

성능 개선 및 측정

성능 향상

001 >> 향상 전: 평균 6.98fps

002 >> YOLO 사용 제한 : 평균 7.5fps

올로의 사용을 1초에 한번, 혹은 3프레임에 한번 등으로 호출 주기를 제한하여 전체 프로그램의 속도를 향상

003 >> 멀티 스레딩 사용 : 평균 10.5fps

올로가 찾아낸 바운딩 박스 내부에서 작동하는 mediapipe holistic 을 각각 스레드에서 작업하도록 하여 속도 향상

항상 전



평균 6.98fps

YOLOv5 사용 제한

```
while cap.isOpened():
    success, image = cap.read()
    if not success:
        continue

    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    image.flags.writeable = False

    if start:
        start = False
        startTime = time.time()
        result = yolo_model(image)

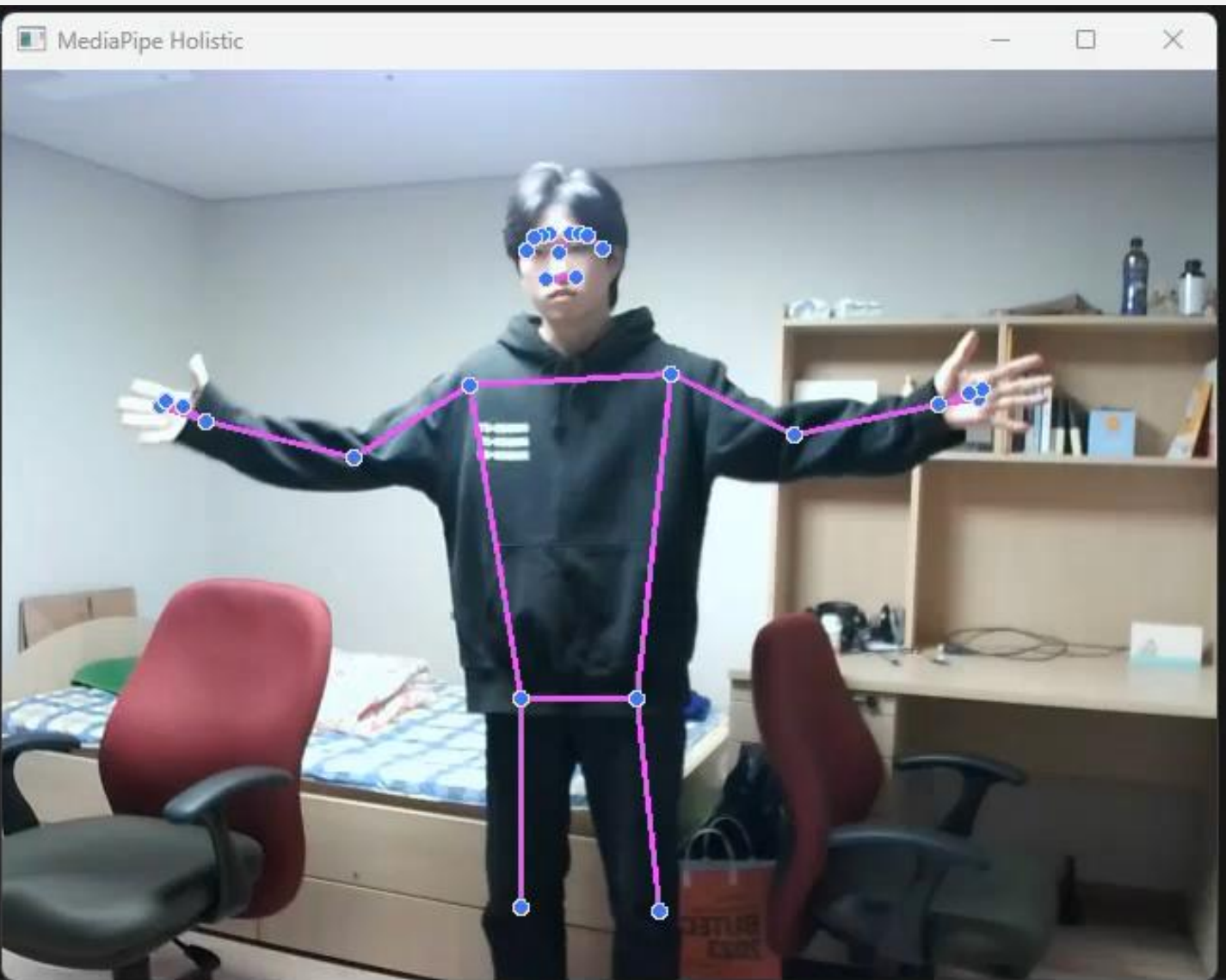
    if (frameCnt == 3):
        result = yolo_model(image)
        frameCnt = 0
    frameCnt += 1
```

YOLOv5 사용 제한

PROBLEMS OUTPUT DEBUG CONSOLE

Frames per second: 11.78
3
Frames per second: 12.47
3
Frames per second: 12.50
3
Frames per second: 11.68
3
Frames per second: 11.62
3
Frames per second: 11.86
3
Frames per second: 12.54
3
Frames per second: 11.66
3
Frames per second: 11.58
3
Frames per second: 11.77
3
Frames per second: 10.98
3
Frames per second: 11.08
3
Frames per second: 10.59
3
Frames per second: 10.89
3

MediaPipe Holistic



평균 7.5fps

멀티 스레딩 사용

```
def holisticProcess(image, i, resultList):
    xmin, ymin, xmax, ymax, confidence, clas = resultList[i]

    results = holistic[i].process(image[int(ymin)+MARGIN:int(ymax)+MARGIN,int(xmin)+MARGIN:int(xmax)+MARGIN:])

    mp_drawing.draw_landmarks(image[int(ymin)+MARGIN:int(ymax)+MARGIN,int(xmin)+MARGIN:int(xmax)+MARGIN:],
                              results.pose_landmarks, mp_holistic.POSE_CONNECTIONS,
                              mp_drawing.DrawingSpec(color=(245,117,66), thickness=2, circle_radius=2),
                              mp_drawing.DrawingSpec(color=(245,66,230), thickness=2, circle_radius=2)
                              )

    data_left = str(results.left_hand_landmarks)
    data_right = str(results.right_hand_landmarks)
    data_pose = str(results.pose_world_landmarks)
    # data_face = str(results.face_landmarks)

    temp = [data_left, data_right, data_pose]

    part_data.append(temp)

for i in range(len(resultList)):

    p = threading.Thread(target=holisticProcess, args=(image, i, resultList, ))
    p.start()
    procs.append(p)


for p in procs:
    p.join() # 프로세스가 모두 종료될 때까지 대기
```


멀티 스테딩 사용

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

Frames per second: 12.07
3
Frames per second: 11.66
3
Frames per second: 12.54
3
Frames per second: 11.71
3
Frames per second: 10.79
3
Frames per second: 11.83
3
Frames per second: 12.64
3
Frames per second: 12.56
3
Frames per second: 13.45
3
Frames per second: 13.69
3
Frames per second: 13.07
3
Frames per second: 13.37
3
Frames per second: 13.48
3
Frames per second: 11.77
3
Frames per second: 13.29
3

MediaPipe Holistic



평균 10.26fps