริเวณการมองเห็นของวัตถุได้แคบกว่าระยะระหว่างกล้องกับวัตถุที่ห่างกันมาก เช่นเดียวกับกรณีของเทอร์โมมิเตอร์ชนิดอินฟราเรด (Infrared Thermometer) โดยภายใน FOV มีค่า IFOV (Instantaneous Field of View เป็นหน่วยย่อย ซึ่งแสดงถึงพื้นที่ที่เล็กที่สุด

2.4 การมองเห็นด้วยเครื่องจักร (Machine Vision)

ในส่วนนี้จะกล่าวถึงหลักการมองเห็นด้วยเครื่องจักร การจัดสภาพแวดล้อม วิธีการดึงข้อมูลภาพเข้าสู่คอมพิวเตอร์ การประมวลผลภาพเบื้องต้นที่มีการทำงานหลายกระบวนการรวมไปถึงการแยกบริเวณของภาพที่มีลักษณะร่วมกันออกเป็นส่วน ๆ

เกริกพงษ์ เกียรติพานิชกิจ (2550) การมองเห็นด้วยเครื่องจักร (Machine Vision) เป็นวิธีการที่ทำให้เครื่องจักรหรืออุปกรณ์ประมวลผลมีความสามารถในการรับรู้ภาพ สามารถตัดสินใจและสั่งงานกลไกส่วนต่าง ๆ จากข้อมูลที่ได้จากภาพนั้น ๆ ซึ่งมีการพัฒนามายาวนานเพื่อที่จะทำให้เครื่องจักรหรืออุปกรณ์ประมวลผลมีความสามารถในการมองเห็นเทียบเท่ามนุษย์แต่ยังไม่ประสบความสำเร็จ แม้อุปกรณ์ประมวลผลจะมีความเร็วในการคำนวณผลทางคณิตศาสตร์มากกว่ามนุษย์ เนื่องจากความแตกต่างกันระหว่างการทำงานของอุปกรณ์ประมวลผลและสมองมนุษย์ โดยการทำงานของอุปกรณ์ประมวลผลจะเป็นการทำงานแบบทีละขั้นตอนซึ่งไม่เหมือนมนุษย์ที่มีหน่วยย่อยของสมองจำนวนมหาศาลและทำงานไปพร้อม ๆ กัน แม้ว่ายังไม่สามารถพัฒนาให้เทียบเท่าการมองเห็นของมนุษย์ได้ แต่ก็สามารถนำไปใช้ได้กับงานหลากหลายรูปแบบ เช่น การนำไปใช้ในการมองเห็นของหุ่นยนต์ การนำไปตรวจสอบชิ้นงานด้วยภาพอัตโนมัติในเชิงอุตสาหกรรม การนับจำนวนชิ้นส่วนต่างๆ บนผลิตภัณฑ์ เป็นต้น จากความต้องการในงานเชิงอุตสาหกรรมที่ ต้องการให้ระบบสามารถทำงานโดยใช้ระยะเวลาสั้นที่สุด จึงจำเป็นต้องนำความรู้ที่ผู้ปฏิบัติงานมีอยู่เข้ามาช่วยเพื่อให้ระบบทำงานได้อย่างดีที่สุด ส่วนประกอบโดยทั่วไปของการมองเห็นด้วยเครื่องจักรมีดังนี้

2.4.1 การจัดสภาพแวดล้อม (Scene Constraint)

การจัดสภาพแวดล้อมเพื่อลดความซับซ้อนในการประมวลผล ลดเวลาในการทำงาน เนื่องจากความสามารถในการมองเห็นของเครื่องจักรมีข้อจำกัดและไม่เท่าเทียม เราจึงต้องช่วยลด

ความยุ่งยากในการประมวลผล ซึ่งสามารถทำได้หลายวิธี เช่น

- การจัดการกับวัตถุ เราต้องจัดการกับวัตถุให้วางไปในทิศทางเดียวกัน เพื่อง่ายต่อการตรวจสอบซึ่งถ้าไม่มีการจัดเรียงวัตถุจะทำให้อุปกรณ์ประมวลผลต้องหาทิศทางของวัตถุในแต่ละชิ้นเอง
- กำหนดระยะระหว่างเลนส์กล้องถึงวัตถุ และทิศทางของกล้อง สิ่งเหล่านี้เป็นการกำหนดขนาดของวัตถุ เช่น ถ้าวางระยะกล้องอยู่ใกล้วัตถุก็จะทำให้เห็น วัตถุใหญ่ขึ้น ในการทำงานค่าระยะห่างควรกำหนดไว้ให้ตายตัว
- การจัดการเรื่องแสงเป็นองค์ประกอบที่สำคัญมาก เนื่องจากการมองเห็นด้วยเครื่องจักรเกิดจากแสงที่มาตกกระทบกับตัววัตถุ แล้วสะท้อนผ่านเลนส์เข้ามาที่เซนเซอร์รับภาพของกล้อง ซึ่งต้องพิจารณาทั้งเรื่องแหล่งกำเนิดแสง การกระเจิงของแสง และคุณสมบัติอื่น ๆ

นอกจากการจัดสภาพแวดล้อมแล้ว งานบางประเภทจำเป็นต้องใช้ภาพจากกล้องจำนวนหลาย ๆ ตัว เพื่อตรวจสอบในหลายมุมมอง หรือเก็บวัตถุชนิดเดียวกันที่ได้จากแหล่งกำเนิดแสงหลายประเภท

2.4.2 การดึงข้อมูลภาพ (Image Acquisition)

การดึงข้อมูลภาพเป็นกระบวนการที่เริ่มตั้งแต่การถ่ายภาพโดยกล้องจนถึงการดึงภาพเข้าสู่คอมพิวเตอร์ เพื่อประมวลผลและสั่งงานต่อไป โดยมีรายละเอียดที่สำคัญดังนี้

- ประเภทของกล้อง กล้องดิจิตอลในปัจจุบันนั้นมีเซนเซอร์รับภาพ (Image Sensor) เป็นสารกึ่งตัวนำ ใช้ในการรับภาพ ในเซนเซอร์จะประกอบด้วยไดโอดที่ถูกเรียกว่า “เซลล์รับภาพ” ที่มีความไวต่อแสงเรียงอยู่เป็นจำนวนมาก ไดโอดแต่ละตัวทำหน้าที่ในการจดจำความเข้มหรือความสว่างของแสงที่ตกกระทบแต่ละตัวไว้และจะถูกแปลงให้อยู่ในรูปข้อมูลดิจิตอลเก็บไว้ในหน่วยความจำกล้อง ซึ่งไดโอดนี้เรียกว่า “เซลล์รับภาพ” หนึ่งเซลล์รับภาพจะให้ค่าความเข้มแสงที่ตกกระทบเพียงหนึ่งค่าเท่านั้น เซลล์รับค่าทั่วไปมีค่าอยู่ระหว่าง 0 ถึง 255 ซึ่งทำให้ได้ภาพโทนขาวดำ (Gray Scale Image) เท่านั้น กรณีรับภาพสีต้องมีการติดตั้งตัวกรองแสงสี (Filter) แต่ละสีไว้หน้าเซลล์รับภาพ ซึ่งสามารถแยกสีได้เพียง 3 สี ได้แก่ สีแดง สีเขียว และสีน้ำเงิน

รูปที่ 2.1 ความเข้มแสงเทียบกับค่าที่ได้จากเซลล์รับภาพ

กล้องสำหรับตรวจชิ้นงานแบ่งเป็น 2 ประเภท คือ กล้อง Area Scan ที่เซลล์รับภาพเรียงตัวกันเป็นพื้นที่ แต่ละเซลล์ทำการแปลงค่าความเข้มแสงพร้อม ๆ กัน แต่จะได้ภาพที่มีความละเอียดน้อย และกล้อง Line Scan ที่เซลล์รับภาพเรียงตัวเป็นแถวยาวจำนวนมากถึง 12,000 เซลล์ ทำให้สามารถจับภาพของทั้งวัตถุได้ ซึ่งกล้องต้องมีการเคลื่อนที่สัมพันธ์กับวัตถุทำให้มีได้ความละเอียดของภาพที่สูงมาก

- ภาพที่อุปกรณ์ประมวลผลมองเห็น มีหลักการมองเห็นของกล้องเช่นเดียวกับการมองเห็นของมนุษย์ คือ ภาพเกิดจากแสงตกระทบวัตถุแล้วมีแสงสะท้อนจากวัตถุผ่านเลนส์เข้ามาตกกระทบเซนเซอร์รับภาพ (Image Sensor) ของกล้อง ซึ่งประกอบด้วยเซลล์รับภาพ (Pixel) จำนวนมากที่ทำหน้าที่แปลงความเข้มแสงให้อยู่ในรูปของสัญญาณแรงดันไฟฟ้า และถูกแปลงเป็นสัญญาณดิจิตอลด้วยตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอลอีกที โดยการ ทำงานจะแตกต่างจากการมองเห็นของมนุษย์ 2 ประการด้วยกัน คือ

1. จำนวนเซลล์รับภาพที่ประกอบกันขึ้นมาเป็นเซนเซอร์รับภาพของกล้องมีจำนวนน้อยกว่าของมนุษย์ ทำให้ได้ภาพที่มีความละเอียดน้อยกว่าของมนุษย์
2. ค่าความเข้มแสงที่ได้จากเซลล์รับภาพของเซนเซอร์รับภาพที่อยู่ในกล้องจะเป็นค่าไม่ต่อเนื่องเพราะเป็นการทำงานของอุปกรณ์ดิจิตอลที่สุ่มขนาดของความเข้มแสงที่ตกกระทบ

จากข้อจำกัดดังกล่าว ทำให้ข้อมูลภาพ Gray Scale ที่อุปกรณ์ประมวลผลมองเห็นเป็นอาร์เรย์ 2 มิติ โดยที่ค่าแต่ละช่องของอาร์เรย์จะแทนความเข้มแสงหรือความเข้มของแสงที่ตรงกระทบเซลล์รับภาพที่ตำแหน่งนั้น ซึ่งค่าความเข้มแสงดังกล่าวจะเป็นค่าที่ไม่ต่อเนื่องและโดยทั่วไปมีค่าระหว่าง 0 ถึง 255 เท่านั้น ดังสมการรูปที่ 2.2 ซึ่งเป็นตัวอย่างภาพที่มีจำนวนแถวหรือความสูงของภาพเท่ากับ m แถว และมีจำนวนหลักหรือความกว้างของภาพเท่ากับ n หลัก

$$\text{Image} = \begin{bmatrix} I(1,1) & I(1,2) & . & . & I(1,n) \\ I(2,1) & . & & & . \\ . & & . & & . \\ . & & & . & . \\ I(m,1) & I(m,2) & . & . & I(m,n) \end{bmatrix}$$

รูปที่ 2.2 สมการของอาร์เรย์ 2 มิติ

มหาวิทยาลัยเทคโนโลยีสุรนารี

เมื่อ I (m, n) คือ ค่าความเข้มแสง ณ แถว m และหลัก n ของเซนเซอร์รับภาพซึ่งเป็นค่าไม่ต่อเนื่อง

วิธีดึงภาพข้อมูลภายในระบบปฏิบัติการวินโดวส์ มีอยู่ด้วยกัน 3 วิธี คือ

1. การดึงข้อมูลโดยการใช้เครื่องมือในการโปรแกรมที่ผู้ผลิตกล้องให้มา เครื่องมือดังกล่าวมีอยู่ 2 รูปแบบ คือ dll และ activeX

2. การดึงข้อมูลโดยการใช้เครื่องมือในการโปรแกรมของระบบปฏิบัติการวินโดวส์ เช่น Video for Window (VFW) เป็นเครื่องมือที่มีมาให้ในระบบปฏิบัติการวินโดวส์อยู่แล้ว มีการใช้งานมายาวนาน ทำให้ข้อบกพร่องปัญหาต่าง ๆ ของโปรแกรมได้ถูกแก้ไขไปเกือบหมด การใช้งานเครื่องมือในการโปรแกรมของระบบปฏิบัติการวินโดวส์สามารถใช้งานได้อย่างยืดหยุ่นมากที่สุด ซึ่งทำให้สามารถเข้าถึงหน่วยความจำทั้งก่อนและหลังการแสดงภาพ

3. การดึงข้อมูลภาพโดยการใช้เครื่องมือในการโปรแกรมที่อยู่ในรูป dll หรือ activeX เครื่องมือโปรแกรมลักษณะนี้ถูกออกแบบให้มีความสามารถหลาย ๆ อย่าง เช่น การดึงข้อมูลภาพ การประมวลผลภาพเบื้องต้น โปรแกรมลักษณะนี้สามารถใช้งานได้ง่ายแต่ขาดความยืดหยุ่นในการปรับแก้การทำงาน

2.4.3 การประมวลผลภาพ (Image Processing)

การประมวลผลภาพมีหลายกระบวนการด้วยกัน ซึ่งเป็นกระบวนการที่สามารถพบได้ในสาขาเรื่อง การประมวลภาพดิจิตอล (Digital Image Processing) เช่น

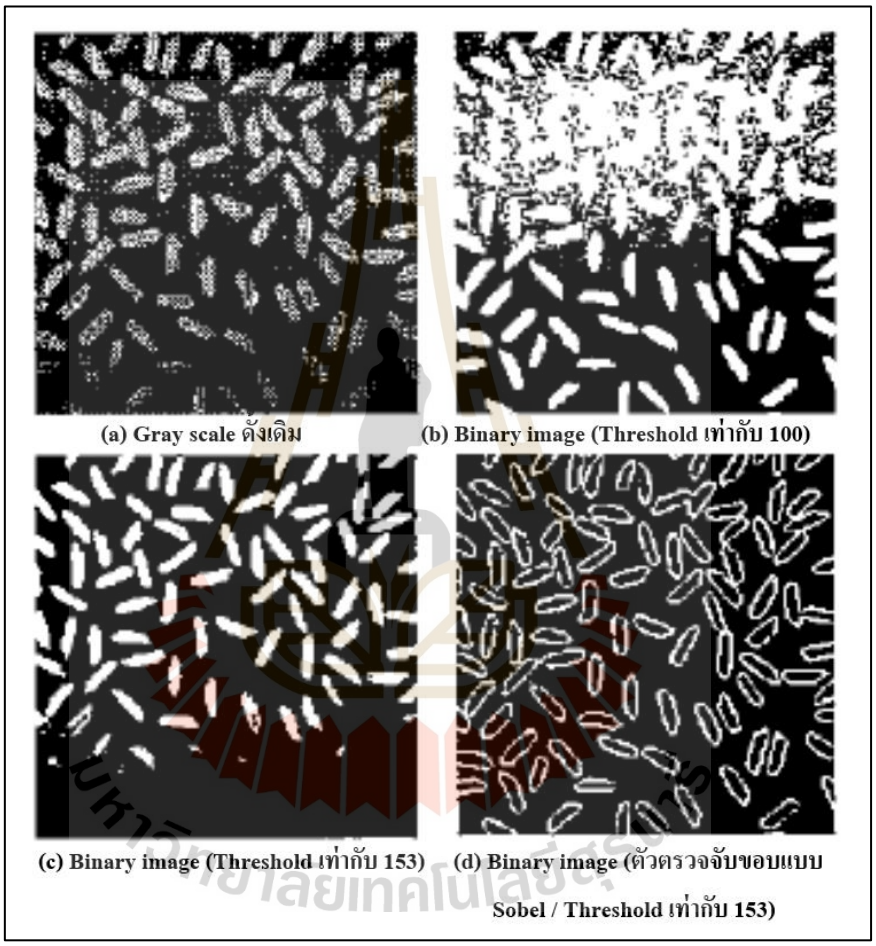
- การลดทอนสัญญาณรบกวนที่ปรากฏขึ้นในภาพ
- การตรวจจับขอบของวัตถุที่อยู่ในภาพ
- การแปลงคุณสมบัติทางกายภาพของภาพ
- การแปลงสี
- การบีบอัดข้อมูลภาพ และอื่น ๆ อีกมากมาย

ซึ่งพบว่า อัลกอริทึม (Algorithm) ของการประมวลผลภาพบางอย่างก็ไม่เหมาะสมกับการตรวจสอบงานแบบอัตโนมัติ เนื่องจากการทำงานส่วนใหญ่ต้องการการทำงานที่รวดเร็วที่สุด จึงจำเป็นต้องหาใช้วิธีที่ง่ายและใช้เวลาน้อยที่สุด ซึ่งสามารถทำได้ด้วยการจัดสภาพแวดล้อมในการจับภาพที่ดี เช่น

- การระบุบริเวณที่ต้องการตรวจสอบ (Region of Interest, ROI) การระบุเฉพาะบริเวณที่สนใจรวมทั้งรูปแบบของการวัดหรือการตรวจสอบที่ใช้กับบริเวณนั้น ๆ ทำให้โปรแกรมสามารถตัดการคำนวณของบริเวณที่ไม่เกี่ยวข้องออกไปได้ ทำให้โปรแกรมมีการทำงานที่เร็วขึ้น
- การนำความรู้เรื่องรูปร่างของผลิตภัณฑ์มาใช้ สามารถทำได้หลายวิธีที่นิยมมากที่สุดคือ การตรวจจับเส้นตรงหรือวงกลม

2.4.4 การแยกบริเวณ (Segmentation)

การแยกบริเวณเป็นการแยกบริเวณของภาพออกเป็นส่วน ๆ โดยมุ้งเน้นจะใช้วิธีใดแยกวัตถุที่สนใจออกจากพื้นหลัง เงื่อนไขใดพิจารณาพิกเซลว่าส่วนใดจัดว่าเป็นวัตถุ (Object) ส่วนใดจัดว่าเป็นพื้นหลัง (Background) และยังต้องคำนึงถึงการเก็บข้อมูลในรูปแบบใดที่เหมาะสมสำหรับกระบวนการแยกภาพ ซึ่งกระบวนการแยกอยู่ 2 วิธี คือ



รูปที่ 2.3 ผลของการแยกบริเวณออกเป็นส่วน ๆ

1. การแยกบริเวณโดยใช้ค่า Threshold คือ ค่าที่เป็นจำนวนเต็มระหว่าง 0 ถึง 255 เช่นเดียวกับค่าความเข้มของแสงในภาพ Gray Scale การแยกภาพโดย Threshold นั้นจะแปลงภาพ Gray Scale ให้เป็นภาพที่มีเพียงสองระดับ โดยให้เงื่อนไขว่าถ้าค่าพิกเซลตำแหน่งใดมีค่าต่ำกว่าหรือเท่าค่า Threshold ให้ค่าพิกเซลตำแหน่งนั้นเป็น 0 หรือด้านมืด และถ้าค่าพิกเซลตำแหน่งใดสูงกว่า

ค่า Threshold ให้มีค่าเป็น 255 หรือด้านสว่าง ซึ่งการแยกภาพแบบ Threshold ยังแบ่งได้อีก 2 ประเภท ได้แก่

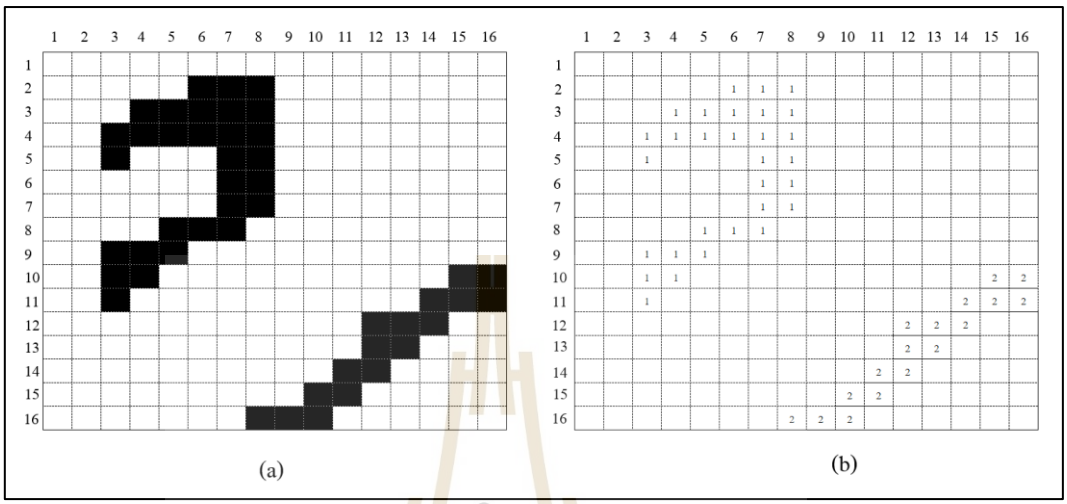
- การใช้ Threshold ค่าเดียวกับภาพ เรียกว่า Global Threshold
- การใช้ Threshold ที่เป็นค่าของตัวเองโดยการแบ่งภาพออกเป็นภาพย่อย ๆ ที่มีขนาดเล็ก ๆ เรียกว่า “Local Threshold” และเนื่องจากการแยกบริเวณโดยการใช้ค่า Threshold นั้น ผลที่ได้คือ บริเวณพื้นที่มีค่าความเข้มแสงใกล้เคียงกัน (Area Based Segmentation)

2. การแยกบริเวณโดยการใช้ขอบของวัตถุ (Edge based segmentation) การแบ่งวิธีนี้ต้องคำนวณหาขอบของวัตถุก่อน ขอบในความหมายการประมวลภาพคือ พิกเซลที่มีการเปลี่ยนแปลงความเข้มแสงเกินค่าที่กำหนด สามารถตรวจจับได้โดยใช้ตัวตรวจจับขอบ (Edge detector)

ผลของการแยกบริเวณออกเป็นส่วน ๆ ทั้ง 2 วิธีนั้น แสดงดังรูปที่ 2.3 ซึ่งสังเกตเห็นว่า การใช้ค่า Threshold ค่าเดียวกันทั้งภาพ (Global Thresholding) เพื่อแยกเม็ดข้าวออกจากฉากหลังนั้น พบว่าไม่ว่าเลือกใช้ค่า Threshold เท่าใดเม็ดข้าวบางเม็ดก็ยังหายไปจาก Binary Image ทั้งนี้เนื่องจากแสงในภาพ Gray Scale นั้นมีการกระจายไม่สม่ำเสมอทำให้ความสว่างของแต่ละบริเวณภายในภาพไม่เท่ากันและในทางกลับกัน การแยกขอบของเม็ดข้าวโดยการใช้ตัวตรวจจับแบบ Sobel นั้นเราจะได้ของของเม็ดครบทุกเม็ดแม้ว่าการใช้ค่า Threshold ค่าเดียวกันทั้งภาพ (Global Thresholding) เพื่อแบ่งวัตถุในภาพออกเป็นส่วน ๆ นั้นจะให้ผลที่ไม่ดีเท่ากับวิธีการแบ่งวัตถุในภาพด้วยการตรวจจับขอบแต่อย่างไรก็ตามจะพบว่าวิธีนี้จะเหมาะสมกับการนำไปใช้เพื่อการตรวจสอบผลิตภัณฑ์มากกว่า เนื่องจากได้ตัดขั้นตอนการตรวจจับขอบ ซึ่งเป็นขั้นตอนที่ต้องเสียเวลาในการคำนวณค่อนข้างมาก และผลที่ได้จากวิธีการใช้ค่า Threshold นั้นสามารถนำไปคำนวณหาคุณสมบัติพื้นฐาน เช่น พื้นที่หรือจุดศูนย์กลางของบริเวณได้อย่างสะดวกมากกว่าด้วย นอกจากนั้น เราสามารถจัดแสงให้มีความสม่ำเสมอมากกว่านี้ เพื่อขจัดปัญหาที่เกิดขึ้นกับการใช้วิธี Global thresholding ได้อีกด้วย อย่างไรก็ตาม การตรวจขอบก็เป็นวิธีการที่ต้องเรียนรู้ควบคู่ไปด้วยเช่นกันเนื่องจากสามารถนำไปใช้ได้ในสภาพแบบเปิดที่มีความแตกต่างระหว่างวัตถุและฉากหลังไม่สม่ำเสมอ เช่น ระบบการมองเห็นของหุ่นยนต์เตะบอลหรือหุ่นยนต์กู้ภัยได้

ผลที่ได้จากกระบวนการแยกบริเวณของภาพที่ถูกแบ่งออกเป็นส่วน ๆ ซึ่งทำให้เกิดการแยกบริเวณที่เป็นวัตถุออกจากบริเวณที่เป็นฉากหลัง (Background) ด้วยซึ่งผลที่ได้จริง ๆ ทางกายภาพคือ ภาพที่มีความเข้มเพียง 2 ระดับ (Binary Image) ทั้งนี้ขึ้นอยู่กับลักษณะงานแต่ละอย่างว่าส่วนที่จัดเป็นวัตถุนั้นจะเป็นสีขาวหรือสีดำและขั้นตอนต่อไปนั้นเราจะทำการพิจารณาว่าพิกเซลใดบ้างที่มีการเชื่อมต่อ (Connect) กัน เพื่อที่จะจัดให้พิกเซลเหล่านั้นให้อยู่ในบริเวณเดียวกันหรือวัตถุเดียวกันกระบวนการย่อยนี้เรียกว่า “Connected Components Labeling” ซึ่งผลของการ

ทำงานของกระบวนการย่อยนี้จะทำให้รู้ว่าพิกเซลในแต่ละตำแหน่งนั้น จัดเป็นของวัตถุชิ้นใด ดังแสดงในรูปที่ 2.4 ซึ่งแสดงหมายเลขของชิ้นวัตถุที่พิกเซลตำแหน่งนั้น ๆ เป็นสมาชิกอยู่



รูปที่ 2.4 ผลการทำงานกระบวนการ Connected Components Labeling

ซึ่งรูปที่ 2.4 (a) แสดงภาพ Binary Image ที่ได้จากวิธี Global Thresholding ซึ่งในที่นี้เป็นการพิจารณาวัตถุสีดำที่มีพื้นหลังเป็นสีขาว และรูปที่ 2.4 (b) ผลที่ได้จากการทำงานของ Connected Components Labeling ที่ทำให้เราทราบว่า พิกเซลในแต่ละตำแหน่งนั้นเป็นของวัตถุชิ้นใด ซึ่งในรูปมีวัตถุอยู่ 2 ชิ้น คือวัตถุหมายเลข 1 และ 2

หลังจากที่เราทราบว่าตำแหน่งใดเป็นวัตถุ ตำแหน่งใดเป็นพื้นหลังแล้ว เราต้องทำการเก็บพิกัดของพิกเซลที่จัดอยู่ในบริเวณเดียวกัน โดยต้องพิจารณาอยู่ 2 เรื่องด้วยกัน คือ

- วิธีการเก็บพิกัดของกลุ่มพิกเซล จัดเป็นการบีบอัดข้อมูลชนิดหนึ่ง ที่จะต้องใช้วิธีที่ใช้เนื้อที่หน่วยความจำน้อยที่สุด ซึ่งมีอยู่หลายวิธี เช่น Chain Code, Run-Length Encoding
- โครงสร้างของข้อมูล (Data Structure) เป็นการเก็บข้อมูลที่ได้จากหัวหน้าก่อนหน้าเข้าด้วยกัน เพื่อความสะดวกในการอ้างถึง

การแยกบริเวณออกเป็นส่วน ๆ นั้นสามารถทำได้อย่างมีประสิทธิภาพ โดยขึ้นอยู่กับ 2 ปัจจัยหลัก คือ การเลือกแหล่งกำเนิดแสงที่มีความยาวคลื่นแสงใกล้เคียงกับความไว (Sensivity) หรือความสามารถในการตอบสนองต่อแสงที่มีความยาวคลื่นต่าง ๆ ของตัวกล้อง และอีกปัจจัยคือ การจัดฉากหลัง (Background) ให้มีความเข้มแสงแตกต่างกันมาก ๆ กับตัววัตถุที่พิจารณา เพื่อให้สามารถแยกส่วนที่เป็นวัตถุและส่วนที่เป็นฉากหลังออกจากกันโดยวิธี Global Thresholding ได้อย่างชัดเจนนั่นเอง

2.4.5 การคำนวณหาคุณสมบัติของวัตถุ (Feature Extraction)

การคำนวณหาคุณสมบัติของวัตถุเป็นการนำภาพที่แบ่งออกเป็นส่วน ๆ ที่มีคุณสมบัติคล้ายคลึงกันและทำการจัดเก็บพิกัดของพิกเซลที่เป็นของบริเวณเดียวกัน โดยเลือกใช้รูปแบบการเก็บที่เหมาะสมแล้วมาคำนวณหรือวัดคุณสมบัติของแต่ละบริเวณในแต่ละชิ้นที่อยู่ในรูป ซึ่งในทางปฏิบัติการคำนวณหาคุณสมบัติบางประการนั้นสามารถทำไปพร้อม ๆ กับกระบวนการ Connect Components Labeling ได้เลย เช่น การคำนวณหาพื้นที่ ซึ่งเป็นการนับจำนวนพิกเซลที่เป็นของวัตถุนั้น ๆ หรือการหาเส้นรอบวงของวัตถุที่เป็นการนับจำนวนพิกเซลของวัตถุที่มีด้านใดด้านหนึ่งติดอยู่กับบริเวณที่เป็นฉากหลัง แต่คุณสมบัติบางประการก็จะต้องทำหลังจากที่เสร็จสิ้นกระบวนการดังกล่าวไปแล้ว เพื่อให้ทราบพื้นที่ทั้งหมดของวัตถุเสียก่อน เช่น การหาจุดศูนย์กลางของวัตถุ ผลที่ได้จากกระบวนการนี้ คือ ค่าคุณสมบัติต่าง ๆ ของวัตถุแต่ละชิ้นที่อยู่ในภาพ ซึ่งถ้ามาจัดเรียงในรูปแบบเวคเตอร์เราจะได้ Feature vector

2.4.6 การจำแนกวัตถุและการแปลความหมาย (Classification and Interpretation)

วัตถุ คือ บริเวณของภาพที่มีค่าความเข้มแสงใกล้เคียงกัน ซึ่งถูกแบ่งออกเป็นบริเวณ ๆ แยกออกจากพื้นหลังด้วยกระบวนการแยกบริเวณ (Segmentation) ซึ่งคุณสมบัติต่าง ๆ ที่ได้มาจากกระบวนการคำนวณหาคุณสมบัติของวัตถุ (Feature Extraction) จะถูกนำมาเขียนให้อยู่ในรูปเวคเตอร์ซึ่งเรียกว่า “Feature Vector” และเมื่อถึงขั้นตอนนี้ เราจะสามารถแทนวัตถุแต่ละชิ้นที่อยู่ในภาพด้วยเวคเตอร์ต่อ 1 วัตถุ ซึ่งการทำงานของระบบต่อจากนี้จะทำโดยการพิจารณาค่าที่อยู่ใน Feature Vector เท่านั้น นอกจากนั้นยังมีข้อสังเกตบางประการคือ เมื่อขั้นตอนก่อนหน้าเสร็จสิ้นลง โปรแกรมจะสามารถแยกวัตถุหรือบริเวณที่สนใจออกจากฉากหลัง และรู้เพียงคุณสมบัติต่าง ๆ ของวัตถุแต่ละชิ้นเท่านั้น แต่ไม่สามารถรู้ได้ว่าวัตถุชิ้นนั้น ๆ คือวัตถุอะไร

การจำแนก (Classification) คือ กระบวนการจัดกลุ่มให้วัตถุที่กำลังพิจารณาอยู่ว่าเป็นวัตถุในกลุ่มใด โดยอาศัยข้อมูลที่ได้จากการวัดหรือการคำนวณค่าคุณสมบัติเปรียบเทียบกับคุณสมบัติของวัตถุตัวอย่างที่อยู่ในแต่ละกลุ่ม ดังนั้น ก่อนที่ระบบจะสามารถทำการตัดสินใจดังกล่าวได้ ระบบจะต้องมีตัวอย่างของวัตถุในแต่ละกลุ่ม ผลของการจำแนกสามารถทำให้ทราบว่าบริเวณที่แยกมานั้นเป็นวัตถุชนิดใด ซึ่งจะทำให้เกิดการตีความหมายของภาพและตัดสินใจสั่งการการทำงานของอุปกรณ์ต่าง ๆ

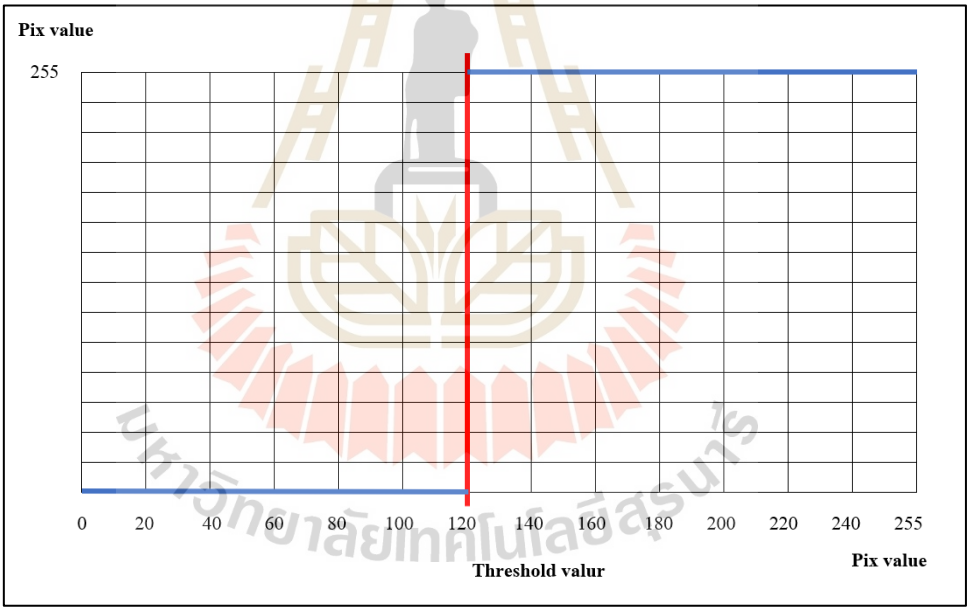
2.4.7 กลไกการเคลื่อนไหว (Actuation)

กระบวนการนี้เป็นการสั่งการทำงานของส่วนกลไกต่าง ๆ ของอุปกรณ์ให้เคลื่อนไหวทำงานต่าง ๆ ตามที่ได้รับคำสั่ง เช่น การสั่งให้สายพานเลื่อนชิ้นงานชิ้นต่อไปเข้ามาหรือสั่งแขนหุ่นยนต์ให้หยิบจับสิ่งของ

จากข้อมูลที่กล่าวมาจะเห็นได้ว่า ข้อมูลที่ได้จากการประมวลภาพเบื้องต้น คือ ภาพที่ได้รับการปรับปรุงคุณภาพแล้ว เมื่อส่งข้อมูลดังกล่าวไปที่กระบวนการแยกบริเวณจะได้บริเวณของวัตถุที่สนใจ นอกจากนั้น แต่ละบริเวณจะมีการบีบอัดพิกัดของพิกเซลไว้ในรูปแบบที่เหมาะสม และส่งต่อไปยังกระบวนการจำแนกและตีความหมาย ก็จะได้ข้อมูลที่สำคัญที่สุดออกมาและตัดสินใจว่าจะจัดการกับวัตถุที่ได้นั้นอย่างไร

2.5 ค่าเริ่มเปลี่ยน (Threshold)

ค่าเริ่มเปลี่ยนเป็นหลักการที่ใช้ค่าคงที่ค่าหนึ่งในการเปรียบเทียบกับค่าของพิกเซล (Pixel) ในแต่ละพื้น ถ้าค่าของพิกเซลในพื้นที่นั้นมีค่าน้อยกว่าค่าคงที่ก็จะเปลี่ยนค่าพิกเซลของพื้นที่นั้นเป็น 0 แต่ถ้าค่าของพิกเซลในพื้นที่นั้นมีค่ามากกว่าก็จะเปลี่ยนค่าพิกเซลของพื้นที่นั้นเป็น 255 โดยภาพ Input จะต้องเป็นภาพ Gray Scale



รูปที่ 2.5 ค่าของพิกเซล

จากภาพสมมุติว่ากำหนดค่า Threshold เท่ากับ 120 จะสังเกตว่าค่าไหนที่มีค่าน้อยกว่า 120 จะถูกปรับให้เป็น 0 แต่ถ้าค่าไหนมีค่ามากกว่าจะถูกปรับให้เป็น 255 ดังนั้นสมการฟังก์ชันของ Threshold แสดงดังสมการที่ 2.1 และสมการที่ 2.2

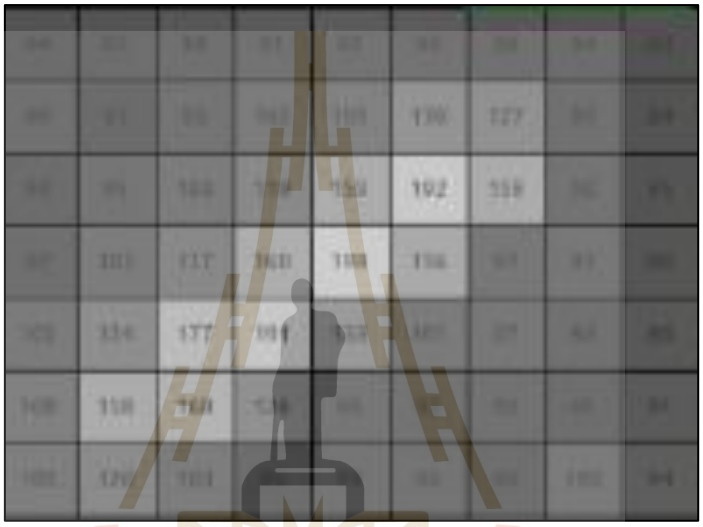
$g(x, y) = 0$ ถ้า $f(x, y)$ มากกว่าหรือเท่ากับ Threshold Value (2.1)

$g(x, y) = 255$ ถ้า $f(x, y)$ น้อยกว่า Threshold Value (2.2)

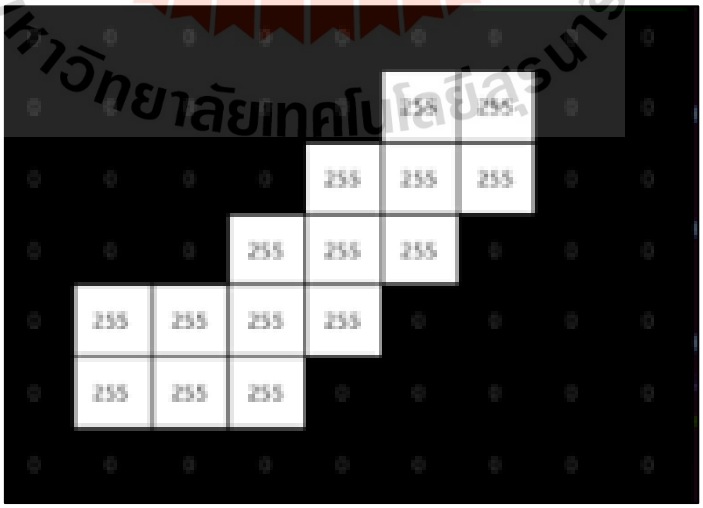
เมื่อ $f(x, y)$ คือ ตำแหน่งพิกเซลของภาพต้นฉบับ

$g(x, y)$ คือ ตำแหน่งพิกเซลของภาพผลลัพธ์

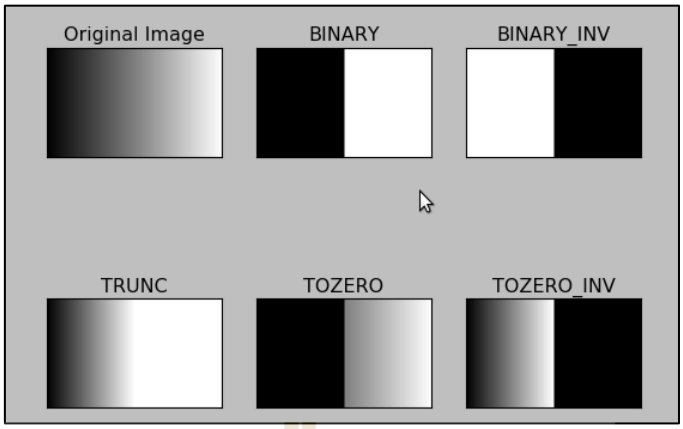
ตัวอย่างการคำนวณด้วยมือเมื่อมีรูปภาพขนาด 10 * 10 และให้ Threshold value เท่ากับ 120



รูปที่ 2.6 ภาพหลังผ่านการทำ Gray Scale

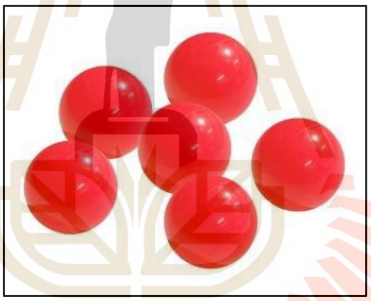


รูปที่ 2.7 ภาพหลังการคำนวน Threshold



รูปที่ 2.8 ฟังก์ชัน Threshold ของแต่ละแบบ

- ภาพเริ่มต้นการจำลองการทำงานฟังก์ชัน Threshold



รูปที่ 2.9 ภาพเริ่มต้นการทำงานฟังก์ชัน Threshold

- ภาพการแปลงจากภาพเริ่มต้นเป็นภาพแบบ Grayscale

รูปที่ 2.10 ภาพ Gray Scale

มหาวิทยาลัยเทคโนโลยีสุรนารี

ใช้ฟังก์ชัน Threshold โดยกำหนดให้

- dst(x,y) = ค่าพิกเซลใหม่ที่ได้จากการคำนวนฟังก์ชัน
- src(x,y) = ค่าเริ่มต้นของพิกเซล
- thresh = ค่าคงที่ที่กำหนดขึ้น
- maxVal = ค่าสูงสุดของพิกเซลเท่ากับ 255

1. แบบที่ 1 Threshold Binary

ประเภทนี้เป็นประเภทที่พบบ่อยมากที่สุดและการทำงานง่ายที่สุด จากสมการฟังก์ชัน Threshold Binary

$$dst(x,y) = \begin{cases} maxVal & \text{if } src(x,y) > thesh \\ 0 & \text{otherwise} \end{cases} \quad (2.3)$$

รูปที่ 2.11 รูปหลังจากใช้ Threshold Binary ในการคำนวน

2. แบบที่ 2 Threshold Binary, Inverted

ประเภทนี้มีการทำงานตรงข้ามกับแบบที่ 1 โดยทำการกำหนดพิกเซลปลายทางให้เป็น 0 เมื่อพิกเซลต้นทางมีค่ามากกว่าค่า Threshold และกำหนดพิกเซลปลายทางให้เท่ากับค่า maxValue ถ้าพิกเซลต้นทางมีค่าน้อยกว่า Threshold ซึ่งสมการฟังก์ชัน Threshold Binary, Inverted แสดงดังสมการที่ 2.4

$$dst(x,y) = \begin{cases} 0 & \text{if } src(x,y) > thesh \\ maxVal & \text{otherwise} \end{cases} \quad (2.4)$$

มหาวิทยาลัยเทคโนโลยีสุรนารี

รูปที่ 2.12 รูปหลังจากใช้ Threshold Binary, Inverted ในการคำนวน

3. แบบที่ 3 Truncate

ประเภทนี้ทำการกำหนดพิกเซลปลายทางให้เป็นค่า Threshold ถ้าพิกเซลต้นทางมีค่ามากกว่าค่า Threshold ไม่เช่นนั้นจะถูกกำหนดค่าให้เป็นพิกเซลต้นทาง โดยประเภทนี้จะไม่สนใจค่า maxValue ซึ่งสมการฟังก์ชัน Truncate แสดงดังสมการที่ 2.5

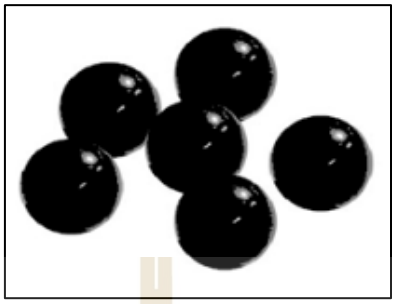
$$dst(x,y) = \begin{cases} threshold & \text{if } src(x,y) > thesh \\ src(x,y) & \text{otherwise} \end{cases} \quad (2.5)$$

รูปที่ 2.13 รูปหลังจากใช้ Truncate ในการคำนวน

4. แบบที่ 4 Threshold to Zero

ประเภทนี้พิกเซลปลายทางจะถูกกำหนดให้มีค่าเท่ากับพิกเซลต้นทางถ้าพิกเซลต้นทางมีค่ามากกว่าค่า Threshold ไม่เช่นนั้นจะถูกกำหนดให้เป็นศูนย์ โดยประเภทนี้จะไม่สนใจค่า maxValue ซึ่งสมการฟังก์ชัน Threshold to Zero แสดงดังสมการที่ 2.6

$$dst(x,y)=\begin{cases} src(x,y) & \text{if } src(x,y) > thesh \\ 0 & \text{otherwise} \end{cases} \quad (2.6)$$



รูปที่ 2.14 รูปหลังจากใช้ Threshold to Zero ในการคำนวน

5. แบบที่ 5 Threshold to Zero, Inverted

ประเภทนี้จะกำหนดให้พิกเซลปลายทางให้มีค่าเป็นศูนย์ถ้าพิกเซลต้นทางมีค่ามากกว่าค่า Threshold ไม่เช่นนั้นจะถูกกำหนดให้มีค่าเท่ากับพิกเซลต้นทาง โดยประเภทนี้จะไม่สนใจค่า maxValue ซึ่งสมการฟังก์ชัน Threshold to Zero แสดงดังสมการที่ 2.7

$$dst(x,y)=\begin{cases} 0 & \text{if } src(x,y) > thesh \\ src(x,y) & \text{otherwise} \end{cases} \quad (2.7)$$

รูปที่ 2.15 รูปหลังจากใช้ Threshold to Zero, Inverted ในการคำนวน

มหาวิทยาลัยเทคโนโลยีสุรนารี

2.6 การลบพื้นหลัง (Background Subtraction)

การลบฉากหลังเป็นขั้นตอนก่อนการประมวลผลที่สำคัญในการวิเคราะห์โปรแกรม ตัวอย่างเช่น การนับคนที่ใช้กล้องถ่ายรูปเพื่อนับคนเข้าและออกจากห้อง หรือการจราจรบนท้องถนน ในกรณีดังกล่าวคุณจะต้องนำบุคคลหรือรถยนต์มาเพียงอย่างเดียวเพื่อนับ จำนวน โดยเทคนิคการแยกภาพเบื้องหน้าออกจากภาพพื้นหลัง หากมีภาพพื้นหลังเพียงภาพเดียว เช่นเดียวกับบุคคลในห้องหรือบนถนนที่ไม่มีรถเลย การลบภาพใหม่จากพื้นหลังนั้นจะเป็นเรื่องง่ายมาก แต่มักจะซับซ้อนขึ้นเมื่อมีเงาของยานพาหนะ ซึ่งการลบนั้นสามารถทำได้หลายวิธี โดย OpenCV คำสั่งและฟังก์ชันไว้ให้แล้ว ดังนี้

- BackgroundSubtractorMOG

BackgroundSubtractorMOG เป็นวิธี Gaussian Mixture-Based สำหรับการแยกอัลกอริธึมการแบ่งส่วนพื้นหลัง/พื้นหน้า ซึ่งเป็น “รูปแบบการผสมที่ดีขึ้นโดยการปรับให้เข้ากับการติดตามแบบเรียลไทม์โดยการตรวจจับเงา” โดยผู้พัฒนาคือ P.LadewTraKuPong และ R.Bowden ในปี ค.ศ. 2001 นั่นคือ การใช้วิธีการจำลองแต่ละพิกเซลของพื้นหลังโดยมีส่วนผสมของการกระจ่ายค่า (Mixture of krasp) (K = 3 ถึง 5) โดยน้ำหนักของส่วนผสมแทนเวลาสัดส่วนที่สีเหล่านั้นอยู่ในที่เกิดเหตุ ซึ่งจำเป็นมากในการสร้างวัตถุพื้นหลังโดยใช้ cv2.createBackgroundSubtractorMOG ()

- BackgroundSubtractorMOG2

นอกจากนี้ในการผสมตามแบบเกาส์ (Gaussian Mixture-Based Background/Foreground) นั้นยังมีงานวิจัยอีกชิ้นหนึ่งที่เขียนโดย Z.Zivkovic “การปรับตัวที่ดีขึ้นของรูปแบบ Gaussian” โดยมีส่วนผสมสำหรับการลบฉากหลังในปี ค.ศ. 2004 และมีประสิทธิภาพประมาณความหนาแน่นโดยหนึ่งในคุณลักษณะที่สำคัญของขั้นตอนวิธีการนี้ก็คือจะเลือกจำนวนที่เหมาะสมของการกระจายแบบ Gaussian สำหรับแต่ละพิกเซล (ในกรณีที่ผ่านมาเรานำ K ไปที่ Gaussian ตลอดกระบวนการ) ซึ่งดีกว่าที่จะปรับให้เข้ากับแต่ฉากที่แตกต่างกัน

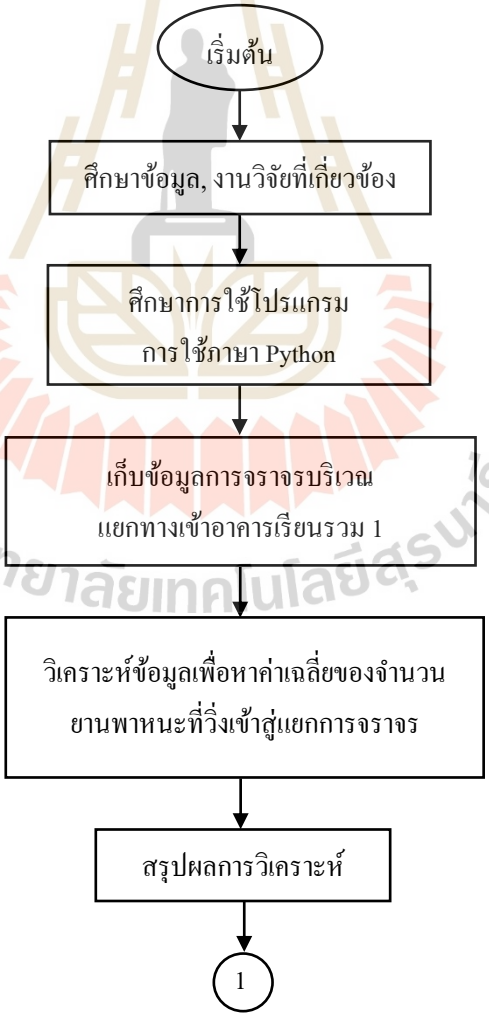
- BackgroundSubtractorGMG

อัลกอริทึมนี้ประกอบด้วยค่า Statistical Background Image โดยจะประเมินค่าพิกเซลต่อพิกเซลและแบ่งส่วน วิธีนี้คิดค้นโดย Andrew B. Godbehere และ Akihiro Matsukawa, Ken Goldberg และอยู่ในงานวิจัยชื่อ “Visual Tracking of Human Visitors under Variable-Lighting Conditions for a Responsive Audio Art Installation” ในปี ค.ศ. 2012 โดยในวิจัยนั้นประสบความสำเร็จ โดยค่ายศิลปะเรียกว่า “Are We There Yet?” จาก 31 มีนาคม ถึง 31 กรกฎาคม ค.ศ. 2011 ที่พิพิธภัณฑ์ยิวร่วมสมัยในซานฟรานซิส, แคลิฟอร์เนีย โดยการใช้งานนั้นจะมีไม่กี่ขั้นตอนโดยเฟรมสำหรับ Background นั้นจะจำลองพื้นหลัง โดยมีส่วนของขั้นตอนวิธีการแบ่งส่วนความน่าจะเป็นของ Foreground โดยที่ระบุวัตถุเบื้องต้นหน้าไปได้

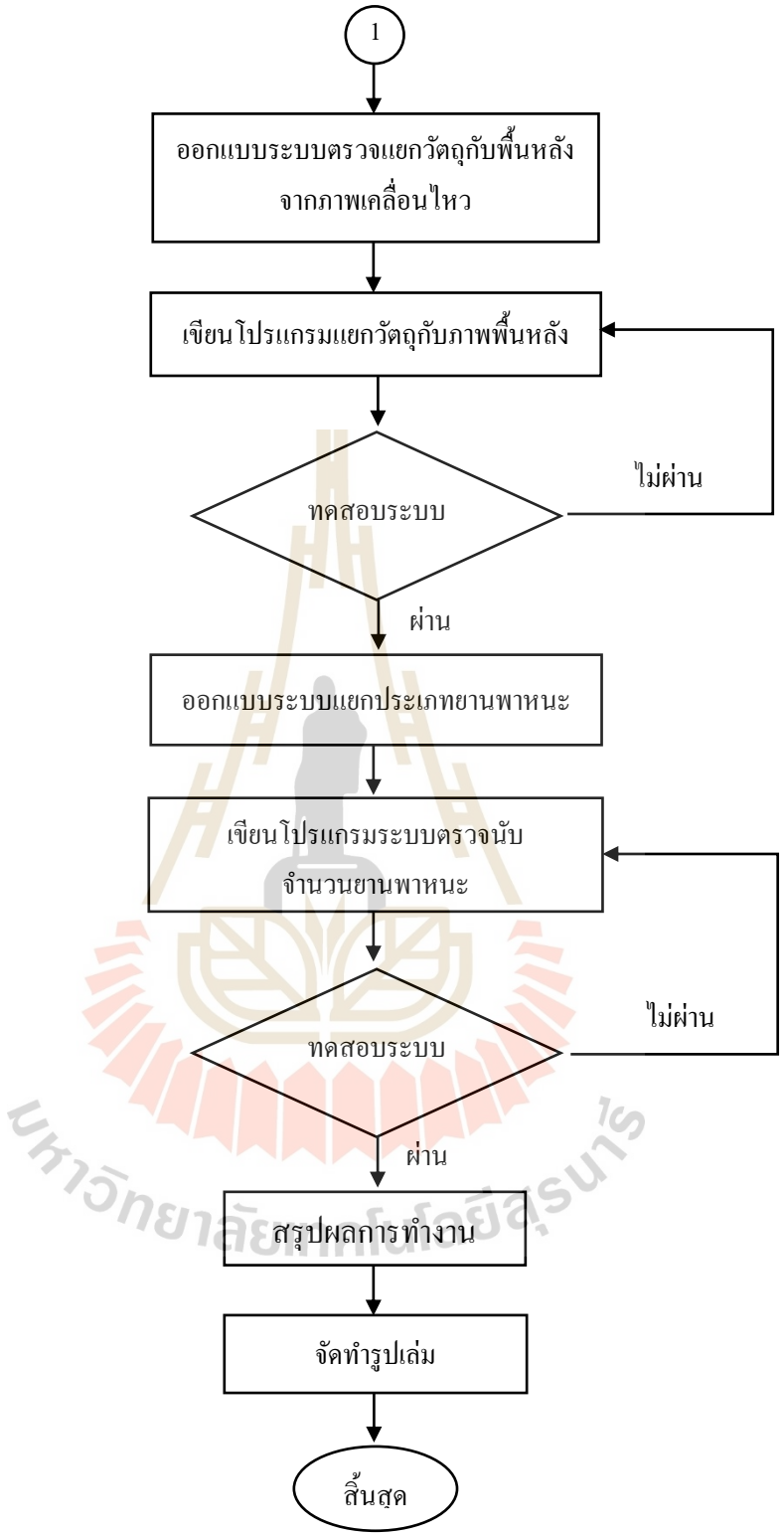
บทที่ 3
วิธีดำเนินงานวิจัย

3.1 กล่าวนำ

งานวิจัยนี้เป็นการนำเสนอการออกแบบระบบนับจำนวนและแยกประเภทยานพาหนะควบคุมผ่านการมองเห็นของเครื่องจักร (Machine Vision) โดยนำภาพเคลื่อนไหวมาวิเคราะห์หาความแตกต่างระหว่างพื้นหลังและวัตถุ เพื่อนับจำนวนยานพาหนะบนท้องถนน แยกประเภทยานพาหนะ ลำดับขั้นตอนการทำงานแสดงเป็นแผนผังได้ดังนี้



รูปที่ 3.1 แผนผังขั้นตอนการทำการวิจัย



รูปที่ 3.1 แผนผังขั้นตอนการทำการวิจัย (ต่อ)

3.2 การเก็บข้อมูลบริเวณแยกการจราจรทางเข้าอาคารเรียนรวม 1

ขั้นตอนนี้จะต้องทำการเก็บภาพหรือวิดีโอที่เราจะนำไปใช้ในการตรวจจับวัตถุ โดยวิดีโอที่นำมาใช้ในงานวิจัยนี้ได้ถูกเก็บจากกล้อง IP camera ที่ติดตั้งอยู่บริเวณแยกทางเข้าอาคารเรียนรวม 1 โดยมีมุมมองของภาพและการมองเห็น ดังรูปที่ 3.2

รูปที่ 3.2 บริเวณที่เก็บข้อมูล

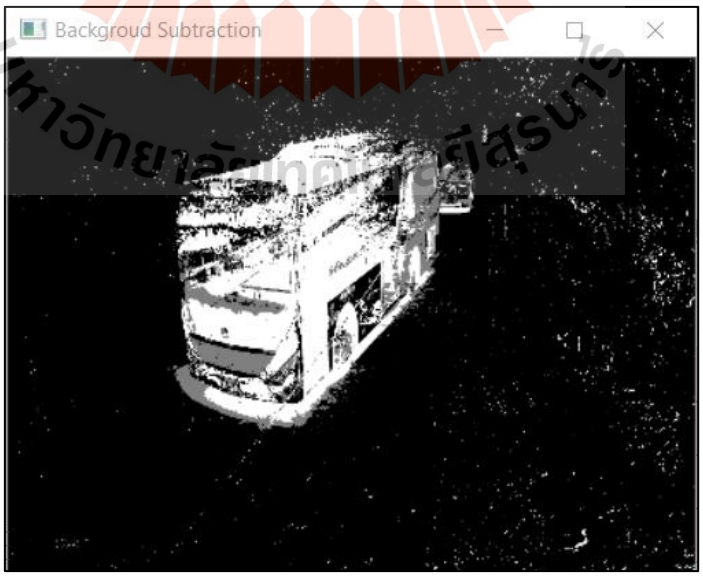
รูปที่ 3.3 บริเวณที่เก็บข้อมูลขณะที่มียานพาหนะเคลื่อนที่ผ่าน

3.3 การออกแบบระบบแยกวัตถุออกจากพื้นหลัง

การทำงานของระบบนี้ถูกสร้างขึ้นด้วยโปรแกรมภาษา Python และมีการใช้ไลบรารี่สำเร็จรูป OpenCV ซึ่งหลังจากที่ได้ภาพบริเวณที่ต้องการแล้ว เริ่มต้นโดยการเขียนโค้ดให้โปรแกรมอ่านค่าจากวิดีโอที่มีอยู่ในคอมพิวเตอร์ และเรียกใช้ฟังก์ชันของ VideoCapture ที่ใช้ในการอ่านเฟรมวิดีโอแล้วแปลงค่าสี RGB ของภาพให้เป็นภาพขาวดำ (Gray scale) แสดงดังรูปที่ 3.4

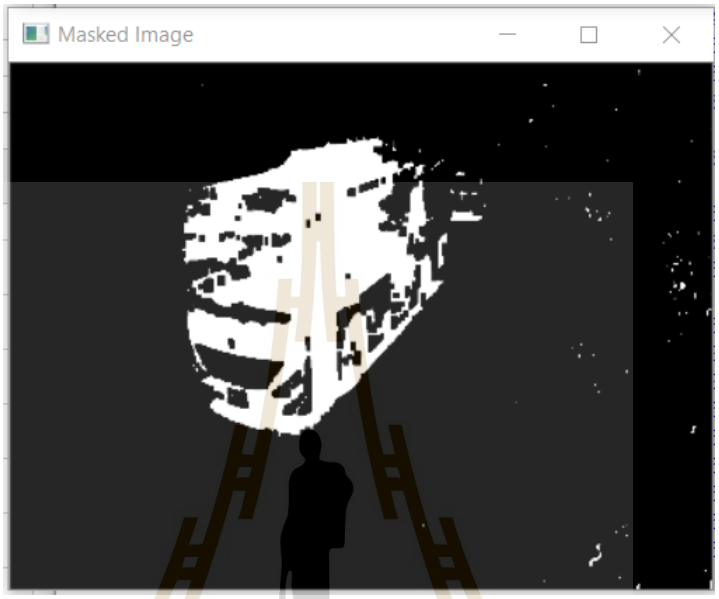
รูปที่ 3.4 ภาพแบบ Gray Scale

จากนั้นเรียกใช้ฟังก์ชัน BackgroundSubtractorMOG2() เป็นฟังก์ชันที่มีหลักการแยก Background และ Foreground โดยอาศัยอัลกอริทึมที่ใช้หลักการของ Gaussian มาช่วยในการแยก และใช้การเปรียบเทียบค่า Threshold เพื่อแยกบริเวณพิกเซลที่เป็นสีขาวและพิกเซลที่เป็นสีดำจะได้ภาพขาวดำที่ถูกลบเงา และลบสัญญาณรบกวนต่าง ๆ (Noise) ออกจากภาพ หลังจากเสร็จสิ้นกระบวนการนี้เราจะได้ผลลัพธ์ของการแยกวัตถุออกจากภาพพื้นหลัง แสดงดังรูปที่ 3.5



รูปที่ 3.5 ภาพการใช้ฟังก์ชัน BackgroundSubtractorMOG2()

เมื่อเสร็จสิ้นกระบวนการนี้จะได้ภาพของยานพาหนะที่สนใจโดยถูกลบแสงและเงา รวมถึงลบสัญญาณรบกวนของภาพ (Noise) ให้เหลือเพียงแค่ส่วนที่เป็นพิกเซลสีขาวและพิกเซลสีดำ ทำให้สามารถแยกยานพาหนะออกจากภาพพื้นหลังได้ แสดงดังรูปที่ 3.6



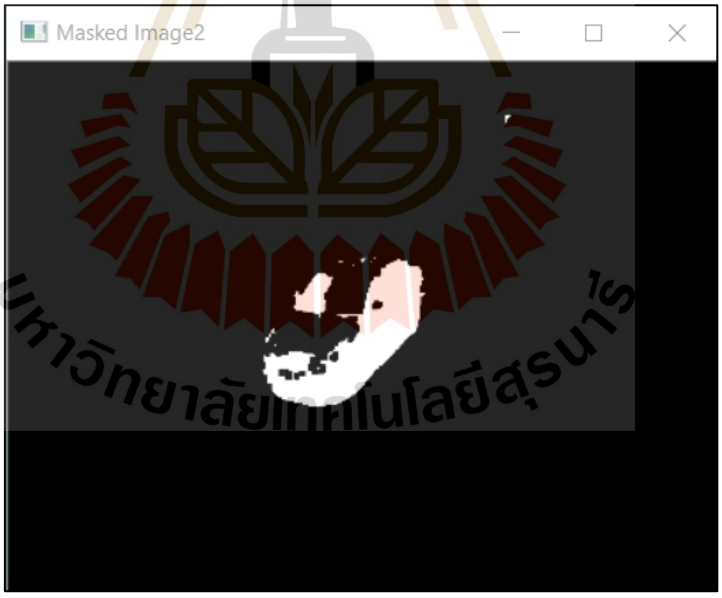
รูปที่ 3.6 ภาพที่ทำการลบเงาและลบสัญญาณรบกวนบนภาพ

3.4 ออกแบบระบบแยกประเภทยานพาหนะ

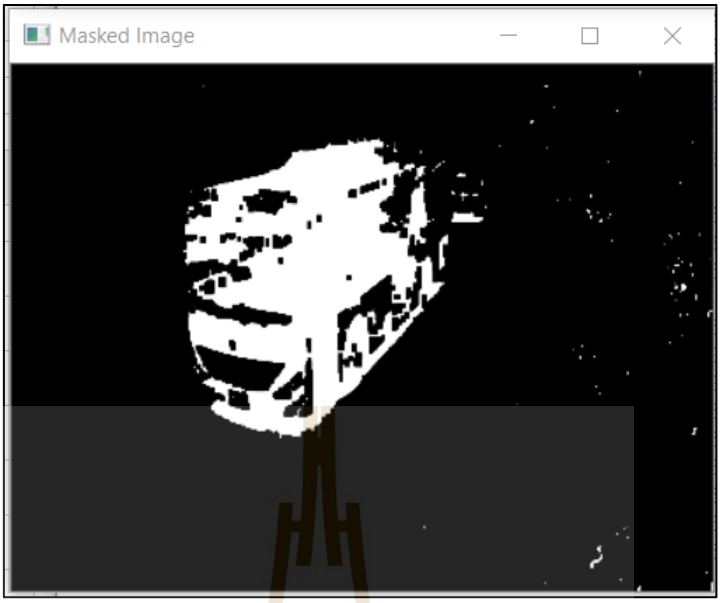
หลังจากที่สามารถแยกยานพาหนะออกจากพื้นหลังได้แล้วเป็นการแยกประเภทให้กับยานพาหนะที่สนใจ ซึ่งยานพาหนะที่สนใจในโปรแกรมนี้แบ่งออกเป็น 3 ประเภท คือ

1. รถขนาดเล็ก 2 ล้อ (มอเตอร์ไซค์หรือจักรยาน)
2. รถขนาดกลาง 4 ล้อ
3. รถขนาดใหญ่ (รถบัสหรือรถบรรทุก)

รูปที่ 3.7 ภาพที่ทำการลบเงาและลบสัญญาณรบกวนบนภาพของรถขนาดเล็ก



รูปที่ 3.8 ภาพที่ทำการลบเงาและลบสัญญาณรบกวนบนภาพของรถขนาดกลาง



รูปที่ 3.9 ภาพที่ทำการลบเงาและลบสัญญาณรบกวนบนภาพของรถขนาดใหญ่

โดยจะเริ่มต้นที่การสร้างเส้นขึ้นมาในโปรแกรมเพื่อใช้ในการอ้างอิงยานพาหนะที่เคลื่อนที่ผ่าน ซึ่งในโปรแกรมนี้จะสร้างขึ้นมา 2 เส้นบนเฟรมของภาพ ดังนี้

1. เส้นที่ 1 เส้นสีน้ำเงิน เพื่อไว้ตรวจนับรถขนาดใหญ่
2. เส้นที่ 2 เส้นสีแดง เพื่อไว้ตรวจนับรถขนาดเล็กและรถขนาดกลาง

รูปที่ 3.10 การวาดเส้นบนเฟรมของภาพ

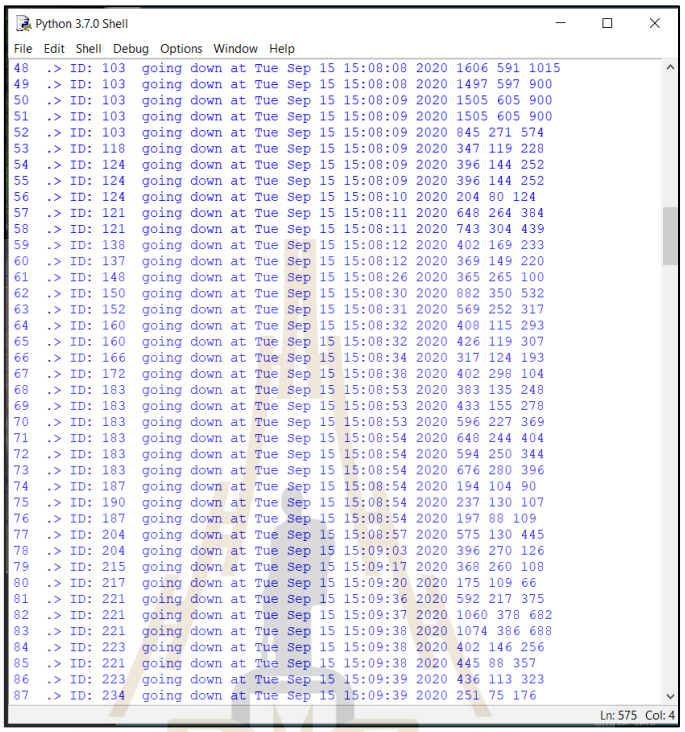
การแยกประเภทของยานพาหนะจะเริ่มต้นโดยการหาขอบเขตให้กับยานพาหนะที่เราสนใจในภาพ โดยที่จะสร้างขอบเขตรอบๆยานพาหนะที่เราสนใจ หลังจากนั้นยานพาหนะจะถูกติดตามด้วยการสร้างกรอบสี่เหลี่ยมครอบยานพาหนะและในกรอบสี่เหลี่ยมจะมีการกำหนดจุดเซนทรอยของยานพาหนะนั้น ๆ ไว้ด้วย โดยกรอบสี่เหลี่ยมนี้ถูกสร้างขึ้นโดยอ้างอิงจากจุดที่มีค่ามากที่สุดเมื่อเทียบกันในแนวแกน x และในแนวแกน y ร่วมกับค่าความกว้างและสูงของรูปภาพ

รูปที่ 3.11 ภาพที่ได้จากการขอบเขตของพาหนะ

เมื่อจุดเซนทรอยของยานพาหนะที่กำหนดไว้ในกรอบเคลื่อนที่ผ่านเส้นที่สร้างไว้ตัวโปรแกรมจะแสดงค่าความกว้างและความสูงของวัตถุออกมา ซึ่งขนาดของยานพาหนะแต่ละประเภทนั้นได้มาจากการนำเอาค่าความกว้างและความสูงรวมกัน และเมื่อได้ขนาดของยานพาหนะมาแล้วก็นำค่าที่ได้นั้นไปกำหนดเพื่อแยกประเภทของยานพาหนะ โดยขนาดของรถแต่ละประเภทมีค่าดังนี้

- รถขนาดเล็ก มีขนาดอยู่ระหว่าง 300-700 โดยที่ความสูงอยู่ระหว่าง 150-400 ความกว้างอยู่ระหว่าง 100-230 และความสูงต้องมากกว่าความกว้าง
- รถขนาดกลาง มีขนาดอยู่ระหว่าง 715-800 โดยที่มีความสูงอยู่ระหว่าง 300-400 และความกว้างอยู่ระหว่าง 400-450 และมีขนาดอยู่ระหว่าง 800-1300 โดยที่มีความสูงอยู่ระหว่าง 300-450 และความกว้างอยู่ระหว่าง 450-600 รถขนาดกลางจำเป็นต้องมีการกำหนดขนาดอยู่ 2 ช่วง เพราะรถขนาดกลางที่มีในปัจจุบันนั้นมีหลากหลายประเภทซึ่งมีขนาดที่แตกต่างกัน

- รถขนาดใหญ่ มีขนาดอยู่ระหว่าง 1,800-2,600 โดยที่มีความสูงมากกว่า 800 ความกว้างน้อยกว่า 1,300 และความสูงต้องน้อยกว่าความกว้าง



รูปที่ 3.12 ค่าขนาดของวัตถุที่สนใจ

3.5 การออกแบบระบบนับจำนวนยานพาหนะ

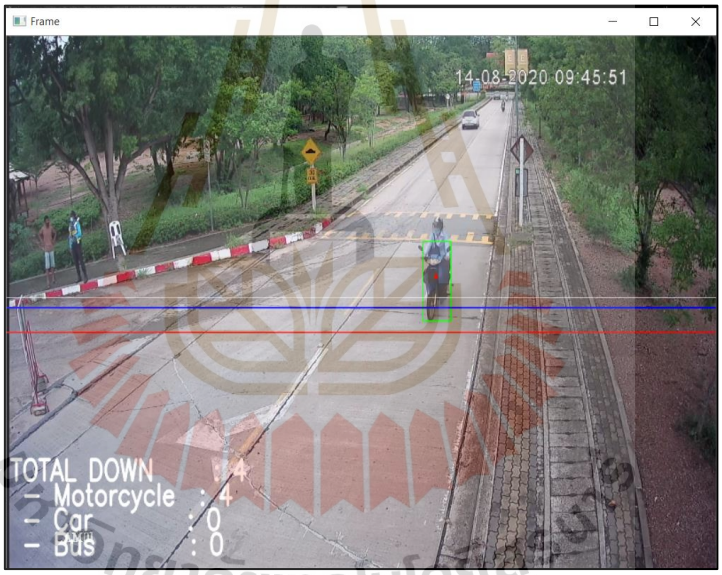
จากที่ได้ทำการแยกยานพาหนะออกจากพื้นหลังและแยกประเภทยานพาหนะแล้วนั้น ต่อมาจะเป็นการนับจำนวนยานพาหนะ ในขณะที่ยานพาหนะถูกติดตามโดยการสร้างกรอบในภาพแบบสีขาวดำแล้วนั้นบนเฟรมของภาพที่นำมาแสดงการทำงานของโปรแกรมก็ได้ทำการสร้างกรอบและกำหนดจุดเซนทรอยของภาพนั้นไว้เช่นกัน หลังจากที่จุดเซนทรอยของภาพได้เคลื่อนที่ผ่านเส้นที่กำหนดไว้ ทำให้ทราบขนาดของยานพาหนะแต่ละประเภทแล้ว ตัวโปรแกรมก็จะทำการนับจำนวนยานพาหนะแต่ละประเภทนั้น ๆ โดยที่ในโปรแกรมนี้จำเป็นที่จะต้องสร้างเส้นขึ้นมา 2 เส้น เพราะเมื่อทำการเปรียบเทียบภาพพื้นหลังออกจากวัตถุที่เราสนใจนั้น เมื่อเวลายานพาหนะเคลื่อนที่ทำให้เกิดภาพที่แตกต่างกันออกไปและทำให้มีการนับจำนวนคลาดเคลื่อนและนับรถผิดประเภท และด้วยความเร็วในการเคลื่อนที่ของรถขนาดเล็กแต่ละคันไม่เท่ากัน จึงทำให้รถขนาดเล็ก ที่เคลื่อนที่มาด้วยความเร็วที่สูงกว่าคันอื่น ๆ โปรแกรมตรวจจับไม่ทัน และเนื่องด้วยวงของการเลี้ยว

ของรถขนาดใหญ่แต่ละคันไม่เท่ากัน จึงทำให้การเคลื่อนที่ผ่านเส้นของรถขนาดใหญ่แต่ละคันไม่เท่ากัน

โดยในโปรแกรมนี้ที่มีการใช้เส้นนับจำนวนเพียงเส้นเดียวจะมีค่าความแม่นยำต่างจากโปรแกรมที่มีการใช้เส้นนับจำนวนสองเส้นคิดเป็นร้อยละ ดังนี้

- รถขนาดเล็ก มีค่าความแม่นยำต่างกันร้อยละ 2.30
- รถขนาดกลาง มีค่าความแม่นยำต่างกันร้อยละ 15.02
- รถขนาดใหญ่ มีค่าความแม่นยำต่างกันร้อยละ 17.94

และในโปรแกรมนี้มีข้อจำกัดในการตรวจจับยานพาหนะเมื่อยานพาหนะเคลื่อนที่เข้ามาพร้อมกันและเกิดการทับซ้อนกันของภาพจะทำให้โปรแกรมแยกยานพาหนะนั้น ๆ ออกจากกันไม่ได้ จึงอาจทำให้เกิดความคลาดเคลื่อนในการนับจำนวนของโปรแกรม



รูปที่ 3.13 การตีกรอบและติดตามยานพาหนะ

เมื่อยานพาหนะเคลื่อนที่ผ่านเส้นแล้วจะได้ค่าแสดงออกที่มอนิเตอร์ โดยมีค่าที่แสดงคือแสดงว่ายานพาหนะเป็นประเภทใดและแสดงขนาดของภาพ ความกว้าง และความสูง และจะแสดงภาพยานพาหนะที่เราสนใจ

```
Motorcycle DOWN
1  .> ID: 1  going down at Tue Sep 15 10:30:32 2020 402 285 117
Motorcycle DOWN
2  .> ID: 2  going down at Tue Sep 15 10:30:37 2020 379 265 114
Motorcycle DOWN
3  .> ID: 4  going down at Tue Sep 15 10:30:40 2020 389 274 115
Motorcycle DOWN
4  .> ID: 5  going down at Tue Sep 15 10:30:43 2020 445 250 195
```

Ln: 292 Col: 0

รูปที่ 3.14 ค่าที่ได้จากการนับยานพาหนะ

มหาวิทยาลัยเทคโนโลยีสุรนารี

รูปที่ 3.15 ยานพาหนะที่สนใจ

บทที่ 4
ผลการดำเนินการวิจัยและการอภิปรายผล

4.1 กล่าวนำ

บทนี้จะกล่าวถึงผลการตรวจสอบความถูกต้องของโปรแกรม หลังจากได้ออกแบบระบบแยกประเภทและนับจำนวนยานพาหนะสมบูรณ์แล้ว โดยผลการทดลองนี้เก็บผลการทดลองเป็นเวลา 5 วัน และแบ่งวิดีโอออกเป็นจำนวน 50 วิดีโอ การทดลองนี้ทดลองเปรียบเทียบการนับจำนวนยานพาหนะของโปรแกรมและการนับยานพาหนะของมนุษย์

4.2 ผลการตรวจสอบความถูกต้องของโปรแกรม

งานวิจัยนี้ได้ตรวจสอบความถูกต้องของโปรแกรมด้วยการเปรียบเทียบการนับจำนวนยานพาหนะแต่ละประเภทของมนุษย์เทียบกับโปรแกรมที่สร้างขึ้น โดยใช้วิดีโอทั้งหมด 5 วัน แบ่งวันละ 10 วิดีโอ รวมทั้งหมด 50 วิดีโอ ซึ่งสามารถแสดงได้ดังนี้

ตารางที่ 4.1 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดเล็ก วันที่ 1

วิดีโอ	นับโดยมนุษย์	นับโดยโปรแกรม
1	995	640
2	157	149
3	827	568
4	16	16
5	129	137
6	721	537
7	36	39
8	277	281
9	90	94
10	258	252
รวม	**3,506**	**2,713**

มหาวิทยาลัยเทคโนโลยีสุรนารี

ตารางที่ 4.2 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดเล็ก วันที่ 2

วิดีโอ	นับโดยมนุษย์	นับโดยโปรแกรม
1	774	644
2	160	150
3	574	489
4	26	27
5	264	246
6	734	505
7	33	33
8	187	179
9	46	46
10	410	358
รวม	**3,208**	**2,677**

ตารางที่ 4.3 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดเล็ก วันที่ 3

วิดีโอ	นับโดยมนุษย์	นับโดยโปรแกรม
1	590	498
2	279	249
3	328	299
4	21	21
5	154	167
6	494	413
7	41	38
8	87	92
9	86	84
10	131	124
รวม	**2,211**	**1,985**

มหาวิทยาลัยเทคโนโลยีสุรนารี

ตารางที่ 4.4 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดเล็ก วันที่ 4

วิดีโอ	นับโดยมนุษย์	นับโดยโปรแกรม
1	171	155
2	52	52
3	51	50
4	2	2
5	42	41
6	137	130
7	20	20
8	5	5
9	36	36
10	17	19
รวม	**533**	**510**

ตารางที่ 4.5 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดเล็ก วันที่ 5

วิดีโอ	นับโดยมนุษย์	นับโดยโปรแกรม
1	408	371
2	73	72
3	373	321
4	1	2
5	40	43
6	442	405
7	17	18
8	202	197
9	26	27
10	286	249
รวม	**1,868**	**1,705**

มหาวิทยาลัยเทคโนโลยีสุรนารี

จากข้อมูลที่ได้จากการวิจัยเพื่อตรวจสอบความถูกต้องของโปรแกรม ด้วยการเปรียบเทียบการนับจำนวนรถขนาดเล็กของมนุษย์เทียบกับโปรแกรมที่สร้างขึ้น โดยใช้วิดีโอทั้งหมด 5 วัน สามารถวิเคราะห์ผลได้ดังนี้

1) ค่าความแม่นยำเฉลี่ย (Accuracy) ของการทำงานโดยรวมวันที่ 1 คิดเป็นร้อยละ 77.38
2) ค่าความแม่นยำเฉลี่ย (Accuracy) ของการทำงานโดยรวมวันที่ 2 คิดเป็นร้อยละ 83.45
3) ค่าความแม่นยำเฉลี่ย (Accuracy) ของการทำงานโดยรวมวันที่ 3 คิดเป็นร้อยละ 89.78
4) ค่าความแม่นยำเฉลี่ย (Accuracy) ของการทำงานโดยรวมวันที่ 4 คิดเป็นร้อยละ 95.68
5) ค่าความแม่นยำเฉลี่ย (Accuracy) ของการทำงานโดยรวมวันที่ 5 คิดเป็นร้อยละ 91.27

ในข้อมูลข้างต้นแสดงค่าการนับจำนวนรถขนาดเล็กเปรียบเทียบระหว่างการนับโดยมนุษย์และการนับโดยโปรแกรมค่าความแม่นยำเฉลี่ยทั้งหมด 5 วัน ของการทำงานของโปรแกรม คิดเป็นร้อยละ 87.51 และค่าส่วนเบี่ยงเบนมาตรฐานของค่าความแม่นยำเท่ากับ 7.16

ตารางที่ 4.6 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดกลาง วันที่ 1

วิดีโอ	**นับโดยมนุษย์**	**นับโดยโปรแกรม**
1	31	29
2	18	15
3	23	19
4	5	5
5	14	13
6	37	28
7	10	9
8	30	28
9	12	10
10	56	42
รวม	**236**	**198**

มหาวิทยาลัยเทคโนโลยีสุรนารี

ตารางที่ 4.7 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดกลาง วันที่ 2

วิดีโอ	นับโดยมนุษย์	นับโดยโปรแกรม
1	21	19
2	20	17
3	39	37
4	2	2
5	16	15
6	52	45
7	5	6
8	13	11
9	10	9
10	29	33
รวม	**207**	**194**

ตารางที่ 4.8 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดกลาง วันที่ 3

วิดีโอ	นับโดยมนุษย์	นับโดยโปรแกรม
1	32	27
2	26	22
3	20	21
4	7	8
5	11	12
6	24	25
7	8	9
8	22	17
9	8	8
10	8	9
รวม	**166**	**158**

มหาวิทยาลัยเทคโนโลยีสุรนารี

ตารางที่ 4.9 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดกลาง วันที่ 4

วิดีโอ	นับโดยมนุษย์	นับโดยโปรแกรม
1	38	32
2	23	21
3	3	3
4	1	1
5	11	11
6	36	33
7	6	6
8	3	3
9	6	6
10	10	10
รวม	**137**	**126**

ตารางที่ 4.10 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดกลาง วันที่ 5

วิดีโอ	นับโดยมนุษย์	นับโดยโปรแกรม
1	57	55
2	29	26
3	23	18
4	3	3
5	11	12
6	66	54
7	3	3
8	21	17
9	9	8
10	23	24
รวม	**245**	**220**

มหาวิทยาลัยเทคโนโลยีสุรนารี

จากข้อมูลที่ได้จากการวิจัยเพื่อตรวจสอบความถูกต้องของโปรแกรม ด้วยการเปรียบเทียบการนับจำนวนรถขนาดกลางของมนุษย์เทียบกับโปรแกรมที่สร้างขึ้น โดยใช้วิดีโอทั้งหมด 5 วัน สามารถวิเคราะห์ผลได้ดังนี้

1) ค่าความแม่นยำเฉลี่ย (Accuracy) ของการทำงานโดยรวมวันที่ 1 คิดเป็นร้อยละ 83.90
2) ค่าความแม่นยำเฉลี่ย (Accuracy) ของการทำงานโดยรวมวันที่ 2 คิดเป็นร้อยละ 93.72
3) ค่าความแม่นยำเฉลี่ย (Accuracy) ของการทำงานโดยรวมวันที่ 3 คิดเป็นร้อยละ 95.18
4) ค่าความแม่นยำเฉลี่ย (Accuracy) ของการทำงานโดยรวมวันที่ 4 คิดเป็นร้อยละ 91.97
5) ค่าความแม่นยำเฉลี่ย (Accuracy) ของการทำงานโดยรวมวันที่ 5 คิดเป็นร้อยละ 90.45

ในข้อมูลข้างต้นแสดงค่าการนับจำนวนรถขนาดกลางเปรียบเทียบระหว่างการนับโดยมนุษย์และการนับโดยโปรแกรมค่าความแม่นยำเฉลี่ยทั้งหมด 5 วัน ของการทำงานของโปรแกรม คิดเป็นร้อยละ 90.91 และค่าส่วนเบี่ยงเบนมาตรฐานของค่าความแม่นยำเท่ากับ 4.41

ตารางที่ 4.11 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดใหญ่ วันที่ 1

วิดีโอ	**นับโดยมนุษย์**	**นับโดยโปรแกรม**
1	12	10
2	4	4
3	6	5
4	6	6
5	3	2
6	6	6
7	3	3
8	5	5
9	5	5
10	4	5
รวม	**54**	**51**

มหาวิทยาลัยเทคโนโลยีสุรนารี

ตารางที่ 4.12 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดใหญ่ วันที่ 2

วิดีโอ	นับโดยมนุษย์	นับโดยโปรแกรม
1	16	14
2	6	5
3	6	5
4	6	6
5	4	3
6	5	4
7	6	6
8	6	4
9	5	5
10	4	2
รวม	**64**	**54**

ตารางที่ 4.13 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดใหญ่ วันที่ 3

วิดีโอ	นับโดยมนุษย์	นับโดยโปรแกรม
1	13	11
2	5	5
3	3	3
4	6	4
5	3	4
6	5	5
7	5	6
8	6	6
9	5	5
10	4	3
รวม	**55**	**52**

มหาวิทยาลัยเทคโนโลยีสุรนารี

ตารางที่ 4.14 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดใหญ่ วันที่ 4

วิดีโอ	นับโดยมนุษย์	นับโดยโปรแกรม
1	2	2
2	2	3
3	2	2
4	2	2
5	3	2
6	2	2
7	2	2
8	2	1
9	2	2
10	2	2
รวม	**21**	**20**

ตารางที่ 4.15 เปรียบการทำงานของโปรแกรมในการนับจำนวนรถขนาดใหญ่ วันที่ 5

วิดีโอ	นับโดยมนุษย์	นับโดยโปรแกรม
1	12	12
2	4	4
3	2	2
4	2	2
5	2	1
6	3	3
7	3	4
8	3	3
9	3	4
10	2	1
รวม	**36**	**36**

มหาวิทยาลัยเทคโนโลยีสุรนารี

จากข้อมูลที่ได้จากการวิจัยเพื่อตรวจสอบความถูกต้องของโปรแกรม ด้วยการเปรียบเทียบการนับจำนวนรถขนาดใหญ่ของมนุษย์เทียบกับโปรแกรมที่สร้างขึ้น โดยใช้วิดีโอทั้งหมด 5 วัน สามารถวิเคราะห์ผลได้ดังนี้

1) ค่าความแม่นยำเฉลี่ย (Accuracy) ของการทำงานโดยรวมวันที่ 1 คิดเป็นร้อยละ 94.44
2) ค่าความแม่นยำเฉลี่ย (Accuracy) ของการทำงานโดยรวมวันที่ 2 คิดเป็นร้อยละ 84.38
3) ค่าความแม่นยำเฉลี่ย (Accuracy) ของการทำงานโดยรวมวันที่ 3 คิดเป็นร้อยละ 94.55
4) ค่าความแม่นยำเฉลี่ย (Accuracy) ของการทำงานโดยรวมวันที่ 4 คิดเป็นร้อยละ 95.24
5) ค่าความแม่นยำเฉลี่ย (Accuracy) ของการทำงานโดยรวมวันที่ 5 คิดเป็นร้อยละ 100

ในข้อมูลข้างต้นแสดงค่าการนับจำนวนรถขนาดใหญ่เปรียบเทียบระหว่างการนับโดยมนุษย์และการนับโดยโปรแกรมค่าความแม่นยำเฉลี่ยทั้งหมด 5 วัน ของการทำงานของโปรแกรม คิดเป็นร้อยละ 93.72 และค่าส่วนเบี่ยงเบนมาตรฐานของค่าความแม่นยำเท่ากับ 5.71

จากผลของการทดลองของโปรแกรมที่ใช้ในการนับจำนวนยานพาหนะและแยกประเภทยานพาหนะทั้ง 3 ประเภท จำนวน 50 วิดีโอนั้น จะเห็นว่าในบางวิดีโอที่มีความหนาแน่นของยานพาหนะสูงจะมีค่าความแม่นยำต่ำกว่าวิดีโอที่มีความหนาแน่นต่ำ ทั้งนี้เนื่องจากมุมของการติดตั้งกล้องที่ทำให้เกิดภาพทับซ้อนกันซึ่งทำให้ตัวโปรแกรมไม่สามารถแยกยานพาหนะนั้น ๆ ออกจากกันได้และด้วยปริมาณแสงและเงาที่เกิดขึ้นแตกต่างกันตามแต่ละช่วงเวลาของวิดีโอก็เป็นผลทำให้เกิดความคลาดเคลื่อนในการทำงานของโปรแกรมนั้น ๆ ด้วย ซึ่งถ้าหาค่าความแม่นยำของความถูกต้องของวิดีโอทั้งหมดนั้นจะเห็นว่าการทำงานของโปรแกรมมีความแม่นยำค่อนข้างสูงเมื่อเทียบกับการนับค่าโดยมนุษย์

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 5
สรุปผลการวิจัย

5.1 กล่าวนำ

ในบทนี้จะกล่างถึงบทสรุปผลการทำงานของโปรแกรมแยกประเภทและนับจำนวนยานพาหนะที่เคลื่อนที่ผ่านบริเวณการจราจรอาคารเรียนรวม 1 โดยใช้หลักการมองเห็นของเครื่องจักร โดยในที่นี้จะกล่าวถึงผลที่ได้จากการทดสอบการทำงานและความถูกต้อง รวมถึงข้อเสนอแนะ

5.2 สรุปผลการวิจัย

ในการสร้างระบบแยกประเภทและนับจำนวนยานพาหนะ โดยการใช้การมองเห็นของเครื่องจักร แสดงให้เห็นว่าระบบสามารถแยกประเภทและนับจำนวนยานพาหนะนั้นมีค่าความถูกต้องแม่นยำที่สูงเมื่อเทียบกับการใช้มนุษย์ในการนับจำนวน แต่เนื่องจากปัญหาทางด้านมุมของการติดตั้งกล้อง และปัจจัยทางด้านสภาพแวดล้อมต่าง ๆ รวมไปถึงความเร็วในการเคลื่อนที่ของยานพาหนะนั้น เป็นผลให้การแยกยานพาหนะออกจากภาพพื้นหลัง อาจเกิดความผิดพลาดในเรื่องของขนาดเนื่องจากแสงและเงาของภาพที่ผิดเพี้ยนไปตามแต่ละช่วงเวลา ทำให้การคำนวณค่าของโปรแกรมนั้นมีข้อผิดพลาดในการแยกประเภทและนับจำนวนของยานพาหนะ

มหาวิทยาลัยเทคโนโลยีสุรนารี

รายการอ้างอิง

เกริกพงษ์ เกียรติพานิชกิจ.(2017). **Introduction to Machine Vision**, สืบค้นเมื่อ 11 มกราคม 2563. จากเว็บไซต์: http://www.geocities.ws/k_kirkpong/chapter01.pdf.

สนั่น งานวิวัฒน์ถาวร. (2551), **Motion Detection By Background Subtraction**. คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์.

ไตรวิทย์ อินทจักร. (2556). ระบบตรวจจับและคัดแยกรถสำหรับกล้องวงจรปิดบนท้องถนน. **วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต, มหาวิทยาลัยสงขลานครินทร์**.

Arya, K. V., Tiwari, S., & Behwalc, S. (2016). **Real-time vehicle detection and tracking**.

Memon, S., Bhatti, S., Ali, L., Talpur, M., & Memon, M. (2018). A Video based Vehicle Detection, Counting and Classification System. **International Journal of Image, Graphics and Signal Processing**, 10, 34-41.

มหาวิทยาลัยเทคโนโลยีสุรนารี

ภาคผนวก ก

การเขียนโปรแกรมควบคุมการทำงานของระบบ

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
import cv2
import numpy as np
import Vehicle
import time
count_down = 0
DownMTR = 0
DownCAR = 0
DownBUS = 0
num_car = 0
cap = cv2.VideoCapture('3_15082020.MP4')
for i in range(19):
        #print properties video
        print (i, cap.get(i))
w = cap.get(3)
print ('Width', w)
h = cap.get(4)
print ('Height', h)
frameArea = h*w
areaTH = frameArea/800
print ('Area Threshold', areaTH)
line_down = 550 # position of line blue
line_down2 = 600 # position of line red
down_limit = line_down + 20
        print ("Blue line y:", str(line_down))
line_down_color = (255,0,0)
        print ("Red line y:", str(line_down2))
line_down_color2 = (0,0,255)
        #set h,w line blue
pt1 = [0, line_down]
pt2 = [w, line_down]
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
pts_L1 = np.array([pt1,pt2], np.int32)
pts_L1 = pts_L1.reshape((-1,1,2))
        #set h,w line red
pt3 = [0, line_down2]
pt4 = [w, line_down2]
pts_L2 = np.array([pt3,pt4], np.int32)
pts_L2 = pts_L2.reshape((-1,1,2))
        #line limit white
pt5 = [0, down_limit]
pt6 = [w, down_limit]
pts_L3 = np.array([pt5,pt6], np.int32)
pts_L3 = pts_L3.reshape((-1,1,2))
        #Create the background subtractor
fgbg = cv2.createBackgroundSubtractorMOG2()
kernelOp = np.ones((3,3), np.uint8)
kernelOp2 = np.ones((5,5), np.uint8)
kernelCl = np.ones((11,11), np.uint8)
        #Variables
font = cv2.FONT_HERSHEY_SIMPLEX
vehicles = []
max_p_avg = 5
pid = 1
        while(cap.isOpened()):
        ret, frame = cap.read()
    for i in vehicles:
        i.avg_one()
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
cv2.imshow('Gray Image',cv2.resize(gray, (400, 300)))
fgmask = fgbg.apply(gray)
fgmask2 = fgbg.apply(gray)
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
try:
ret, imBin = cv2.threshold(fgmask, 200, 255, cv2.THRESH_BINARY)
ret, imBin2 = cv2.threshold(fgmask2, 200, 255, cv2.THRESH_BINARY)
mask = cv2.morphologyEx(imBin, cv2.MORPH_OPEN, kernelOp)
mask2 = cv2.morphologyEx(imBin2, cv2.MORPH_OPEN, kernelOp)
mask = cv2.morphologyEx(mask, cv2.MORPH_CLOSE, kernelCl)
mask2 = cv2.morphologyEx(mask2, cv2.MORPH_CLOSE, kernelCl)
cv2.imshow('Background Subtraction', cv2.resize(fgmask, (400, 300)))
cv2.imshow('Masked Image', cv2.resize(mask, (400, 300)))
cv2.imshow('Masked Image2', cv2.resize(mask2, (400, 300)))
except:
        print ('Number of Motorcycle Down:', DownMTR)
        print ('Number of Car Down:', DownCAR)
        print ('Number of Bus Down:', DownBUS)
        print ('TOTAL DOWN:', count_down)
        break
contours, hierarchy = cv2.findContours(mask, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_NONE)
    for cnt in contours:
        #create lines to track moving objects
cv2.drawContours(gray, cnt, -1, (0,255,0), 3, 8)
area = cv2.contourArea(cnt)
    if area > areaTH:
        M = cv2.moments(cnt)
        cx = int(M['m10']/M['m00'])
        cy = int(M['m01']/M['m00'])
        x,y,w,h = cv2.boundingRect(cnt)
    def pause():
programPause = raw_input("Press the <ENTER> key to continue...")
new = True
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
        for i in vehicles:
            if abs(x-i.getX()) <= w and abs(y-i.getY()) <= h:
                new = False
                i.updateCoords(cx,cy)
            if i.going_DOWN(line_down2,line_down2) == True:
                obj = gray[y:y+h, x:x+w]
                rectangle = cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
                height = y + h
                width = x + w
                Size_DOWN = h + w
            if 715 < Size_DOWN < 800:    # car
            if 300 < h < 400 and 400 < w < 450:
                        DownCAR += 1
                        count_down += 1
                        print("Car DOWN")
                        num_car += 1
print(num_car," .> ID:", i.getId(), ' going down at', time.strftime("%c"),Size_DOWN, h, w)
             elif 800 < Size_DOWN < 1300:    #car
            if 300 <= h < 450 and 450 < w < 600:
                        DownCAR += 1
                        count_down += 1
print("Car DOWN")
                        num_car += 1
print(num_car," .> ID:", i.getId(), ' going down at', time.strftime("%c"),Size_DOWN, h, w)
            elif 300 < Size_DOWN < 700:     #motorcycle
            if 150 < h < 390 and 100 < w < 230:
            if h > w:
                        DownMTR += 1
                        count_down += 1
print("Motorcycle DOWN")
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
                num_car += 1
print(num_car," .> ID:", i.getId(), ' going down at', time.strftime("%c"),Size_DOWN, h, w)
cv2.imshow('Object Detection', obj)
        elif i.going_DOWN(line_down,line_down) == True:
                obj = gray[y:y+h, x:x+w]
                rectangle = cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
                height = y + h
                width = x + w
                Size_DOWN = h + w
        if 1800 < Size_DOWN < 2600:    #bus
        if h > 800 and w < 1310:
        if h < w :
                DownBUS += 1
                count_down += 1
                print("Bus DOWN")
                num_car += 1
print(num_car," > ID:", i.getId(), ' going down at', time.strftime("%c"),Size_DOWN, h, w)
        cv2.imshow('Object Detection', obj)
        break
        if i.getState() == '1':
        if i.getDir() == 'down' and i.getY() > down_limit:
                i.setDone()
        if i.timedOut():
                index = vehicles.index(i)
                vehicles.pop(index)
        del i
        if new == True:
                p = Vehicle.MyVehicle(pid,cx,cy, max_p_avg)
                vehicles.append(p)
        pid += 1
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
cv2.circle(gray, (cx,cy),5, (0,0,255), -1)
img = cv2.rectangle(gray, (x,y),(x+w,y+h),(0,255,0),2)
cv2.drawContours(gray, cnt, -1, (0,255,0), 3)
cv2.imshow('Image', cv2.resize(img, (400, 300)))
cv2.circle(frame, (cx,cy),5, (0,0,255), -1)
img = cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,0),2)
            for i in vehicles:
      """
            if len(i.getTracks()) >= 2:
                        pts = np.array(i.getTracks(), np.int32)
                        pts = pts.reshape((-1,1,2))
                        frame = cv2.polylines(frame, [pts], False, i.getRGB())
            if i.getId() == 9:
print str(i.getX()), ',', str(i.getY())
      """
cv2.putText(gray, str(i.getId()), (i.getX(),i.getY()),font,0.3,i.getRGB(),1,cv2.LINE_AA)
str_down = 'TOTAL DOWN     : ' + str(count_down)
MTR_down = ' - Motorcycle  : ' + str(DownMTR)
CAR_down = ' - Car         : ' + str(DownCAR)
BUS_down = ' - Bus         : ' + str(DownBUS)
            #lines
frame = cv2.polylines(frame, [pts_L1], False, line_down_color, thickness=2)
frame = cv2.polylines(frame, [pts_L2], False, line_down_color2, thickness=2)
frame = cv2.polylines(frame, [pts_L3], False, (255,255,255), thickness=1)
cv2.putText(frame, str_down, (10,900), font, 2, (255,255,255),5, cv2.LINE_AA)
cv2.putText(frame, MTR_down, (10, 950), font, 2, (255, 255, 255), 5, cv2.LINE_AA)
cv2.putText(frame, CAR_down, (10, 1000), font, 2, (255, 255, 255), 5, cv2.LINE_AA)
cv2.putText(frame, BUS_down, (10, 1050), font, 2, (255, 255, 255), 5, cv2.LINE_AA)
cv2.imshow('Frame', cv2.resize(frame, (800, 600)))
k = cv2.waitKey(10) & 0xff
```

```
            if k == 27:
             break
cap.release()
cv2.destroyAllWindows()
```

ประวัติผู้เขียน

นางสาวฐานิดา ศรีชัยเพชร เกิดเมื่อวันที่ 7 พฤษภาคม พ.ศ. 2538 ที่จังหวัดนครราชสีมา เริ่มการศึกษาในระดับอนุบาลศึกษาถึงประถมศึกษาปีที่ 1 ที่โรงเรียนมารีย์ธงชัย จังหวัดนครราชสีมา ได้เข้าศึกษาในระดับประถมศึกษาปีที่ 2 ตลอดจนสำเร็จการศึกษาการศึกษาในระดับมัธยมศึกษาสายวิทย์คณิตที่โรงเรียนมารีย์วิทยา จังหวัดนครราชสีมา ได้สำเร็จการศึกษาวิศวกรรมศาสตรบัณฑิต (วิศวกรรมคอมพิวเตอร์) สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา เมื่อ พ.ศ. 2560 และในปีเดียวกันได้ศึกษาต่อระดับวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมเมคคาทรอนิกส์ ขณะศึกษาได้เป็นผู้สอนปฏิบัติการของสาขาวิศวกรรมเครื่องกล จำนวน 2 รายวิชา ได้แก่ (1) ระบบอัตโนมัติอุตสาหกรรม และ (2) ปฏิบัติการระบบควบคุมและอัตโนมัติ เป็นเวลา 2 ปี

มหาวิทยาลัยเทคโนโลยีสุรนารี