

การพัฒนาโปรแกรมประยุกต์และปัญญาประดิษฐ์ เพื่อการมองเห็นของเครื่องจักร
Computer Programming and Artificial Intelligence in Machine Vision

3/4 – Machine Vision and Image Processing

- การประมวลผลภาพเคลื่อนไหว
- เครื่องมือทั่วไปที่ใช้ในการประมวลผลภาพ
- การค้นหาจุดด้วยการจับคู่ระหว่างจุดสำคัญ (Key point Matching)
- Mini Project_1: Pokémon Matching Image Project
- คำถ้ามหำบบเพื่อทดสอบความเข้าใจ

1/6 – การประมวลผลภาพเคลื่อนไหว

1. การบันทึกภาพ

เนื่องจากหลักการประมวลผลภาพ จะเป็นการประมวลผล จากภาพนิ่ง หากแต่การนำเข้าข้อมูลเป็นภาพเคลื่อนไหว เช่น ไฟล์วีดีโอ หรือการสตรีมวีดีโอด้วยกล้องเว็บแคม หรือกล้องออนไลน์จากเครือข่ายที่สามารถเข้าถึงได้จาก IP Address ดังนั้น การประมวลผลภาพจะทำได้ด้วยวิธีการประมวลผลภาพจากภาพนิ่งแบบต่อเนื่องกัน

Lab101_อ่านภาพนิ่งจากไฟล์วีดีโอด้วยภาษา Python

1. การอ่านภาพนิ่งจากไฟล์วีดีโอด้วยภาษา Python

```

1 import cv2
2 cap = cv2.VideoCapture('./image/move.avi')
3 while cap.isOpened():
4     ret, image = cap.read()
5     if ret == False:
6         break
7     cv2.imshow ('Image Show', image)
8     if cv2.waitKey(1) == ord('q'):
9         break
10 cap.release()
11 cv2.destroyAllWindows ()
```

2. การบันทึกภาพนิ่งจากไฟล์วิดีโอ-1

| | |
|--|--|
|  | สร้าง folder frame ก่อน |
| <pre> 1 import cv2 2 cap = cv2.VideoCapture('./image/move.avi') 3 while cap.isOpened(): 4 ret, image = cap.read() 5 if ret == False: 6 break 7 cv2.imshow ('Image Show', image) 8 cv2.imwrite ('./frame/result.jpg', image) ← 9 10 if cv2.waitKey(1) == ord('q'): 11 break 12 13 cap.release() 14 cv2.destroyAllWindows () 15 </pre> | บันทึกรูปชื่อ result.jpg ในแฟ้ม frame |
|  | ผลการทำงาน |

3. การบันทึกภาพนิ่งจากไฟล์วิดีโอ-1

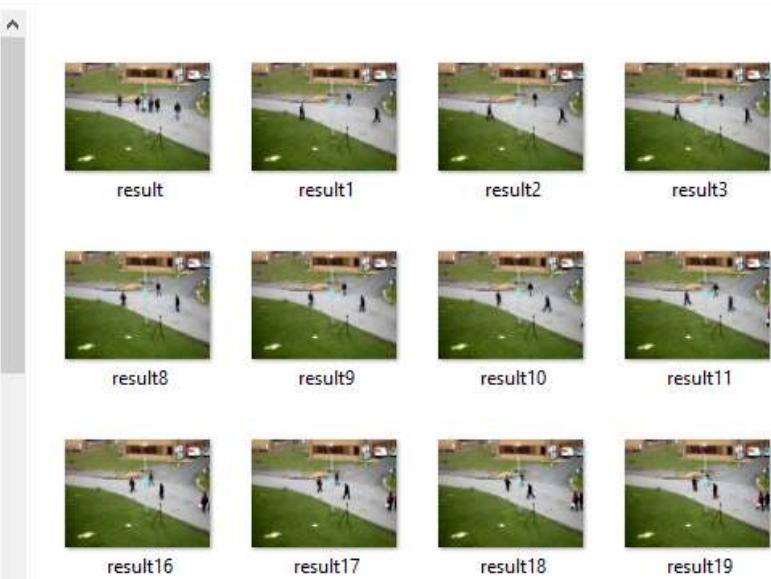
```

1 import cv2
2 cap = cv2.VideoCapture('./image/move.avi')
3 i = 1
4 while cap.isOpened():
5     ret, image = cap.read()
6     if ret == False:
7         break
8     cv2.imshow('Image Show', image)
9     cv2.imwrite('./frame/result'+str(i)+'.jpg', image)
10    i += 1
11    if cv2.waitKey(1) == ord('q'):
12        break
13 cap.release()
14 cv2.destroyAllWindows()

```

กำหนดตัวแปร i เพื่อใช้คำนวน
เปลี่ยนตัวเลขเป็นตัวอักษรเพื่อใช้ในการตั้งชื่อไฟล์
เพิ่มค่า i ขึ้น 1 ค่า

Week08 > frame

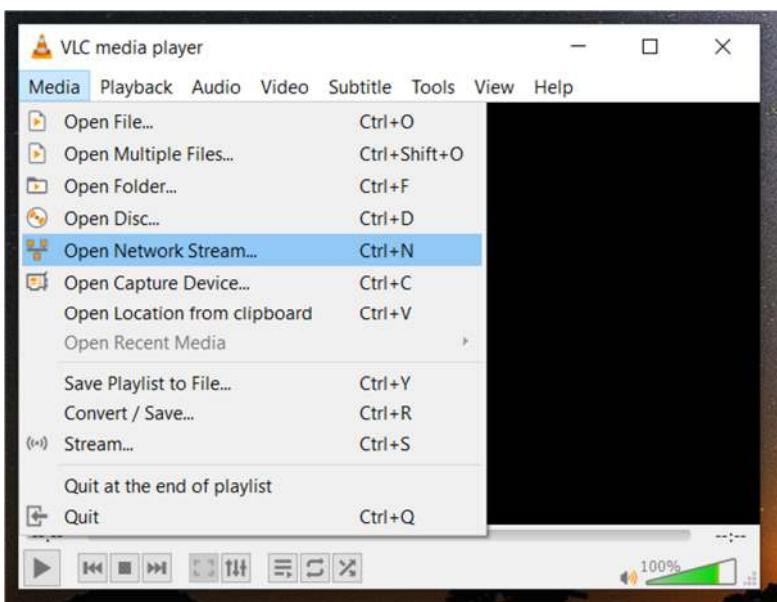


ผลการทำงาน

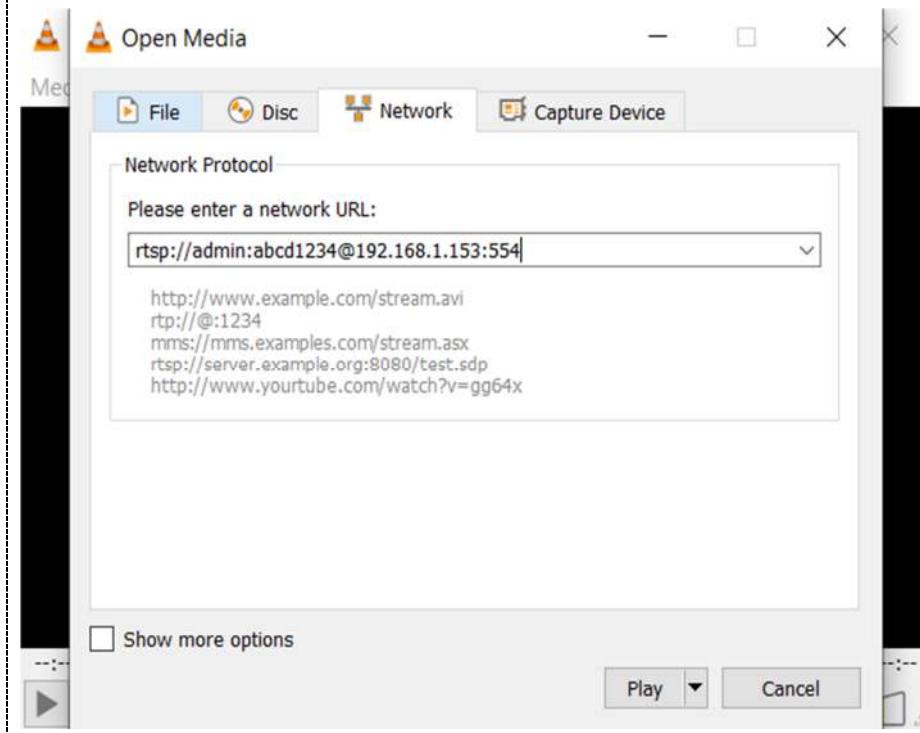
Lab102a อ่านภาพผ่านจากวิดีโอสตรีม

4. ทดสอบวิดีโอสตรีม

ดาวน์โหลด VLC จาก www.videolan.org



rtsp://admin:abcd1234@192.168.1.153:554



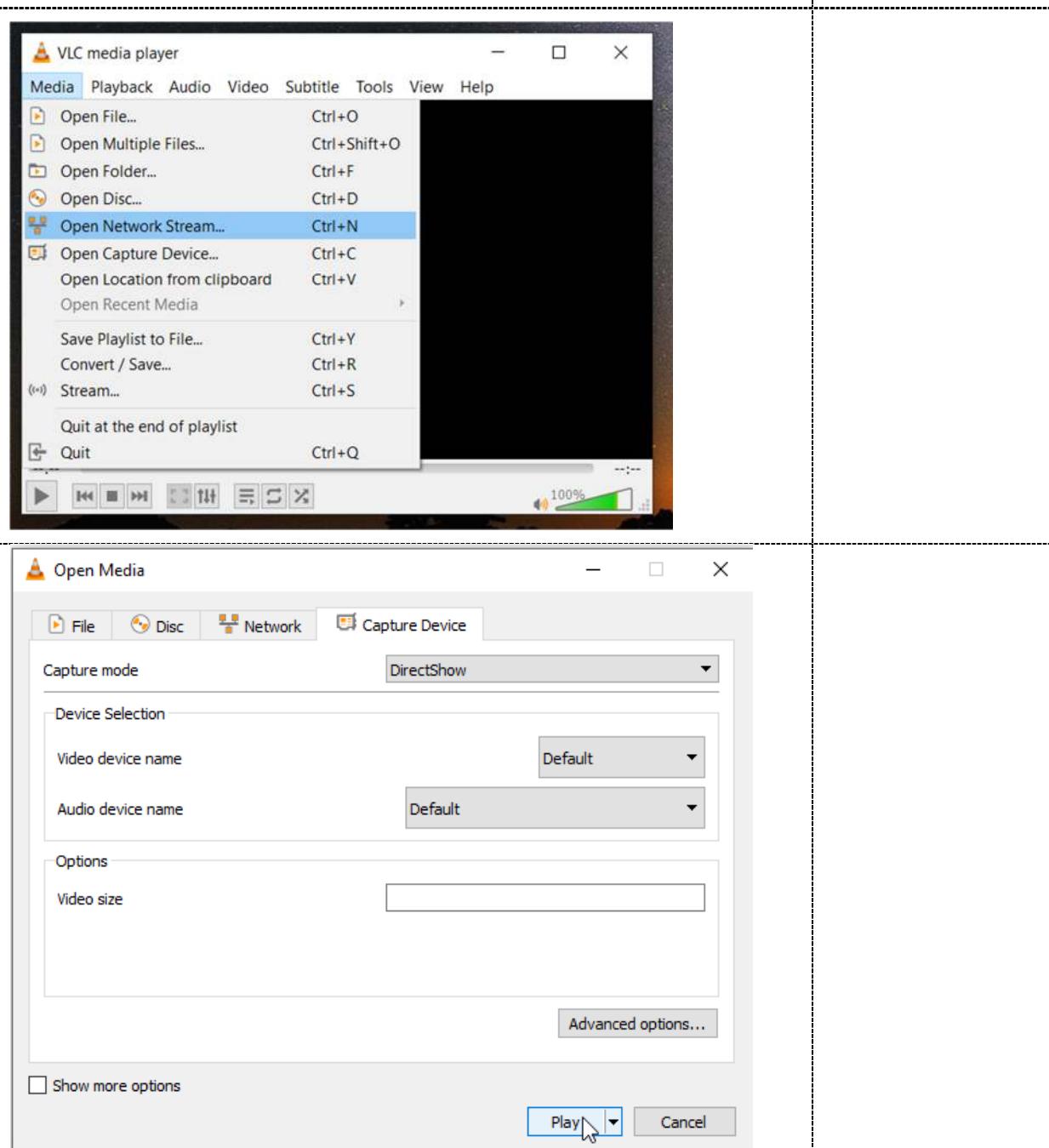
5. การบันทึกภาพนิ่งจากการสตรีมวีดีโอ

```
1 import cv2
2 import datetime
3 #cap = cv2.VideoCapture('rtsp://admin: abcd1234@192.168.1.64:554')
4 cap = cv2.VideoCapture(0)
5 while True:
6     ret, image = cap.read()
7     cv2.imshow('Image Show', image)
8     pressedKey = cv2.waitKey(1)
9     if pressedKey == ord('q'):
10         break
11     elif pressedKey == ord('s'):
12         now = datetime.datetime.now().strftime ('%y%d%m_%H%M%S')
13         cv2.imwrite('./save_'+str(now)+'.jpg', image)
14 cap.release()
15 cv2.destroyAllWindows ()
```

Lab102b อ่านภาพนิ่งจาก WebCam

6. ทดสอบ WebCam

ดาวน์โหลด VLC จาก www.videolan.org



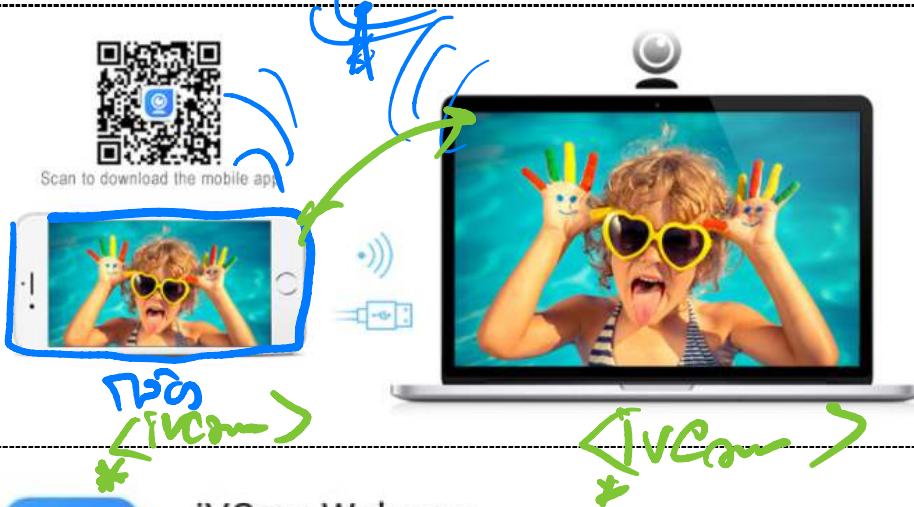
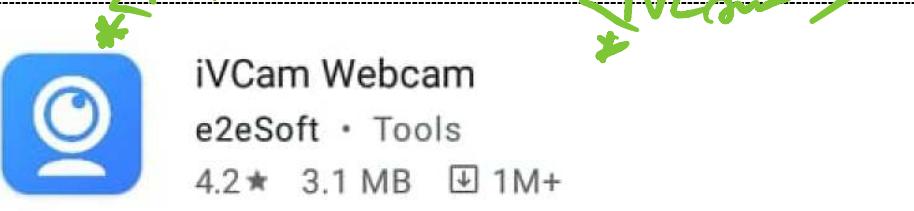
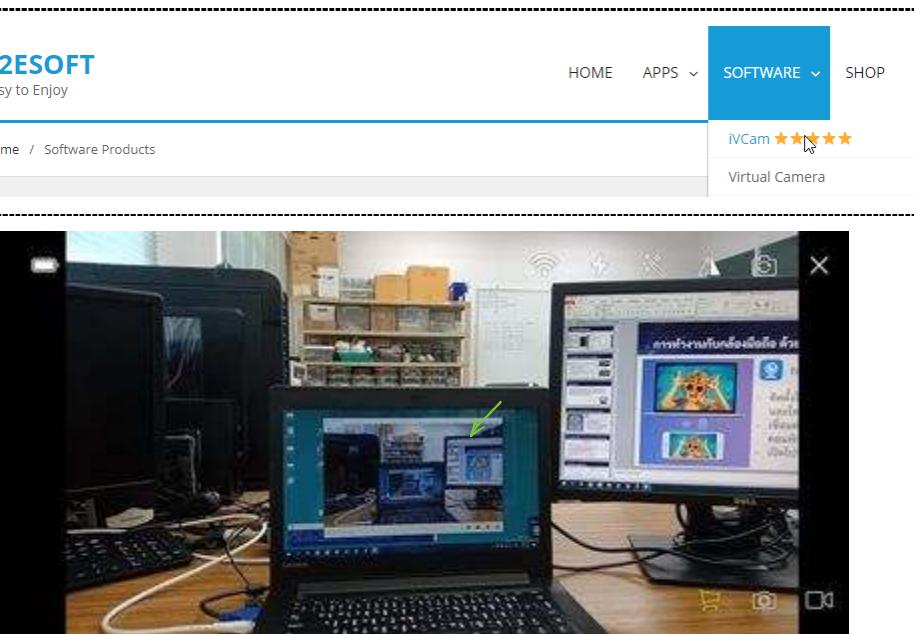
7. การบันทึกภาพนิ่งจาก WebCam

```
1 import cv2
2 import datetime
3 #cap = cv2.VideoCapture('rtsp://admin: abcd1234@192.168.1.64:554')
4 cap = cv2.VideoCapture(0)
5 while True:
6     ret, image = cap.read()
7     cv2.imshow('Image Show', image)
8     pressedKey = cv2.waitKey(1)
9     if pressedKey == ord('q'):
10         break
11     elif pressedKey == ord('s'):
12         now = datetime.datetime.now().strftime ('%y%d%m_%H%M%S')
13         cv2.imwrite('./save_'+str(now)+'.jpg', image)
14 cap.release()
15 cv2.destroyAllWindows ()
```

Lab103_การส่งภาพจากกล้อง Webcam ไปยังโทรศัพท์มือถือ

8. การทำงานกับกล้องมือถือ ด้วยโปรแกรม ivCam >> <https://www.e2esoft.com/>

- ติดตั้งโปรแกรมบนคอมพิวเตอร์ และโทรศัพท์มือถือ
- เชื่อมต่อเครือข่ายเดียวกันสำหรับ คอมพิวเตอร์ กับโทรศัพท์มือถือ
- เปิดโปรแกรมเพื่อเชื่อมต่อ

| | |
|--|--|
|  | Notebook Webcam To Mobile |
|  | Android App |
|  | Notebook Program ผลการทำงาน ภาพจาก โทรศัพท์มือถือ |

Lab104_การบันทึกภาพเคลื่อนไหว

9. การโหลดไฟล์วิดีโอ และการอ่านค่าคุณสมบัติของไฟล์

```

1 import cv2
2 cap = cv2.VideoCapture('./image/move.avi')
3
4 frame_number = cap.get(cv2.CAP_PROP_FRAME_COUNT)
5 h = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
6 w = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
7 fps = int(cap.get(cv2.CAP_PROP_FPS))
8
9 print(frame_number, h, w, fps)

```

795.0 576 768 10

H W 10fps

กำหนดชื่อไฟล์วิดีโอ

จำนวนเฟรม

ความสูง

ความกว้าง

จำนวนเฟรมต่อวินาที

10. การบันทึกภาพเคลื่อนไหว

- กำหนดรูปแบบไฟล์

`fourcc.VideoWriter_fourcc(*'XVID')`

รูปแบบการบีบอัดไฟล์

- `cv2.VideoWriter_fourcc(*'XVID')` for avi
- `cv2.VideoWriter_fourcc(*MP4V)` for mp4

- กำหนดรูปแบบการบันทึกไฟล์วิดีโอ

`output = cv2.VideoWriter(filename, fourcc, fps, frameSize)`

ชื่อไฟล์, วิดีโอฟอร์แมท, จำนวนเฟรม, (กว้าง,สูง)

- บันทึกรูปลงไฟล์วิดีโอ

`output.write(image)`

11. การบันทึกภาพเคลื่อนไหวจากไฟล์วีดีโอ

```

1 import cv2
2 cap = cv2.VideoCapture('./image/move.avi')
3 h = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
4 w = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
5 fps = int(cap.get(cv2.CAP_PROP_FPS))
6
7 fourcc = cv2.VideoWriter_fourcc(*'XVID')
8 out = cv2.VideoWriter('./image/save1.avi', fourcc, fps, (w,h))
9
10 while(cap.isOpened()):
11     ret, frame = cap.read()
12     if ret == True:
13         out.write(frame) ←
14         cv2.imshow('frame', frame)
15         if cv2.waitKey(1) & 0xFF == ord('q'):
16             break
17     else:
18         break
19 cap.release()
20 out.release()
21 cv2.destroyAllWindows()
22

```

กำหนด
รูปแบบไฟล์

กำหนด
รูปแบบการ
บันทึกไฟล์
วีดีโอ

บันทึกรูปลง
ไฟล์วีดีโอ

12. การบันทึกภาพเคลื่อนไหวจากกล้อง WebCam

```

1 import cv2
2 video = cv2.VideoCapture(0)
3 fourcc = cv2.VideoWriter_fourcc(*'XVID') ←
4 h = 640
5 w = 480 ←
6 fps = 20.0
7 result = cv2.VideoWriter('./image/save2.avi', fourcc, fps, (w,h))
8
9 while(True):
10     ret, frame = video.read()
11     if ret == True:
12         result.write(frame) ←
13         cv2.imshow('frame', frame)
14         if cv2.waitKey(1) & 0xFF == ord('s'):
15             break
16     else:
17         break
18 video.release()
19 result.release()
20 cv2.destroyAllWindows()
21

```

กำหนด
รูปแบบไฟล์

กำหนด
รูปแบบการ
บันทึกไฟล์
วีดีโอ

บันทึกรูปลง
ไฟล์วีดีโอ

Lab105_การทำงานกับภาพต่างฟอร์แมท (PIL <- -> cv2)Logo create → <https://onlinepngtools.com/create-transparent-png>

13. ทำไม OpenCV ใช้ BGR แทนที่จะใช้ RGB ?

- รูปแบบสี BGR จึงเป็นที่นิยมในหมู่ผู้ผลิตกล้องถ่ายรูปและผู้ให้บริการซอฟต์แวร์ภาพ
- ช่องสีแดงต้องเป็นหนึ่งในช่องสีที่สำคัญน้อยที่สุดดังนั้นจึงอยู่ในรายการสุดท้ายและบิตแมปจำนวนมากใช้รูปแบบ BGR สำหรับการจัดเก็บภาพ

14. การเปิดภาพด้วย PIL

- ให้แน่ใจว่า มี PIL ใน python3 หากไม่มีให้เพิ่ม < **conda install -c anaconda pillow** >

```

1 # Open PIL image
2 import cv2
3 import numpy as np
4 from PIL import Image
5 image = Image.open(r"C:\Users\Pk007_20210109Bit64\Desktop\Week08\image\rsu.png")
6 image.show()
7

```

15. การแปลงข้อมูลภาพ cv2 เป็น PIL

```

1 # Convert CV2 to PIL image
2 import cv2
3 import numpy as np
4 from PIL import Image
5 opencv_image=cv2.imread("C:/Users/Pk007_20210109Bit64/Desktop/Week08/image/lena.jpg")
6
7 color_cverted = cv2.cvtColor(opencv_image, cv2.COLOR_BGR2RGB)
8 pil_image=Image.fromarray(color_cverted)
9
10 pil_image.show()
11

```

เปลี่ยนรูปแบบสีจาก BGR เป็น RGB

เปลี่ยนรูปแบบของภาพ

16. การแปลงข้อมูลภาพ PIL เป็น cv2

```

1 # Convert PIL to CV2 image
2 import cv2
3 import numpy as np
4 from PIL import Image
5 pil_img=Image.open("C:/Users/Pk007_20210109Bit64/Desktop/Week08/image/lena.jpg")
6
7 numpy_image=np.array(pil_img)
8 opencv_image=cv2.cvtColor(numpy_image, cv2.COLOR_RGBA2BGR)
9
10 cv2.imshow("image",opencv_image)
11 cv2.waitKey(0)
12 cv2.destroyAllWindows()

```

เปลี่ยนรูปแบบของภาพ

เปลี่ยนรูปแบบสีจากRGB เป็น BGR

17. การทำงานกับภาพที่ต่างฟอร์แมท

```

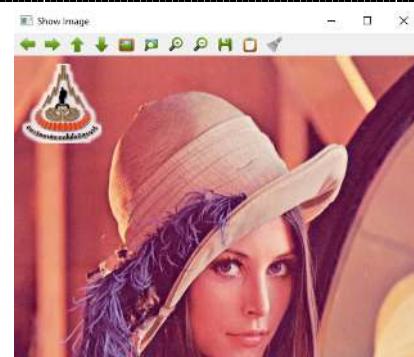
1 # Insert Logo to Picture
2 import cv2
3 import numpy as np
4 from PIL import Image
5 image = cv2.imread("C:/Users/Pk007_20210707Bit64/Desktop/MV30/image/lena.jpg")
6 logo = Image.open(r"C:\Users\Pk007_20210707Bit64\Desktop\MV30\image\SUT_PNG.png")
7 logo.thumbnail((100, 100))
8
9 color_covered = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
10 pil_image=Image.fromarray(color_covered)
11
12 pil_image.paste(logo, (0+10,0+10),logo)
13
14 numpy_image=np.array(pil_image)
15 opencv_image=cv2.cvtColor(numpy_image, cv2.COLOR_RGB2BGR)
16
17 cv2.imshow('Show Image', opencv_image)
18 cv2.waitKey()
19 cv2.destroyAllWindows()

```

แปลง PIL เป็น cv2

แทรกภาพโลโก้ที่ตำแหน่ง 0+10, 0+10

แปลง cv2 เป็น PIL



18. การแทรกโลโก้ลงในภาพเคลื่อนไหว

```

1 # Insert Logo to Movies
2 import cv2
3 import numpy as np
4 from PIL import Image
5 cap = cv2.VideoCapture('C:/Users/Pk007_20210707Bit64/Desktop/MV30/image/move.avi')
6 logo = Image.open('C:/Users/Pk007_20210707Bit64/Desktop/MV30/image/SUT_Trans.png')
7 logo.thumbnail((150, 150))
8 while(True):
9     ret, image = cap.read()
10
11     color_couvered = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
12     pil_image=Image.fromarray(color_couvered)
13
14     pil_image.paste(logo, (0+10,0+10),logo)
15
16     numpy_image=np.array(pil_image)
17     opencv_image=cv2.cvtColor(numpy_image, cv2.COLOR_RGBA2BGR)
18
19     cv2.imshow('Show Image', opencv_image)
20     if cv2.waitKey(1) & 0xFF == ord('q'):
21         break
22 cap.release()
23 cv2.destroyAllWindows()

```

Line

- <5> อ่านวีดีโอไฟล์
- <8> วนซ้ำเพื่ออ่านภาพ
- <9> อ่านภาพที่ลิสเฟรม
- <20> ตรวจสอบเงื่อนไขเพื่อจบการวนซ้ำ



กิจกรรมที่ 1/6 – Webcam to Video with Logo

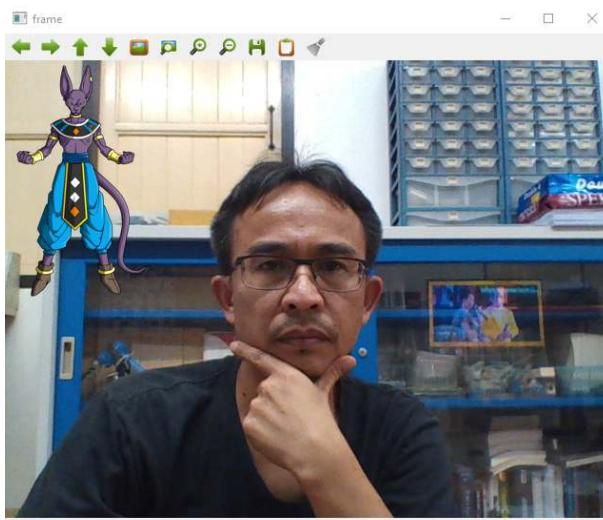
จะเขียนโปรแกรมเพื่อบันทึกวิดีโอด้วยกล้องเวปแคม ทำการใส่โลโก้ของตัวเอง และระบุวันที่ในภาพ

```

1 import cv2
2 import numpy as np
3 from PIL import Image
4 cap = cv2.VideoCapture(0)
5 pil_logo = Image.open(r"C:\Users\Pk007_20210707Bit64\Desktop\MV30\image\Bill_DragonBall.png")
6 pil_logo.thumbnail((250, 250))

7
8
9 while(cap.isOpened()):
10     ret, opencv_image = cap.read()
11
12     color_cverted = cv2.cvtColor(opencv_image, cv2.COLOR_BGR2RGB)
13     pil_image=Image.fromarray(color_cverted)
14
15     pil_image.paste(pil_logo, (0,0),pil_logo)
16
17     numpy_image=np.array(pil_image)
18     image=cv2.cvtColor(numpy_image, cv2.COLOR_RGB2BGR)
19
20     cv2.imshow('frame',image)
21     if cv2.waitKey(1) == ord('q'):
22         break
23
24 cap.release()
25 cv2.destroyAllWindows()

```



← Capture

YouTube Link (หรือลิงค์ที่ฝากวิดีโอ อื่นๆ) ของงานที่ทำได้

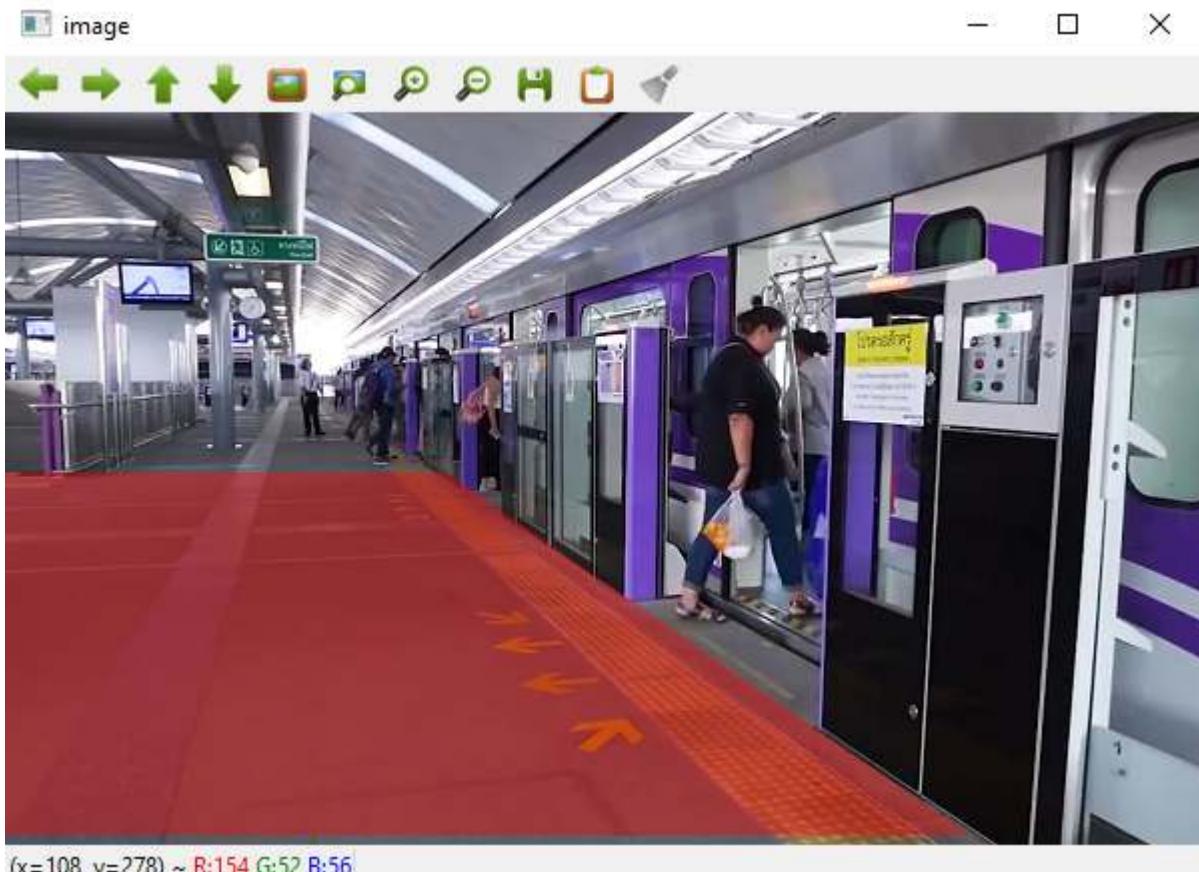
Lab106_การแทรกรูปทรงหลายเหลี่ยมด้วย fillPoly

19. รูปทรงหลายเหลี่ยมด้วย fillPoly

การปรับค่าความโปร่งใสโดยใช้ค่า alpha และใช้คำสั่ง
addWeighted() เพื่อปรับให้ภาพเกิดการซ้อนกัน

```

1 # Insert fillPoly to Picture
2 import cv2
3 import numpy as np
4 img = cv2.imread('./image/thai-subway.png')
5 overlay = img.copy()
6 pts = np.array([[0,360],[0,180],[220,180],[480,360]], np.int32)
7 pts = pts.reshape((-1, 1, 2))
8 cv2.fillPoly(overlay, [pts], (0,0,255))
9 alpha = 0.4
10 image_new = cv2.addWeighted(overlay, alpha, img, 1 - alpha, 0)
11 cv2.imshow('image',image_new)
12 cv2.waitKey(0)
13 cv2.destroyAllWindows()
```



20. การวัดรูปทรงแบบโปร่งใสบนวีดีโอ

```
1 # Insert fillPoly to Video
2 import cv2
3 import numpy as np
4 cap = cv2.VideoCapture('./image/video.avi')
5 while(cap.isOpened()):
6     ret, frame = cap.read()
7     overlay = frame.copy()
8     pts = np.array([[80, 180],[180,100],[310,100],[290, 180]], np.int32)
9     pts = pts.reshape((-1, 1, 2))
10    cv2.fillPoly(overlay, [pts], (0,0,255))
11    alpha = 0.4
12    imageOverlay = cv2.addWeighted(overlay, alpha, frame, 1 - alpha, 0)
13    cv2.imshow('image', imageOverlay)
14    if cv2.waitKey(1) & 0xFF == ord('q'):
15        break
16 cap.release()
17 cv2.destroyAllWindows()
```

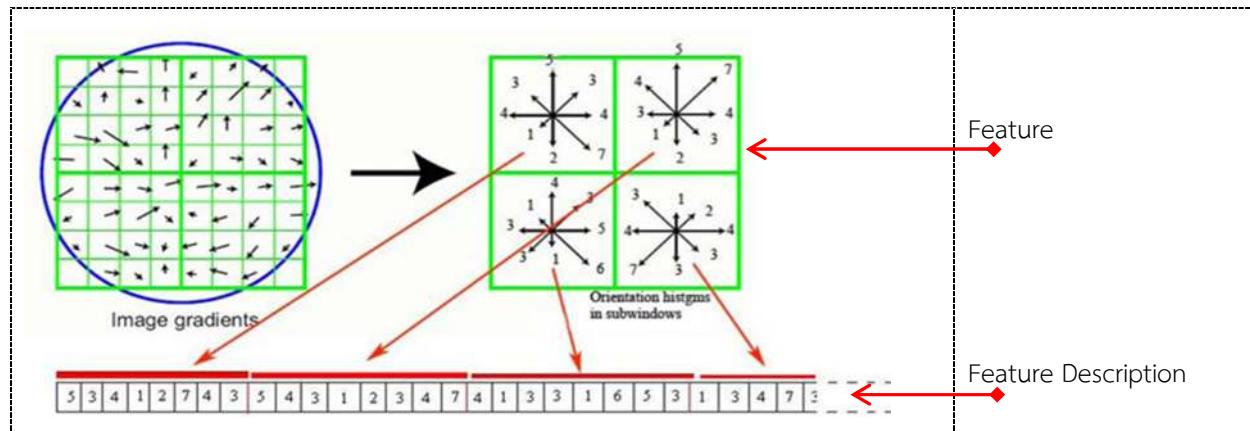


2/5 – เครื่องมือทั่วไปที่ใช้ในการประมวลผลภาพ

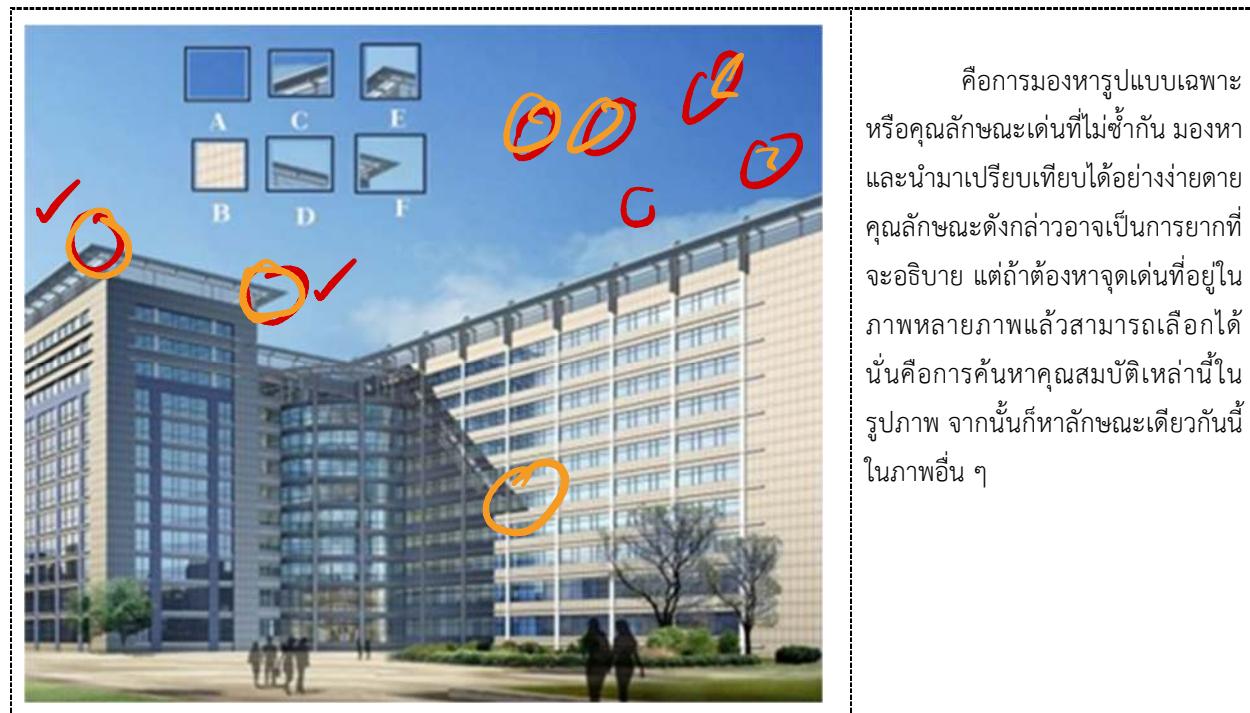
1. การวิเคราะห์คุณลักษณะเด่น (Feature Extraction)

วิธีการวิเคราะห์คุณลักษณะเด่น (Feature) และคำอธิบายคุณลักษณะเด่น (Feature Description) ซึ่งประกอบไปด้วยขั้นตอนและวิธีการต่าง ๆ เช่น การตรวจจับมุม การเพิ่มประสิทธิภาพ การเพิ่มความเร็วในการตรวจจับคุณลักษณะ การนำคุณลักษณะที่ได้ไปใช้ในการจับคู่เพื่อการค้นหาสิ่งที่ต้องการภายในภาพอื่น ๆ

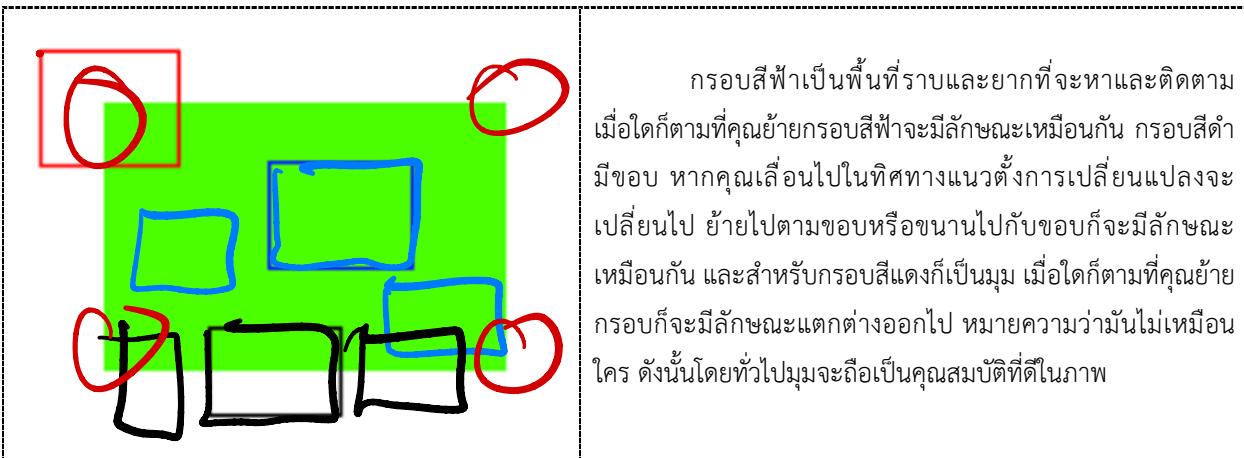
2. Feature และ Feature Description



3. การมองหาลักษณะเด่น



4. วิธีการอย่างง่ายในการหาคุณลักษณะเด่น



Lab201- Canny Edge Detection

5. การหาขอบภาพด้วย Canny Edge Detection

```
1 import cv2
2 myName = 'B3601234 Wichai Srisuruk'
3 img = cv2.imread('./image/lena.jpg',0)
4
5 #edges = cv2.Canny (img, 50, 100, apertureSize=3, L2gradient=True)
6 edges = cv2.Canny (img, 50, 100, apertureSize=3)
7 cv2.imshow('Edges-' +myName, edges)
8
9 cv2.waitKey (0)
10 cv2.destroyAllWindows()
```

Minimum intensity gradient

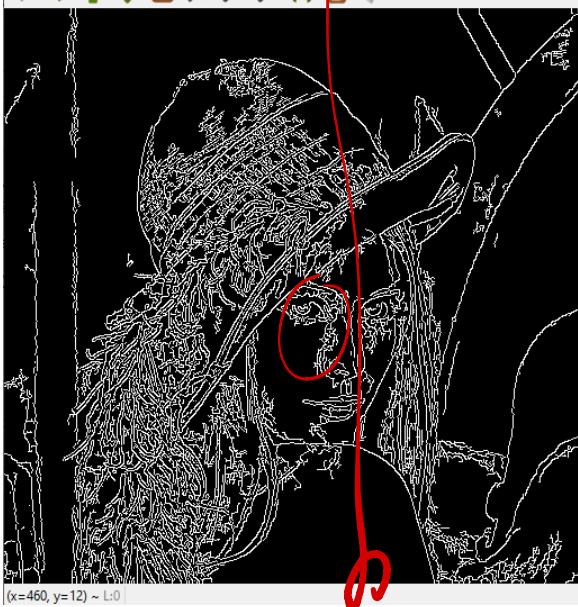
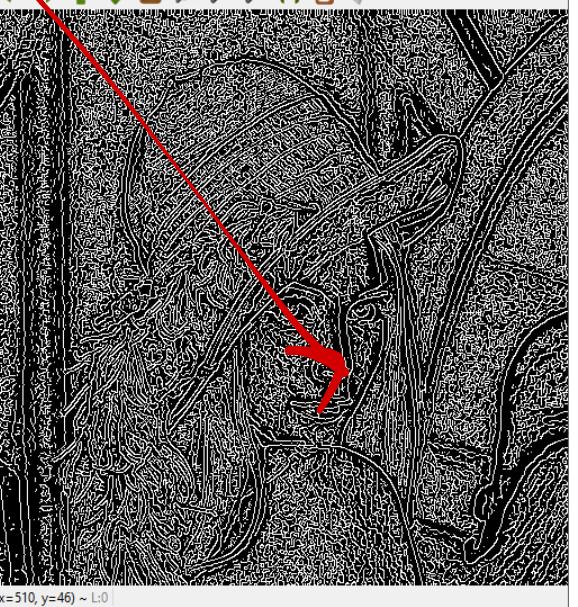
Maximum intensity gradient

edges = cv2.Canny(img,100, 200, apertureSize=3,L2gradient=True)

finding gradient magnitude

Filter Size

{50,100}

| | |
|---|--|
|  |  |
| <pre>edges = cv2.Canny (img, 50, 100, apertureSize=3, L2gradient=True)</pre> | |
|  |  |
| <pre>edges = cv2.Canny (img, 10, 100, apertureSize=3, L2gradient=True)</pre> | |

กิจกรรมที่ 2/6 – Canny Edge Detection from Webcam to Video with Logo

จงเขียนโปรแกรมเพื่ออ่านภาพวิดีโอจากสตรีมวิดีโอของกล้องเวปแคม และ Capture เพื่อทำ Canny Edge Detection

< Code > - Step 1

```

1 # Mission-1/5 Step-1: Test Video Stream
2 import cv2
3 import datetime
4 cap = cv2.VideoCapture(1)
5 while True:
6     ret, image = cap.read()
7     cv2.imshow('Image Show', image)
8     pressedKey = cv2.waitKey(1)
9     if pressedKey == ord('q'):
10         break
11     elif pressedKey == ord('c'):
12         now = datetime.datetime.now().strftime ('%y%d%m_%H%M%S')
13         cv2.imwrite('./save_'+str(now)+'.jpg', image)
14 cap.release()
15 cv2.destroyAllWindows ()
16

```

→ ต่อ Video CAM

C

→ บันทึก Capture

< Code > - Step 2 (Finish)

```

1 # Mission-1/5 Step-2: Edge
2 import cv2
3 myName = 'B3601234 Wichai Srisuruk'
4 cap = cv2.VideoCapture(1)
5 while True:
6     ret, colorImg = cap.read()
7     cv2.imshow('Image Show', colorImg)
8     pressedKey = cv2.waitKey(1)
9     if pressedKey == ord('q'):
10         break
11     elif pressedKey == ord('c'):
12         edges = cv2.Canny (colorImg, 50, 100, apertureSize=3, L2gradient=True)
13         cv2.imshow('Original-' + myName, colorImg)
14         cv2.imshow('Edge-' + myName, edges)
15
16 cap.release()
17 cv2.destroyAllWindows ()
18

```

C

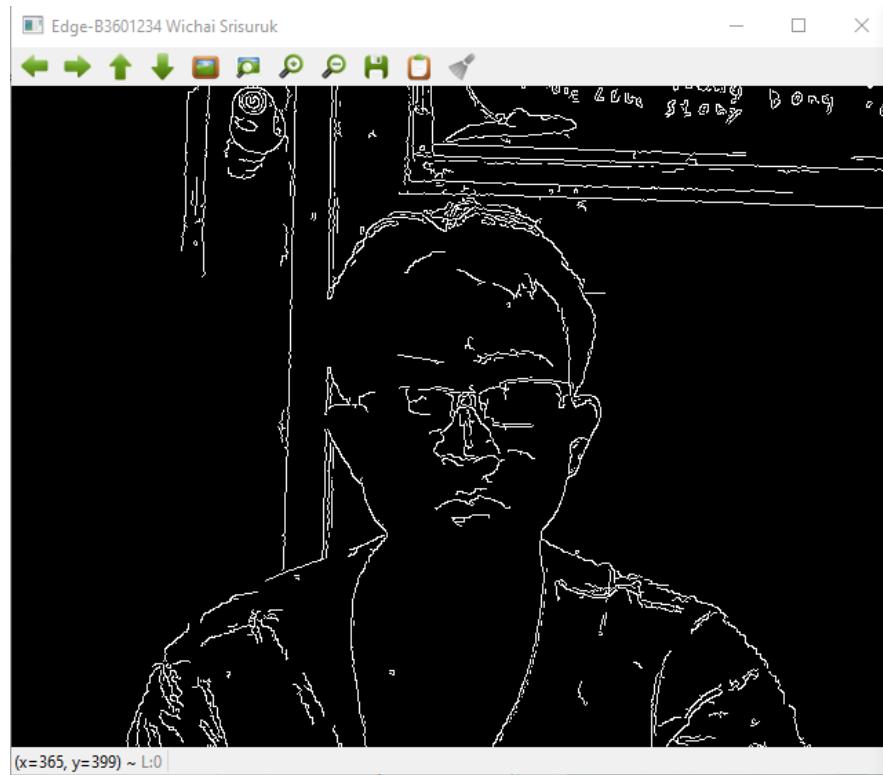
→ บันทึก Canny.

→ รูป Canny

< picture input >



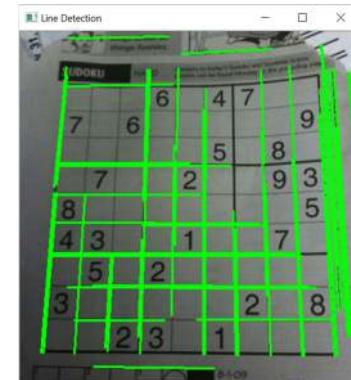
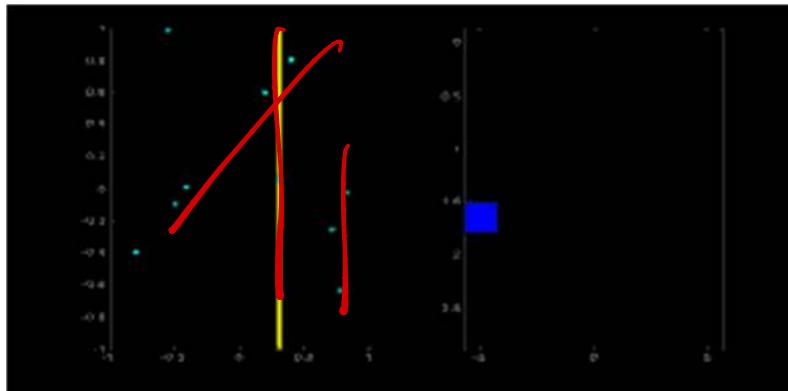
< Edge output >



Lab202- Hough Transform

6. การตรวจจับเส้นด้วย Hough Transform

Hough Transform เป็นเทคนิคยอดนิยมในการตรวจจับรูปทรง ที่นี่หากสามารถแสดงรูปร่างนั้นในรูปแบบทางคณิตศาสตร์ สามารถตรวจจับรูปร่างได้แม้จะหักหรือบิดเบี้ยวเล็กน้อยก็ตาม



7. การตรวจจับเส้นด้วย Probabilistic Hough Line

```
lines = cv2.HoughLinesP(edges, 1, np.pi/180, 100, minLineLength=100, maxLineGap=10)
```

Resolution : ความละเอียดระยะห่างของตัวสะสมเป็นพิกเซล

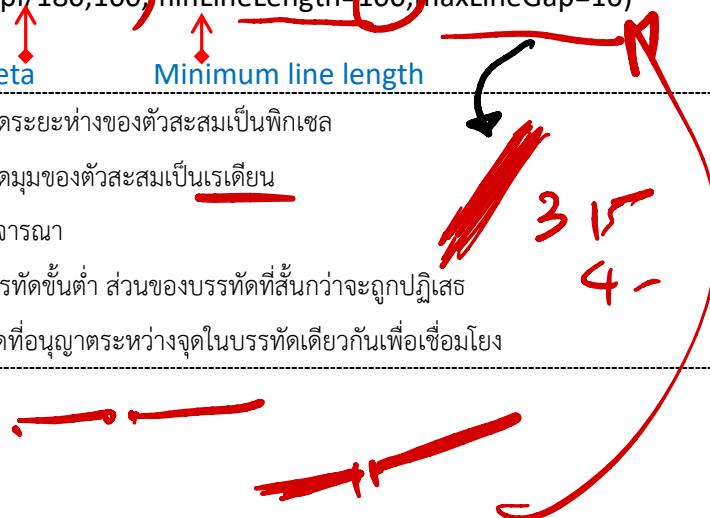
Theta : ความละเอียดมุมของตัวสะสมเป็นเรเดียน

Threshold : เกณฑ์การพิจารณา

Minimum line length : ความยาวบรรทัดขั้นต่ำ ส่วนของบรรทัดที่สั้นกว่าจะถูกปฏิเสธ

Maximum allowed gap : ช่องว่างสูงสุดที่อนุญาตระหว่างจุดในบรรทัดเดียวกันเพื่อเชื่อมโยง

- Resolution : ความละเอียดระยะห่างของตัวสะสมเป็นพิกเซล
- Theta : ความละเอียดมุมของตัวสะสมเป็นเรเดียน
- Threshold : เกณฑ์การพิจารณา
- Minimum line length : ความยาวบรรทัดขั้นต่ำ ส่วนของบรรทัดที่สั้นกว่าจะถูกปฏิเสธ
- Maximum allowed gap : ช่องว่างสูงสุดที่อนุญาตระหว่างจุดในบรรทัดเดียวกันเพื่อเชื่อมโยง

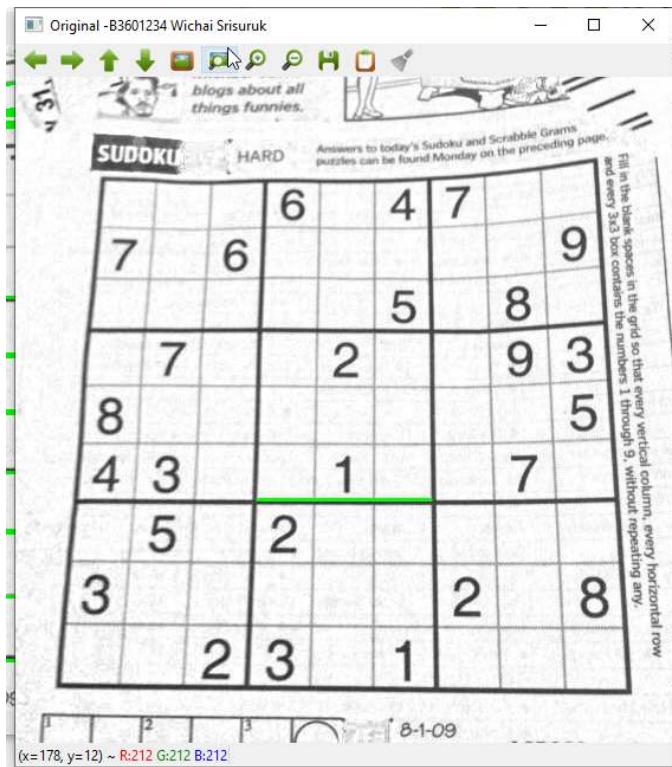


8. การตรวจจับเส้นด้วย Hough Line Transform

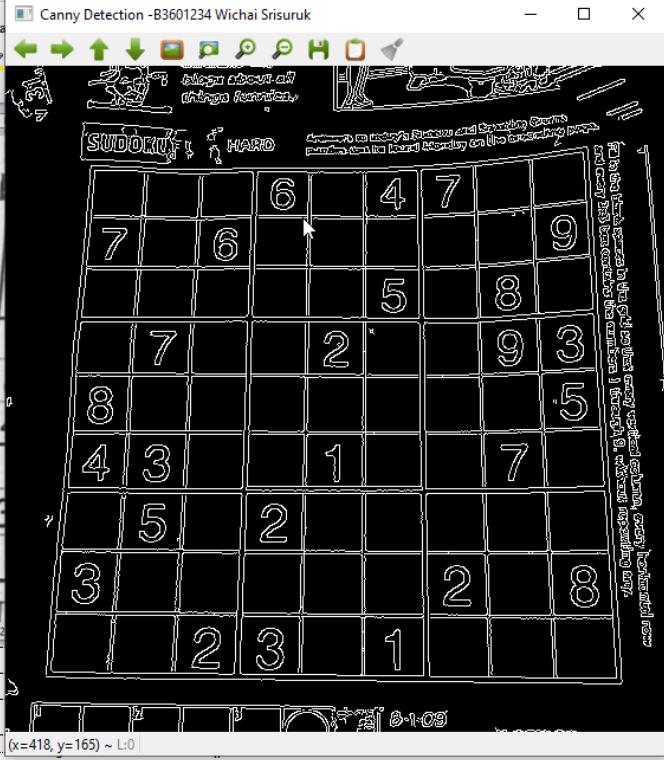
```

1 # Hough Line Transform
2 import cv2
3 import numpy as np
4 myName = 'B3601234 Wichai Srisuruk'
5
6 img = cv2.imread('./image/sudoku.jpg')
7 cv2.imshow('Original - ' + myName, img)
8
9 gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
10 edges = cv2.Canny(gray, 50, 150)
11 lines = cv2.HoughLinesP(edges, 1, np.pi/180, 100, minLineLength=100, maxLineGap=10)
12 for line in lines:
13     x1,y1,x2,y2 = line[0]
14     cv2.line(img, (x1,y1), (x2,y2), (0,255,0), 2)
15
16 cv2.imshow('Canny Detection - ' + myName, edges)
17 cv2.imshow('Line Detection - ' + myName, img)
18 cv2.waitKey(0)
19 cv2.destroyAllWindows()
20

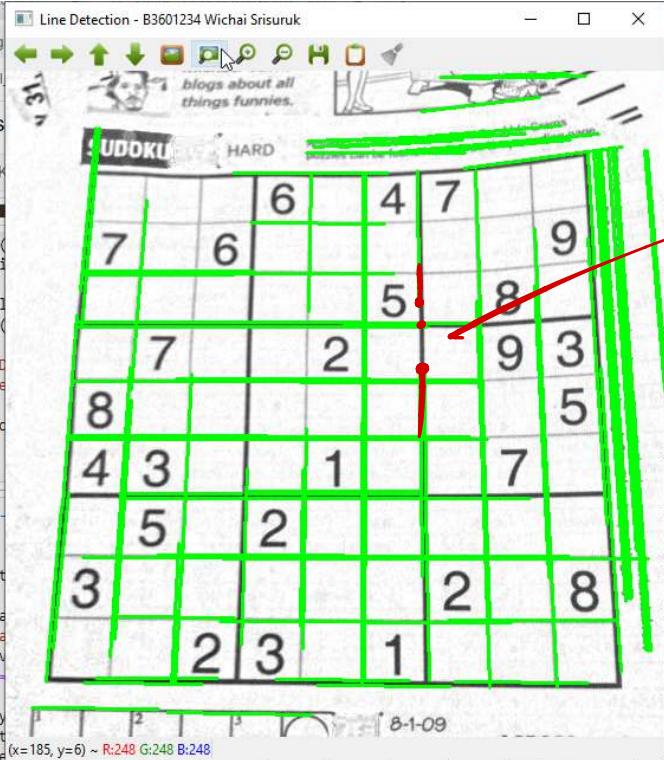
```



Original



Edge Detection



Hough Line Transform

Low Gap = 10
100

Lab203- Hough Circle

9. การตรวจจับวงกลมด้วย Hough Circle

| gap | Minimum distance | Param2 | Maximum allowed |
|-----|--|---|-----------------|
| | circles = cv2.HoughCircles(img, cv2.HOUGH_GRADIENT, 1, 100, 100, 30, minRadius=0, maxRadius=0) | ↓ Inverse ratio ↑ Param1 ↓ Minimum line length | |

- Inverse ratio : อัตราส่วนผกผันของความละเอียดของตัวสะสหมกับความละเอียดของภาพ
- Minimum distance : ระยะห่างขั้นต่ำระหว่างศูนย์กลางของวงกลมที่ตรวจสอบ
- Param1 : Higher threshold เกณฑ์การพิจารณาสำหรับตรวจจับเส้นขอบ
- Param2 : Accumulator threshold หรือ เกณฑ์การสะสม
- Minimum circle radius : ความยาวของรัศมีขั้นต่ำ ส่วนของรัศมีที่สั้นกว่าจะถูกปฏิเสธ
- Maximum circle radius : ช่องว่างสูงสุดที่อนุญาตระหว่างจุดในบรรทัดเดียวกันเพื่อเข้มโถง

10. คำสั่งการตรวจจับ cv2.circle()

Python OpenCV → cv2.circle() method

OpenCV-Python is a library of Python bindings designed to solve computer vision problems. cv2.circle() method is used to draw a circle on any image.

Syntax: cv2.circle(image, center_coordinates, radius, color, thickness)

Parameters:

- image: It is the image on which circle is to be drawn.
- center_coordinates: It is the center coordinates of circle. The coordinates are represented as tuples of two values i.e. (X coordinate value, Y coordinate value).
- radius: It is the radius of circle.
- color: It is the color of border line of circle to be drawn. For BGR, we pass a tuple. eg: (255, 0, 0) for blue color.
- thickness: It is the thickness of the circle border line in px. Thickness of -1 px will fill the circle shape by the specified color.

Return Value: It returns an image.

11. คำสั่งการทำภาพเบลอแบบ Median Blurring ด้วยคำสั่ง cv.medianBlur()

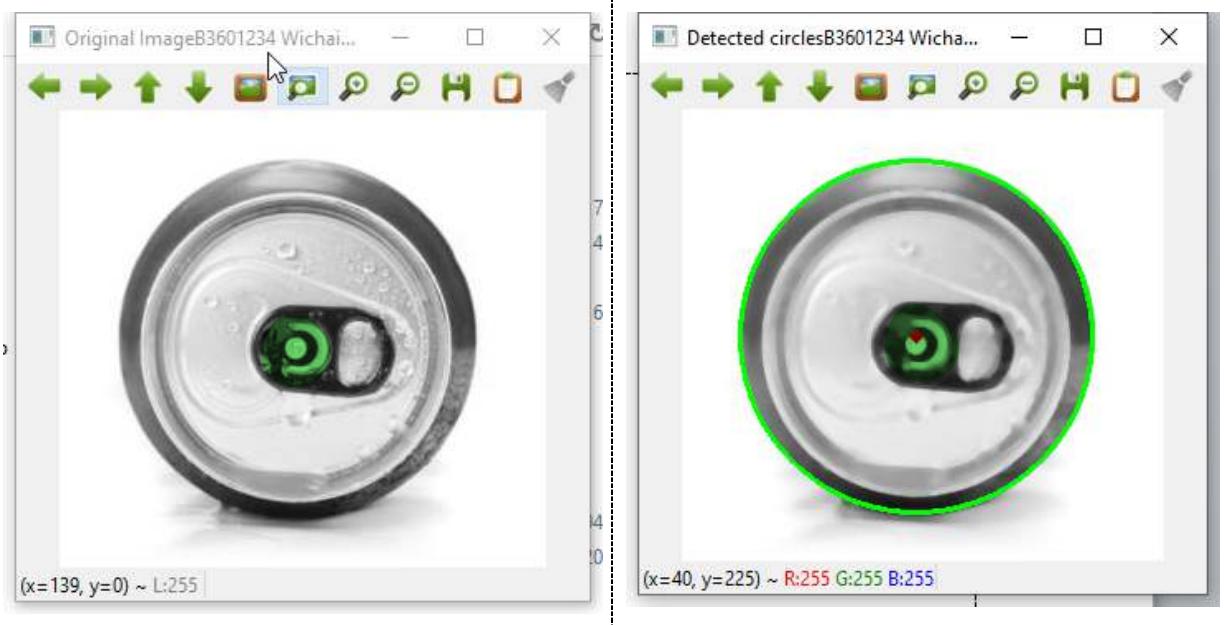
- <https://phyblas.hinaboshi.com/oshi08>
- https://docs.opencv.org/master/d4/d13/tutorial_py_filtering.html

12. การตรวจจับวงกลมด้วย Hough Circle Transform

```

1 # Hough Circle Transform
2 import cv2
3 import numpy as np
4 myName = 'B3601234 Wichai Srisuruk'
5 img = cv2.imread('./image/can.jpg',0)
6 cv2.imshow ('Original Image' + myName, img)
7
8 img = cv2.medianBlur(img,5)
9 cimg = cv2.cvtColor(img,cv2.COLOR_GRAY2BGR)
10 circles = cv2.HoughCircles(img,cv2.HOUGH_GRADIENT,1,100, param1=100, param2=30, minRadius=0,maxRadius=0)
11 circles = np.uint16(np.around(circles))
12 for i in circles[0,:]:
13     cv2.circle(cimg, (i[0],i[1]),i[2],(0,255,0),2)
14     cv2.circle(cimg, (i[0],i[1]),2 ,(0,0,255),3)
15
16 cv2.imshow ('Detected circles' + myName,cimg)
17 cv2.waitKey (0)
18 cv2.destroyAllWindows()
19

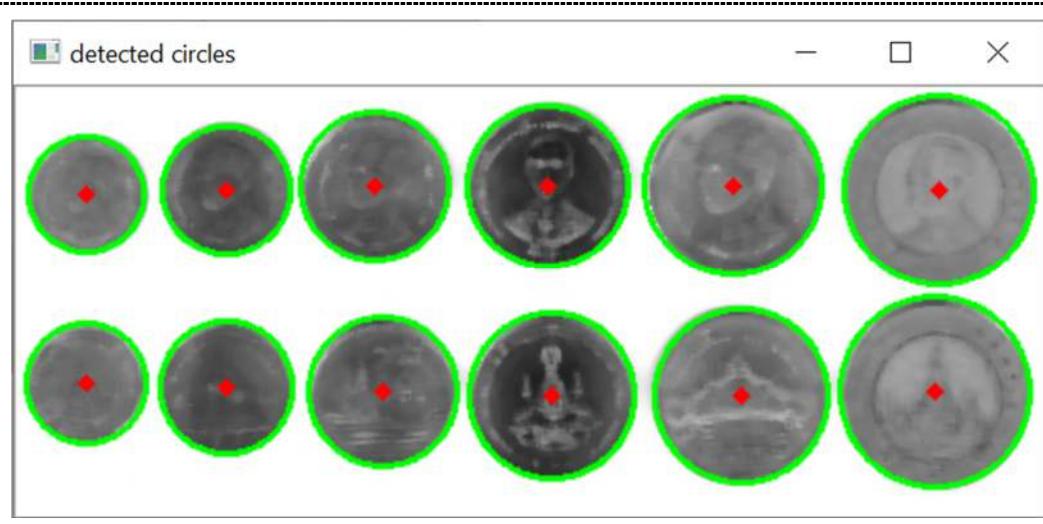
```



กิจกรรมที่ 3/6 – ทำการตรวจจับเหรียญด้วย Hough Circle Transform

ให้ถ่ายรูปเหรียญของตัวเอง จำนวน 12 เหรียญ เมื่อันตัวอย่าง แล้วทดสอบทำ Hough Circle Transform

< Code > - Step 1



Lab204- Optical Flow

13. การไหลแบบอพติคอล (Optical Flow)

การเคลื่อนที่ของวัตถุภาพระหว่างสองเฟรมต่อเนื่องที่เกิดจากการเคลื่อนย้ายวัตถุหรือกล้องถ่ายภาพ เป็นเวกเตอร์แบบสองมิติ ซึ่งแต่ละเวกเตอร์เป็นเวกเตอร์ที่แสดงการเคลื่อนที่ของจุดจากเฟรมแรกไปเฟรมที่สอง จากความเข้มของจุดภาพของวัตถุที่ไม่เปลี่ยนแปลงระหว่างเฟรมติดต่อกัน จุดภาพที่อยู่ใกล้เคียงมีการเคลื่อนไหวที่คล้ายคลึงกัน เบื้องต้นจะเป็นการตั้งสมมติฐานก่อนหน้านี้ว่าจุดภาพที่อยู่ใกล้เคียงทั้งหมดจะมีการเคลื่อนไหวที่คล้ายคลึงกัน การวิเคราะห์การไหลของแสงแบบเบาบาง ใช้หน้าต่าง 3×3 ดังนั้นจะเป็นการพิจารณาจุดมีการเคลื่อนไหว โดยทำการหาค่าความเข้มของจุดภาพสำหรับ 9 จุดเหล่านี้



14. การไหลแบบอพติคอล (Optical Flow)



```

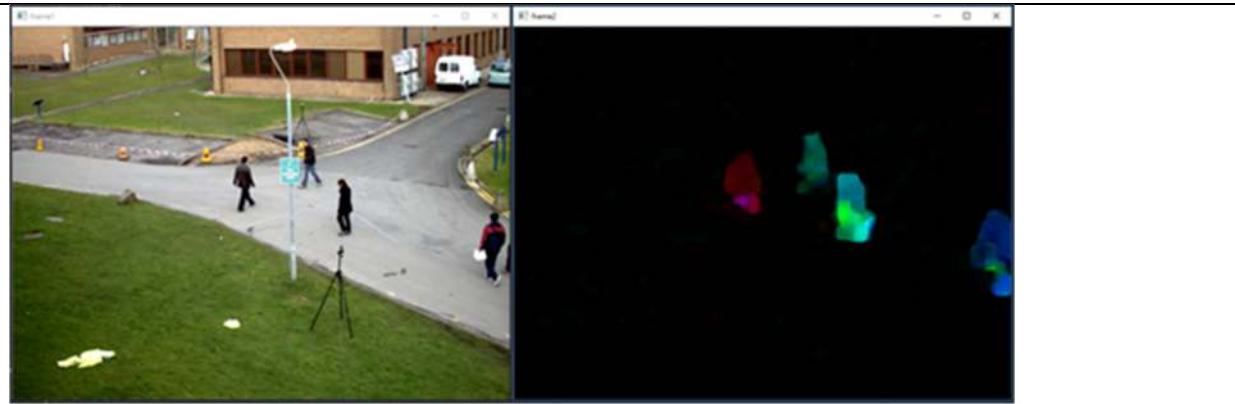
1  # Optical Flow
2  import numpy as np
3  import cv2
4  myName = 'B3601234 Wichai Srisuruk'
5  cap = cv2.VideoCapture('./image/move.avi')
6  feature_params = dict(maxCorners=100,qualityLevel=0.3,minDistance=7,blockSize=7)
7  lk_params = dict( winSize  = (15,15),maxLevel = 2,
8                   criteria = (cv2.TERM_CRITERIA_EPS | cv2.TERM_CRITERIA_COUNT, 10, 0.03))
9  color = np.random.randint(0,255,(100,3))
10 color = np.random.randint(0,255,(100,3))
11 ret, old_frame = cap.read()
12 old_gray = cv2.cvtColor(old_frame, cv2.COLOR_BGR2GRAY)
13 p0 = cv2.goodFeaturesToTrack(old_gray, mask = None, **feature_params)
14 mask = np.zeros_like(old_frame)
15 while(1):
16     ret,frame = cap.read()
17     frame_gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
18     p1, st, err = cv2.calcOpticalFlowPyrLK(old_gray, frame_gray, p0, None, **lk_params)
19     good_new = p1[st==1]
20     good_old = p0[st==1]
21     for i,(new,old) in enumerate(zip(good_new,good_old)):
22         a,b = new.ravel()
23         c,d = old.ravel()
24         a,b,c,d = int(a),int(b),int(c),int(d)
25         mask = cv2.line(mask, (a,b),(c,d), color[i].tolist(), 2)
26         frame = cv2.circle(frame,(a,b),5,color[i].tolist(),-1)
27     img = cv2.add(frame,mask)
28     cv2.imshow('OpticalFlow-' + myName, img)
29     k = cv2.waitKey(30) & 0xff
30     if k == 27:      # Esc Key for Exit
31         break
32     old_gray = frame_gray.copy()
33     p0 = good_new.reshape(-1,1,2)
34
35 cap.release()
36 cv2.destroyAllWindows()
37
38

```

Lab205- Optical Density Flow

15. การวิเคราะห์การไหลของแสงแบบหนาแน่น

การวิเคราะห์การไหลของแสงแบบหนาแน่น จะคำนวณการไหลของแสงทุกจุดในเฟรม โดยประมาณการเคลื่อนไหวสองเฟรมตามการขยายตัวของพื้นที่ ตัวอย่างด้านล่างแสดงวิธีค้นหาการไหลของแสงแบบหนาแน่นด้วยอาร์เรย์ 2 ช่องที่มีเวกเตอร์การไหลของแสง โดยทางขนาดและทิศทาง เพื่อแสดงเป็นโค๊ดสี

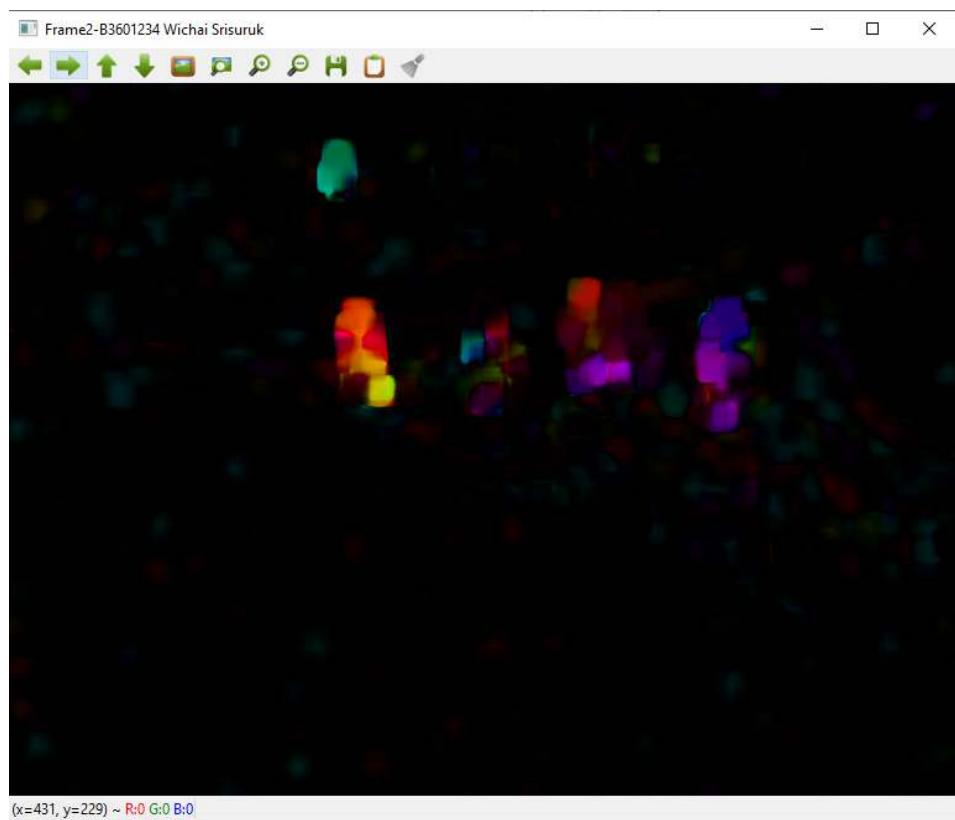
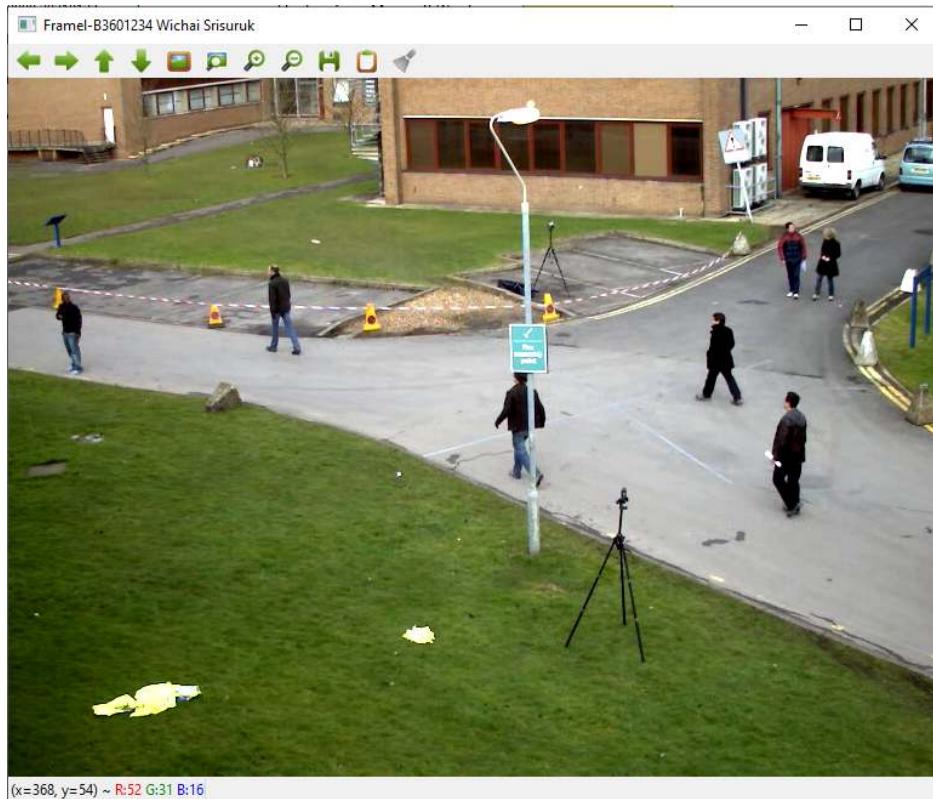


16. การวิเคราะห์การไหลของแสงแบบหนาแน่น

```

1 # Optical Density Flow
2 import cv2
3 import numpy as np
4 myName = 'B3601234 Wichai Srisuruk'
5 cap = cv2.VideoCapture('./image/move.avi')
6 ret, frame1 = cap.read()
7 prvs = cv2.cvtColor(frame1, cv2.COLOR_BGR2GRAY)
8 hsv = np.zeros_like(frame1)
9 hsv[:, :, 1] = 255
10
11 while(1):
12     ret, frame2 = cap.read()
13     next = cv2.cvtColor(frame2, cv2.COLOR_BGR2GRAY)
14     flow = cv2.calcOpticalFlowFarneback(prvs, next, None, 0.5, 3, 15, 3, 5, 1.2, 0)
15     mag, ang = cv2.cartToPolar(flow[:, :, 0], flow[:, :, 1])
16     hsv[:, :, 0] = ang * 180 / np.pi / 2
17     hsv[:, :, 2] = cv2.normalize(mag, None, 0, 255, cv2.NORM_MINMAX)
18     bgr = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
19     cv2.imshow('Frame1-' + myName, frame1)
20     cv2.imshow('Frame2-' + myName, bgr)
21     k = cv2.waitKey(30) & 0xff
22     if k == 27: # ECS for Exit
23         break
24     prvs = next
25 cap.release()
26 cv2.destroyAllWindows()
27

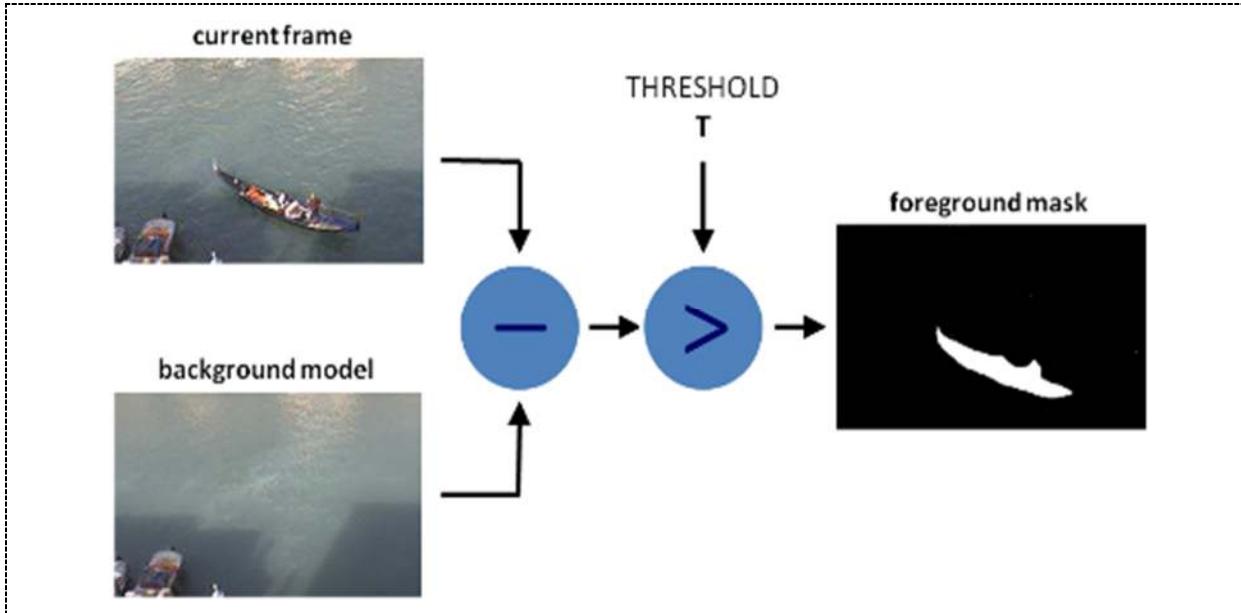
```



Lab206 - การแยกส่วนพื้นหลังกับเบื้องหน้า

17. การแยกส่วนพื้นหลังกับเบื้องหน้า

การลบพื้นหลังของภาพเป็นขั้นตอนการประมวลผลเบื้องต้นที่สำคัญ เช่น กรณีพิจารณาจำนวนผู้เข้าหรือออกจากห้อง การใช้กล้องจราจรแยกข้อมูลเกี่ยวกับยานพาหนะ กรณีเหล่านี้ต้องแยกพื้นหลังที่เคลื่อนที่จากพื้นหลังแบบคงที่



18. การแยกส่วนพื้นหลังกับเบื้องหน้า

```

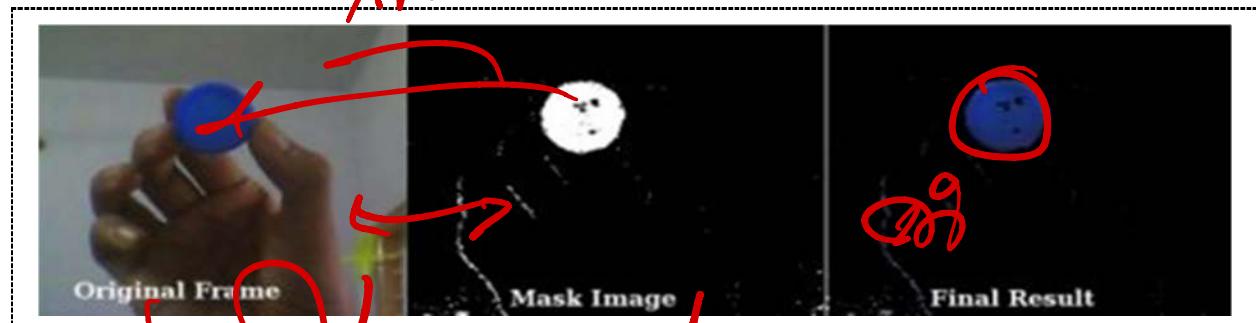
1 # Background-Subtraction
2 import cv2
3 import numpy as np
4 myName = 'B3601234 Wichai Srisuruk'
5 cap = cv2.VideoCapture('./image/move.avi')
6
7 fgbg = cv2.createBackgroundSubtractorMOG2() ← การสร้างภาพ พื้น
8
9 while(True):
10     ret, frame = cap.read()
11     if ret == True:
12         fgmask = fgbg.apply(frame) ← การซ้อนทับกับชั้นที่
13         cv2.imshow('Background-Subtraction'+myName, fgmask) ← เป็น Mask
14         k = cv2.waitKey(30)
15         if k == 27: # กด ESC
16             break
17     else:
18         break
19 cap.release()
20 cv2.destroyAllWindows()
21
  
```



Lab207 – Color Detection

19. การตรวจวัดถูกจากการจับค่าสี Color Detection

การพิจารณาค่าสีภาพในภาพ นับเป็นอีกหนึ่งวิธีการที่นำมาใช้ในการติดตาม หรือตรวจจับวัตถุภายในภาพ ทั้งนี้การวิเคราะห์หาค่าสีที่สามารถนำพิจารณาการตรวจจับสี ต้องใช้ค่าสีที่มีรูปแบบเป็น HSV หรือ เฉดสี Hue range is [0,179] ค่าความอิ่มตัวของสี Saturation range is [0,255] และค่าความสว่าง Value range is [0,255].



20. ระบบมาตราแบบ HSV Image

ระบบสี HSV (Hue, Saturation, Value) หรือ ~~HSB~~ (Hue, Saturation, Brightness) เป็นระบบสีที่นิยมใช้กันในหมู่นักวิจัย เนื่องจากเป็นระบบสีที่ใกล้เคียงกับความคิดของมนุษย์ได้ดีกว่าระบบสี RGB โดย Hue คือสีของภาพ, Saturation คือปริมาณความอิ่มตัวของสี ยิ่งมีค่าน้ำมาก ภาพจะมีสีสดใสยิ่ง มีร่องรอยของสีน้อยลง จนในที่สุดจะกลายเป็นรูปที่ลักษณะแบบ Grayscale และ Value หรือ Brightness เป็นค่าที่แสดงถึงปริมาณความสว่างของภาพ หากมีค่ามากภาพจะยิ่งมีความสว่างมาก ในการตัดพื้นหลังออกจากภาพใช้วิธีการแปลงค่าสีจาก RGB เป็น HSV และทำการหาช่วงของสีของพื้นหลัง

For HSV, Hue range is [0,179], Saturation range is [0,255] and Value range is [0,255]. Different softwares use different scales. So if you are comparing OpenCV values with them, you need to normalize these ranges.

- <http://fivedots.coe.psu.ac.th/~montri/Teaching/image/chap1.htm>
- <https://nextsoftwares.wordpress.com/2014/05/22/ความรู้เบื้องต้นเกี่ยวกับHSV/>

21. OpenCV - cv2.inRange() function

Thresholding using cv2.inRange() function

In the previous blogs, we discussed various thresholding methods such as Otsu, adaptive, BHT, etc. In this blog, we will learn how to segment out a particular region or color from an image. This is naively equivalent to multiple thresholding where we assign a particular value to the region falling in between the two thresholds. Remaining region is assigned a different value. OpenCV provides an inbuilt function for this as shown below

`cv2.inRange(src, lowerb, upperb)`

22. การตรวจจับสี Color Detection

```

1 #Color-Detection
2 import cv2
3 import numpy as np
4 myName = 'B3601234 Wichai Srisuruk'
5 cap = cv2.VideoCapture(1)
6
7 while(1):
8     ret, frame = cap.read()
9     hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
10    lower_blue = np.array([94, 80, 2])
11    upper_blue = np.array([126, 255, 255])
12    mask = cv2.inRange(hsv, lower_blue, upper_blue)
13    res = cv2.bitwise_and(frame, frame, mask= mask)
14    cv2.imshow('Frame-' +myName, frame)
15    cv2.imshow('Mask-' +myName, mask)
16    cv2.imshow('Res-' +myName, res)
17    k = cv2.waitKey(5)
18    if k == 27:
19        break
20 cap.release()
21 cv2.destroyAllWindows()
22
23 #    low_red = np.array([161, 155, 84])
24 #    high_red = np.array([179, 255, 255]) R
25
26 #    low_green = np.array([25, 52, 72])
27 #    high_green = np.array([102, 255, 255]) G
28
29 #    lower_blue = np.array([110, 50, 50])
30 #    upper_blue = np.array([130, 255, 255]) B
31

```

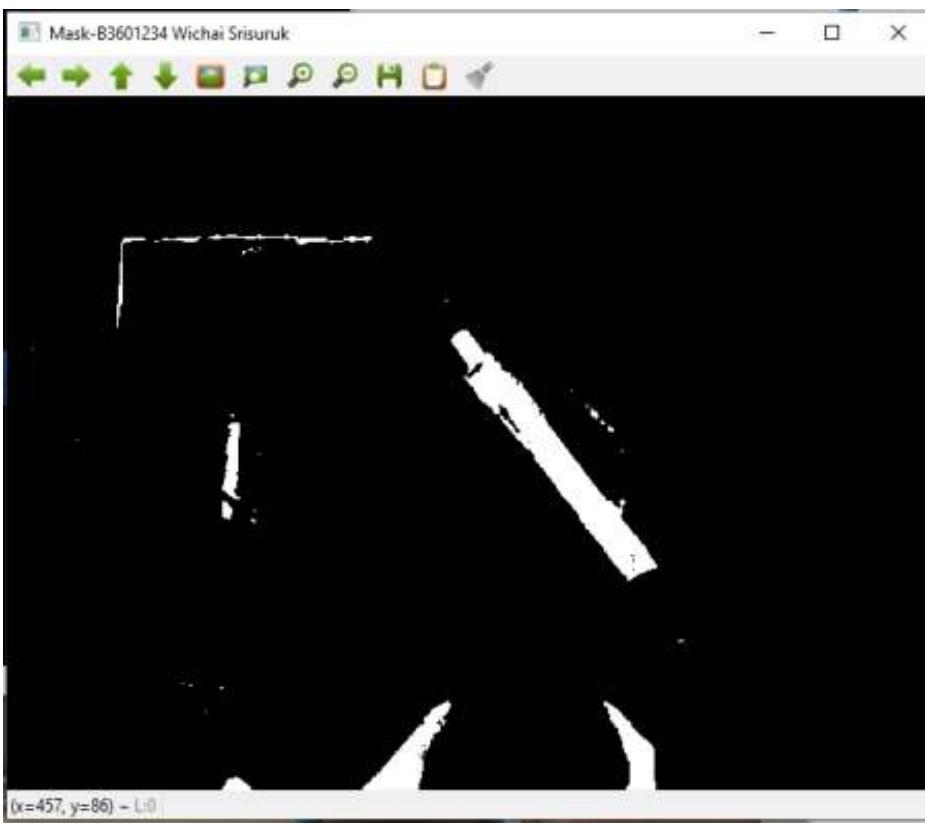
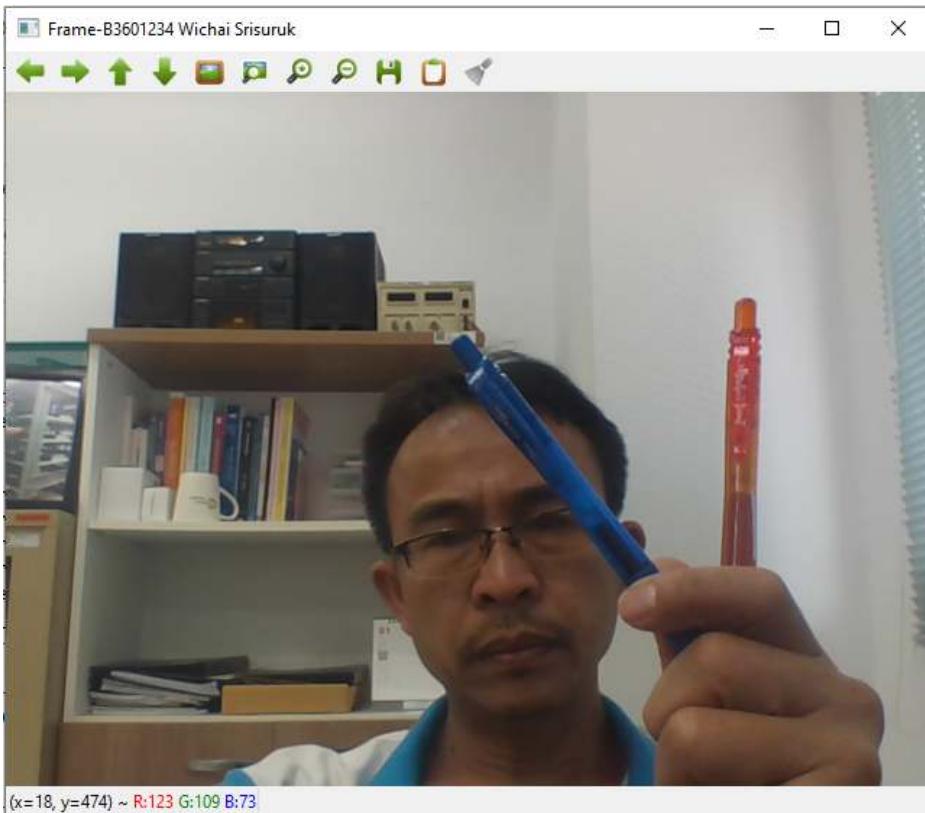
เปลี่ยนค่าสี BGR เป็น HSV

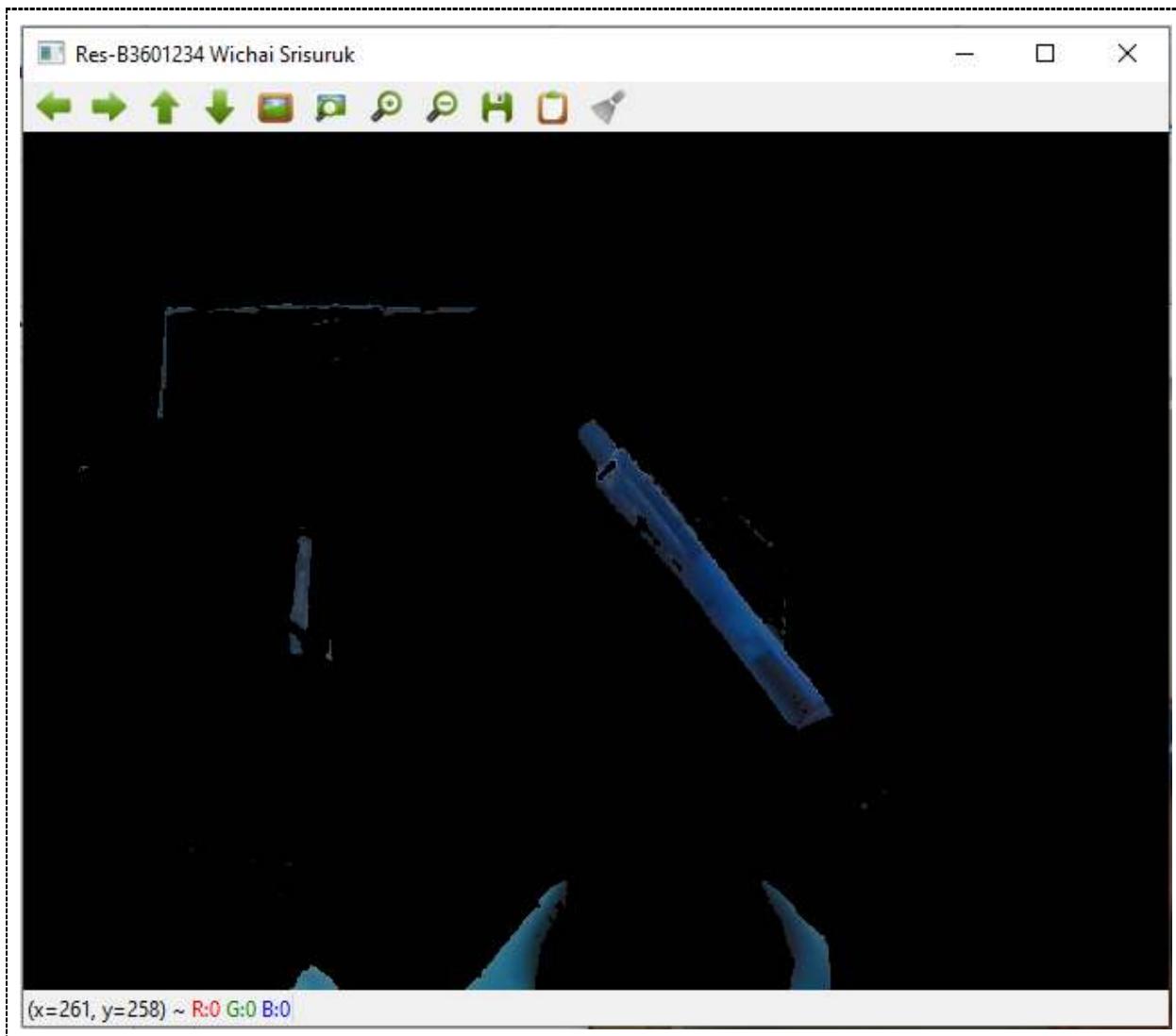
ค่าสีต่ำสุด {

ค่าสีสูงสุด }

การสร้างภาพที่เป็น Mask

การซ่อนทับกับชั้นที่เป็น Mask





กิจกรรมที่ 4/6 – จะทำการ detect ภาพจากกล้องเพื่อหาภาพโค้ดกระป๋อง(สีแดง)

จะเขียนโปรแกรมเพื่ออ่านภาพวิดีโอจากสตรีมวิดีโอด้วยของกล้องเวปแคมที่มีภาพโค้ดกับแบบสีแล้วทำการแสดงเฉพาะโค้ด

< Code > - Step 1

< Real Image >

< Result Image >

ตัวอย่างภาพทดสอบ



Lab208 - Image Overlays using Bitwise Operations

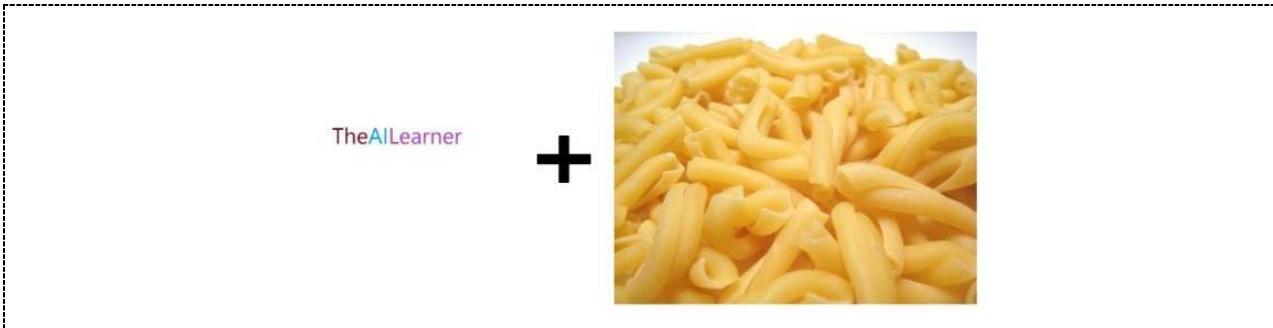
23. Image Overlays using Bitwise Operations OpenCV-Python

<https://theailearner.com/2019/03/26/image-overlays-using-bitwise-operations-opencv-python/>

In the previous blog, we learned how to overlay an image to another image using OpenCV cv2.addWeighted() function. But this approach is limited to rectangular ROI. In this blog, we will learn how to overlay non-rectangular ROI to another image.

Task:

Put the TheAIlearner text image(shown in the left) above an image (Right one).



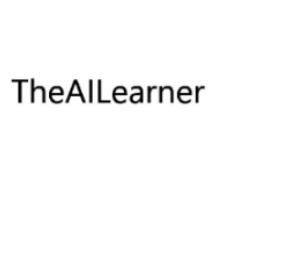
- Because the TheAIlearner text is non-rectangular, we will be using OpenCV cv2.bitwise_and(img1, img2, mask) where the mask is an 8-bit single channel array, that specifies elements of the output array to be changed.
- For Bitwise_and you need to know the following two rules

Black + Any Color = Black

White + Any Color = That Color

- Now, let's see step by step how to do this

| | |
|--|---|
| First load the two images | <pre>img1 = cv2.imread('D:/downloads/pasta_screen.jpg') img2 = cv2.imread('D:/downloads/logo.jpg')</pre> |
| Select the region in the image where you want to put the logo. Here, I am putting this in the top left corner. | <pre>rows,cols,channels = img2.shape roi = img1[0:rows, 0:cols]</pre> |
| Now, we will create a mask. You can create a mask by a number of ways but here we will be using thresholding for this as shown in the code below. We will also create an inverse mask. Depending on the image you need to change the thresholding function parameters. | <pre>img2gray = cv2.cvtColor(img2,cv2.COLOR_BGR2GRAY) ret, mask = cv2.threshold(img2gray, 200, 255, cv2.THRESH_BINARY_INV) mask_inv = cv2.bitwise_not(mask)</pre> |

| | | |
|---|--|---|
| The mask and mask_inv looks like this |  Mask |  Inverse Mask |
| Now black out the area of logo in the roi created above using the bitwise_and as shown in the code below | | <code>img1_bg = cv2.bitwise_and(roi,roi,mask = mask_inv)</code> |
| This looks like this |  | |
| Now, we will extract the logo region (with colors) from the logo image using the following code | | <code>img2_fg = cv2.bitwise_and(img2,img2,mask = mask)</code> |
| The output looks like this |  | |
| Now, we will simply add the above two images because black has intensity 0 so adding this doesn't change anything and outputs the same color. This is done using the following code | | <code>out_img = cv2.add(img1_bg,img2_fg) img1[0:rows, 0:cols] = out_img</code> |

The final output looks like this



- So, using these simple bitwise operations we can overlay an image to another. Be careful while creating the mask as it entirely depends on the image. According to the image you need to make adjustments to the thresholding function parameters.

มีวิธี

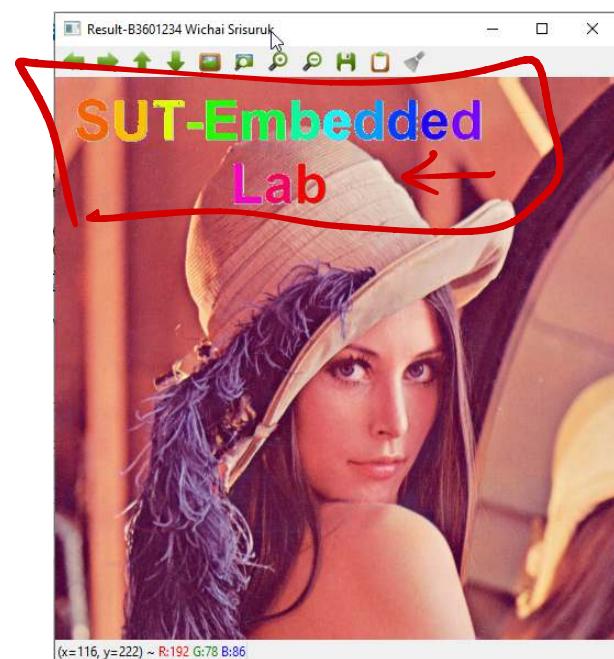
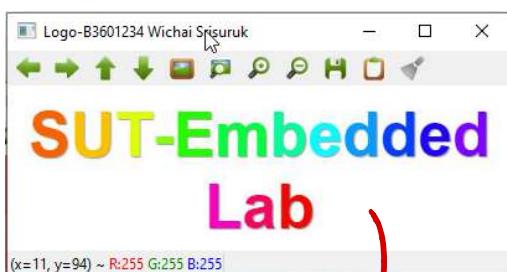
24. สร้างข้อความจาก <https://www.html-code-generator.com/html/rainbow-text-generator>

25. Image Overlays using Bitwise Operations OpenCV-Python

```

1 # Image Overlays using Bitwise Operations OpenCV-Python
2 import cv2
3 myName = 'B3601234 Wichai Srisuruk'
4 img1 = cv2.imread('./image/lena.jpg')
5 img2 = cv2.imread('./image/SEmb.jpg')4↙↙↙↙
6 cv2.imshow('Original-' + myName, img1)
7 cv2.imshow('Logo-' + myName, img2)
8
9 rows,cols,channels = img2.shape
10 roi = img1[0:rows, 0:cols] ↑↑↑↑
11
12 img2gray = cv2.cvtColor(img2,cv2.COLOR_BGR2GRAY)
13 ret, mask = cv2.threshold(img2gray, 220, 255, cv2.THRESH_BINARY_INV)
14 mask_inv = cv2.bitwise_not(mask)
15
16 img1_bg = cv2.bitwise_and(roi,roi,mask = mask_inv) AND
17 img2_fg = cv2.bitwise_and(img2,img2,mask = mask)
18 out_img = cv2.add(img1_bg,img2_fg)
19 img1[0:rows, 0:cols ] = out_img
20
21 cv2.imshow('Result-' + myName, img1)
22
23 cv2.waitKey (0)
24 cv2.destroyAllWindows()

```



กิจกรรมที่ 5/6 – Graphic Text

ใช้ภาพถ่ายของตัวเอง สร้างข้อความ และเติมข้อความในภาพถ่าย บุคลากรหัวเมือง

< Code > - Step 1

< Real Image >

< Result Image >

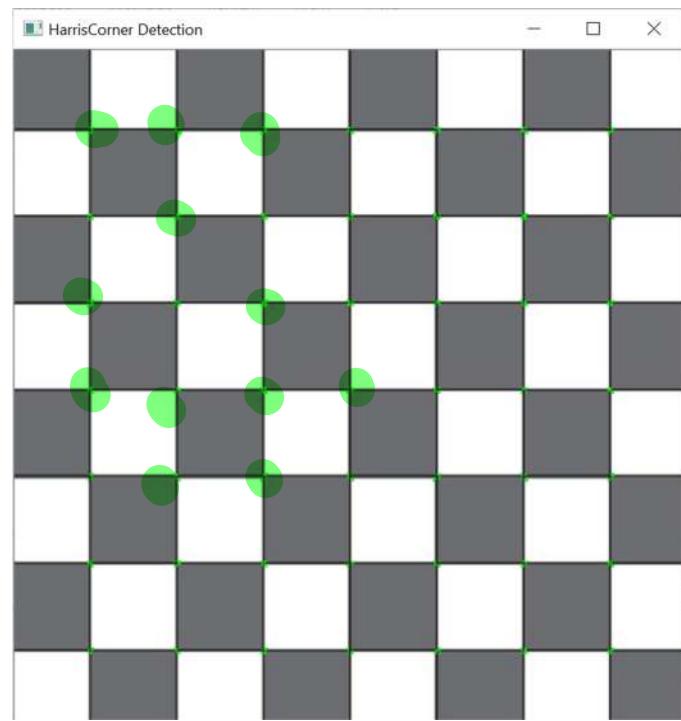
3/5 – การค้นหาวัตถุด้วยการจับคู่ระหว่างจุดสำคัญ (Keypoint Matching)

การค้นหาวัตถุหรือสิ่งที่ต้องการภายในภาพ ทำได้ด้วยการจับคู่ระหว่างจุดสำคัญ (Keypoint Matching) ของภาพทั้งสองภาพว่าตรงกันหรือไม่ โดยพิจารณาจากพื้นที่ใกล้เคียง โดยใช้อัตราส่วนระยะใกล้เคียงกับระยะทางที่ใกล้ที่สุด หากอัตราส่วนมีค่ามากกว่าที่จะยอมรับได้ ก็จะถูกปฏิเสธ ซึ่งมีความหมายว่าไม่สามารถจับคู่กันได้ หรือไม่สามารถหาสิ่งที่ต้องการภายในภาพได้

Lab301 - Harris Corner Detection

1. การตรวจจับมุมแวร์ริส (Harris Corner Detection)

มุมเป็นคุณลักษณะเด่นที่ดีสำหรับใช้ในการตรวจจับคุณลักษณะภายในภาพ ซึ่งถูกค้นพบโดยคริสแวร์ริสและไมค์สตีเฟนส์ OpenCV ได้มีการเตรียมฟังก์ชัน `cv2.cornerHarris()` สำหรับใช้ในการตรวจจับมุมแวร์ริส การตรวจจับมุม



```

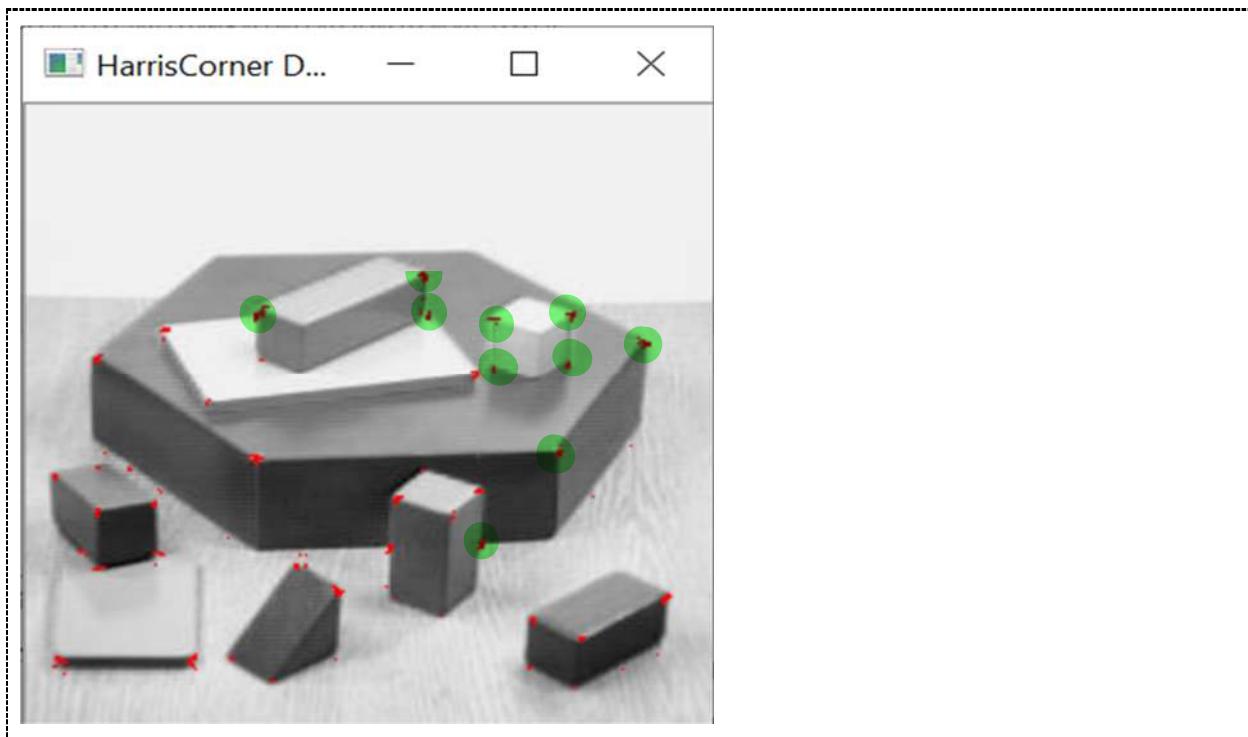
import cv2
import numpy as np
img = cv2.imread('./image/blox.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
gray = np.float32(gray)

dst = cv2.cornerHarris(gray, 2, 3, 0.04)

img[dst > 0.01 * dst.max()] = [0,0,255]

cv2.imshow("HarrisCorner Detection", img)
cv2.waitKey()
cv2.destroyAllWindows()

```



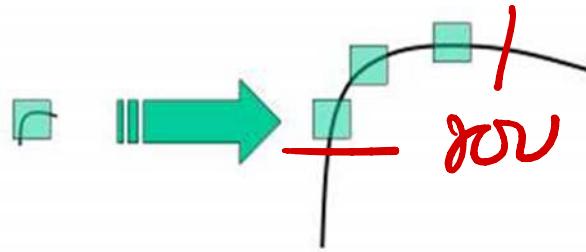
Lab302 – SHIFT (Scale-Invariant Feature Transform)

2. คุณลักษณะจากการเปลี่ยนแปลงแบบไม่ต่อเนื่อง

การหาคุณลักษณะจากการเปลี่ยนแปลงแบบไม่ต่อเนื่อง (Scale-Invariant Feature Transform)

SIFT โดยการตรวจหาจุดสำคัญที่จะมีลักษณะพิเศษจากพิกัด (x, y)

โดยพิจารณาจากจุดที่อยู่ในลักษณะเดียวกัน
โดยใช้กรอบขนาด 16×16 รอบจุดสำคัญ แบ่งออกเป็น 16 บล็อกอย่างขนาด 4×4 สำหรับแต่ละบล็อกอย่างจะมีการสร้างกราฟแยกแจ้งความถี่ของจุดภาพ



```
import cv2
import numpy as np
img = cv2.imread('./image/house.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

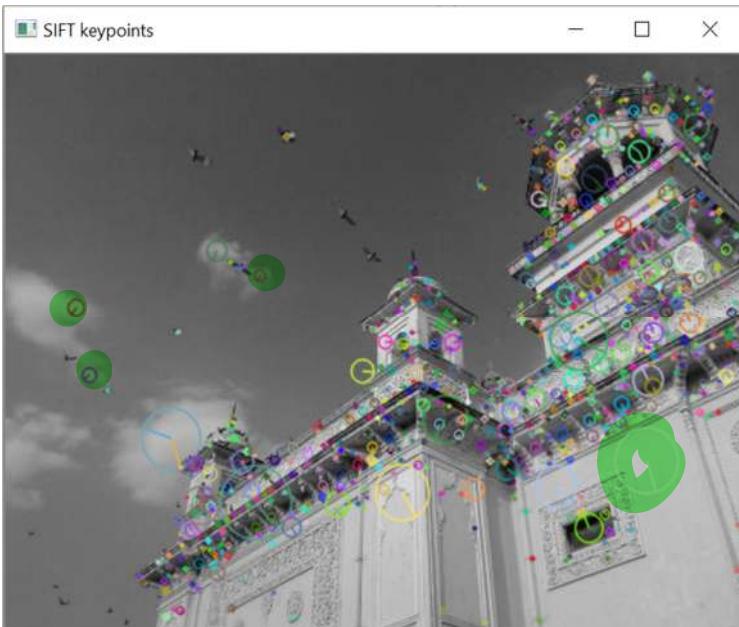
sift = cv2.xfeatures2d.SIFT_create()

kp = sift.detect(gray, None)

img = cv2.drawKeypoints(gray, kp, img, flags=cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)

cv2.imshow('SIFT keypoints', img)

cv2.waitKey()
cv2.destroyAllWindows()
```



3. เพิ่มส่วนเสริม

```
conda install -c menpo opencv
```

4. เพิ่มความเร็วและเสถียรในการตรวจจับคุณลักษณะ

การเพิ่มความเร็วและความเสถียรภาพในการตรวจจับคุณลักษณะ (Speeded-Up Robust Features) SURF ซึ่งทำงานได้เร็วขึ้น ด้วยวิธีปรับปรุงความเร็วในทุกขั้นตอน เนื่องจากใช้ภาพรวมและตัวกรอง โดย สามารถจัดการภาพที่เบลอและหมุนได้ ซึ่งพบร้าเร็วกว่า SIFT ถึง 3 เท่าในขณะที่ประสิทธิภาพเทียบเท่ากัน

```
▶ import cv2
import numpy as np
img = cv2.imread('./image/house.jpg')
gray= cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

surf = cv2.xfeatures2d.SURF_create()

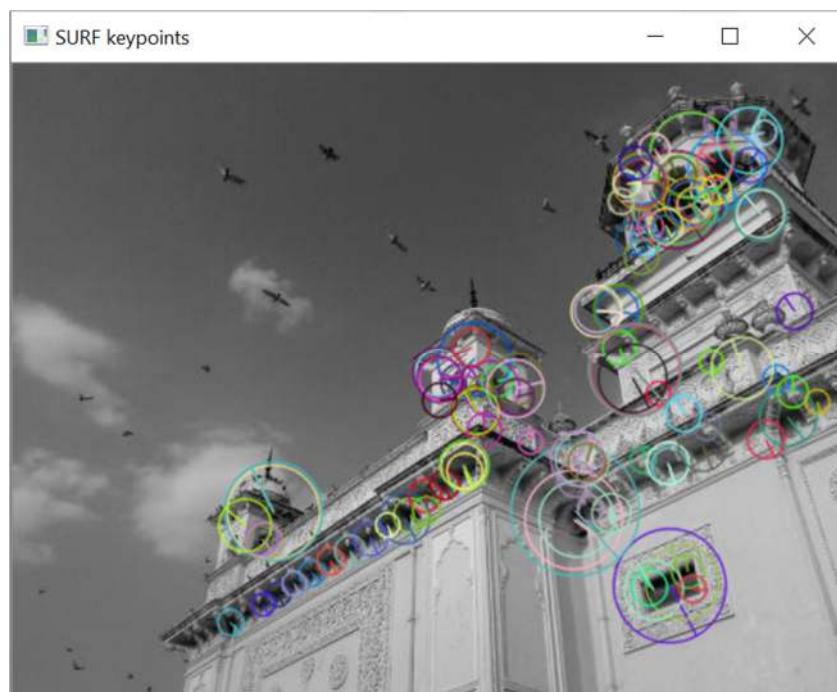
surf.setHessianThreshold(5000)

kp, des = surf.detectAndCompute(img,None)

img=cv2.drawKeypoints(gray,kp,img,flags=cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)

cv2.imshow('SURF keypoints',img)

cv2.waitKey()
cv2.destroyAllWindows()
```



Lab303 – Key point Matching

5. การตรวจจับวัตถุด้วยการเปรียบเทียบจากแม่แบบ

```

import cv2
import numpy as np

# from matplotlib import pyplot as plt

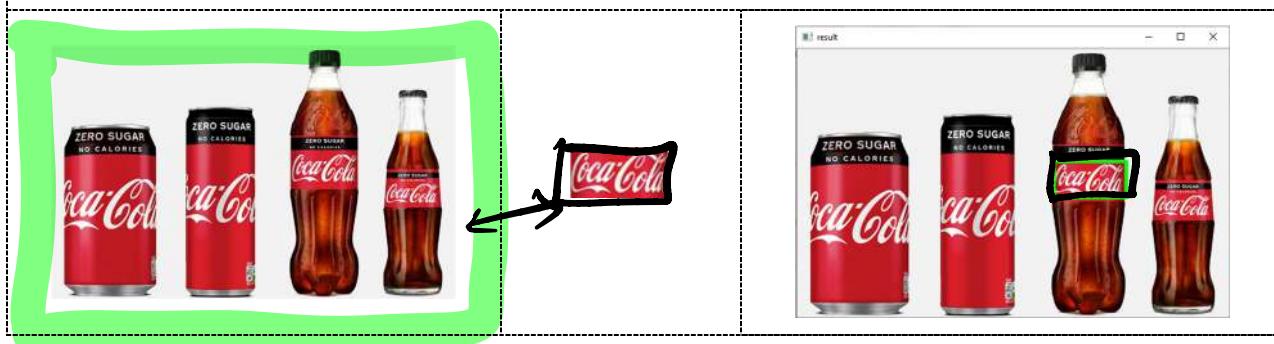
img_rgb = cv2.imread('./image/coca-cola.jpg')
img_gray = cv2.cvtColor(img_rgb, cv2.COLOR_BGR2GRAY)
template = cv2.imread('./image/cola.jpg', 0)
w, h = template.shape[::-1]

res = cv2.matchTemplate(img_gray, template, cv2.TM_CCOEFF_NORMED)
threshold = 0.8
loc = np.where(res >= threshold)
print(len(loc))

for pt in zip(*loc[::-1]):
    cv2.rectangle(img_rgb, pt, (pt[0] + w, pt[1] + h), (0, 255, 0), 2)
    print("rectangle 1")

# cv2.imwrite('res.png',img_rgb)
cv2.imshow("result", img_rgb)
cv2.waitKey(0)

```



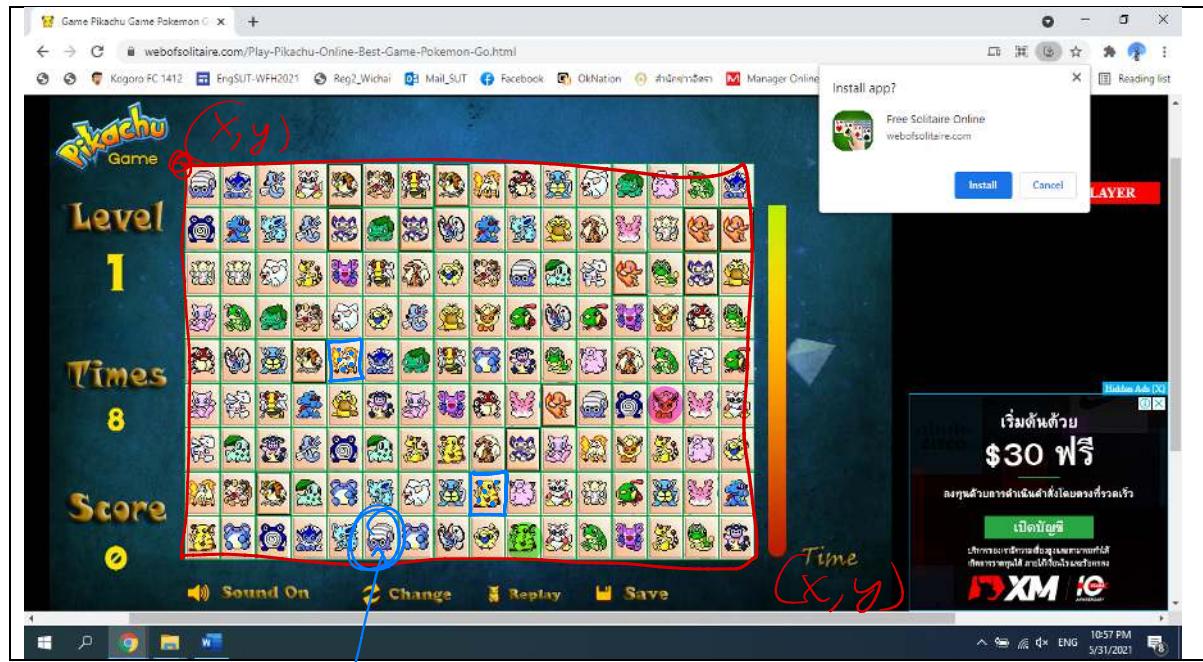
4/5 -- Mini Project_1: Pokémon Matching Image Project

<https://phyblas.hinaboshi.com/20180722>

1. Capture Pokémon Game from >> <https://webofsolitaire.com/Play-Pikachu-Online-Best-Game-Pokemon-Go.html>

↑ Done

1



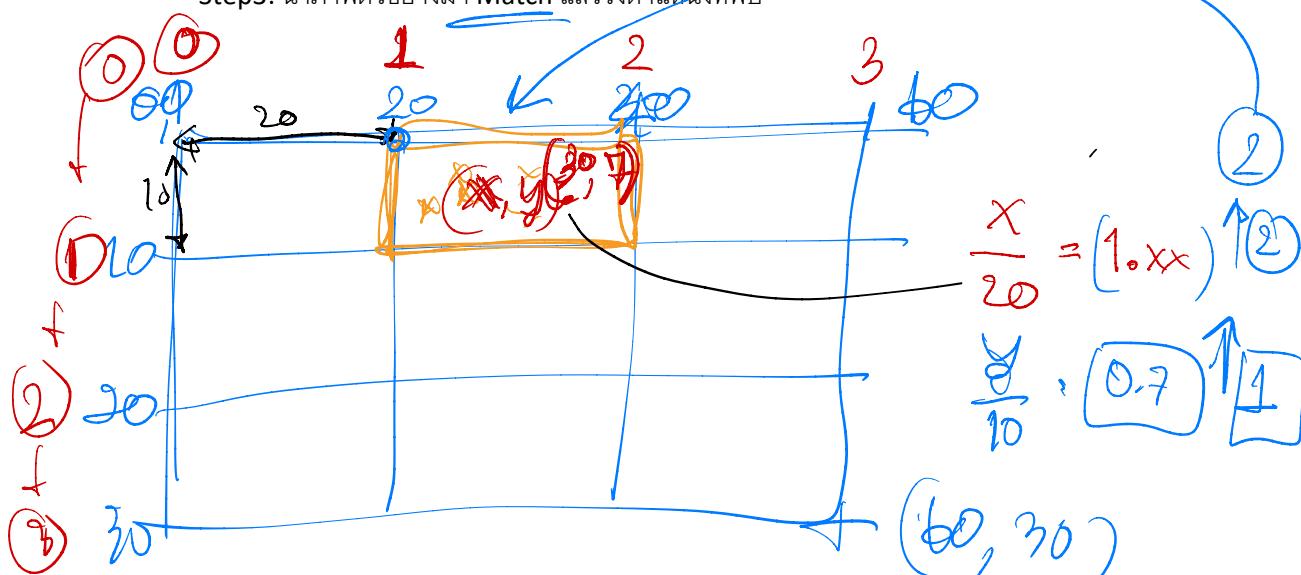
2. Experiment < with python code >

values
Step0: จากวุปด้านแบบ ทดสอบเปิด, ทดสอบโปรแกรม Key point Matching

wave
Step1: หาตำแหน่งของภาพ, ตัดเอาเฉพาะพื้นที่ที่ทำงาน

Marker
Step2: ใช้ Mouse Click เพื่อเอาภาพตัวอย่าง

Marker
Step3: นำภาพตัวอย่างมา Match และวัดตำแหน่งที่พบ



กิจกรรมที่ 6/6 – Pokémon Matching Image Project

ศึกษาและปรับแก้การทำงานของโปรแกรมเพื่อ

1. แก้ไขให้โปรแกรมทำงานให้ถูกต้องทำอย่างไร < img_rgb = testImg.copy() >
2. ให้ระบบายสีแดงแทนที่จะตีกรอบเขียว
3. หาตัวนี้ แล้วไม่ครบ 4 ตัว ทำอย่างไร



4. ทดสอบกับโจทย์ใหม่ที่สร้างเองจาก <https://webofsolitaire.com/Play-Pikachu-Online-Best-Game-Pokemon-Go.html>

การพัฒนาโปรแกรมประยุกต์และปัญญาประดิษฐ์ เพื่อการมองเห็นของเครื่องจักร
Computer Programming and Artificial Intelligence in Machine Vision

ชื่อ-สกุล :

5/5 -- คำถ้ามท้ายบทเพื่อทดสอบความเข้าใจ

กิจกรรมที่ 1/6 – Webcam to Video with Logo

จะเขียนโปรแกรมเพื่อบันทึกวิดีโอจากสตรีมวิดีโอของกล้องเว็บแคม ทำการใส่โลโก้ XXX และระบุวันที่ในภาพ

< Capture Jupyter Code >



YouTube Link (หรือลิงค์ที่ฝากวิดีโอ อื่นๆ) ของงานที่ทำได้

กิจกรรมที่ 2/6 – Canny Edge Detection from Webcam to Video with Logo

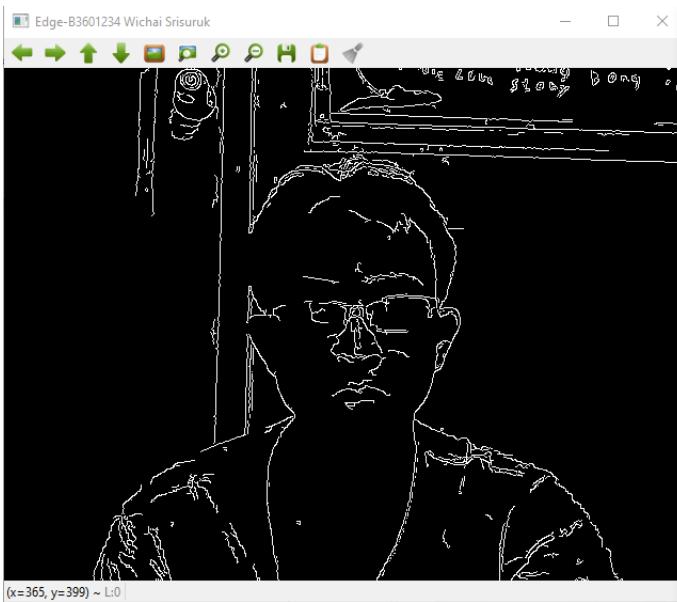
จะเขียนโปรแกรมเพื่ออ่านภาพวิดีโอจากสตรีมวิดีโอของกล้องเวปแคม และ Capture เพื่อทำ Canny Edge Detection

< Capture Jupyter Code >

< picture input >



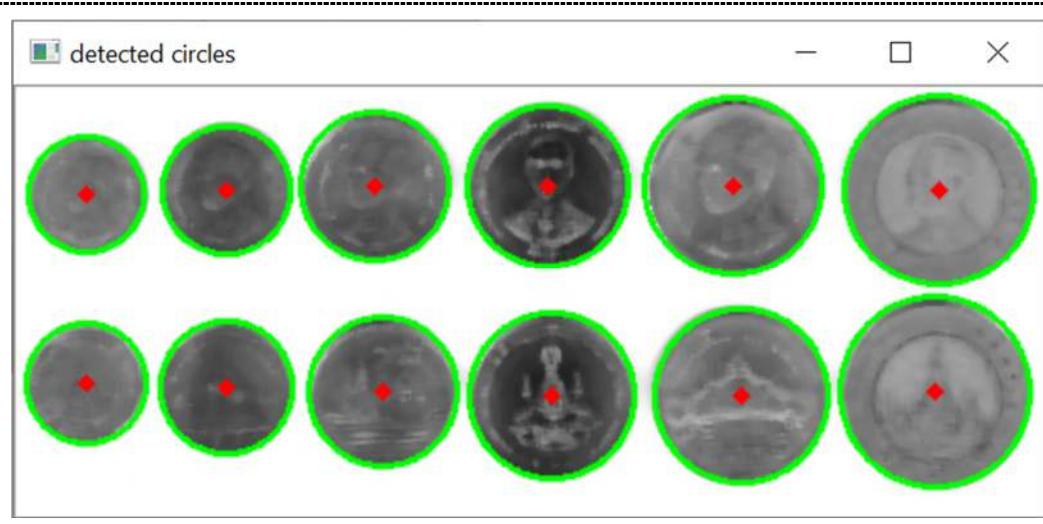
< Edge output >



กิจกรรมที่ 3/6 – ทำการตรวจจับเหรียญด้วย Hough Circle Transform

ให้ถ่ายรูปเหรียญของตัวเอง จำนวน 12 เหรียญ เมื่อันตัวอย่าง แล้วทดสอบทำ Hough Circle Transform

< Code > - Step 1



กิจกรรมที่ 4/6 – จะทำการ detect ภาพจากกล้องเพื่อหาภาพโคล์กกระป๋อง(สีแดง)

จะเขียนโปรแกรมเพื่ออ่านภาพวิดีโอจากสตรีมวิดีโอของกล้องเวปแคมที่มีภาพโคล์กแบบสีแล้วทำการแสดงผลทางโค้ก

< Code > - Step 1

< Real Image >

< Result Image >

ตัวอย่างภาพทดสอบ



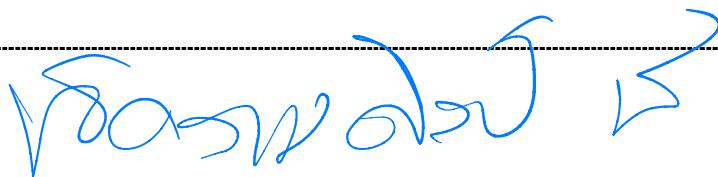
กิจกรรมที่ 5/6 – Graphic Text

ใช้ภาพถ่ายของตัวเอง สร้างข้อความ แล้วเติมเข้าความในภาพถ่ายมุมล่างขวาเมื่อ

< Code > - Step 1

< Real Image >

< Result Image >



กิจกรรมที่ 6/6 – Pokémon Matching Image Project

ศึกษาและปรับแก้การทำงานของโปรแกรมเพื่อ

1. แก้ไขให้โปรแกรมทำงานให้ถูกต้องทำอย่างไร → 1. 修正 . Copy ()

2. ให้ระบบายสีแดงแทนที่จะตีกรอบเขียว

3. หาตัวนี้ แล้วไม่ครบ 4 ตัว ทำอย่างไร



修正

. Copy ()

修正 Temple so 10%

4. ทดสอบกับโจทย์ใหม่ที่สร้างเองจาก <https://webofsolitaire.com/Play-Pikachu-Online-Best-Game-Pokemon-Go.html>