

ข้อเสนอโครงการ
Advanced Image Processing วิชา 2102514

ระบบตรวจสอบสภาพจราจรติดขัดด้วย
การประมวลผลภาพจากกล้องวงจรปิด

The Traffic Congestion Investigating System by
Image Processing from CCTV Camera

จัดทำโดย
นายธีรวัฒน์ เลิศอัมพรวิทย์ เลขประจำตัว 6430183721

เสนอ
รศ. ดร. ชานูชัย ปลื้มปิติวิริยะเวช

ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2567

สารบัญ

1. บทนำ	3
1.1 ที่มาและความสำคัญของโครงการ	3
1.2 วัตถุประสงค์ของโครงการ	3
1.3 ขอบเขตของโครงการ	3
1.4 ผลลัพธ์ที่คาดหวังจากโครงการ	4
1.5 องค์ความรู้ทางวิศวกรรมที่นำมาประยุกต์ใช้	4
2. หลักการและทฤษฎีที่เกี่ยวข้อง	5
2.1 ภาพระดับเทา (Grayscale Image)	5
2.2 การคำนวณความแตกต่างของพิกเซล (Pixel Difference)	5
2.3 เทคนิคการลบพื้นหลัง (Background Subtraction Technique)	5
2.4 การกรองค่ามัธยฐาน (Median Filtering)	6
2.5 ROI (Region of Interest)	6
2.6 การขยายภาพ (Dilation Operation)	7
2.7 การกร่อนภาพ (Erosion Operation)	8
2.8 การปิดภาพ (Closing Operation)	9
2.9 การเปิดภาพ (Opening Operation)	9
2.10 การตรวจจับขอบภาพ (Edge Detection)	10
2.11 วิธีการตรวจจับขอบเขตวัตถุ (Contour-Based Method)	11
3. แนวทางการดำเนินงาน	13
3.1 การตรวจสอบสภาพการจราจรโดยใช้ภาพจราจรจำนวน 1 ภาพ (Single Image)	13
3.2 การตรวจสอบสภาพการจราจรโดยใช้ภาพจราจรจำนวนหลายภาพ (Multiple Images)	16

4. ผลลัพธ์จากการดำเนินการเบื้องต้น	17
4.1 การตรวจสอบสภาพการจราจรโดยใช้ภาพจราจรจำนวน 1 ภาพ (Single Image)	17
4.2 การตรวจสอบสภาพการจราจรโดยใช้ภาพจราจรจำนวนหลายภาพ (Multiple Images)	22
5. บทสรุป	24
5.1 บทสรุปการทำโครงการจนถึงปัจจุบัน	24
5.2 แผนการดำเนินงาน	24
5.3 ปัญหา อุปสรรค และแนวทางแก้ไข	25
6. เอกสารอ้างอิง	26

1. บทนำ

1.1 ที่มาและความสำคัญของโครงการ

ปัจจุบันปัญหาการจราจรติดขัดมีความรุนแรงมากขึ้น เนื่องมาจากการเพิ่มขึ้นของประชากร ส่งผลให้การเดินทางใช้เวลานานขึ้น แม้จะมีระบบเทคโนโลยีที่ช่วยอธิบายสภาพการจราจรและแนะนำเส้นทางเดินทาง แต่การตรวจสอบสภาพการจราจรติดขัดที่มีอยู่ในปัจจุบันยังมีข้อจำกัดหลายประการ

วิธีการตรวจสอบสภาพการจราจรมีหลายรูปแบบ เช่น การใช้อุปกรณ์ตรวจจับปริมาณและความเร็วของรถยนต์ที่ติดตั้งข้างถนน ซึ่งมีค่าใช้จ่ายสูงและติดตั้งยาก เนื่องจากต้องใช้เซ็นเซอร์หนึ่งตัวสำหรับการนับรถ และอีกสองตัวสำหรับการตรวจจับความเร็ว อีกวิธีหนึ่งคือการใช้กล้อง CCTV ถ่ายภาพจราจร แล้วให้เจ้าหน้าที่ตำรวจวิเคราะห์สภาพการจราจรจากภาพถ่ายเหล่านั้น ซึ่งต้องใช้ทรัพยากรบุคคลและเวลามาก

จากข้อจำกัดดังกล่าว ผู้จัดทำจึงมีแนวคิดในการสร้างระบบที่สามารถแจ้งเตือนและวิเคราะห์สภาพการจราจร โดยระบบจะประมวลผลด้วยคอมพิวเตอร์และส่งผลลัพธ์ของสภาพการจราจรไปยังระบบ ซึ่งเจ้าหน้าที่สามารถนำผลลัพธ์นั้นไปใช้ในการวางแผนหรือควบคุมระบบสัญญาณไฟจราจรได้

การพัฒนาระบบตรวจสอบสภาพการจราจรติดขัดโดยใช้การประมวลผลภาพจากกล้อง CCTV จึงเป็นการนำเทคโนโลยีการประมวลผลภาพมาประยุกต์ใช้เพื่อแก้ไขปัญหาการจราจร ช่วยลดต้นทุนในการติดตั้งอุปกรณ์ตรวจจับพิเศษ และลดภาระงานของเจ้าหน้าที่ในการวิเคราะห์ภาพจราจร

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อพัฒนาระบบตรวจสอบสภาพการจราจร โดยใช้การประมวลผลภาพจากกล้อง CCTV ที่สามารถวิเคราะห์และจำแนกสภาพการจราจรออกเป็นสามระดับ ได้แก่ การจราจรคล่องตัว (Flow) การจราจรหนาแน่น (Heavy) และการจราจรติดขัด (Jammed)
2. เพื่อสร้างระบบที่สามารถนำไปใช้ในการควบคุมการจราจรที่ทางแยกได้อย่างมีประสิทธิภาพ โดยอาศัยเทคนิคการประมวลผลภาพ ได้แก่ Image Segmentation, Morphological Image Processing ในการวิเคราะห์สภาพการจราจรบนท้องถนน

1.3 ขอบเขตของโครงการ

1. ใช้เทคนิคการประมวลผลภาพ (Image Processing) ได้แก่ Image Segmentation, Morphological Image Processing ในการตรวจจับยานพาหนะที่แล่นผ่านบนท้องถนน

2. ใช้ข้อมูลนำเข้าเป็นภาพจากกล้อง CCTV ที่ถ่ายบริเวณท้องถนน ได้แก่
 - ภาพพื้นหลัง (Background) เป็นภาพถนนที่ไม่มียานพาหนะสัญจร
 - ภาพการจราจร (Image) เป็นภาพถนนที่ใช้ในการจำแนกสภาพการจราจร
3. การจำแนกสภาพการจราจร แบ่งเป็น 3 กลุ่ม ได้แก่
 - การจราจรคล่องตัว (Flow) มียานพาหนะสัญจรบนถนนจำนวนน้อย
 - การจราจรหนาแน่น (Heavy) มียานพาหนะสัญจรบนถนนจำนวนปานกลาง
 - การจราจรติดขัด (Jammed) มียานพาหนะสัญจรบนถนนจำนวนมาก

1.4 ผลลัพธ์ที่คาดหวังจากโครงการ

สามารถวิเคราะห์และจำแนกสภาพการจราจรออกเป็นสามระดับ ได้แก่ การจราจรคล่องตัว (Flow) การจราจรหนาแน่น (Heavy) และการจราจรติดขัด (Jammed)

1.5 องค์ความรู้ทางวิศวกรรมที่นำมาประยุกต์ใช้

1. ความรู้เกี่ยวกับการประมวลผลภาพ (Image Processing) ได้แก่ Image Segmentation, Morphological Image Processing
2. ทักษะการเขียนโปรแกรมคอมพิวเตอร์ ได้แก่ ภาษา Python

2. หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 ภาพระดับเทา (Grayscale Image)

ภาพระดับเทา คือ ภาพที่เกิดจากการสุ่มสีเพียงหนึ่งช่องสัญญาณในแต่ละพิกเซล ภาพประเภทนี้ประกอบด้วยสีดำ เฌดสีเทา และสีขาว ซึ่งแตกต่างจากภาพขาวดำที่มีเพียงสีดำและสีขาวเท่านั้นในการเก็บข้อมูลภาพ ภาพระดับเทาเป็นผลลัพธ์จากการวัดความเข้มของแสงในแต่ละพิกเซลในคลื่นแม่เหล็กไฟฟ้า เช่น แสงขาว ภาพระดับเทาแบบที่มองเห็นได้โดยทั่วไปใช้ข้อมูล 8 บิตในการเก็บค่าแต่ละพิกเซล ซึ่งมีระดับความเข้มของแสง 256 ระดับ อย่างไรก็ตาม สำหรับการใช้งานทางเทคนิคอื่นๆ ภาพอาจใช้ 10 หรือ 12 บิตสำหรับการเก็บข้อมูล [1]

ภาพระดับเทาเป็นพื้นฐานสำคัญในการประมวลผลภาพ เนื่องจากทำให้การคำนวณและการวิเคราะห์ง่ายขึ้น ลดความซับซ้อนของข้อมูลเมื่อเทียบกับภาพสี ซึ่งมีข้อมูล 3 ช่องสัญญาณ (แดง, เขียว, น้ำเงิน) ในแต่ละพิกเซล

2.2 การคำนวณความแตกต่างของพิกเซล (Pixel Difference)

การคำนวณความแตกต่างของพิกเซลหรือการลบพิกเซล เป็นกระบวนการหาภาพผลลัพธ์ที่สามจากภาพนำเข้าสู่สองภาพ วิธีการนี้ทำการลบค่าพิกเซลของภาพแรกด้วยค่าพิกเซลของภาพที่สอง ในบางกรณี อาจเป็นไปได้ที่จะมีภาพนำเข้าเพียงภาพเดียวแล้วลบด้วยค่าทางสถิติ [1]

การคำนวณความแตกต่างของพิกเซลมีความสำคัญในการตรวจจับการเคลื่อนไหวและการเปลี่ยนแปลงระหว่างเฟรมภาพที่ต่อเนื่องกัน ซึ่งเป็นพื้นฐานสำคัญของเทคนิคการลบพื้นหลัง (Background Subtraction) ที่ใช้ในการตรวจจับวัตถุเคลื่อนไหว

2.3 เทคนิคการลบพื้นหลัง (Background Subtraction Technique)

การลบพื้นหลังเป็นเทคนิคที่ใช้กันอย่างแพร่หลายในการประมวลผลภาพ โดยมีวัตถุประสงค์หลักเพื่อแยกวัตถุที่เคลื่อนไหวออกจากฉากหลังในลำดับของเฟรมจากกล้องนิ่ง เทคนิคนี้ช่วยให้สามารถแยกส่วนของภาพเบื้องหน้า (Foreground: วัตถุที่เคลื่อนไหว) ออกจากพื้นหลัง (Background: วัตถุที่อยู่นิ่ง) ได้อย่างชัดเจน ซึ่งสามารถนำไปใช้ประมวลผลต่อในขั้นตอนถัดไป เช่น การจดจำวัตถุ [2]

เทคนิคการลบพื้นหลัง (Background Subtraction) อาศัยสมมติฐานสำคัญว่า “พื้นหลังของฉากมีลักษณะคงที่” ดังนั้นเทคนิคนี้อาจไม่เหมาะสมในกรณีที่ฉากมีการเปลี่ยนแปลงตลอดเวลา เช่น มีเงาเปลี่ยนทิศทางตามแสง มีแสงไฟกระพริบ หรือมีการเคลื่อนไหวของวัตถุในพื้นหลัง เช่น ใบไม้เคลื่อนไหว

อย่างไรก็ตาม เทคนิคการลบพื้นหลังสามารถนำไปประยุกต์ในชีวิตประจำวัน เช่น ระบบกล้องวงจรปิดภายในบ้าน ซึ่งไม่จำเป็นต้องบันทึกภาพตลอดเวลา แต่เน้นการตรวจจับและบันทึกเฉพาะช่วงเวลาที่มีการเคลื่อนไหวเกิดขึ้น ช่วยลดปริมาณข้อมูลที่ต้องจัดเก็บ และเพิ่มประสิทธิภาพในการเฝ้าระวังเหตุการณ์ที่สำคัญ



รูปที่ 1 เทคนิคการลบพื้นหลัง [2]

2.4 การกรองค่ามัธยฐาน (Median Filtering)

การกรองค่ามัธยฐานเป็นวิธีการในเทคนิคการประมวลผลภาพ วิธีการนี้ช่วยลดสัญญาณรบกวนหรือจุดเล็กๆ ในภาพ โดยเฉพาะสัญญาณรบกวนแบบเกลือและพริกไทย (salt and pepper noise) โดยใช้หน้ากาก (mask) ขนาดที่เป็นเลขคี่วางบนภาพ จากนั้นเรียงลำดับพิกเซลในกรอบภาพ แล้วนำค่าพิกเซลที่เป็นค่ากลางมาแทนที่พิกเซลในตำแหน่งที่พิจารณา [1]

การกรองค่ามัธยฐานเป็นเทคนิคที่มีประสิทธิภาพในการกำจัดสัญญาณรบกวนแบบจุด (impulse noise) โดยไม่ทำให้ขอบภาพเบลอนมากเกินไป ซึ่งแตกต่างจากการกรองค่าเฉลี่ย (mean filtering) ที่มักทำให้ขอบภาพเบลอ การกรองค่ามัธยฐานสามารถรักษารายละเอียดของภาพได้ดีกว่า

2.5 ROI (Region of Interest)

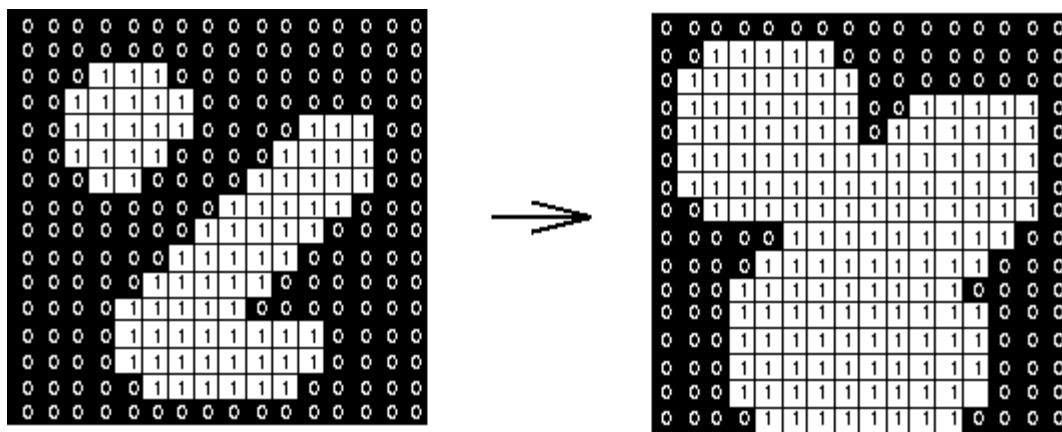
ROI (Region of Interest) คือ พื้นที่หรือส่วนหนึ่งของภาพ ที่เราสนใจหรือให้ความสำคัญในการประมวลผลภาพ (image processing) หรือการวิเคราะห์ข้อมูลในลักษณะเฉพาะ โดยไม่จำเป็นต้องพิจารณาทุกส่วนของภาพทั้งหมด เช่น หากเราต้องการตรวจจับวัตถุในภาพ เราสามารถเลือกเพียงบางส่วนของภาพที่มีวัตถุที่สนใจและดำเนินการประมวลผลแค่ในพื้นที่นั้นเพื่อประหยัดเวลาและทรัพยากรการคำนวณ

การใช้ ROI ช่วยให้สามารถจำกัดขอบเขตการทำงานและโฟกัสไปที่ส่วนที่สำคัญในภาพ เช่น การตรวจจับใบหน้าในภาพถ่าย หรือการตรวจจับรถยนต์ในภาพจากกล้องวงจรปิด

2.6 การขยายภาพ (Dilation Operation)

การขยายภาพเป็นการเปรียบเทียบระหว่างองค์ประกอบโครงสร้าง (structuring element) ในขนาดต่างๆ เช่น 3x3 หรือ 5x5 กับแต่ละพิกเซลในภาพ กระบวนการนี้ปรับปรุงค่าพิกเซลในตำแหน่งศูนย์กลางของหน้าต่าง โดยพิจารณาพิกเซลที่มีค่าสูงสุดเพื่อใส่ในตำแหน่งศูนย์กลางของหน้าต่าง สำหรับภาพขาวดำหรือภาพไบนารี จะพิจารณาเฉพาะพิกเซลที่มีค่าเป็น 1 แต่ไม่ใช่ 0 [1]

การขยายภาพทำให้วัตถุในภาพใหญ่ขึ้น โดยเติมช่องว่างเล็กๆ และเชื่อมต่อองค์ประกอบที่แยกกัน ซึ่งเป็นประโยชน์ในการเชื่อมต่อส่วนของวัตถุที่อาจถูกแยกออกจากกันในระหว่างกระบวนการประมวลผลภาพ



รูปที่ 2 การขยายภาพ (Dilation Operation) [3]

2.6.1 ลักษณะของการขยายภาพ [4]

1. การขยายจะเพิ่มพื้นที่ของวัตถุในภาพ ในขณะที่การกัดกร่อนจะลดขนาดวัตถุลง
 2. ขนาดของภาพหลังการขยายยังคงเหมือนเดิมกับภาพต้นฉบับ โดยไม่เปลี่ยนแปลงขนาดของภาพ
 3. การขยายจะเพิ่มขนาดของวัตถุ ซึ่งช่วยให้วัตถุในภาพดูใหญ่ขึ้น
 4. ความหนาของการขยายจะขึ้นอยู่กับขนาดและรูปร่างขององค์ประกอบที่โครงสร้างที่ใช้
 5. การขยายสามารถใช้ได้ทั้งกับภาพไบนารี และภาพระดับสีเทา
- การขยายภาพไบนารี ค่าพิกเซลจะถูกตั้งค่าเป็น 1 หากพิกเซลข้างเคียงมีค่าเป็น 1 ซึ่งทำให้ส่วนที่มีค่าเป็น 1 ขยายออกไป

- การขยายภาพระดับสีเทา ค่าของพิกเซลเอาต์พุตจะเป็นค่าสูงสุดของพิกเซลทั้งหมดในบริเวณใกล้เคียง ซึ่งช่วยให้ภาพดูสว่างขึ้นในบางพื้นที่

2.6.2 การประยุกต์ใช้การขยายภาพ [4]

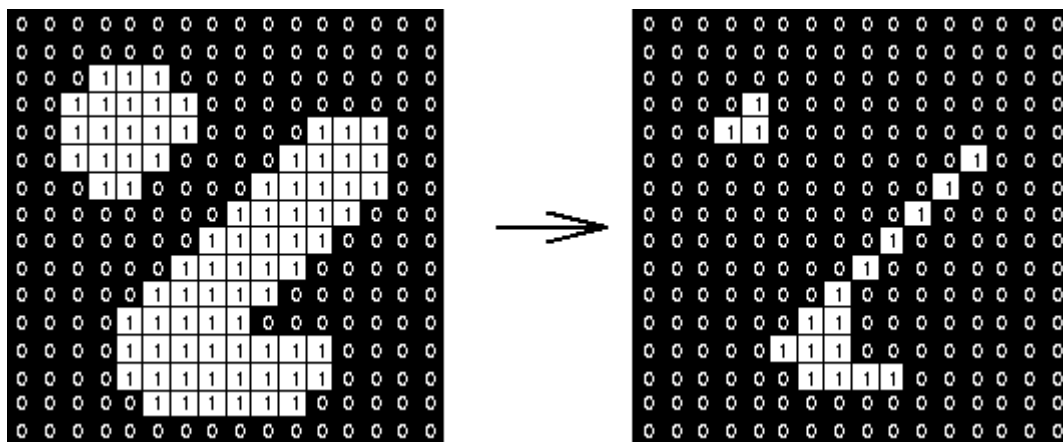
1. **Removing Pepper Noise** กระบวนการขยายเหมาะสมสำหรับการเติมเต็มช่องว่างเล็กๆ (สัญญาณรบกวนแบบพริกไทย) ในภาพ

2. **Edge Detection** กระบวนการขยายสามารถใช้ในการตรวจจับขอบในภาพอินพุตได้ โดยการทำการขยายภาพอินพุตและนำภาพที่ขยายแล้วมาลบออกจากภาพต้นฉบับ ซึ่งจะเหลือเพียงขอบเขตที่เกิดจากพิกเซลพื้นหลังที่เพิ่มขึ้นจากกระบวนการขยาย

2.7 การกร่อนภาพ (Erosion Operation)

การกร่อนภาพมีลักษณะคล้ายกับการขยายภาพ แต่มีความแตกต่างคือ การกร่อนให้ความสนใจกับค่าต่ำสุดของพิกเซล ซึ่งหมายความว่าพิกเซลจะมีค่าเป็น 0 หาก มีค่าในกรอบภาพเป็น 0 ในภาพไบนารี [1]

การกร่อนภาพทำให้วัตถุในภาพเล็กลง กำจัดพิกเซลสัญญาณรบกวนเล็กๆ และแยกวัตถุที่เชื่อมต่อกันเล็กน้อย นอกจากนี้ยังช่วยลบส่วนที่ยื่นออกมาที่บางจากวัตถุ ทำให้รูปร่างของวัตถุเรียบขึ้น



รูปที่ 3 การกร่อนภาพ (Erosion Operation) [5]

2.7.1 ลักษณะของการกร่อนภาพ [6]

1. การกร่อนภาพ จะทำให้วัตถุในภาพเล็กลงหรือหดตัว ซึ่งตรงข้ามกับการขยายที่ทำให้วัตถุขยายใหญ่ขึ้น
2. ขนาดของภาพหลังการกร่อนภาพ ยังคงเหมือนเดิมกับภาพต้นฉบับ โดยไม่เปลี่ยนแปลงขนาดของภาพ

3. การกร่อนภาพ จะทำให้ขอบของวัตถุในภาพเล็กลงและลดขนาดของวัตถุ
4. ลักษณะของการกร่อนภาพขึ้นอยู่กับขนาดและรูปร่างขององค์ประกอบโครงสร้างที่ใช้
5. การกร่อนภาพสามารถใช้ได้ทั้งกับภาพไบนารี และภาพระดับสีเทา
 - การกร่อนภาพของภาพไบนารี ค่าพิกเซลจะถูกตั้งค่าเป็น 1 หากพิกเซลข้างเคียงทั้งหมดมีค่าเป็น 1 ซึ่งทำให้พื้นที่ที่มีค่าเป็น 1 ลดขนาดลง
 - การกร่อนภาพของภาพระดับสีเทา ค่าของพิกเซลเอาต์พุตจะเป็นค่าต่ำสุดของพิกเซลทั้งหมดในบริเวณใกล้เคียง ซึ่งช่วยให้ภาพดูมืดลงในบางพื้นที่

2.7.2 การประยุกต์ใช้การกร่อนภาพ [6]

1. **Counting Objects** การกร่อนภาพสามารถใช้ในการแยกวัตถุที่เชื่อมต่อกันออกจากกัน เพื่อให้สามารถแยกและนับจำนวนวัตถุได้โดยใช้อัลกอริธึมการระบุ (labelling algorithms)
2. **Removing Salt Noise** กระบวนการกัดเซาะสามารถใช้เพื่อลบสัญญาณรบกวนแบบเกลือ (จุดสีขาว) ออกจากภาพได้

2.8 การปิดภาพ (Closing Operation)

การปิดภาพ เป็นกระบวนการขยายภาพ (Dilation) แล้วทำการกร่อนภาพ (Erosion) ตามลำดับ

ประโยชน์ของการปิดภาพ

- เชื่อมต่อวัตถุที่แยกกัน: เช่น วัตถุที่มีช่องว่างเล็กๆ ระหว่างกัน
- เติมเต็มรูเล็กๆ ในวัตถุ: เช่น ช่องว่างที่เกิดจาก noise หรือจุดที่หายไป
- ปรับปรุงขอบเขตและรูปร่างของวัตถุ: ทำให้ขอบเขตของวัตถุดูชัดเจนและสมบูรณ์

2.9 การเปิดภาพ (Opening Operation)

การเปิดภาพ เป็นกระบวนการกร่อนภาพ (Erosion) แล้วทำการขยายภาพ (Dilation) ตามลำดับ

ประโยชน์ของการเปิดภาพ

- กำจัด noise แบบเกลือ (Salt Noise): เช่น จุดขาวที่ไม่ต้องการในภาพ

- แยกวัตถุที่เชื่อมต่อกัน: ทำให้วัตถุที่เชื่อมกันอยู่แยกออกจากกัน

2.10 การตรวจจับขอบภาพ (Edge Detection)

การตรวจจับขอบเป็นเทคนิคในกระบวนการประมวลผลภาพ ซึ่งมีบทบาทสำคัญในการระบุและกำหนดตำแหน่งของขอบเขตหรือขอบของวัตถุภายในภาพ เทคนิคนี้ทำหน้าที่ตรวจจับความไม่ต่อเนื่องของค่าความเข้มของพิกเซลในภาพ เพื่อแยกโครงสร้างของวัตถุที่ปรากฏออกมาอย่างชัดเจน ขอบของวัตถุ มักปรากฏในบริเวณที่ค่าความเข้มของภาพเปลี่ยนแปลงอย่างฉับพลัน [7]

2.10.1 เทคนิคการตรวจจับขอบในภาพ

เทคนิคที่ใช้สำหรับการตรวจจับขอบในภาพ ได้แก่

- Sobel Edge Detection
- Canny Edge Detection
- Laplacian Edge Detection
- Prewitt Edge Detection
- Roberts Cross Edge Detection
- Scharr Edge Detection

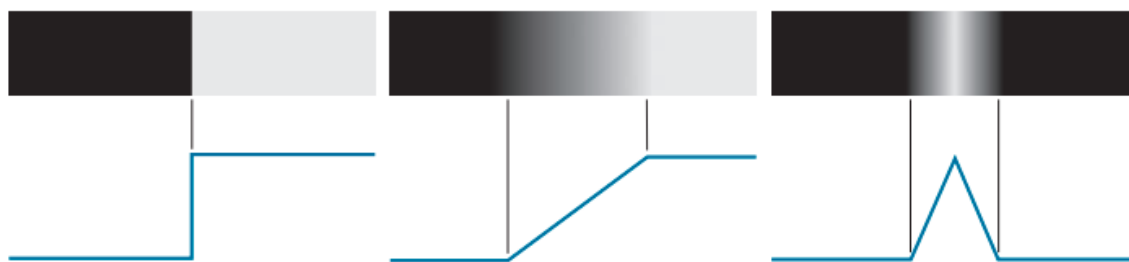
เป้าหมายหลักของเทคนิคเหล่านี้ คือการตรวจจับขอบที่สำคัญภายในภาพ แล้วทำการเชื่อมโยงขอบที่พบเพื่อสร้างเส้นหรือขอบเขตที่มีความหมาย ส่งผลให้ได้ภาพที่แบ่งส่วนเป็นพื้นที่ต่างๆ ซึ่งสามารถนำไปใช้ในงานประมวลผลภาพขั้นสูง เช่น การนับวัตถุ การวัดขนาด การสกัดคุณลักษณะ และการจำแนกประเภทวัตถุ

2.10.2 แนวคิดหลักในการตรวจจับขอบ

1. แบบจำลองขอบ (Edge Models)

แบบจำลองขอบคือแนวคิดเชิงทฤษฎีที่ใช้เพื่ออธิบายลักษณะของขอบที่พบในภาพ ซึ่งช่วยในการพัฒนาอัลกอริทึม โดยแบบจำลองพื้นฐานแบ่งออกเป็น 3 ประเภท

1. ขอบขั้นบันได (Step Edge) แสดงถึงการเปลี่ยนแปลงค่าความเข้มอย่างกะทันหัน
2. ขอบแบบลาดเอียง (Ramp Edge) แสดงถึงการเปลี่ยนแปลงค่าความเข้มอย่างค่อยเป็นค่อยไป
3. ขอบหลังคา (Ridge Edge) แสดงถึงจุดสูงสุดของความเข้ม แล้วค่อยๆ ลดลง



รูปที่ 4 แบบจำลองขอบ (Edge Models) [7]

2. ฟังก์ชันความเข้มของภาพ (Image Intensity Function)

ฟังก์ชันความเข้มแสดงถึงระดับความสว่างของแต่ละพิกเซลในภาพสีเทา สำหรับภาพสี เช่น RGB ความเข้มจะพิจารณาในแต่ละช่องสี (เช่น สีแดง, สีเขียว, สีน้ำเงิน ในภาพ RGB)

3. อนุพันธ์อันดับแรกและอันดับสอง

1) อนุพันธ์อันดับแรก (First Derivative)

ใช้ในการตรวจจับขอบโดยวัดอัตราการเปลี่ยนแปลงของความเข้ม เช่น การใช้ตัวดำเนินการ Sobel, Prewitt, หรือ Scharr ซึ่งสามารถระบุขอบได้จากจุดที่ความเข้มเปลี่ยนแปลงอย่างรวดเร็ว

2) อนุพันธ์อันดับสอง (Second Derivative)

ตรวจจับขอบโดยระบุจุดตัดศูนย์ (zero crossing) ของอนุพันธ์อันดับแรก ตัวอย่างเช่นการใช้ตัวดำเนินการ Laplacian เหมาะกับการหาขอบที่มีลักษณะบางและคมชัด

2.11 วิธีการตรวจจับขอบเขตวัตถุ (Contour-Based Method)

การตรวจจับขอบเขตวัตถุเป็นกระบวนการที่มุ่งหาขอบของวัตถุในภาพ โดยอาศัยการวิเคราะห์พิกเซลที่มีระดับสีเดียวกัน เพื่อแยกขอบของวัตถุออกจากพื้นหลัง วิธีนี้สามารถหาขอบทั้งหมดในภาพ และทำการลบพิกเซลที่อยู่บนขอบเหล่านั้น ยกเว้นพิกเซลที่จำเป็นต่อความต่อเนื่องของภาพ กระบวนการใช้เทมเพลตขนาด 3x3 ในการตรวจสอบพิกเซลที่ควรถูกลบออก

กระบวนการทำให้ขอบบางลง (Thinning) โดยใช้วิธีการตรวจจับขอบเขตวัตถุประกอบด้วย 4 ขั้นตอนหลัก

1. หาพิกเซลขอบเขตทั้งหมด
2. ระบุพิกเซลขอบเขตใดๆ ที่ไม่สามารถลบได้
3. ลบพิกเซลทั้งหมดในขอบเขตยกเว้นพิกเซลในขั้นตอนที่สอง

4. หากมีพิกเซลใดๆ ถูกลบในขั้นตอนที่สาม ขั้นตอนวิธีจะทำซ้ำในขั้นตอนแรก

ในส่วนของการตรวจจับขอบนอก กระบวนการนี้คล้ายกับการหารหัสลูกโซ่ (Chain Code) โดยเริ่มจากพิกเซลดำที่มีพิกเซลรอบข้างเป็นพื้นหลังที่คล้ายกัน แล้วนับพิกเซลโดยวนตามเข็มนาฬิกา จนกลับมายังพิกเซลเริ่มต้นอีกครั้ง พิกเซลที่นับได้จะถูกจัดเก็บไว้ในรายการสำหรับใช้งานในภายหลัง

หลังจากนั้น จะมีการระบุขอบภายในของวัตถุ โดยกระบวนการนี้จะทำซ้ำเพื่อค้นหาขอบด้านใน เช่น รูภายในวัตถุ จนกว่าจะไม่เหลือพิกเซลเริ่มต้นอีกต่อไป

วิธีการตรวจจับขอบเขตวัตถุนี้มีบทบาทสำคัญในการระบุและติดตามวัตถุในภาพ ช่วยให้สามารถวิเคราะห์รูปร่างและลักษณะของวัตถุได้อย่างมีประสิทธิภาพ

3. แนวทางการดำเนินงาน

การดำเนินงานระบบตรวจสอบสภาพจราจรติดขัดด้วยการประมวลผลภาพจากกล้องวงจรปิด โดยใช้การเขียนโปรแกรมภาษา python ด้วยโปรแกรม vs code โดยมีข้อมูลนำเข้าเป็นรูปภาพ 2 รูป ได้แก่

- ภาพพื้นหลัง (Background) เป็นภาพถนนที่ไม่มียานพาหนะสัญจร
- ภาพการจราจร (Image) เป็นภาพถนนที่ใช้ในการจำแนกสภาพการจราจร

โดยใช้ข้อมูลการจราจรจากกล้องวงจรปิด CCTV ที่สามารถเข้าถึงได้จาก

<https://www.youtube.com/watch?v=xIV8HucebVI>

3.1 การตรวจสอบสภาพการจราจรโดยใช้ภาพจราจรจำนวน 1 ภาพ (Single Image)

3.1.1 ใช้เทคนิค Masking เพื่อลบส่วนที่ไม่จำเป็น

เป็นการกำหนดขอบเขตของพื้นที่ที่ต้องการเก็บไว้ เพื่อกำจัดส่วนของภาพที่ไม่ต้องการใช้ เช่น ต้นไม้หรือบริเวณที่ไม่ใช่ถนน โดยภาพที่เหลือจะเป็นส่วนของถนนที่ต้องการวิเคราะห์

Define the region to keep (road area)

```
region = np.array([[200, 400], [900, 400], [1400, 150], [2000, 150], [2000, 1000], [450, 1000], [200, 850]], dtype=np.int32)
```

Create a black mask

```
mask = np.zeros(background.shape[:2], dtype=np.uint8)
```

Fill the road region with white

```
cv2.fillPoly(mask, [region], 255)
```

3.1.2 แปลงภาพเป็นระดับสีเทา (Grayscale)

เป็นการช่วยลดความซับซ้อนของข้อมูล ทำให้ภาพง่ายต่อการประมวลผลและลดข้อมูลที่ไม่จำเป็น โดยจะได้ภาพระดับสีเทาที่สามารถนำไปใช้ตรวจจับวัตถุได้ง่ายขึ้น

```
gray_background = cv2.cvtColor(masked_background, cv2.COLOR_BGR2GRAY)
```

```
gray_image = cv2.cvtColor(masked_image, cv2.COLOR_BGR2GRAY)
```

3.1.3 ใช้เทคนิค Background Subtraction เพื่อตรวจจับวัตถุ

เป็นการลบพื้นหลังออกจากภาพ เพื่อแยกส่วนที่เป็นรถออกจากพื้นหลังของถนน ได้ผลลัพธ์เป็นภาพไบนารีที่แสดงเฉพาะส่วนของรถที่แตกต่างจากพื้นหลัง

```
fgbg = cv2.createBackgroundSubtractorMOG2(detectShadows=False)
```

```
fgbg.apply(gray_background)
```

```
fgmask = fgbg.apply(gray_image)
```

3.1.4 ลดสัญญาณรบกวนด้วย Erosion และ Dilation

เป็นการลบจุดรบกวนและเชื่อมต่อพิกเซลของวัตถุที่ติดกัน โดยทำการ Erosion ก่อนแล้วทำการ Dilation เรียกว่า การเปิดภาพ (Opening Operation) ได้ผลลัพธ์เป็นภาพที่มีการแยกวัตถุชัดเจนขึ้น และลดปัญหาการตรวจจับผิดพลาด

```
kernel = np.ones((3, 3), np.uint8)
```

```
eroded = cv2.erode(fgmask, kernel, iterations=5)
```

```
dilated = cv2.dilate(eroded, kernel, iterations=40)
```

3.1.5 ใช้เทคนิค Contour เพื่อตรวจจับขอบของวัตถุ

เป็นหาขอบเขตของรถในภาพ โดยตรวจจับขอบของวัตถุ แล้วนำเส้นขอบวาดลงบนภาพ ได้ผลลัพธ์เป็นภาพที่มีขอบเขตของรถแสดงออกมาเพื่อใช้ในการวิเคราะห์

```
contours, _ = cv2.findContours(dilated, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
```

```
contour_image = image.copy()
```

```
cv2.drawContours(contour_image, contours, -1, (0, 255, 0), 2)
```

3.1.6 วิเคราะห์ขนาดของยานพาหนะบนท้องถนน

วิเคราะห์ขนาดของยานพาหนะทั้งหมด โดยคำนวณจากผลรวมของพื้นที่ของยานพาหนะแต่ละคัน จะได้ผลลัพธ์เป็นจำนวนยานพาหนะบนท้องถนน และพื้นที่ของยานพาหนะที่ใช้

```
vehicle_area = 0
```

```
vehicle_count = 0
```

for c in contours:

area = cv2.contourArea(c)

x, y, w, h = cv2.boundingRect(c)

aspect_ratio = w / h if h != 0 else 0

vehicle_area += area

vehicle_count += 1

3.1.7 ประเมินระดับความหนาแน่นของจราจรบนท้องถนน

คำนวณระดับความหนาแน่นของจราจรบนท้องถนนได้จากสูตร

$$\text{ความหนาแน่นของการจราจร} = \frac{\text{พื้นที่ของยานพาหนะที่ใช้}}{\text{พื้นที่ที่ได้กำหนดขอบเขตทั้งหมด}}$$

โดยกำหนดเงื่อนไขในการประเมินระดับความหนาแน่นได้ดังนี้

- 1) **Flow** เมื่อความหนาแน่นของการจราจรมีค่าต่ำกว่า 0.4
- 2) **Heavy** เมื่อความหนาแน่นของการจราจรมีค่าอยู่ระหว่าง 0.4 - 0.7
- 3) **Jammed** เมื่อความหนาแน่นของการจราจรมีค่าสูงกว่า 0.7

total_area = np.count_nonzero(mask)

congestion_ratio = vehicle_area / total_area

if congestion_ratio > 0.7:

status = "Jammed"

elif congestion_ratio >= 0.4:

status = "Heavy"

else:

status = "Flow"

3.1.8 ส่งออกข้อมูลการจราจร

ส่งออกข้อมูลการจราจรเป็นภาพที่มีขอบเขตของรถโดยใช้เทคนิค Contour และมีการระบุ (Label) ค่าความหนาแน่นของการจราจรลงไปในการ์ดส่งออก (Output Image)

3.2 การตรวจสอบสภาพการจราจรโดยใช้ภาพจราจรจำนวนหลายภาพ (Multiple Images)

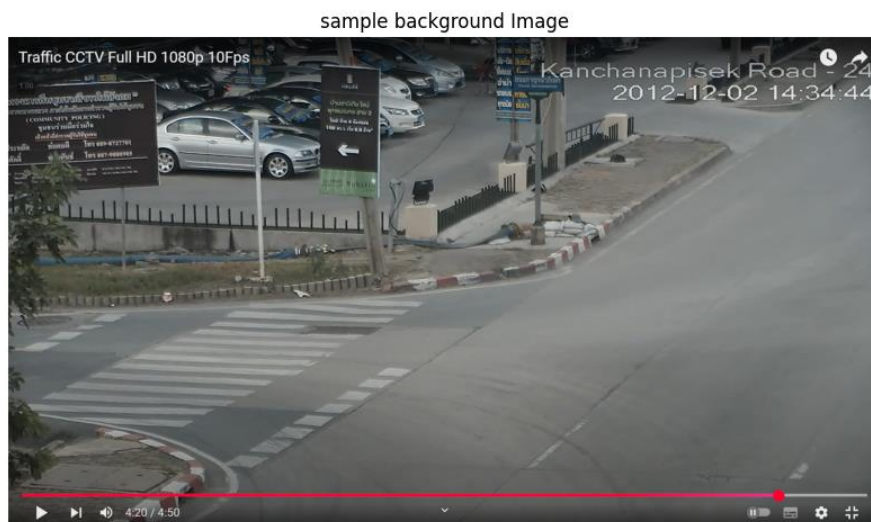
เป็นการประยุกต์การดำเนินงาน โดยใช้โปรแกรมการตรวจสอบสภาพการจราจรโดยใช้ภาพจราจร (Image) จำนวน 1 ภาพ ตั้งแต่กระบวนการที่ 3.1.1 จนถึง 3.1.8 มาแก้ไขโปรแกรมเพื่อให้สามารถรับภาพการจราจร (Image) จำนวนหลายภาพได้

4. ผลลัพธ์จากการดำเนินการเบื้องต้น

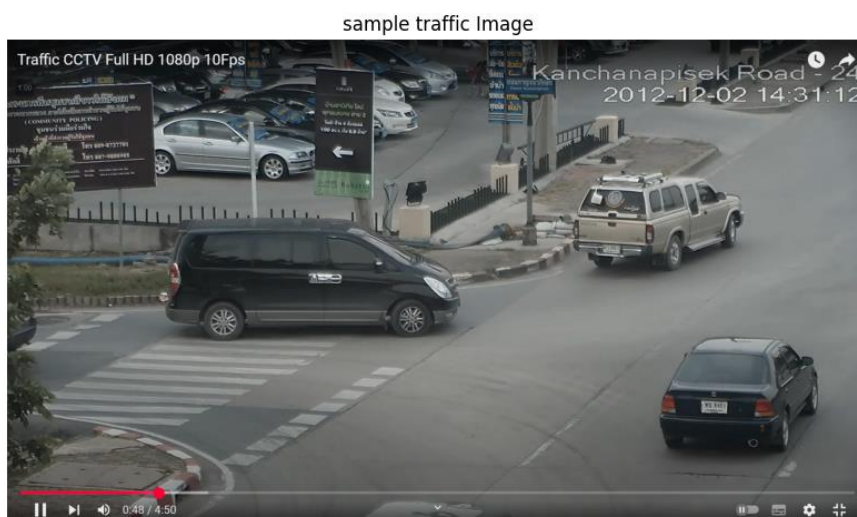
การดำเนินงานระบบตรวจสอบสภาพจราจรติดขัดด้วยการประมวลผลภาพจากกล้องวงจรปิด โดยใช้การเขียนโปรแกรมภาษา python ด้วยโปรแกรม vs code มีผลลัพธ์จากการดำเนินการเบื้องต้น ดังนี้

4.1 การตรวจสอบสภาพการจราจรโดยใช้ภาพจราจรจำนวน 1 ภาพ (Single Image)

ใช้ข้อมูลนำเข้าเป็นภาพจากกล้อง CCTV ที่ถ่ายบริเวณท้องถนน ได้แก่ ภาพพื้นหลัง (Background) ภาพการจราจร (Image) สามารถแสดงได้ดังรูปที่ 5 และ 6 ดังนี้



รูปที่ 5 ภาพพื้นหลัง (Background) ที่เป็นข้อมูลนำเข้า



รูปที่ 6 ภาพการจราจร (Image) ที่เป็นข้อมูลนำเข้า

4.1.1 ใช้เทคนิค Masking เพื่อลบส่วนที่ไม่จำเป็น

ผลลัพธ์ของการใช้เทคนิค Masking เพื่อลบส่วนที่ไม่จำเป็น สามารถแสดงได้ดังรูปที่ 7 และ 8 ดังนี้

Masked Background



รูปที่ 7 ภาพพื้นหลัง (Background) ที่ผ่านเทคนิค Masking

Masked Image



รูปที่ 8 ภาพการจราจร (Image) ที่ผ่านเทคนิค Masking

4.1.2 แปลงภาพเป็นระดับสีเทา (Grayscale)

ผลลัพธ์ของการแปลงภาพเป็นระดับสีเทา สามารถแสดงได้ดังรูปที่ 9 และ 10 ดังนี้

Grayscale Background



รูปที่ 9 ภาพพื้นหลัง (Background) ที่ผ่านการแปลงภาพเป็นระดับสีเทา

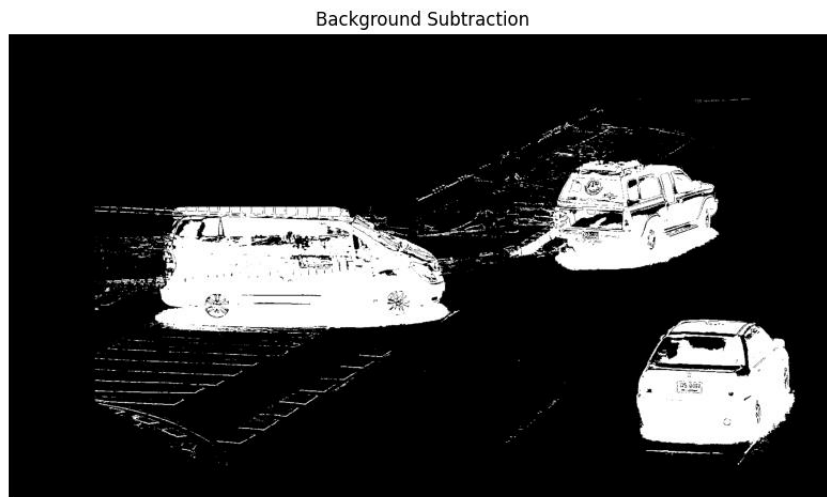
Grayscale Image



รูปที่ 10 ภาพการจราจร (Image) ที่ผ่านการแปลงภาพเป็นระดับสีเทา

4.1.3 ใช้เทคนิค Background Subtraction เพื่อตรวจจับวัตถุ

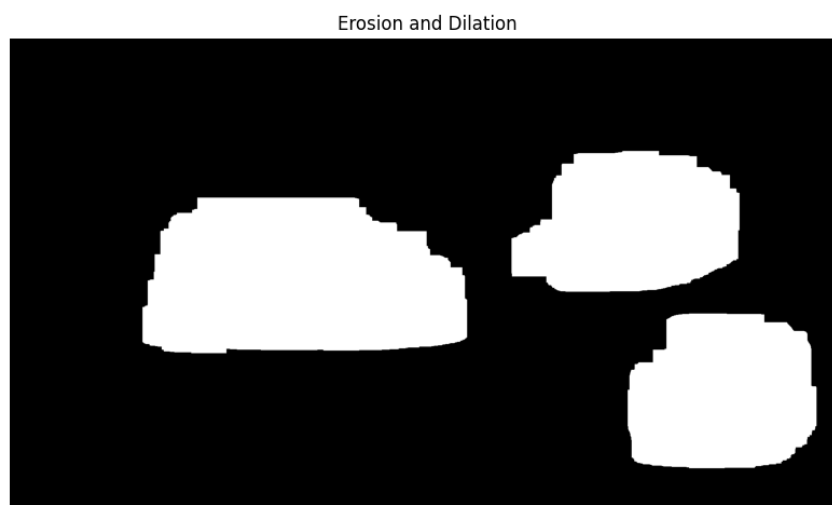
ผลลัพธ์ของการใช้เทคนิค Background Subtraction เพื่อตรวจจับวัตถุ สามารถแสดงได้ดังรูปที่ 11
ดังนี้



รูปที่ 11 ภาพการจราจร (Image) ที่ผ่านการใช้เทคนิค Background Subtraction

4.1.4 ลดสัญญาณรบกวนด้วย Erosion และ Dilation

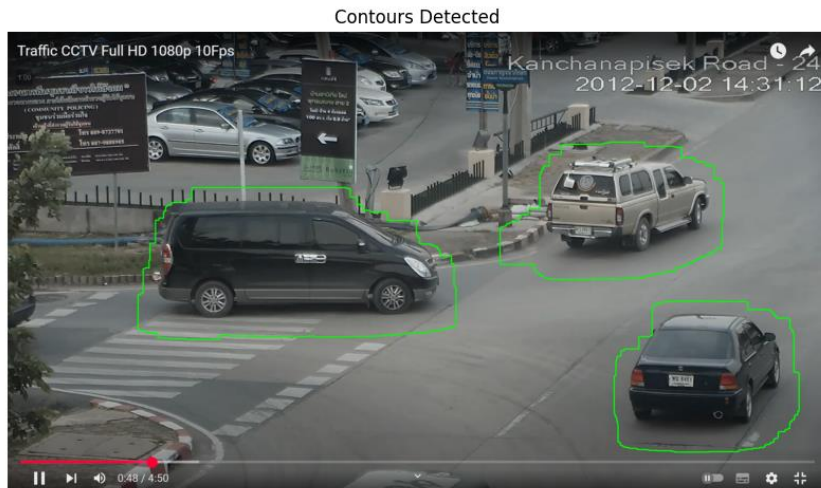
ผลลัพธ์ของการลดสัญญาณรบกวนด้วย Erosion และ Dilation สามารถแสดงได้ดังรูปที่ 12 ดังนี้



รูปที่ 12 ภาพการจราจร (Image) ที่ผ่านการลดสัญญาณรบกวนด้วย Erosion และ Dilation

4.1.5 ใช้เทคนิค Contour เพื่อตรวจจับขอบของวัตถุ

ผลลัพธ์ของการใช้เทคนิค Contour เพื่อตรวจจับขอบของวัตถุ สามารถแสดงได้ดังรูปที่ 12 ดังนี้



รูปที่ 12 ภาพการจราจร (Image) ที่ผ่านการใช้เทคนิค Contour เพื่อตรวจจับขอบของวัตถุ

4.1.6 วิเคราะห์ขนาดของยานพาหนะบนท้องถนน

ผลลัพธ์ของการวิเคราะห์ขนาดของยานพาหนะบนท้องถนน สามารถแสดงได้ดังนี้

Estimated Vehicle Count: 3

Vehicle Area: 488774.00 pixels

4.1.7 ประเมินระดับความหนาแน่นของจราจรบนท้องถนน

ผลลัพธ์ของการประเมินระดับความหนาแน่นของจราจรบนท้องถนน สามารถแสดงได้ดังนี้

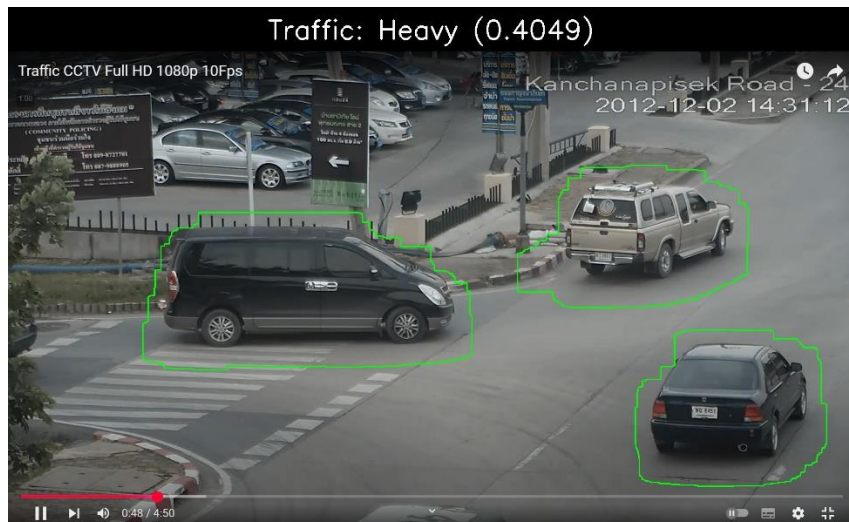
Total Area: 1207145

Congestion Ratio: 0.4049

Traffic Status: Heavy

3.1.8 ส่งออกข้อมูลการจราจร

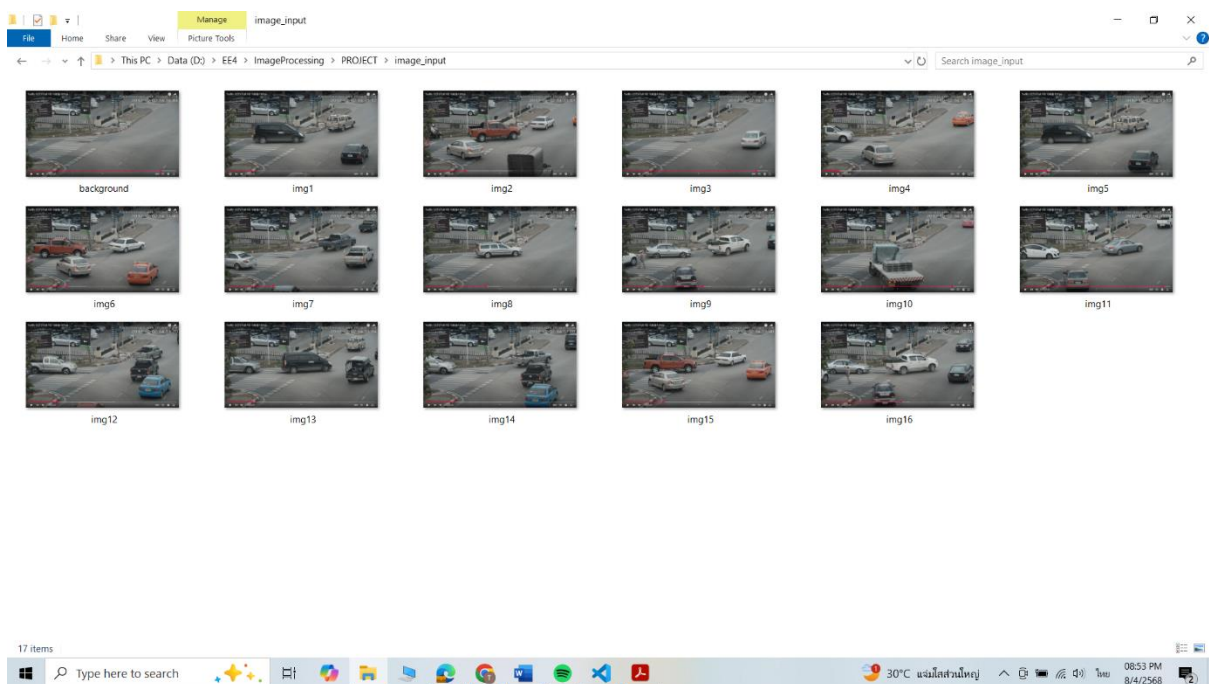
ผลลัพธ์ของการส่งออกข้อมูลการจราจร สามารถแสดงได้ดังรูปที่ 13



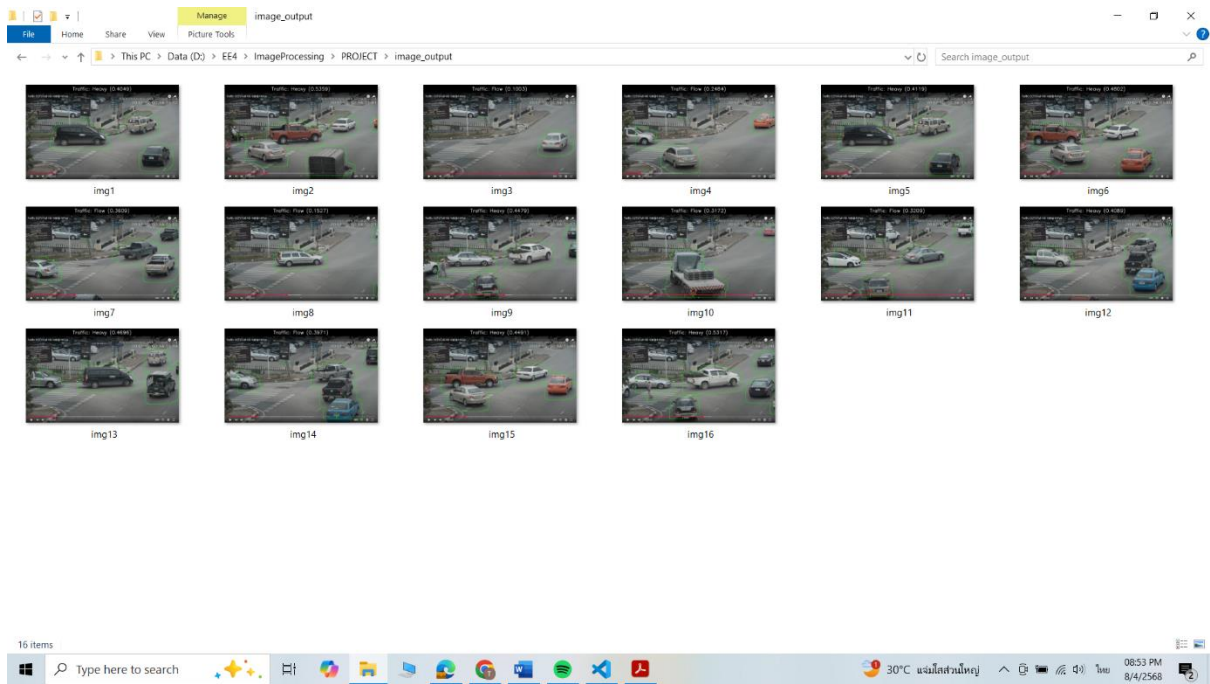
รูปที่ 13 ผลลัพธ์ของการส่งออกข้อมูลการจราจร

4.2 การตรวจสอบสภาพการจราจรโดยใช้ภาพจราจรจำนวนหลายภาพ (Multiple Images)

ผลลัพธ์ของการตรวจสอบสภาพการจราจรโดยใช้ภาพจราจรจำนวนหลายภาพ สามารถแสดงผลของข้อมูลนำเข้าได้ดังรูปที่ 14 และผลลัพธ์ของข้อมูลส่งออกได้ดังรูปที่ 15 ดังนี้



รูปที่ 14 ผลลัพธ์ของข้อมูลนำเข้า (Input Image) จำนวนหลายภาพ



รูปที่ 15 ผลลัพธ์ของข้อมูลส่งออก (Output Image) จำนวนหลายภาพ

5. บทสรุป

5.1 บทสรุปการทำโครงการ

จากการดำเนินงาน พบว่าสามารถออกแบบระบบตรวจสอบสภาพจราจรติดขัดด้วยการประมวลผลภาพจากกล้องวงจรปิด โดยใช้การเขียนโปรแกรมภาษา python ด้วยโปรแกรม vs code โดยใช้ข้อมูลนำเข้าเป็นภาพจากกล้อง CCTV ที่ถ่ายบริเวณท้องถนน ได้แก่

- ภาพพื้นหลัง (Background) เป็นภาพถนนที่ไม่มียานพาหนะสัญจร
- ภาพการจราจร (Image) เป็นภาพถนนที่ใช้ในการจำแนกสภาพการจราจร

ซึ่งสามารถการจำแนกสภาพการจราจร แบ่งเป็น 3 กลุ่ม ได้แก่

- การจราจรคล่องตัว (Flow) เมื่อความหนาแน่นของการจราจรมีค่าต่ำกว่า 0.4
- การจราจรหนาแน่น (Heavy) เมื่อความหนาแน่นของการจราจรมีค่าอยู่ระหว่าง 0.4 - 0.7
- การจราจรติดขัด (Jammed) เมื่อความหนาแน่นของการจราจรมีค่าสูงกว่า 0.7

5.2 แผนการดำเนินงาน

ตารางที่ 1 แผนการดำเนินงาน

ขั้นตอนการดำเนินงาน	เดือน	
	มี.ค.	เม.ย.
1. ศึกษาข้อมูลจากเอกสารที่เกี่ยวข้อง		
2. เขียนโปรแกรมตรวจสอบสภาพการจราจร		
3. จัดทำรายงานฉบับสมบูรณ์		

หมายเหตุ สีเทา คือ ความก้าวหน้าที่วางแผนไว้
สีดำ คือ ความก้าวหน้าปัจจุบัน

5.3 ปัญหา อุปสรรค และแนวทางแก้ไข

1. ปัญหาที่เกิดจากการตรวจจับขอบของวัตถุ โดยใช้เทคนิค Contour ที่ไม่สามารถครอบคลุมยานพาหนะทั้งคัน แก้ไขได้โดยการปรับค่าพารามิเตอร์ในขั้นตอนของ Background Subtraction และขั้นตอนการลดสัญญาณรบกวนด้วย Erosion และ Dilation
2. ปัญหาที่เกิดจากการคำนวณระดับความหนาแน่นของจราจรบนท้องถนน ที่มีความคลาดเคลื่อนกับความหนาแน่นที่เกิดขึ้นจริง แก้ไขได้โดยใช้เทคนิค Masking เพื่อกำจัดส่วนพื้นที่ที่ไม่ต้องการใช้ออกไป

6. เอกสารอ้างอิง

- [1] B. Eamthanakul, M. Ketcham and N. Chumuang, "The Traffic Congestion Investigating System by Image Processing from CCTV Camera," 2017 IEEE International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), 2017, pp. 978-1-5090-5210-3.
- [2] Muhammad Sabih, " Background subtraction in computer vision ", [Online]. Available <https://medium.com/@muhammadsabih56/background-subtraction-in-computer-vision-402ddc79cb1b>. [Accessed 14 April 2025]
- [3] R. Fisher, S. Perkins, A. Walker and E. Wolfart., "Dilation", [Online]. Available: <https://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm>. [Accessed 14 April 2025]
- [4] Anshul Sachdev, "Dilation (Morphological Operation)— Image Processing", [Online]. Available: <https://medium.com/@anshul16/dilation-morphological-operation-image-processing-82d16a619f59>. [Accessed 14 April 2025]
- [5] R. Fisher, S. Perkins, A. Walker and E. Wolfart., "Erosion", [Online]. Available: <https://homepages.inf.ed.ac.uk/rbf/HIPR2/erode.htm>. [Accessed 14 April 2025]
- [6] Anshul Sachdev, "Erosion (Morphological Operation) — Image Processing", [Online]. Available: <https://medium.com/@anshul16/erosion-morphological-operation-image-processing-18537f7c66cd>. [Accessed 14 April 2025]
- [7] Timothy Malche, " Edge Detection in Image Processing: An Introduction", [Online]. Available: <https://blog.roboflow.com/edge-detection/>. [Accessed 14 April 2025]