

ระบบตรวจสอบสภาพจราจรติดขัดด้วย การประมวลผลภาพจากกล้องวงจรปิด

The Traffic Congestion Investigating System by
Image Processing from CCTV Camera

นายธีรวัฒน์ เลิศอัมพรวิทย์ เลขประจำตัว 6430183721

บทที่ 1 บทนำ

- ที่มาและความสำคัญของโครงการ
- วัตถุประสงค์ของโครงการ
- ขอบเขตของโครงการ
- ผลลัพธ์ที่คาดหวังจากโครงการ

ที่มาและความสำคัญของโครงการ

- ปัญหาการจราจรติดขัดมีแนวโน้มรุนแรงขึ้นจากการเพิ่มขึ้นของประชากร
- วิธีตรวจสอบสภาพการจราจรในปัจจุบันยังมีข้อจำกัดหลายด้าน
- วิธีตรวจสอบที่ใช้กัน เช่น การใช้อุปกรณ์ตรวจจับปริมาณและความเร็วของการใช้
กล้อง CCTV และให้เจ้าหน้าที่วิเคราะห์ภาพ
- แนวคิดของโครงการนี้เป็นการสร้างระบบแจ้งเตือนและวิเคราะห์สภาพจราจรโดยใช้
คอมพิวเตอร์ระบบจะประมวลผลภาพจากกล้อง CCTV และส่งผลลัพธ์ให้เจ้าหน้าที่

วัตถุประสงค์ของโครงการ

1. เพื่อพัฒนาระบบตรวจสอบสภาพการจราจรที่สามารถวิเคราะห์และจำแนกสภาพการจราจรออกเป็นสามระดับ ได้แก่ การจราจรคล่องตัว (Flow) การจราจรหนาแน่น (Heavy) และการจราจรติดขัด (Jammed)
2. เพื่อสร้างระบบที่สามารถนำไปใช้ในการควบคุมการจราจรที่ทางแยกได้อย่างมีประสิทธิภาพ โดยอาศัยเทคนิคการประมวลผล ในการวิเคราะห์สภาพการจราจรบนท้องถนน

ขอบเขตของโครงการ

1. ใช้เทคนิคการประมวลผล ได้แก่ Image Segmentation, Morphological Image Processing ในการตรวจจับยานพาหนะที่แล่นผ่านบนท้องถนน
2. ใช้ข้อมูลนำเข้าเป็นภาพจากกล้อง CCTV ที่ถ่ายบริเวณท้องถนน ได้แก่ ภาพพื้นหลัง (Background) และ ภาพการจราจร (Image)
3. การจำแนกสภาพการจราจร แบ่งเป็น 3 กลุ่ม ได้แก่ การจราจรคล่องตัว (Flow) การจราจรหนาแน่น (Heavy) และ การจราจรติดขัด (Jammed)



ผลลัพธ์ที่คาดหวังจากโครงการ

สามารถวิเคราะห์และจำแนกสภาพการจราจรออกเป็นสามระดับ ได้แก่ การจราจร
คล่องตัว (Flow) การจราจรหนาแน่น (Heavy) และการจราจรติดขัด (Jammed)

บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง

- Basic Image Processing
- Neighborhood Processing
- Morphological Image Processing
- Image Segmentation

Basic Image Processing

2.1 ภาพระดับเทา (Grayscale Image)

2.2 การคำนวณความแตกต่างของพิกเซล (Pixel Difference)

2.3 เทคนิคการลบพื้นหลัง (Background Subtraction Technique)



Neighborhood Processing

2.4 การกรองค่ามัธยฐาน (Median Filtering)

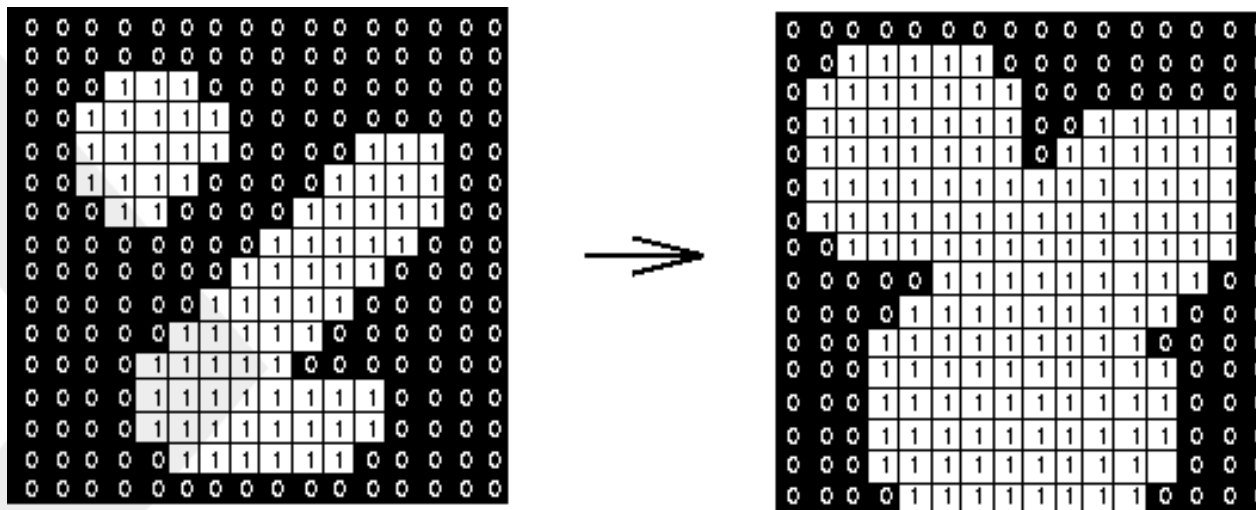
- ช่วยลดสัญญาณรบกวนหรือจุดเล็กๆ ในภาพ
- กำจัดสัญญาณรบกวนแบบจุด (impulse noise)

2.5 ROI (Region of Interest)

- พื้นที่หรือส่วนหนึ่งของภาพที่เราสนใจ

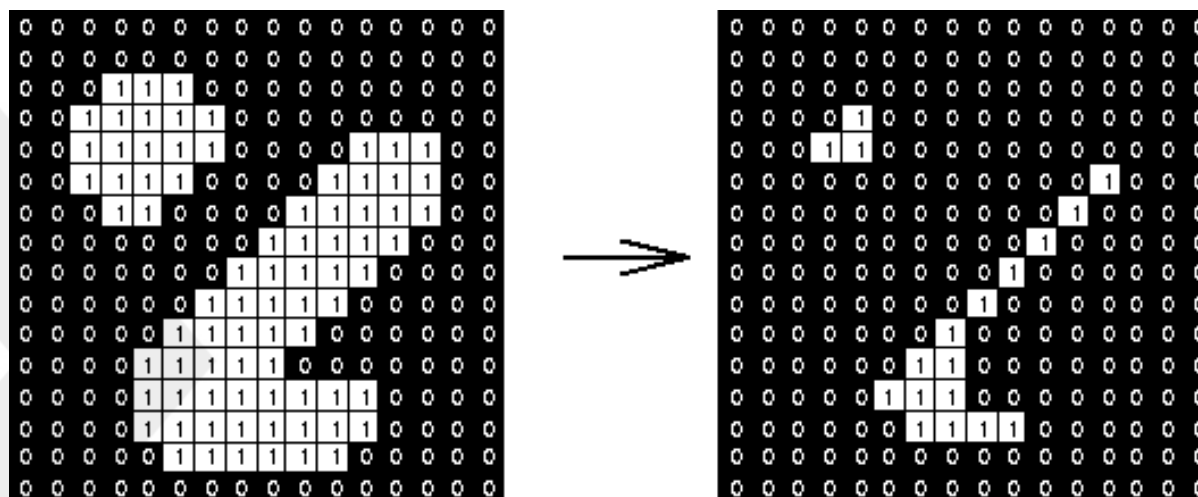
Morphological Image Processing

2.6 การขยายภาพ (Dilation Operation)



Morphological Image Processing

2.7 การกร่อนภาพ (Erosion Operation)





Morphological Image Processing

2.8 การปิดภาพ (Closing Operation)

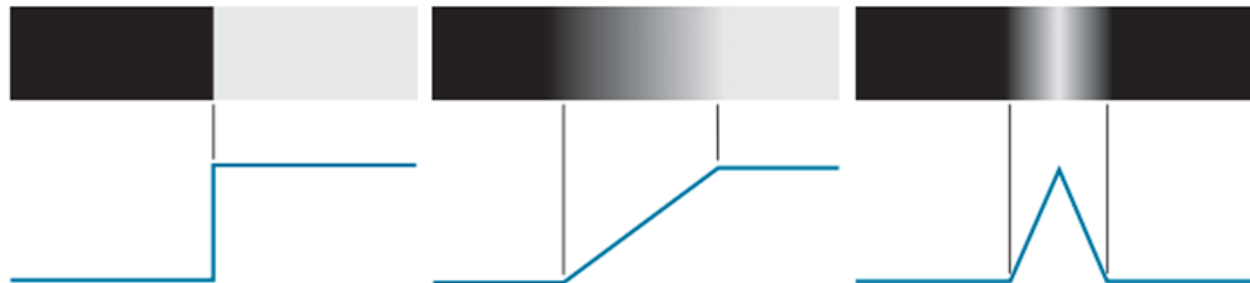
เป็นกระบวนการขยายภาพ (Dilation) แล้วทำการกร่อนภาพ (Erosion)

2.9 การเปิดภาพ (Opening Operation)

เป็นกระบวนการกร่อนภาพ (Erosion) แล้วทำการขยายภาพ (Dilation)

Image Segmentation

2.10 การตรวจจับขอบภาพ (Edge Detection)



2.11 วิธีการตรวจจับขอบเขตวัตถุ (Contour-Based Method)

วิเคราะห์พิกเซลที่มีระดับสีเดียวกัน เพื่อแยกขอบของวัตถุออกจากพื้นหลัง

บทที่ 3 แนวทางการดำเนินงาน

- การตรวจสอบสภาพการจราจรโดยใช้ภาพ
จราจรจำนวน 1 ภาพ (Single Image)
- การตรวจสอบสภาพการจราจรโดยใช้ภาพ
จราจรจำนวนหลายภาพ (Multiple Images)

3.1.1 ใช้เทคนิค Masking เพื่อลบส่วนที่ไม่จำเป็น

```
# 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)

```
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

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

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

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

3.1.4 ลดสัญญาณรบกวนด้วย Erosion และ Dilation

```
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

```
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

```
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

```
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) ค่าความหนาแน่นของการจราจรลงไปในภาพส่งออก (Output Image)



3.2 การตรวจสอบสภาพการจราจรโดยใช้ภาพจราจรจำนวน หลายภาพ (Multiple Images)

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

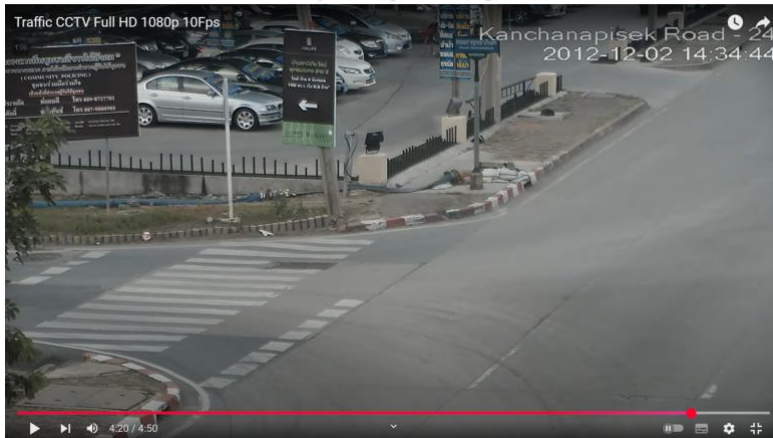
บทที่ 4

ผลลัพธ์จากการดำเนินการเบื้องต้น

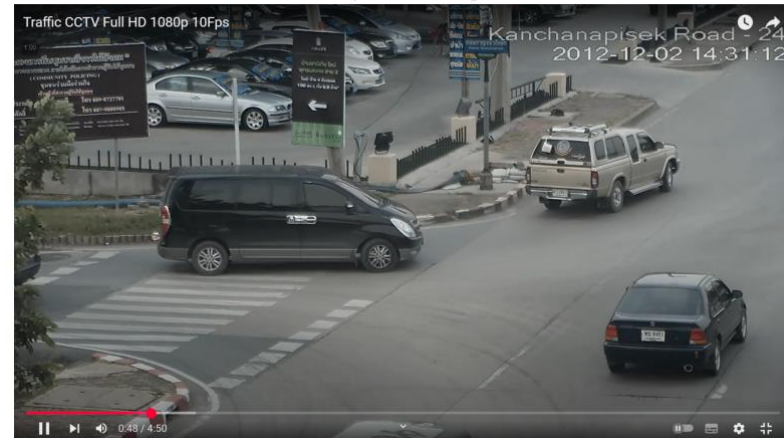
- การตรวจสอบสภาพการจราจรโดยใช้ภาพ
จราจรจำนวน 1 ภาพ (Single Image)
- การตรวจสอบสภาพการจราจรโดยใช้ภาพ
จราจรจำนวนหลายภาพ (Multiple Images)

4.1 การตรวจสอบสภาพการจราจรโดยใช้ภาพจราจร จำนวน 1 ภาพ (Single Image)

sample background Image

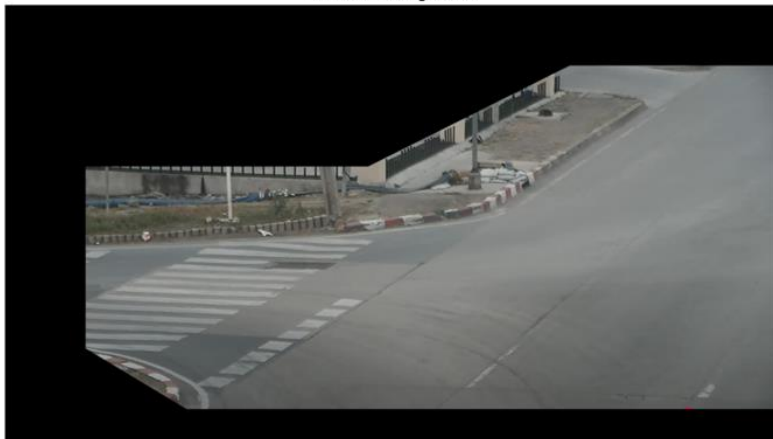


sample traffic Image

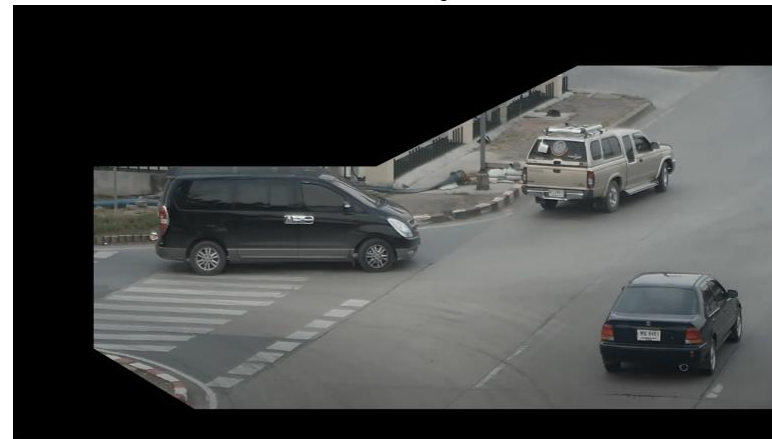


4.1.1 ใช้เทคนิค Masking เพื่อลบส่วนที่ไม่จำเป็น

Masked Background



Masked Image



4.1.2 แปลงภาพเป็นระดับสีเทา (Grayscale)

Grayscale Background



Grayscale Image



4.1.3 ใช้เทคนิค Background Subtraction เพื่อตรวจจับวัตถุ

Background Subtraction

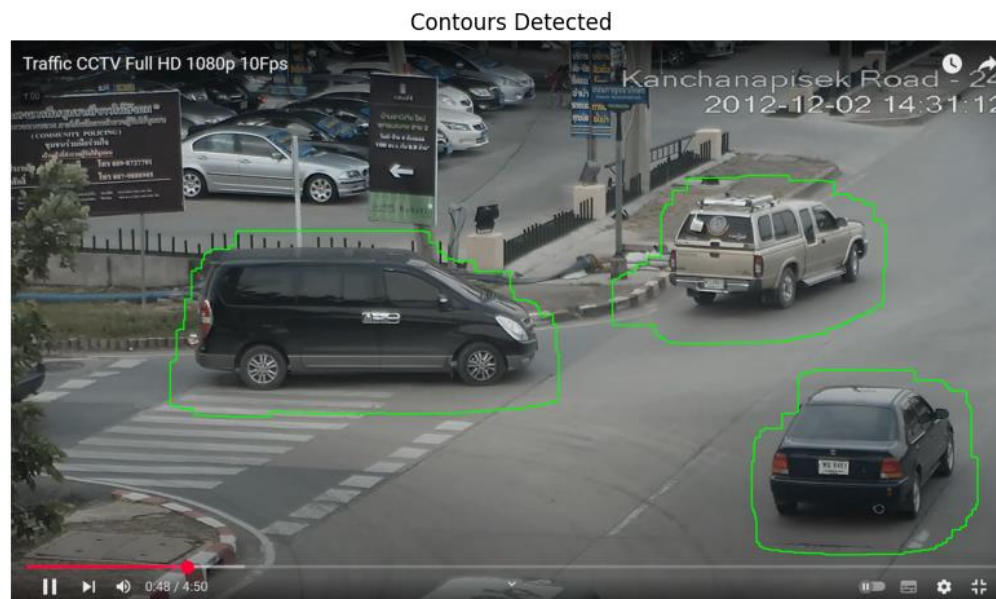


4.1.4 ลดสัญญาณรบกวนด้วย Erosion และ Dilation

Erosion and Dilation



4.1.5 ใช้เทคนิค Contour เพื่อตรวจจับขอบของวัตถุ



4.1.6 วิเคราะห์ขนาดของยานพาหนะบนท้องถนน

Estimated Vehicle Count: 3

Vehicle Area: 488774.00 pixels



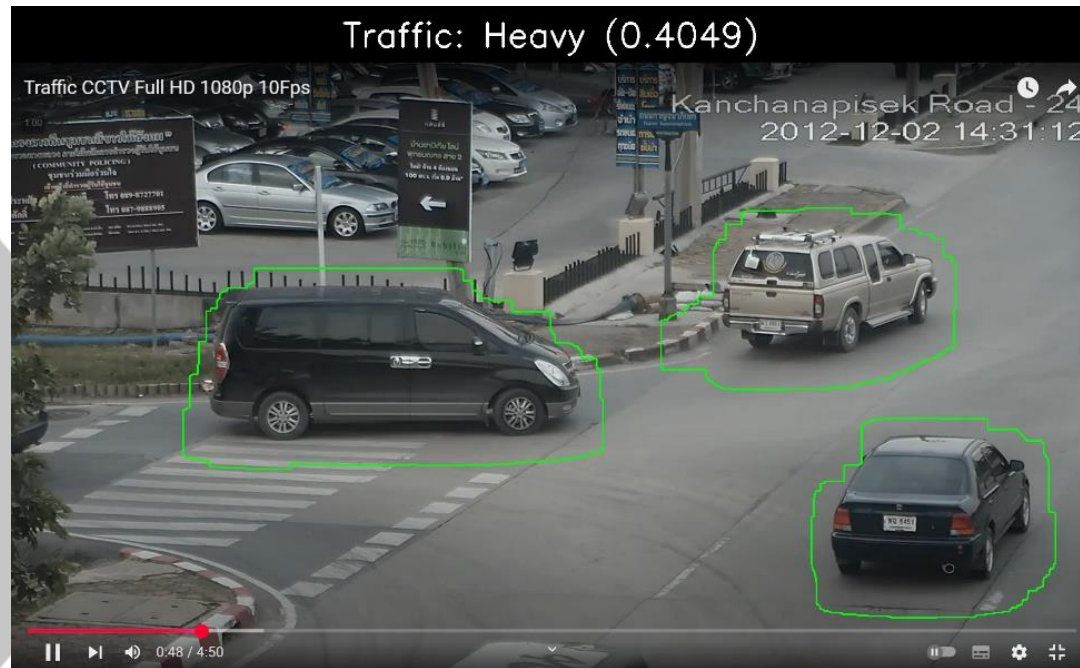
4.1.7 ประเมินระดับความหนาแน่นของจราจร บนท้องถนน

Total Area: 1207145

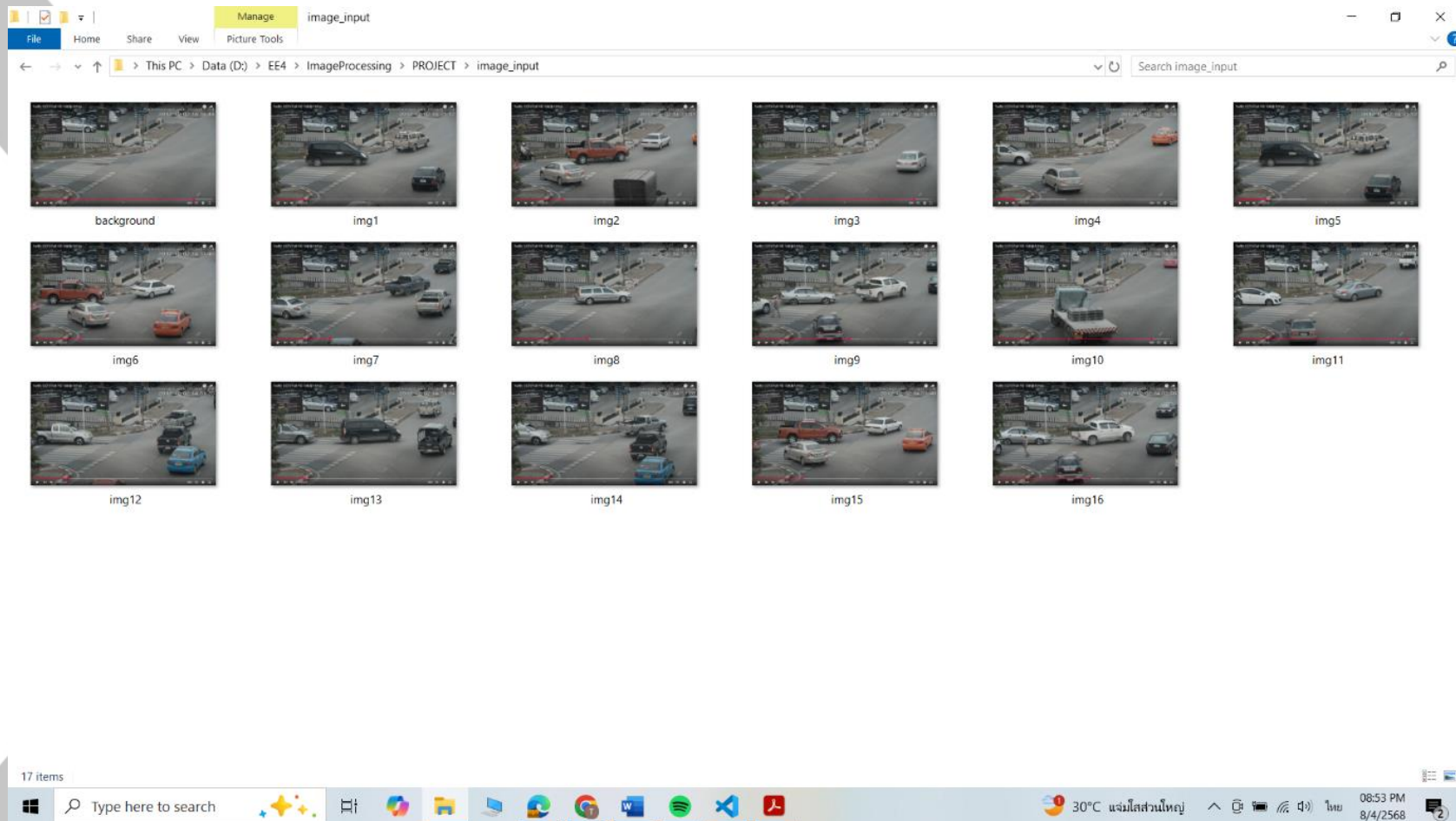
Congestion Ratio: 0.4049

Traffic Status: Heavy

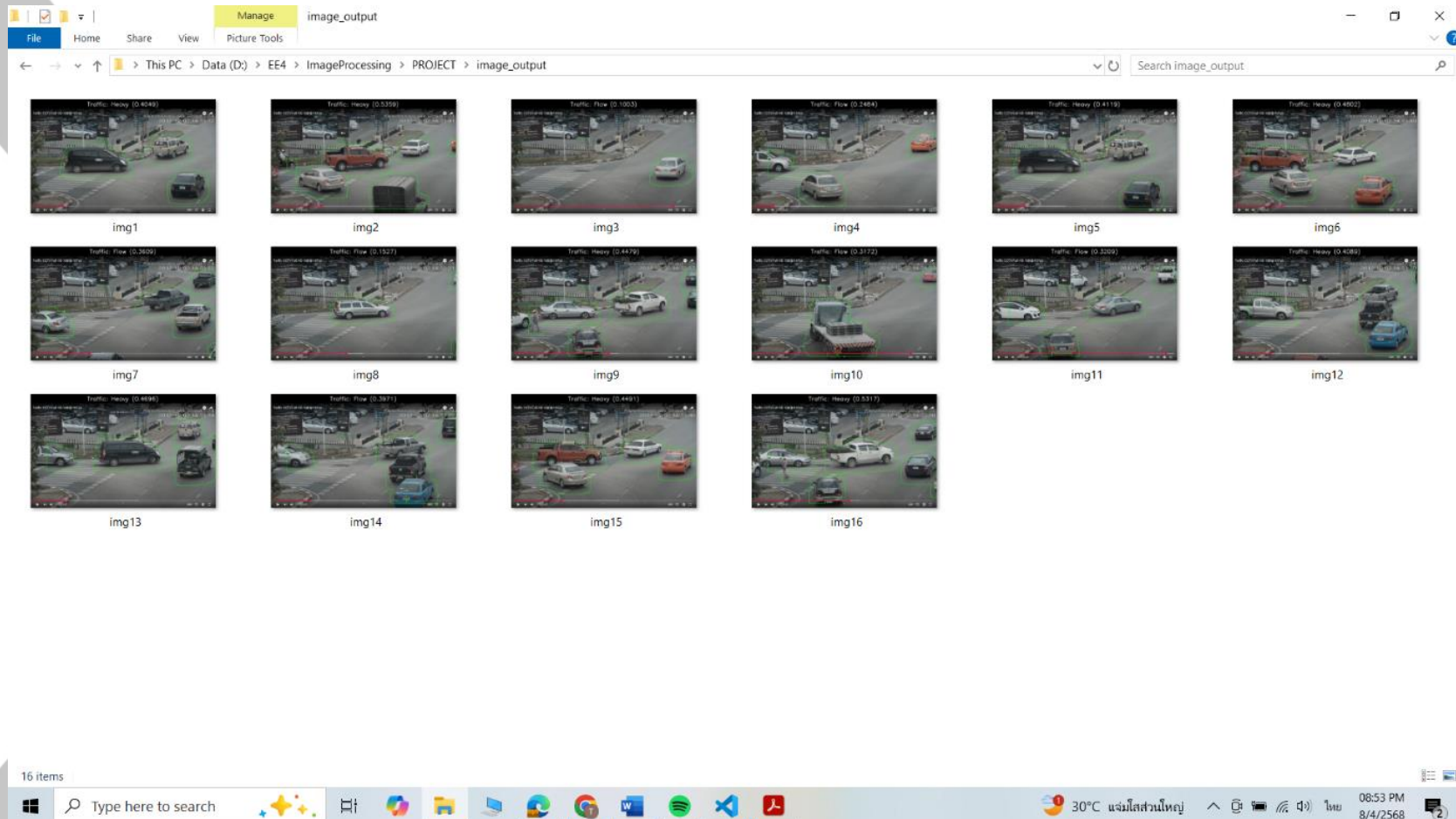
4.1.8 ส่งออกข้อมูลการจราจร



4.2 การตรวจสอบสภาพการจราจรโดยใช้ภาพจราจรจำนวน หลายภาพ (Multiple Images)



4.2 การตรวจสอบสภาพการจราจรโดยใช้ภาพจราจรจำนวน หลายภาพ (Multiple Images)



บทที่ 5 บทสรุป

- บทสรุปการทำโครงการ
- แผนการดำเนินงาน
- ปัญหา อุปสรรค และแนวทางแก้ไข

บทสรุปการทำโครงการ

- สามารถออกแบบระบบตรวจสอบสภาพจราจรติดขัดด้วยการประมวลผลภาพจากกล้องวงจรปิด โดยการใช้การเขียนโปรแกรมภาษา python ด้วยโปรแกรม vs code
- ใช้ข้อมูลนำเข้าเป็นภาพจากกล้อง CCTV ที่ถ่ายบริเวณท้องถนน ได้แก่ ภาพพื้นหลัง (Background) และภาพการจราจร (Image)
- ซึ่งสามารถการจำแนกสภาพการจราจร แบ่งเป็น 3 กลุ่ม ได้แก่ การจราจรคล่องตัว (Flow) การจราจรหนาแน่น (Heavy) และการจราจรติดขัด (Jammed)



แผนการทำงาน

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

ปัญหา อุปสรรค และแนวทางแก้ไข

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



THANK YOU

2102514 Advanced Image Processing