



ชื่อ Water image processing

ผู้ร่วมงาน

- 1) ภัทรชนน เมธาวุฒิยาภรณ์ รหัสนักศึกษา 65010814
- 2) สรยุทธ เปี่ยมรอด รหัสนักศึกษา 65011076

อาจารย์ที่ปรึกษา

- 1) ผศ. ไพศาล สิทธิโยภาสกุล
- 2) บุญยชนะ ภูระหงษ์

ภาควิชาวิศวกรรมคอมพิวเตอร์ หลักสูตรระบบไอโอทีและสารสนเทศ

รายงานนี้เป็นส่วนหนึ่งของวิชา 01236256 MICROCONTROLLER AND EMBEDDED SYSTEMS

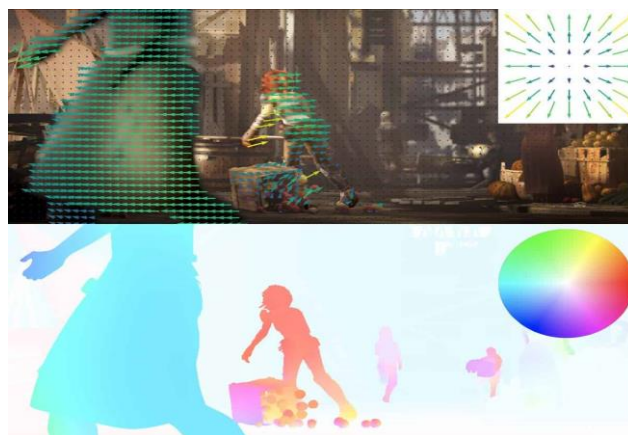
ปีการศึกษา 2566

วัตถุประสงค์

เนื่องจากปัญหาน้ำป่าไหลหลากที่เกิดขึ้นในประเทศไทย เกิดขึ้นในทุกๆปีในช่วงฤดูฝนส่งผลกระทบต่อประชากรไทยในหลายพื้นที่ โดยเฉพาะพื้นที่ราบสูงติดป่าหรือน้ำตกซึ่งน้ำป่าไหลหลาก ส่งผลกระทบต่อชีวิตความเป็นอยู่ส่งผลกระทบต่อ พืชผลทางการเกษตรและชีวิตของประชากรในบริเวณโดยรอบ ทางผู้จัดทำจึงได้คำนึงถึงปัญหาและอยากจะช่วยเพิ่มการป้องกัน เพื่อให้ประชากรโดยรอบได้รับรู้ถึงน้ำป่าไหลหลากจากต้นทางหรือเพื่อให้ทราบว่าน้ำที่ไหลทางต้นทางมีโอกาสทำให้เกิดน้ำป่าไหลหลาก เพื่อทำการรับมือป้องกันหรือทำการนำประชากรที่ดูแลตัวเองไม่ได้หรือ ลำบากให้อพยพออกจากพื้นที่เหล่านั้นไปก่อน โดยเครื่องมือที่ทางผู้จัดทำได้คิดขึ้นมาคือ กล้องตรวจจับการเคลื่อนไหวของน้ำ เพื่อนำลักษณะคลื่น และทิศทางการไหลของน้ำเพื่อหาความเร็วและทิศทาง แจ้งเตือนไปที่ศูนย์ในชุมชน เพื่อให้สามารถวิเคราะห์ว่า มีโอกาสเกิดน้ำป่าไหลหลากหรือไม่เพื่อทำการรับมือได้ทันทางที่

หลักการทำงาน

ในการทำงานสิ่งที่ผู้ทำการทดลองได้ใช้คือ หลักการ image processing โดยใช้หลักการเรื่อง Dense Optical Flow in OpenCV โดย Lucas-Kanade คำนวณการไหลของแสงสำหรับชุดคุณลักษณะแบบกระจาย (ในตัวอย่างของเรา มุมที่ตรวจพบโดยใช้อัลกอริทึม Shi-Tomasi) OpenCV มีอัลกอริทึมอื่นในการค้นหาการไหลของแสงที่หนาแน่น โดยจะคำนวณการไหลของแสงสำหรับจุดทั้งหมดในเฟรม ขึ้นอยู่กับอัลกอริทึมของกุนนาร์ ฟาร์เนแบ็ก ซึ่งอธิบายไว้ใน "การประมาณค่าการเคลื่อนที่แบบสองเฟรมตามการขยายตัวพหุนาม" โดยกุนนาร์ ฟาร์เนแบ็ก ในปี พ.ศ. 2546 ตัวอย่างด้านล่างแสดงวิธีค้นหาการไหลเชิงแสงหนาแน่นโดยใช้อัลกอริทึมด้านบน เราได้รับอาเรย์ 2 แชนเนลพร้อมเวกเตอร์โฟลว์ออปติคัล (u,v) เราพบขนาดและทิศทางของมัน เราใส่รหัสสีให้กับผลลัพธ์เพื่อให้เห็นภาพได้ชัดขึ้น ทิศทางสอดคล้องกับค่าฮิวของรูปภาพ ขนาดสอดคล้องกับระยะนำค่า



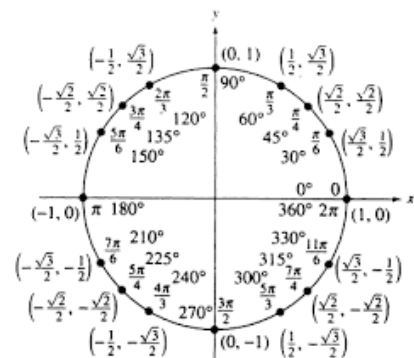
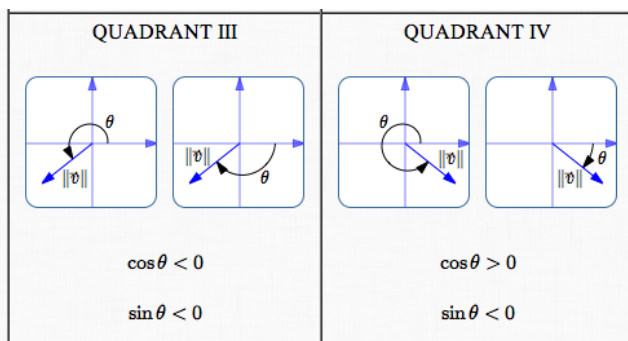
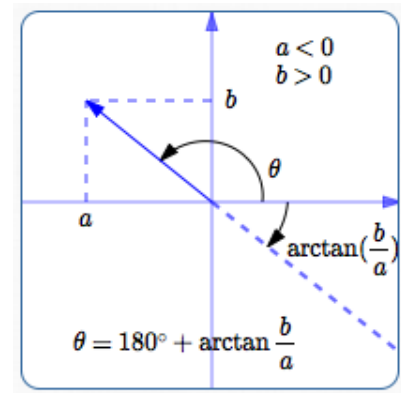
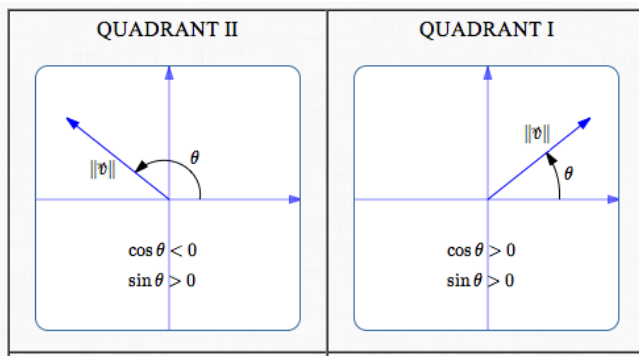
การคำนวณหาความเร็วและทิศทางการไหลของน้ำ สามารถคำนวณได้จากการหาขนาดและมุมเฉลี่ยของแต่ละเวกเตอร์จากสนามเวกเตอร์ที่ประมวลผลผ่านฟังก์ชันสำเร็จรูปจาก module opencv นั่นคือ `cv2.calcOpticalFlowFarneback()` ฟังก์ชันนี้จะรีเทิร์นค่าเป็น 2D array ของคู่อันดับ x_1, y_1 กับ x_2, y_2 ทำให้สามารถคำนวณได้จากการหาขนาดด้วยสมการ magnitude vector formula ดังต่อไปนี้

The **magnitude** or **length** of the vector $\mathbf{v} = \overrightarrow{PQ}$ is the nonnegative number

$$|\mathbf{v}| = \sqrt{v_1^2 + v_2^2 + v_3^2} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

(See Figure 12.10.)

และคำนวณหาแนวโน้มของทิศทางน้ำได้โดยอาศัยหลักการของตรีโกณมิติและวงกลม 1 หน่วยดังภาพเมื่อ a และ b เป็นขนาดของเวกเตอร์เฉลี่ยที่ทำได้ และนำไปพล็อตแบบเชิงขั้วโดย module matplotlib



การวิเคราะห์ความแรงของคลื่นน้ำและความรุนแรงของการไหล จะอาศัยหลักการทางสถิติทำการวิเคราะห์จากค่าความแปรปรวนของขนาดเวกเตอร์ในสนามเวกเตอร์ ที่มีค่าที่กระจุกกระจายไม่เป็นระเบียบว่ามีค่ามากหรือน้อยเพียงใด ดังนี้

	Population	Sample
Variance	$\sigma^2 = \frac{\sum_{i=1}^N (x_i - \mu)^2}{N}$	$S^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}$
Standard deviation	$\sigma = \sqrt{\frac{\sum_{i=1}^N (x_i - \mu)^2}{N}}$	$S = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$

เปรียบเทียบการคำนวณกับโปรแกรมที่เขียนจริงดังนี้

การทำ magnitude vector

```
#vector
U = np.array([lines[t][0][0]-lines[t][1][0] for t in range(len(y))])
V = np.array([lines[t][0][1]-lines[t][1][1] for t in range(len(y))])
ang = []
for i in range(len(V)):
    agcal = V[i] / U[i]
    er = ["-inf", "inf", "nan"]
    if str(agcal) in er: agcal = 0
    ang.append(agcal)

qd[0] = np.mean(U)
qd[1] = np.mean(V)
mgnitue = (U**2 + V**2)**0.5
angl = np.array([np.arctan(ag) for ag in ang])
MeanAngle = np.mean(angl)* 57.2957795
MeanVelocity = np.mean(mgnitue)
```

การหา องศาและทิศทางการน้ำ

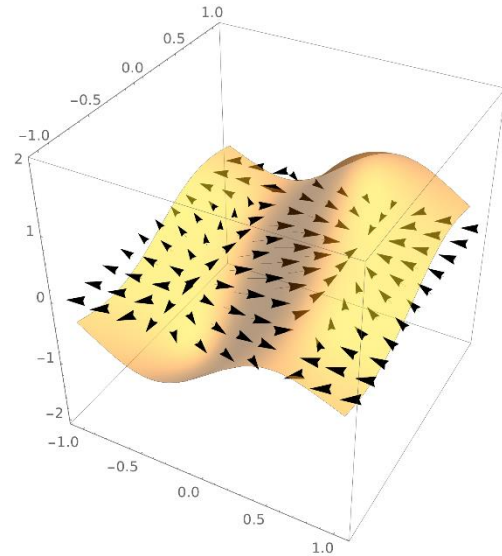
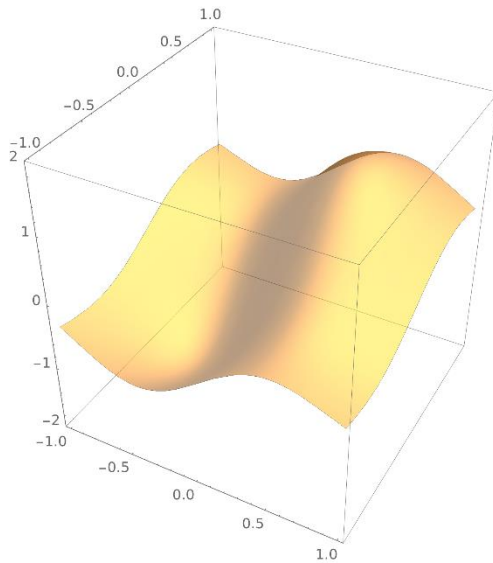
```
icons = ['right',
          'up',
          'left',
          'down']
icon = 'w:direction-'
deg = abs(MeanAngle)

if qd[0] >= 0 and qd[1] >= 0: # (+,+) quadrant 1
    deg = 0 + deg
    id = int(qd[0]<qd[1])
    icon += icons[id]
elif qd[0] <= 0 and qd[1] >= 0: # (-,+) quadrant 2
    deg = 180 - deg
    id = int(abs(qd[0]) > qd[1])
    icon += icons[id+1]
elif qd[0] <= 0 and qd[1] <= 0: # (-,-) quadrant 3
    deg = 180 + deg
    id = int(abs(qd[0]) < abs(qd[1]))
    icon += icons[id + 2]
elif qd[0] >= 0 and qd[1] <= 0: # (+,-) quadrant 4
    deg = 360 - deg
    id = int(qd[0] > abs(qd[1]))
    icon += icons[id - 1]
```

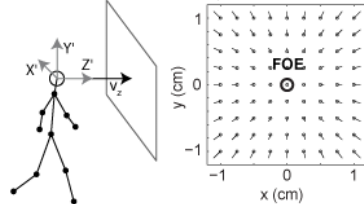
การหาความรุนแรงการไหลของน้ำในที่นี้จะใช้ฟังก์ชันสำเร็จรูปของ module เพื่อการวิเคราะห์ทางคณิตศาสตร์และวิทยาศาสตร์อย่าง numpy ด้วยฟังก์ชัน `numpy.var()` ดังนี้

```
MeanVelocity = np.mean(mgnitue)
VarianFlow = np.var(U)
```

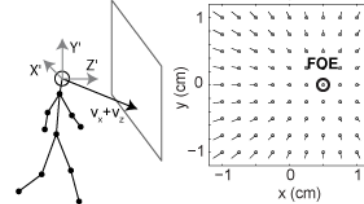
นอกจากนี้ยังใช้ความรู้เรื่อง vector field วิเคราะห์ในระดับที่สูงขึ้นไปเพื่อให้สามารถประยุกต์ใช้งานได้หลากหลาย วิเคราะห์ในหลายมุมมองทำให้มีประสิทธิภาพมากขึ้น เช่น อาจสามารถจำลองพื้นผิวของน้ำและการเคลื่อนไหวในแบบสามมิติได้ เป็นต้น



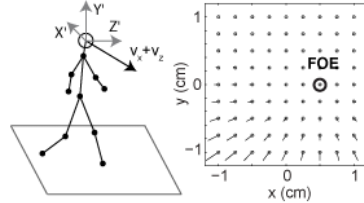
a Forward translation & back-plane



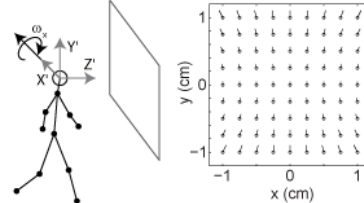
b Forward/sideward translation & back-plane



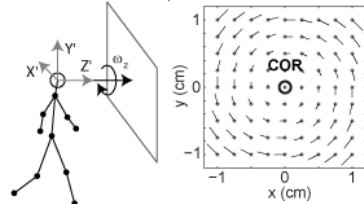
c Forward translation & ground-plane



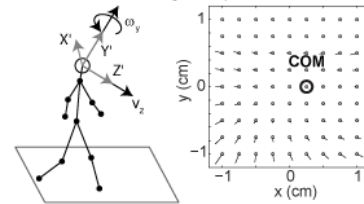
d Pitch rotation & back-plane



e Roll rotation & back-plane



f Curvilinear motion & ground-plane



เมื่อคำนวณค่าต่างๆแล้วจึงทำการส่งข้อมูลขึ้น cloud platform เพื่อเก็บข้อมูลและแสดงผลแบบเรียลไทม์ในที่นี้ใช้เป็น IoT platform ของ adafruit สื่อสารด้วย MQTT protocol



โค้ดส่วนการทำงานเป็นดังนี้

```
def Up2Adafruit(i):
    feedsname = ['velocity-percent', 'varian-flow', 'velocity', 'trend-direction']
    value = [get_VperMax(), VarianFlow, MeanVelocity, get_direction()][1]

    #for i in range(k):
    feed = aio.feeds(feedsname[i])
    # aio.send_data(feed.key, value[i])
    aio.append(feed.key, value[i])

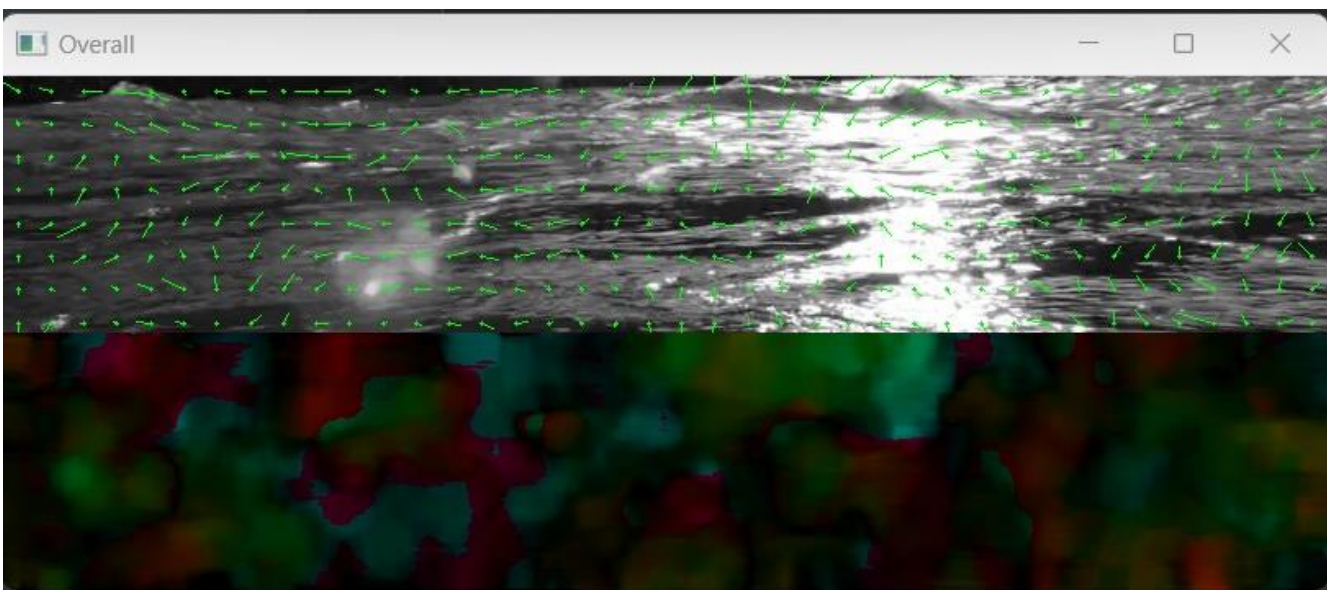
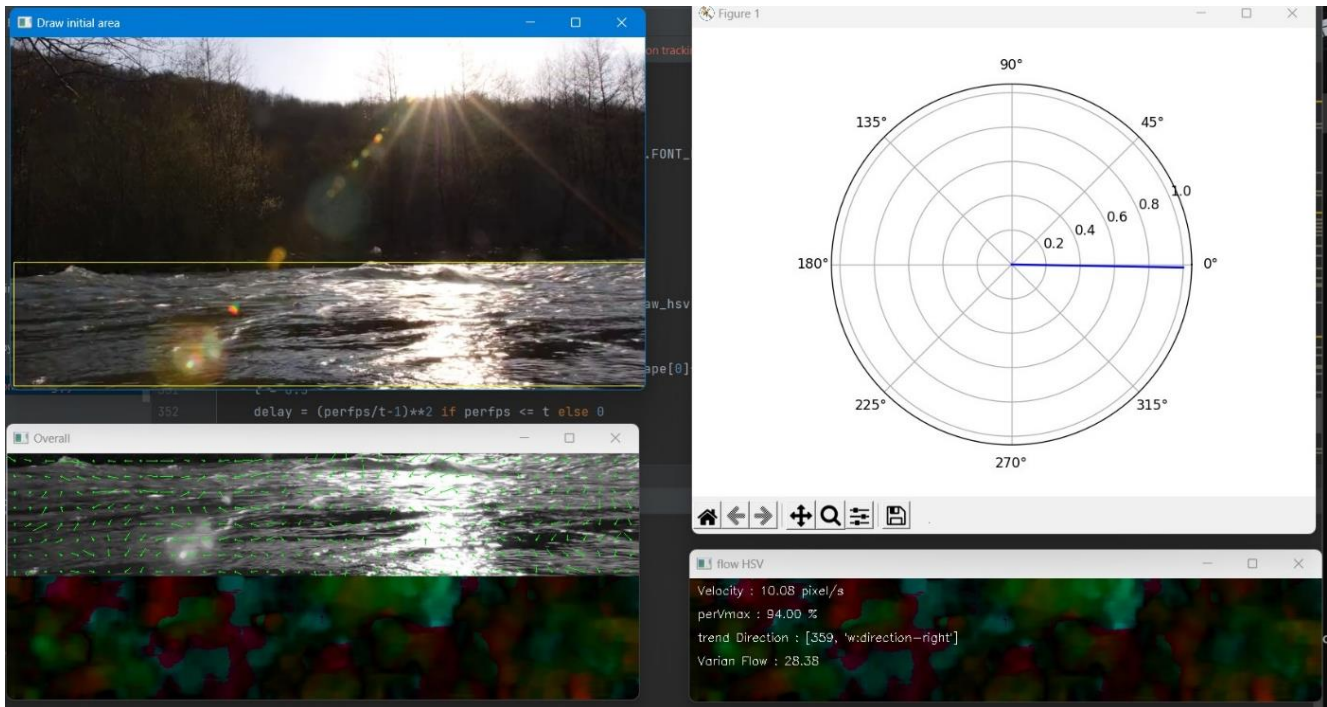
    print('\n-----status-----')
    print(datetime.datetime.now().strftime("%d %b %Y %I:%M:%S%p"))
    print(f'\nVector mean : {qd[0]:.2f} i + {qd[1]:.2f} j')
    print(f'MeanAng : {MeanAngle:.2f}')

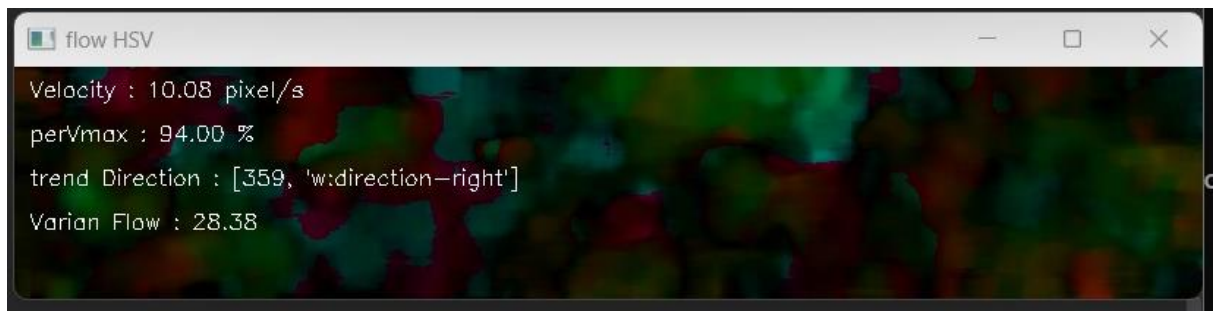
    print(f'\nMeanV : {MeanVelocity:.2f} pixel/s')
    print(f'perVmax : {get_VperMax():.2f} %')
    print(f'trend Direction : {get_direction()}')
    print(f'Var Flow : {VarianFlow:.2f}')
```


ผลการทำงาน

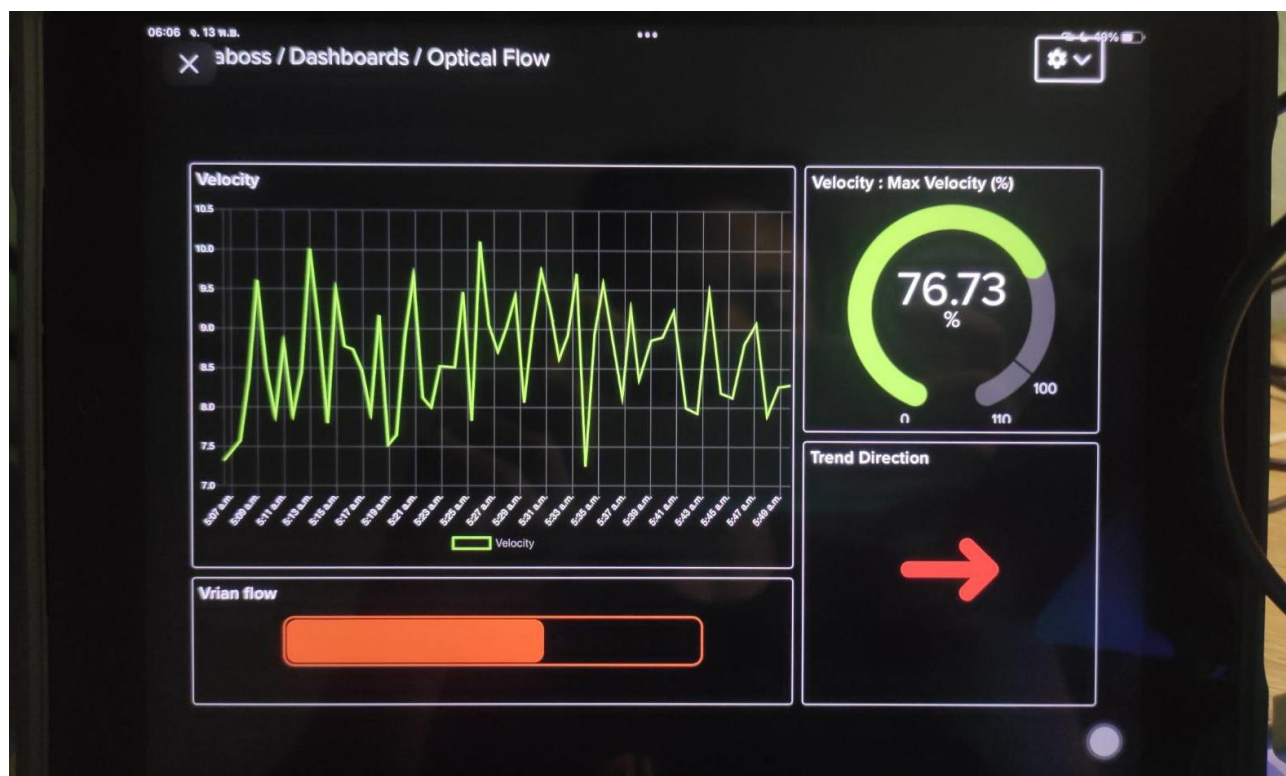
ทดลองการทำงานโดยใช้คลิปวิดีโอตัวอย่างจาก อินเทอร์เน็ต 1 วิดีโอ และบันทึกภาพจากสถานที่จริง บริเวณคลองประเวศคณะสถาปัตยกรรมสถาบันเทคโนโลยีเจ้าคุณทหารบากระบัง มี 2 วิดีโอ มีผลลัดังนี้

1. sample.mp4 (น้ำไหลแรง ทิศทางไปทางขวา)

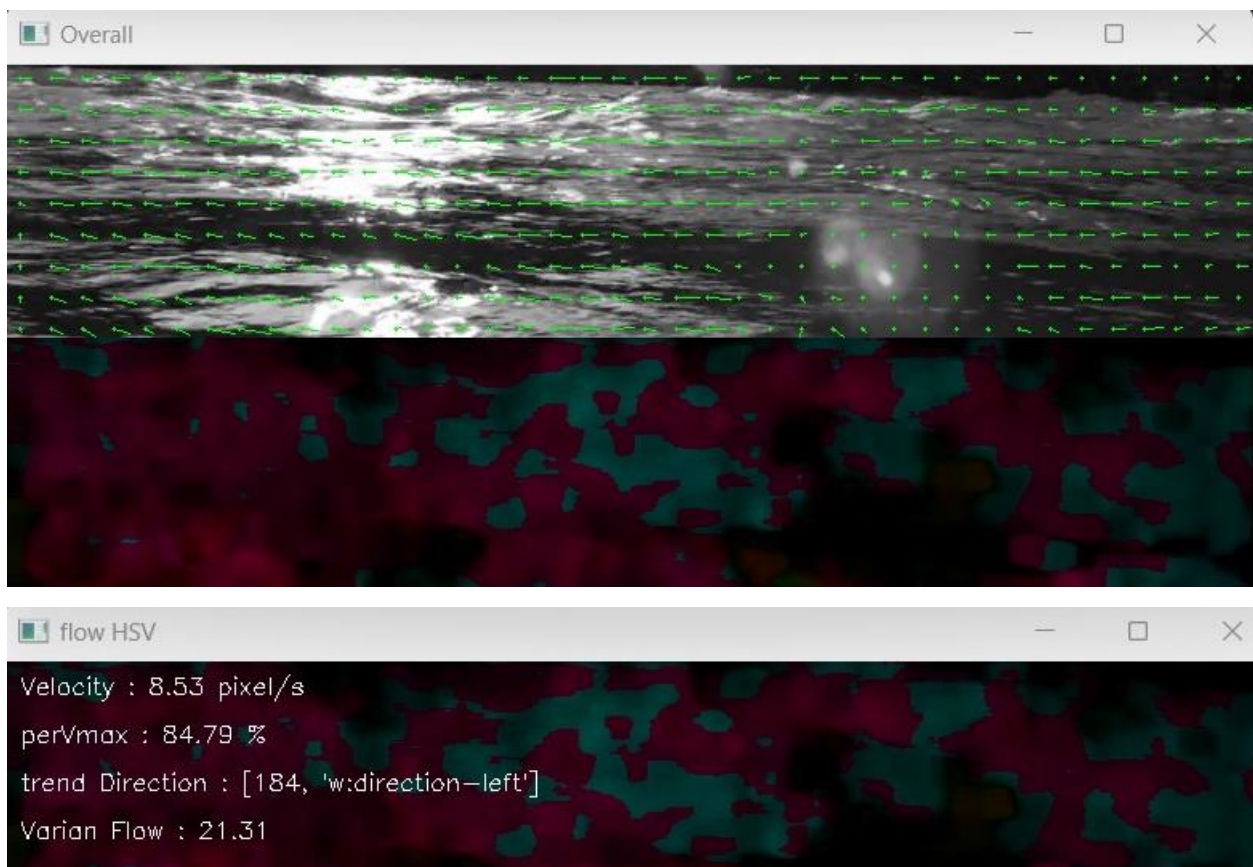
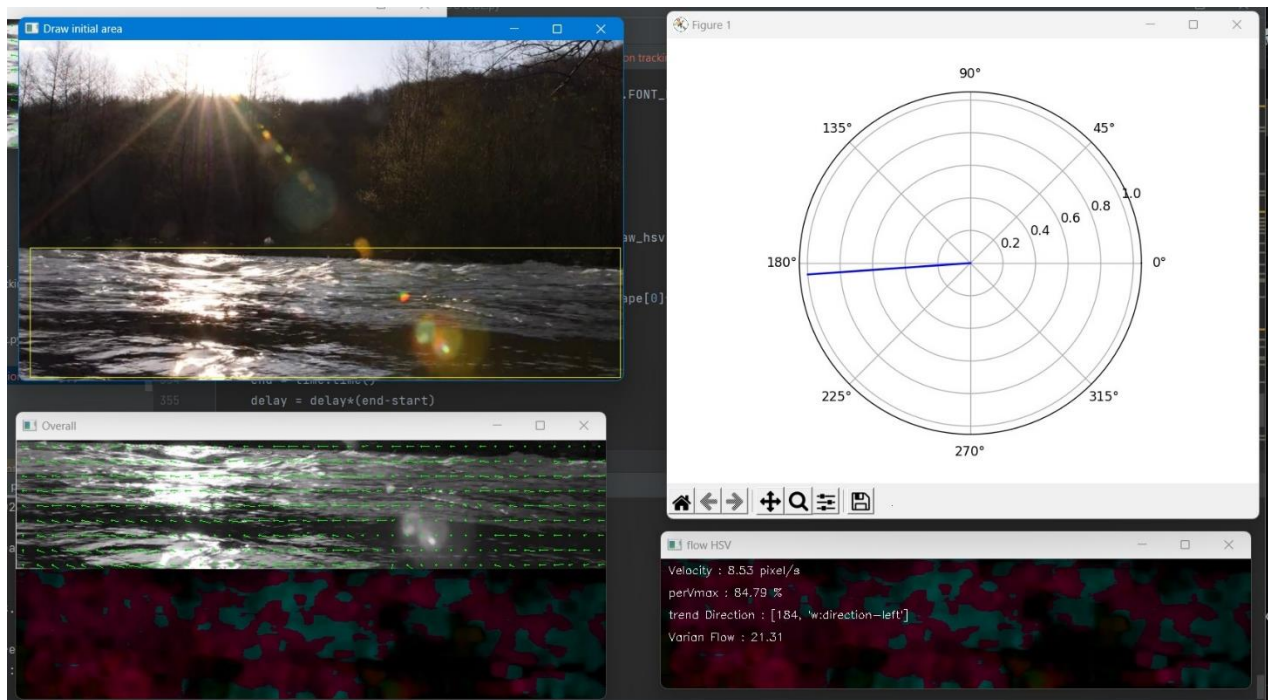




แดชบอร์ดแสดงสถานะต่างๆ



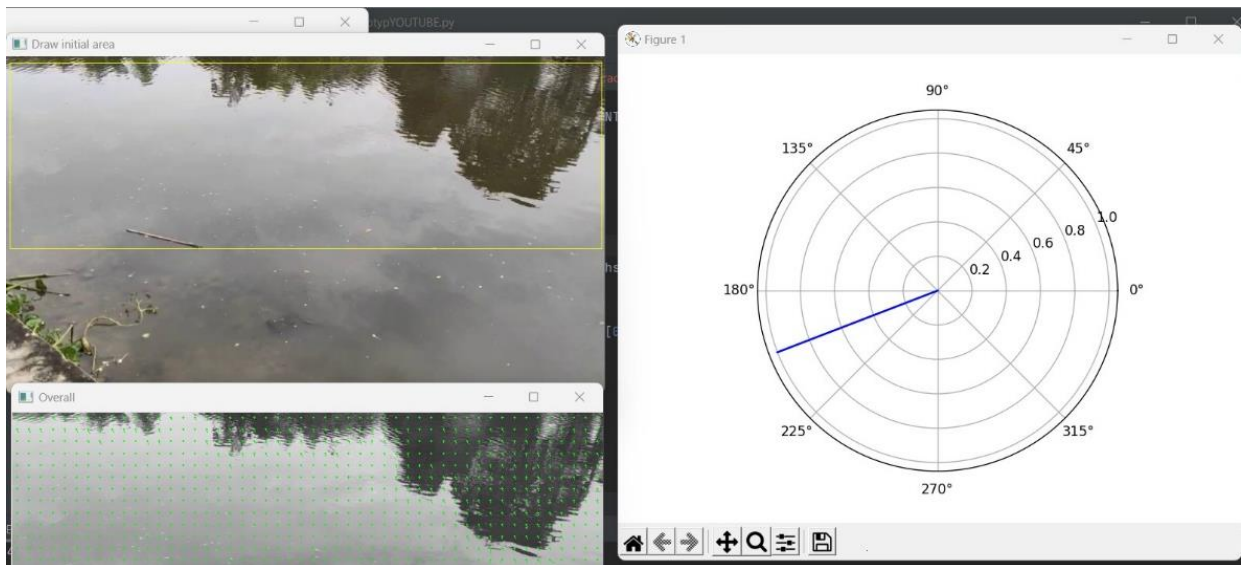
2. sample.mp4 (น้ำไหลแรง ทิศทางไปทางซ้าย) flip Video



แดชบอร์ดแสดงสถานะต่างๆ



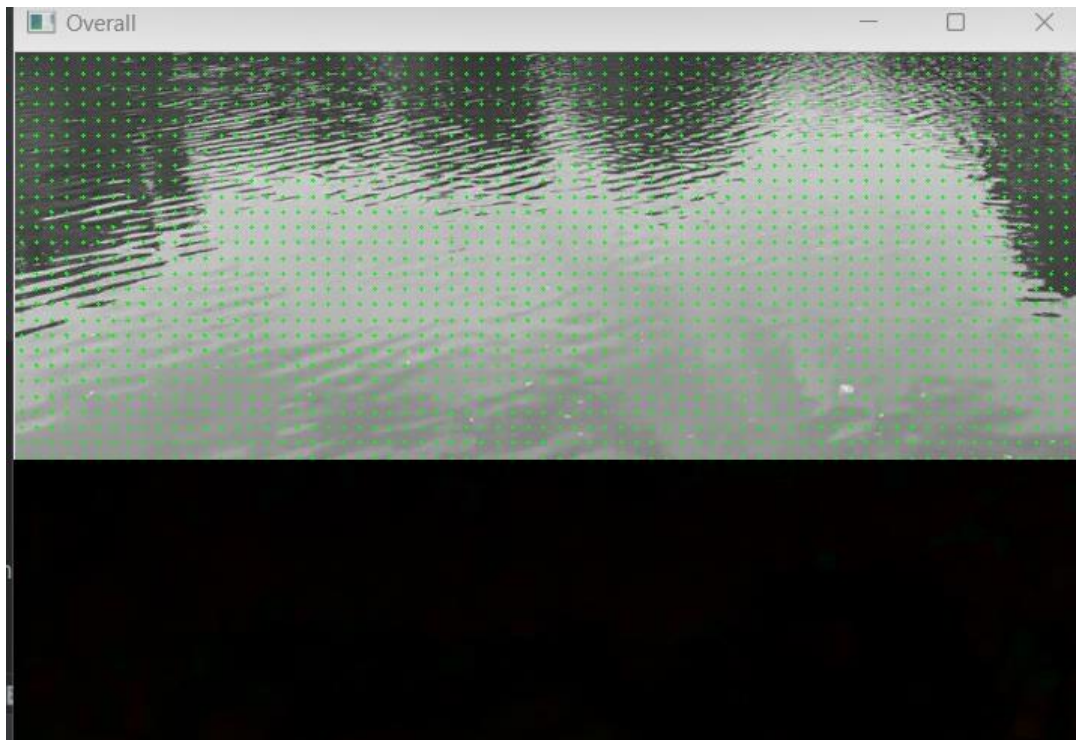
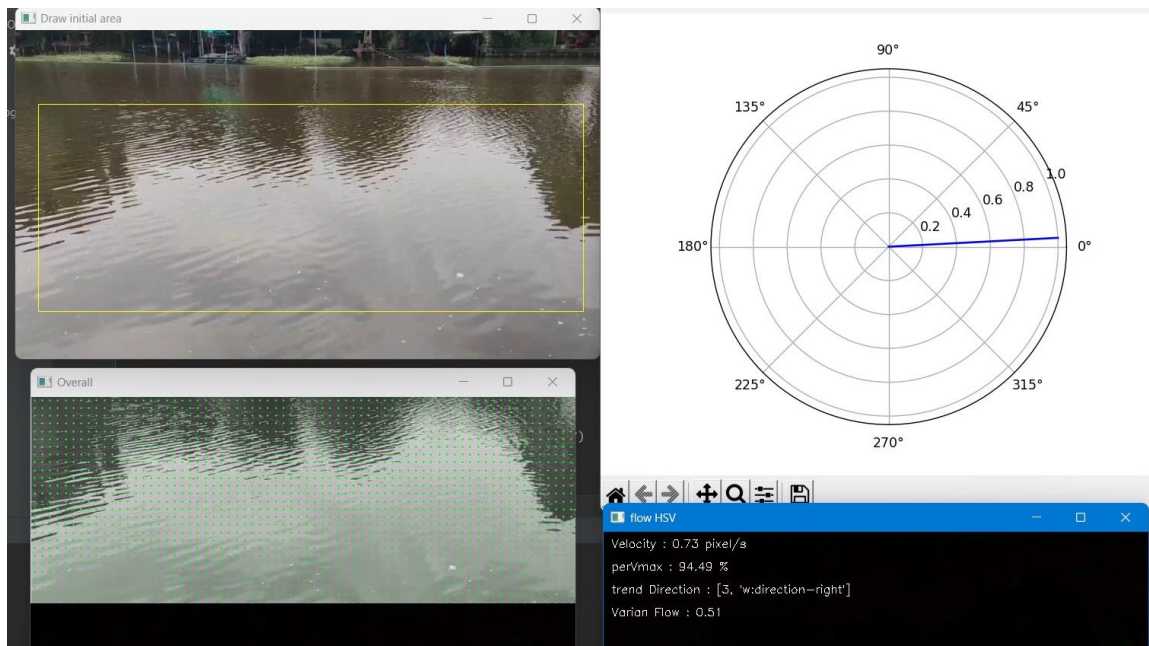
3. sample2.mp4 (น้ำค่อนข้างนิ่ง ไหลไปทางซ้าย) (คลองประเวศ)

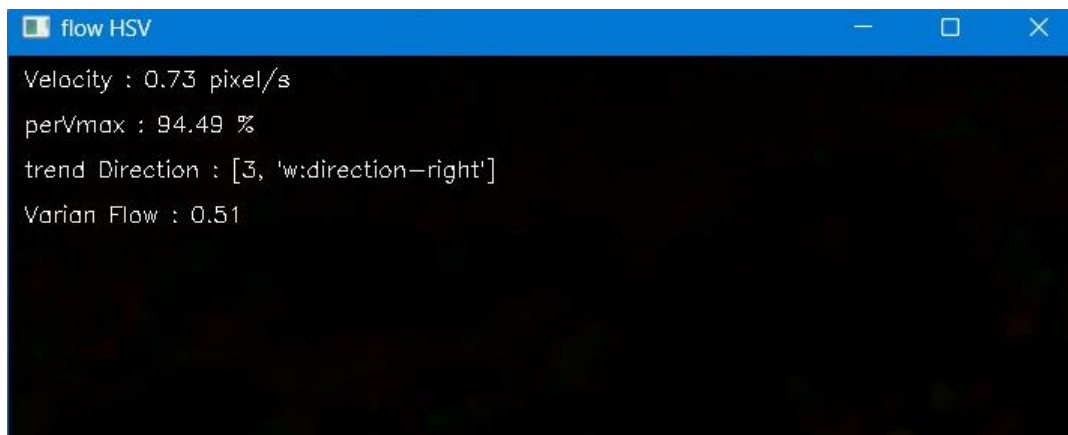


แดชบอร์ดแสดงสถานะต่างๆ



4. sample3.mp4 (น้ำค่อนข้างนิ่ง ไหลไปทางขวา) (คลองประเวศ)





แดชบอร์ดแสดงสถานะต่างๆ

