

3D Keypoints

1. Image to 3D keypoints

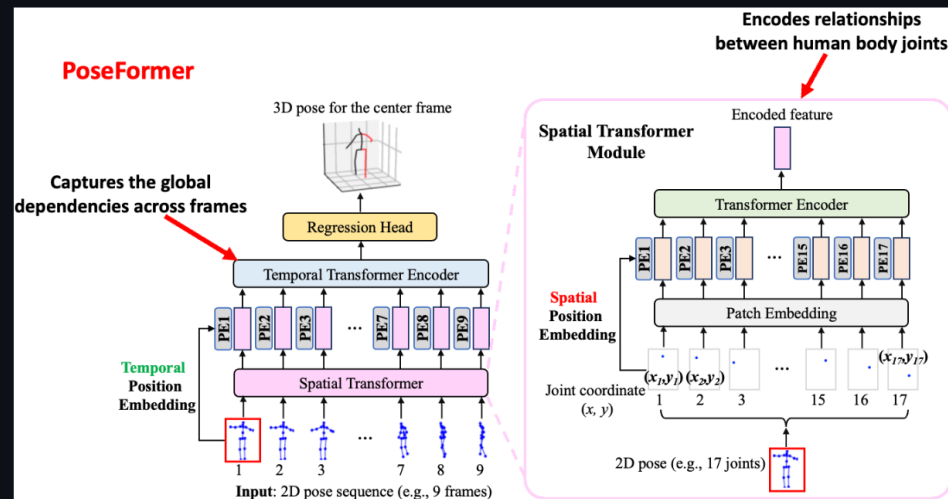
3D Human Pose Estimation with Spatial and Temporal Transformers

This repo is the official implementation for 3D Human Pose Estimation with Spatial and Temporal Transformers. The paper is accepted to ICCV 2021.

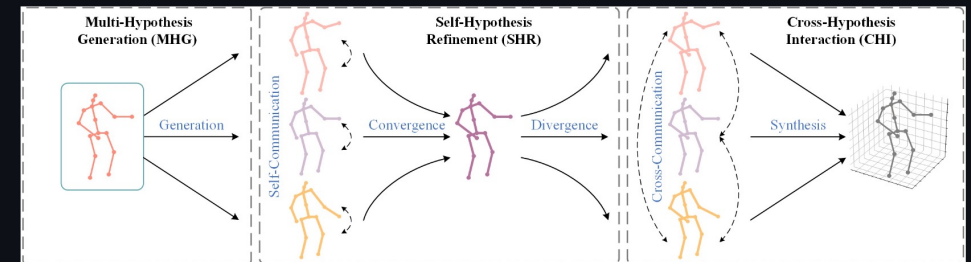
- Welcome to check our CVPR 2023 work: [PoseFormerV2](#)
- Visualization code for in-the-wild videos can be found here [PoseFormer_demo](#)

[Video Demonstration](#)

⌚ PoseFormer Architecture

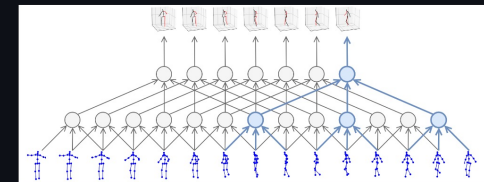


⌚ MHFormer: Multi-Hypothesis Transformer for 3D Human Pose Estimation [CVPR 2022]



MHFormer: Multi-Hypothesis Transformer for 3D Human Pose Estimation,
Wenhao Li, Hong Liu, Hao Tang, Pichao Wang, Luc Van Gool,
In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2022

3D human pose estimation in video with temporal convolutions and semi-supervised training

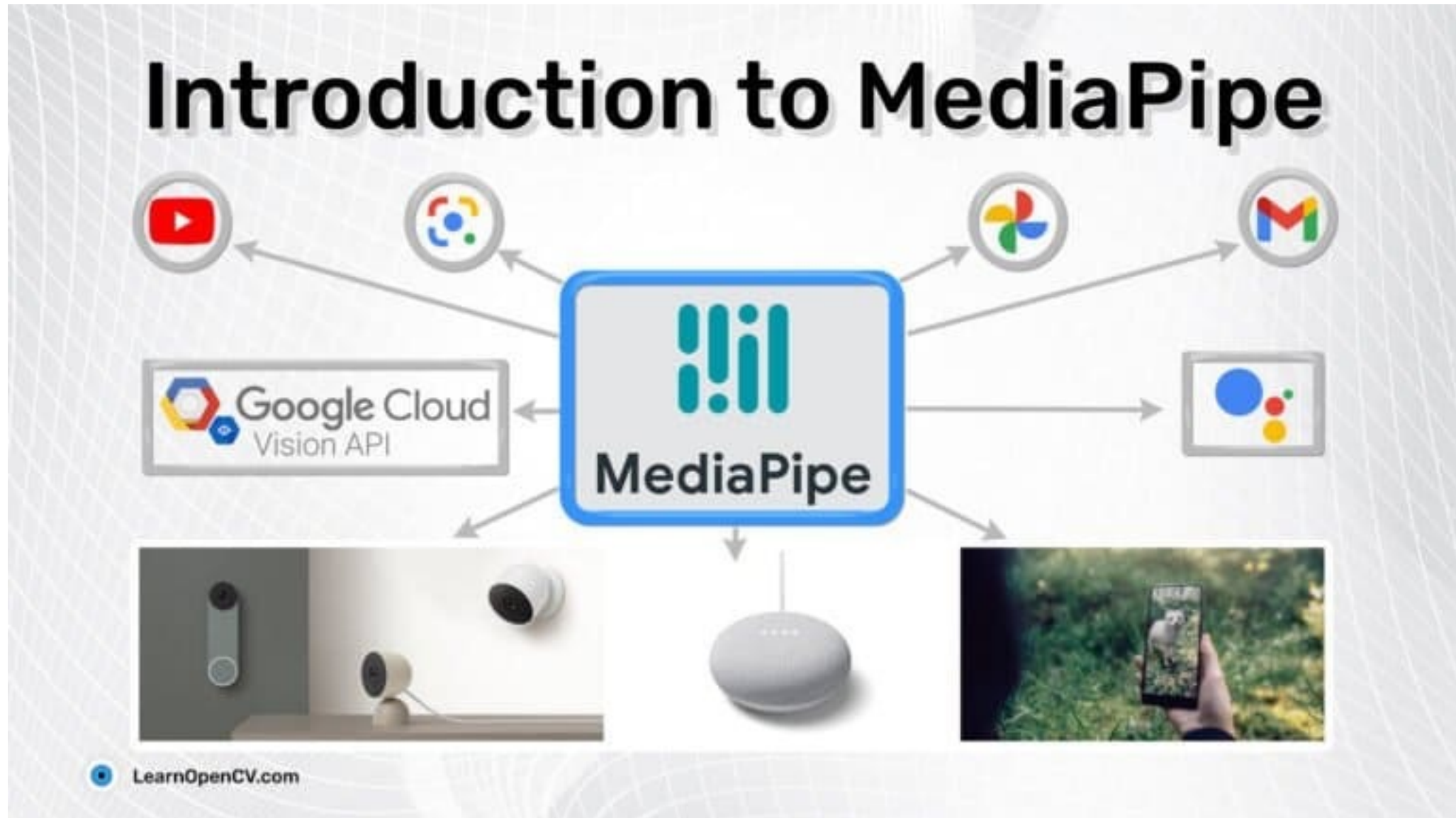


This is the implementation of the approach described in the paper:

Dario Pavlo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3D human pose estimation in video with temporal convolutions and semi-supervised training. In Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

More demos are available at <https://dariopavlo.github.io/VideoPose3D>

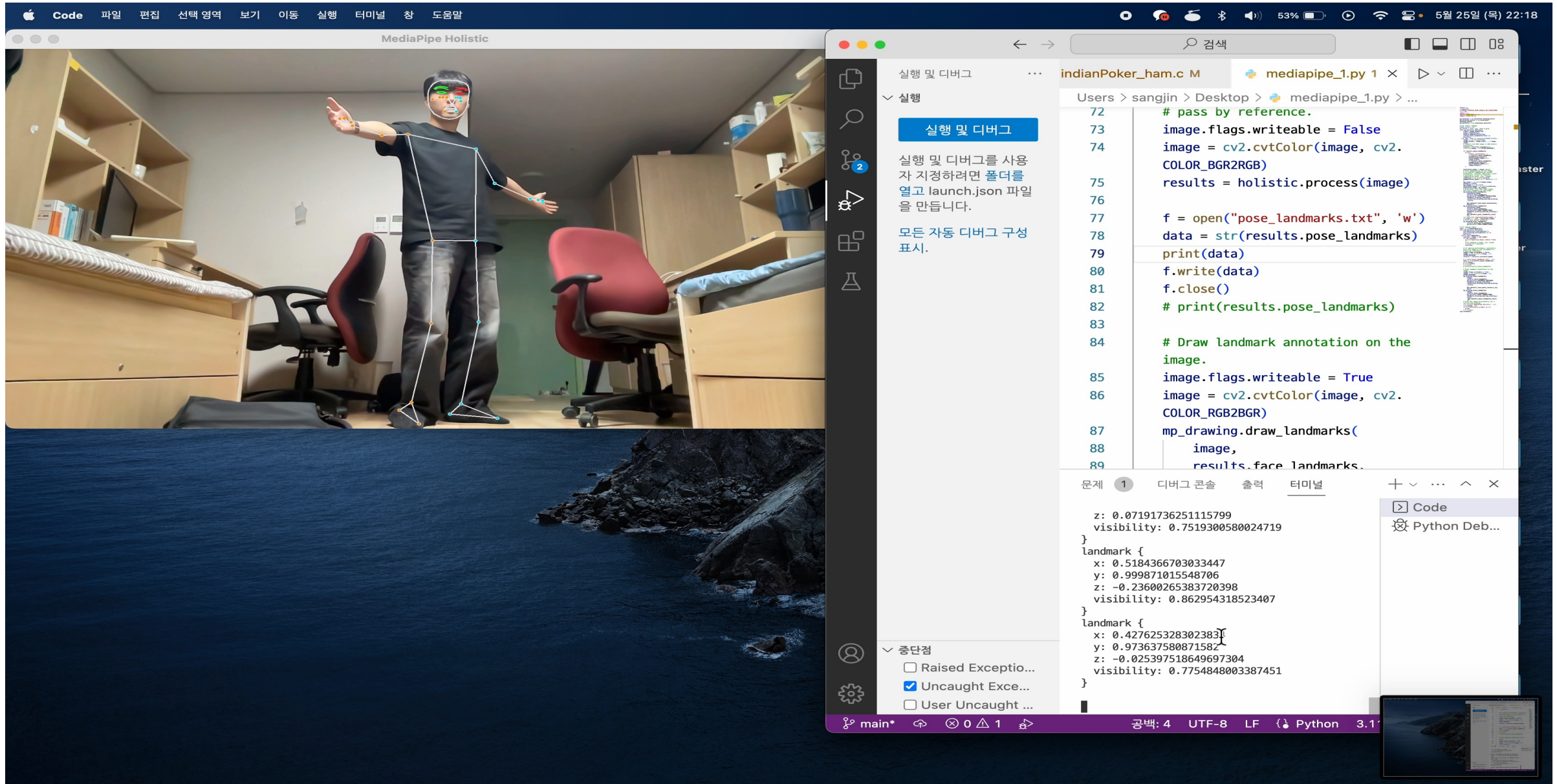
2. MediaPipe



2. MediaPipe (Holistic) – 성능 확인



2. MediaPipe (Holistic) – 좌표 값 확인



The screenshot displays the VS Code interface with the MediaPipe Holistic application running. The left pane shows a video of a person with pose landmarks. The right pane shows the Python code for mediapipe_1.py. The bottom pane shows the terminal output with pose landmarks.

MediaPipe Holistic

실행 및 디버그

실행 및 디버거를 사용자 지정하려면 폴더를 열고 launch.json 파일을 만듭니다.

모든 자동 디버그 구성 표시.

mediapipe_1.py

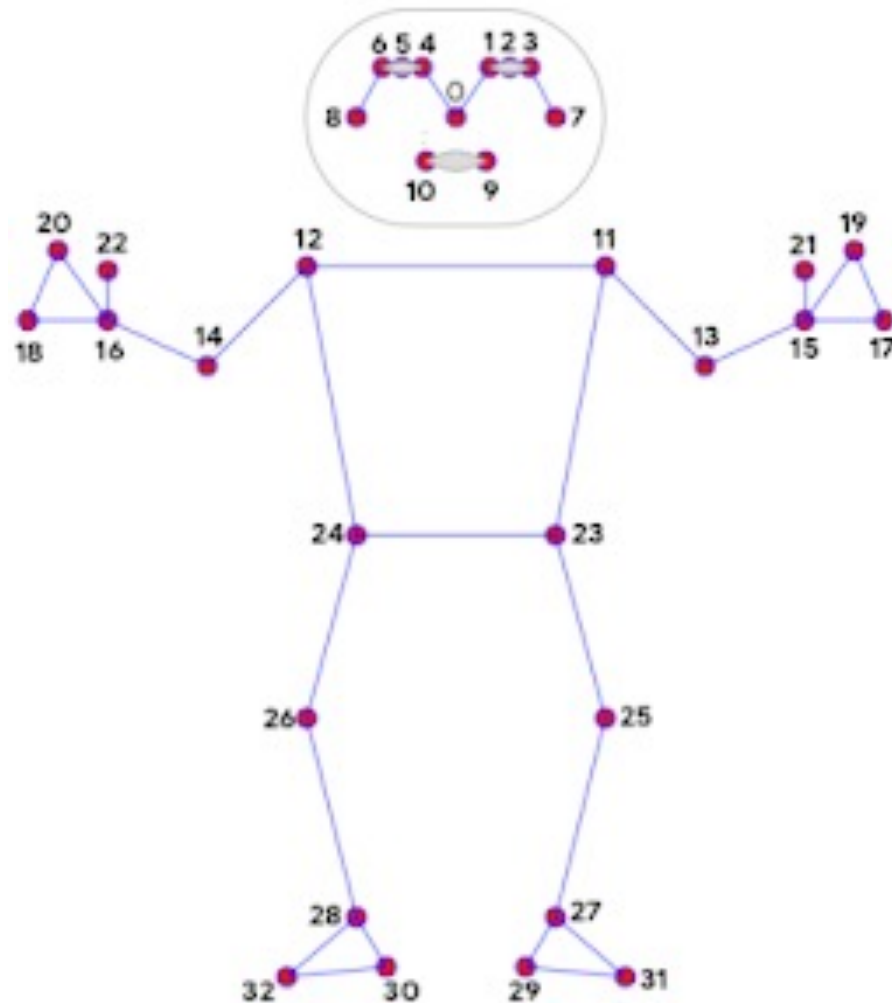
```
72 # pass by reference.
73 image.flags.writeable = False
74 image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
75 results = holistic.process(image)
76
77 f = open("pose_landmarks.txt", 'w')
78 data = str(results.pose_landmarks)
79 print(data)
80 f.write(data)
81 f.close()
82 # print(results.pose_landmarks)
83
84 # Draw landmark annotation on the image.
85 image.flags.writeable = True
86 image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
87 mp_drawing.draw_landmarks(
88     image,
89     results.face_landmarks,
```

문제 1 디버그 콘솔 출력 터미널

```
z: 0.07191736251115799
visibility: 0.7519300580024719
}
landmark {
  x: 0.5184366703033447
  y: 0.999871015548706
  z: -0.23600265383720398
  visibility: 0.862954318523407
}
landmark {
  x: 0.427625328302383
  y: 0.973637580871582
  z: -0.025397518649697304
  visibility: 0.7754848003387451
}
```

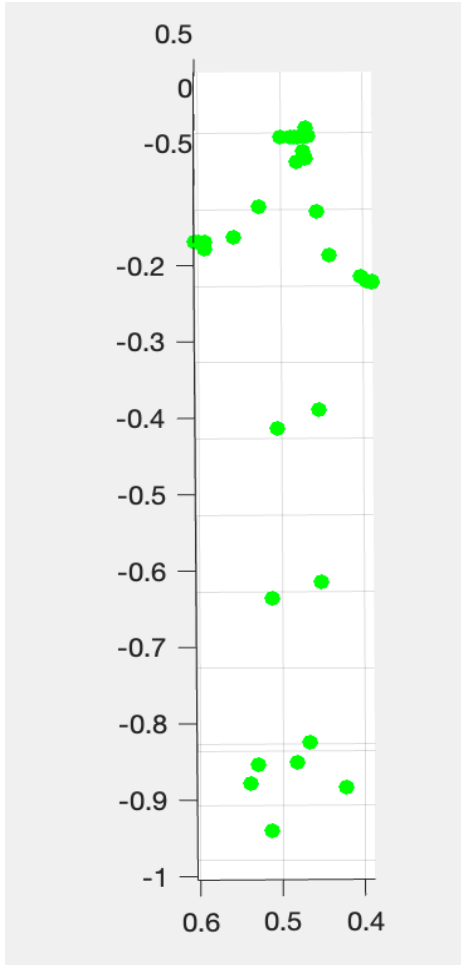
main* 공백: 4 UTF-8 LF Python 3.1

2. MediaPipe (Holistic) – 좌표 값 확인

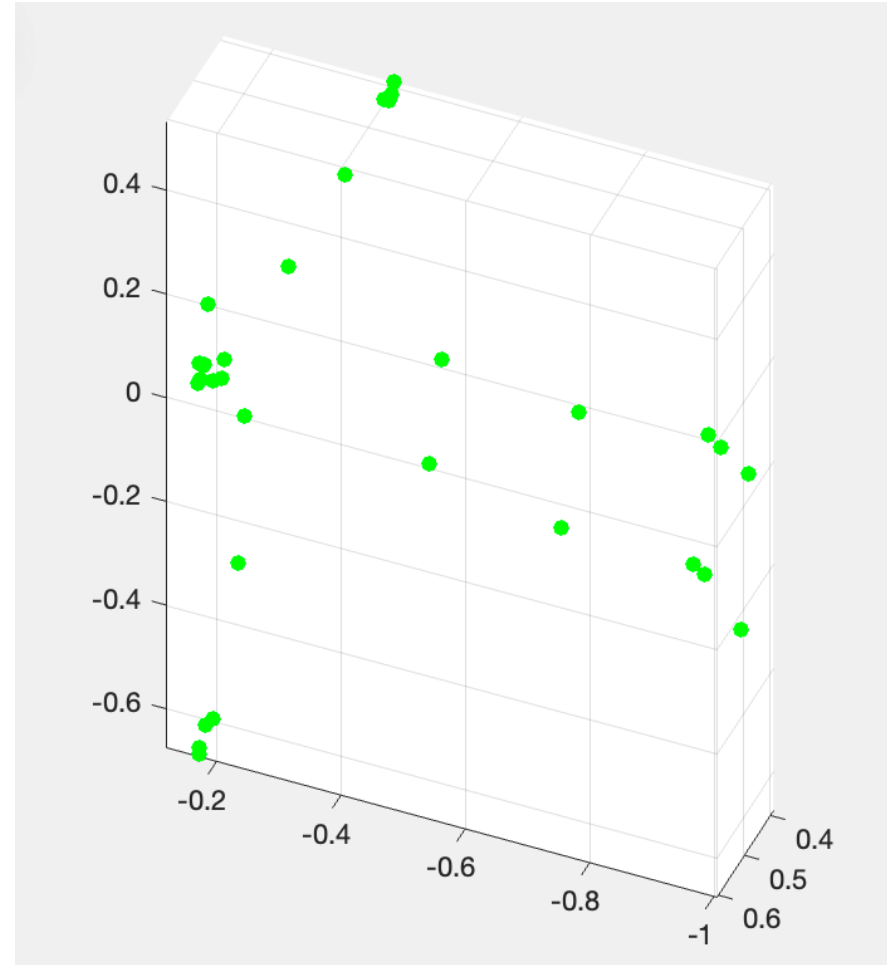


- | | |
|--------------------|----------------------|
| 0. nose | 17. left_pinky |
| 1. left_eye_inner | 18. right_pinky |
| 2. left_eye | 19. left_index |
| 3. left_eye_outer | 20. right_index |
| 4. right_eye_inner | 21. left_thumb |
| 5. right_eye | 22. right_thumb |
| 6. right_eye_outer | 23. left_hip |
| 7. left_ear | 24. right_hip |
| 8. right_ear | 25. left_knee |
| 9. mouth_left | 26. right_knee |
| 10. mouth_right | 27. left_ankle |
| 11. left_shoulder | 28. right_ankle |
| 12. right_shoulder | 29. left_heel |
| 13. left_elbow | 30. right_heel |
| 14. right_elbow | 31. left_foot_index |
| 15. left_wrist | 32. right_foot_index |
| 16. right_wrist | |

2. MediaPipe (Holistic) – 좌표 값 렌더링



정면



측면

결론

1

1명만 인식하는 mediapipe 모델의 단점

2

Human 3.6M 활용 모델 사용