

Human Skeleton model

and Color Image Segmentation by Thresholding



01204483 การประมวลผลสัญญาณภาพดิจิทัล

อาจารย์ที่ปรึกษาโครงการ

ดร.นิตยา เมืองนาค

ผู้จัดทำ

- นางสาว กนกวรรณ บัวภาคำ 6040200049
- นายธรรมธัช ตันติไพธรรม 6040201983
- นาย อีระวัฒน์ ชรินทร์ 6040202203
- นางสาวรัตดาพร อักษรทอง 6040204010
- นายอนุชา ศรีลาแก้ว 6040205407



ที่มาและความสำคัญ

“

เนื่องจากในปัจจุบันนี้มีการค้นคว้าและพัฒนาเทคโนโลยีต่างๆ
เกี่ยวกับการวิเคราะห์ความเคลื่อนไหวของมนุษย์และการปฏิสัมพันธ์
ระหว่างมนุษย์กับคอมพิวเตอร์ (HCI) เพื่อนำมาประยุกต์ใช้ในการ
อำนวยความสะดวกหรือตอบสนองความต้องการของมนุษย์ ซึ่งระบบ
จับการเคลื่อนไหวที่มีวางจำหน่ายส่วนใหญ่ต้องการให้ผู้ผู้ใช้ใส่
เครื่องหมาย แขนข้อต่อหรือส่วนต่างๆ ของร่างกาย และเทคโนโลยี
พื้นฐานที่ใช้อาจจะมีราคาแพงและอาจไม่ตรงวัตถุประสงค์สำหรับการ
นำมาวิเคราะห์การเคลื่อนไหวของมนุษย์



จากการศึกษา เทคนิค Color Image Segmentation by Thresholding

จากการที่ได้ศึกษาเกี่ยวกับเทคนิคต่างๆ ในการปรับปรุงคุณภาพของภาพทางคณะผู้จัดทำ ได้เลือกเทคนิค **Color Image Segmentation by Thresholding** เพื่อศึกษาการแบ่งส่วนและตรวจจับของรูปภาพในการวิเคราะห์หารูปร่างและความเป็นไปได้ของโครงสร้างร่างกายมนุษย์ โดยเริ่มจากเขียนโปรแกรมใน ภาษา **Python** ใช้ **opencv-python** เข้ามาช่วยในการใช้งานโปรแกรม การทำงานของโปรแกรม จะมีการกำหนดลักษณะต่างๆของร่างกายมนุษย์โดยทั้งหมด 19 องค์ประกอบที่ไล่จากศีรษะจนถึงเท้า และลักษณะท่าทางที่เป็นส่วนขยับของร่างกายเช่น คอ มือ แขน เป็นต้น

วัตถุประสงค์



ทางคณะผู้จัดทำ ได้จัดทำขึ้นเพื่อ

- เพื่อศึกษาเกี่ยวกับ เทคนิค Color Image Segmentation by Thresholding
- เพื่อศึกษาการแบ่งส่วนและตรวจจับของรูปภาพ
- โปรแกรมสามารถวิเคราะห์หารูปร่างและความเป็นไปได้ของโครงสร้างร่างกายมนุษย์ได้



วิธีการจัดทำ

การได้มาซึ่งรูปภาพ

ภาพที่ใช้ในการทดสอบโปรแกรม คือ ภาพที่นำมาจากอินเทอร์เน็ต ซึ่งใช้ทั้งหมด 5 ภาพ และภาพที่ถ่ายเองอีก 1 ภาพ

การเตรียมภาพ

นำภาพที่ได้มาจากอินเทอร์เน็ต และภาพที่ถ่ายเอง มาปรับปรุงให้คุณสมบัติของภาพนั้นๆ เด่นขึ้น



ภาพต้นฉบับ



รูปที่ 2 ภาพที่ใช้ในการทดสอบจากอินเทอร์เน็ต



รูปที่ 3 ภาพที่ใช้ในการทดสอบจากอินเทอร์เน็ต



ภาพต้นฉบับ



รูปที่ 4 ภาพที่ใช้ในการทดสอบจากอินเทอร์เน็ต



รูปที่ 5 ภาพที่ใช้ในการทดสอบจากอินเทอร์เน็ต



ภาพต้นฉบับ



รูปที่ 6 ภาพที่ใช้ในการทดสอบจากอินเทอร์เน็ต



รูปที่ 7 ภาพที่ใช้ในการทดสอบจากการถ่ายเอง

วิธีการจัดทำ





วิธีการจัดทำ

การประมวลผลตามวัตถุประสงค์

โดยเริ่มแรก การทำงานโปรแกรมจะมีการตรวจจับรูปภาพเข้ามาทาง input เพื่อทำการ scan ว่ารูปภาพที่เข้ามาลักษณะอย่างไร (ซึ่งใช้ Ai ในการตรวจจับมันเร็วมาก) ถ้าหากการตรวจสอบผ่าน โปรแกรมก็จะเอาภาพ input ที่ได้มานำไปประมวลผลต่อและแสดงออกมาเป็น output ที่ detect 19 องค์ประกอบ ของร่างกายมนุษย์นั้น คือ skeleton แต่จะตรวจจับถูกต้องหรือไม่ จึงต้องใช้ฟังก์ชัน assert เข้ามาช่วยในการตรวจสอบว่าองค์ประกอบที่ตรวจสอบ เข้ามานั้นถูกต้องและผิดพลาดน้อยที่สุดหลังจากผ่านการทดสอบจะเก็บค่า ไว้ที่ Point เพื่อรอคำสั่ง ค้นหาชิ้นส่วนของร่างกายนั้นก็คือลักษณะท่าทางของมนุษย์ว่า ณ รูปภาพนั้นกำลังทำท่าทางอะไรอยู่ และถ้าหากมีความสอดคล้องกับตัวแปรที่กำหนดไว้ข้างต้นจะทำการเช็คค่าของ threshold ในท่าทางและลักษณะของร่างกายมนุษย์ทั้งหมดแต่ละส่วนเพื่อค่าสูงสุดและต่ำสุดเพื่อทำการโยกหากัน ถ้าหากค่า threshold ที่ได้มีค่าสูงกว่าปกติ การทำ Skeleton นั้นก็จะไม่เกิดขึ้นเพราะเนื่องจากค่า threshold สูงเกินไป และหากทำการ threshold เรียบร้อย จะเข้ามาใน loop ของการค้นหาองค์ประกอบและลักษณะท่าทางที่มีความสัมพันธ์กัน และเกิดการทดสอบขึ้นอีกครั้ง จากนั้นเมื่อทำการทดสอบเสร็จสมบูรณ์ก็จะเก็บค่าไว้ที่ตัวแปรตัวถัดไปก็คือ idFrom และ idTo



วิธีการจัดทำ

หมายความว่า ai จะเริ่ม detect ว่าทั้งหมดของร่างกายมนุษย์ตรงไหนสัมพันธ์กันมั้งและเริ่มโยงเส้น Skeleton จากตรงนี้ ถึงตรงนั้น จากนั้นจะใช้คำสั่ง if เพื่อทำการตรวจสอบว่า จุดนี้ไปตรงนั้นและจุดนั้นมาตรงนี้ ถูกหรือไม่ จากนั้นเขียนหน้าตาหรือที่เรียกว่า Figure แบบใน matlab เพื่อจะแสดง output ออกมาให้เห็นก็จะ กำหนดค่าต่างๆ จากนั้นก็จะแสดง Output ออกมา 2 รูปแบบ gray-scale นั่นก็คือ รูปแรกเป็นรูปปกติ รูปที่สองเป็น รูปที่ผ่านการตรวจสอบมาทุกวิธี



ผลลัพธ์

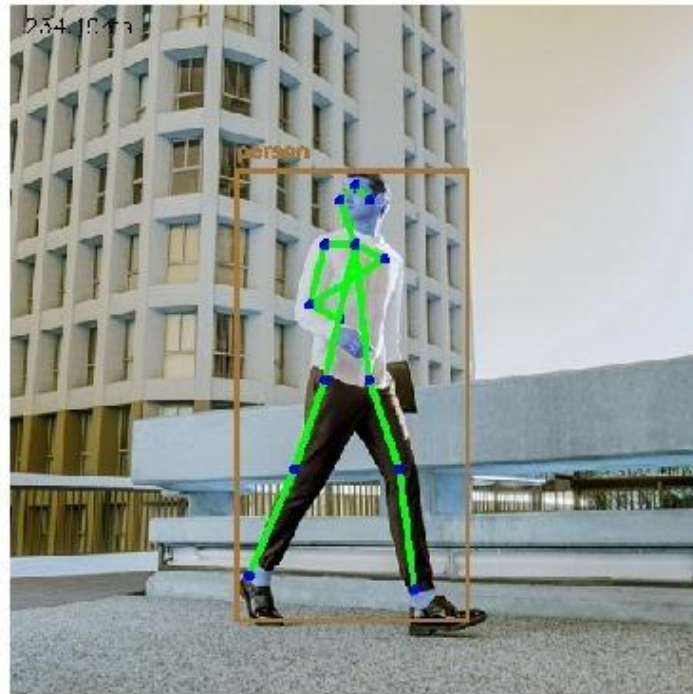
หลังจากผ่านกระบวนการใช้

เทคนิค Color Image Segmentation by Thresholding

Binary Image



Skeleton

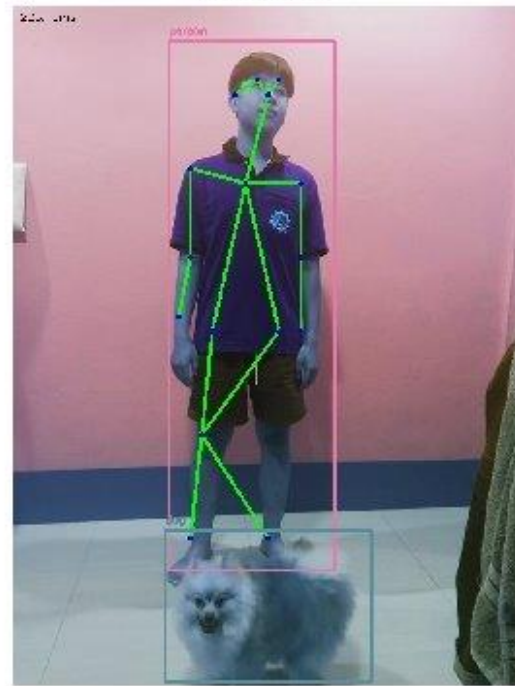


ผลลัพธ์จากการทำ *color image segmentation* โดยภาพที่ได้จากอินเทอร์เน็ต

Binary Image



Skeleton



ผลลัพธ์จากการทำ *color image segmentation* โดยภาพที่ถ่ายเอง



การประเมินและวิเคราะห์ผล

output ออกมาให้เห็นก็จะกำหนดค่าต่างๆ จากนั้นก็จะแสดง Output ออกมา 2 รูป แบบ gray-scale นั่นก็คือ รูปแรกเป็นรูปปกติ รูปที่สองเป็นรูปที่ผ่านการตรวจสอบมาทุกวิธี

ซึ่ง output ที่ได้ สามารถวิเคราะห์หารูปร่างและความเป็นไปได้ของ โครงสร้างร่างกายมนุษย์ได้ตรงตามที่คณะผู้จัดทำได้ศึกษามา



สรุปผลและข้อเสนอแนะ

สรุปผล

จากการศึกษาการแบ่งส่วนและตรวจจับของรูปภาพในการวิเคราะห์หารูปร่างและความเป็นไปได้ของโครงสร้างร่างกายมนุษย์ โดยเริ่มจากเขียนโปรแกรมใน ภาษา Python ใช้ opencv-python เข้ามาช่วยในการใช้งานโปรแกรม การทำงานของโปรแกรม และการทดลองใช้โปรแกรม ผลที่ได้คือ ตรงตามวัตถุประสงค์ของผู้จัดทำ เนื่องจาก โปรแกรมที่ทางผู้จัดทำได้ทำขึ้นนั้น สามารถวิเคราะห์หารูปร่างและความเป็นไปได้ของโครงสร้างร่างกายมนุษย์ได้



สรุปผลและข้อเสนอแนะ

■ ข้อเสนอแนะ

เนื่องจาก เป็นการเรียกใช้การทำของ Image Segmentation โดยแบ่งหลักการให้ส่งข้อมูลเป็นรูปภาพที่มีร่างกายเป็นมนุษย์และใช้ opencv-python ในการตรวจจับเป็นภาพ Skeleton ขึ้นมา ยังมีปัญหาติดขัดอยู่ เนื่องจากเพิ่งได้ศึกษาการเขียน Opencv-python ส่วนใหญ่จะใช้ Video Capture ในการตรวจจับวัตถุและแสดงออกมาในลักษณะร่างของ Skeleton

ควรใช้ Video Capture ในการตรวจจับวัตถุและแสดงออกมาในลักษณะร่างของ Skeleton จะสามารถวิเคราะห์หารูปร่างและความเป็นไปได้ของโครงสร้างร่างกายมนุษย์ได้ดีกว่า



โค้ดแสดงการทำงานของโปรแกรม

Terminal Help

skeletonrgb.py - Project Image - Visual Studio Code

ai.py

skeletonrgb.py ×

skeletonrgb.py > ...

```
1  import cv2
2  import numpy as np
3  import argparse #เป็น module ที่กำหนด argument ที่จะเรียกใช้งานได้จาก console
4  from skimage.morphology import skeletonize
5  import matplotlib.pyplot as plt
6  import cvlib as cv
7  from cvlib.object_detection import draw_bbox
8  parser = argparse.ArgumentParser() #ตัวกำหนดเรียกใช้ที่ console
9  parser.add_argument('--input') #เพิ่มรูปภาพโดยใช้ argument สิ่งที่ console
10 parser.add_argument('--thr', default=0.2, type=float) #ปรับระดับค่า threshold
11 parser.add_argument('--width', default=368, type=int) #กำหนดความกว้างของ figure
12 parser.add_argument('--height', default=368, type=int) #กำหนดความสูงของ figure
13 im = cv2.imread("make.jpg", 0) # ภาพ .jpg ไร้เทส
14 args = parser.parse_args() #ตัวกำหนดเรียกใช้ argument ทั้งหมด
15
```

```

16  #กำหนดส่วนลักษณะต่างๆของร่างกายมนุษย์
17  BODY_PARTS = { "Nose": 0, "Neck": 1, "RShoulder": 2, "RElbow": 3, "RWrist": 4,
18  |             |             |             "LShoulder": 5, "LElbow": 6, "LWrist": 7, "RHip": 8, "RKnee": 9,
19  |             |             |             "RAnkle": 10, "LHip": 11, "LKnee": 12, "LAnkle": 13, "REye": 14,
20  |             |             |             "LEye": 15, "REar": 16, "LEar": 17, "Background": 18 }
21
22  #กำหนดท่าทางให้คล้องกับร่างกายมนุษย์
23  POSE_PAIRS = [ ["Neck", "RShoulder"], ["Neck", "LShoulder"], ["RShoulder", "RElbow"],
24  |             |             |             ["RElbow", "RWrist"], ["LShoulder", "LElbow"], ["LElbow", "LWrist"],
25  |             |             |             ["Neck", "RHip"], ["RHip", "RKnee"], ["RKnee", "RAnkle"], ["Neck", "LHip"],
26  |             |             |             ["LHip", "LKnee"], ["LKnee", "LAnkle"], ["Neck", "Nose"], ["Nose", "REye"],
27  |             |             |             ["REye", "REar"], ["Nose", "LEye"], ["LEye", "LEar"] ]
28
29
30
31  #แสดงหน้าต่าง gui
32  inWidth = args.width
33  inHeight = args.height
34  #เรียกใช้ไฟล์ graph_opt.pb
35  net = cv2.dnn.readNetFromTensorflow("graph_opt.pb")

```

```

36
37 #ai ตรวจสอบรูปภาพที่เข้ามาทาง input
38 cap = cv2.VideoCapture(args.input if args.input else 0)
39
40 #หลังจากตรวจสอบและผ่านการตรวจสอบจะเข้า loop while เพื่อทำไฟล์รูปภาพขึ้นมาให้ชื่อว่า frame ถ้าหากไม่ผ่านก็อาจจะ error หรือรอการตรวจสอบต่อไป
41 while cv2.waitKey(1) < 0:
42     hasFrame, frame = cap.read()
43     if not hasFrame:
44         cv2.waitKey()
45         break
46
47     #แสดงรูปร่างที่ได้ในหน้าต่าง gui
48     frameWidth = frame.shape[1]
49     frameHeight = frame.shape[0]
50
51     #เป็น output ที่เรียกใช้ 19 องค์ประกอบของร่างกายมนุษย์
52     net.setInput(cv2.dnn.blobFromImage(frame, 1.0, (inWidth, inHeight), (127.5, 127.5, 127.5), swapRB=True, crop=False))
53     out = net.forward()
54     out = out[:, :19, :, :]
55     #assert เป็นฟังก์ชันที่ไว้ทดสอบว่าองค์ประกอบร่างกายของมนุษย์ตรงตาม output ที่เข้คเข้ามาไหม
56     assert(len(BODY_PARTS) == out.shape[1])
57
58     #กำหนดตัวแปร points ขึ้นมาค่าว่างเปล่า
59     points = []
60

```

```

60
61 #คำสั่งวนซ้ำให้ค้นหาชิ้นส่วนของร่างกายให้ตรงตามองค์ประกอบ 19 ส่วน
62 for i in range(len(BODY_PARTS)):
63     heatMap = out[0, i, :, :] #ตรวจสอบชิ้นส่วนร่างกายที่มีความสอดคล้องกัน
64
65     #กำหนดตัวแปรเช็คค่าสูงสุดและค่าต่ำสุดของ input
66     _, conf, _, point = cv2.minMaxLoc(heatMap)
67     x = (frameWidth * point[0]) / out.shape[3]
68     y = (frameHeight * point[1]) / out.shape[2]
69
70     #จุดที่มีการเช็คค่า ถ้าหากค่าของ threshold มีค่าที่สูงกว่าปกติ
71     points.append((int(x), int(y)) if conf > args.thr else None)
72
73 #คำสั่ง loop ที่จะค้นหาองค์ประกอบให้ครบสมบูรณ์ที่สุด
74 for pair in POSE_PAIRS:
75     partFrom = pair[0]
76     partTo = pair[1]
77     assert(partFrom in BODY_PARTS)
78     assert(partTo in BODY_PARTS)
79
80     #กำหนดตัวแปรเพิ่มเติม
81     idFrom = BODY_PARTS[partFrom]
82     idTo = BODY_PARTS[partTo]
83     #ใช้คำสั่ง if เพื่อตรวจสอบข้อมูลของรูปภาพที่เข้ามาและเช็คแต่ละเฟรม

```

```

84     if points[idFrom] and points[idTo]:
85         cv2.line(frame, points[idFrom], points[idTo], (0, 255, 0), 3)
86         cv2.ellipse(frame, points[idFrom], (3, 3), 0, 0, 360, (0, 0, 255), cv2.FILLED)
87         cv2.ellipse(frame, points[idTo], (3, 3), 0, 0, 360, (0, 0, 255), cv2.FILLED)
88
89     #กำหนดค่าหน้าต่างของ figure แบบใน matlab
90     fig, axes = plt.subplots(1, 2, figsize=(8, 4), sharex=True, sharey=True)
91     ax = axes.ravel()
92     skeleton = skeletonize(frame)
93
94     #แสดงรูปภาพที่ถูกประมวลผล
95     ax[0].imshow(im)
96     ax[0].set_title('threshold')
97     ax[0].axis('off')
98
99     ax[1].imshow(frame, cmap=plt.cm.gray)
100    ax[1].set_title('skeleton')
101    ax[1].axis('off')
102
103    #ตัวแปรที่พาเข้าไลบรารีที่จะทำการอ่านภาพจากหน่วยเก็บข้อมูลทำการตรวจสอบวัตถุบนภาพและแสดงออกมาเป็นกล่องข้อความ
104    bbox, label, conf = cv.detect_common_objects(im)
105    output_image = draw_bbox(frame, bbox, label, conf)

```




โค้ดแสดงการทำงานของโปรแกรม

```
106
107     #คำนวณเวลาในการประมวลผลรูปภาพ
108     t, _ = net.getPerfProfile()
109     freq = cv2.getTickFrequency() / 1000
110     cv2.putText(frame, '%.2fms' % (t / freq), (10, 20), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 0))
111     print('Success')
112     cv2.imshow('Human Skeleton using OpenCV', frame) #ตัวแสดงทั้งกล้องและ gui
113
114     fig.tight_layout()
115     plt.imshow(output_image)
116     plt.show()
```

จบการทำงาน



THANKS!

ขอบ
คุณ
ค่ะ

