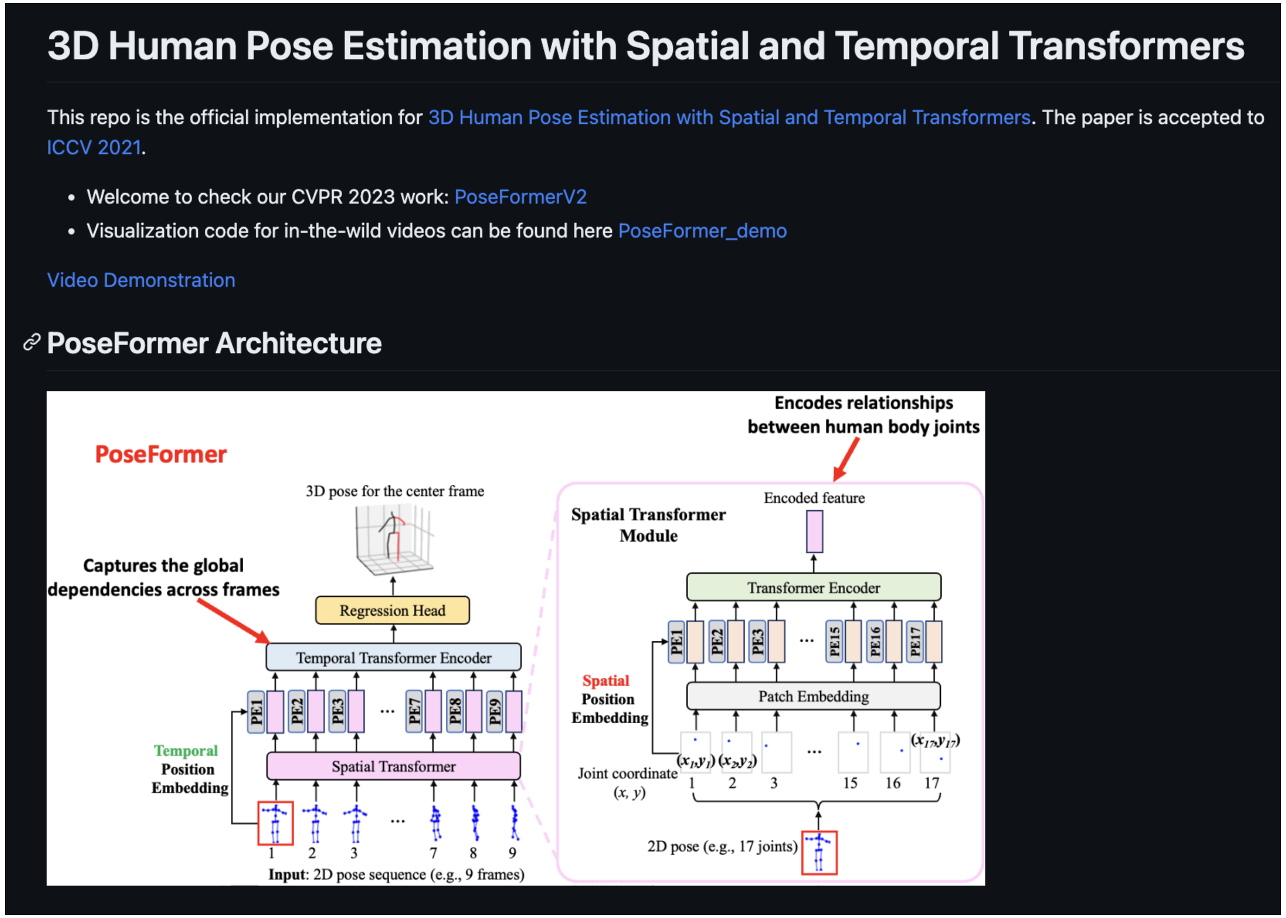
3D Keypoints

2018250033 유규빈
2018250056 함상진

1. Image to 3D keypoints



3D Human Pose Estimation with Spatial and Temporal Transformers

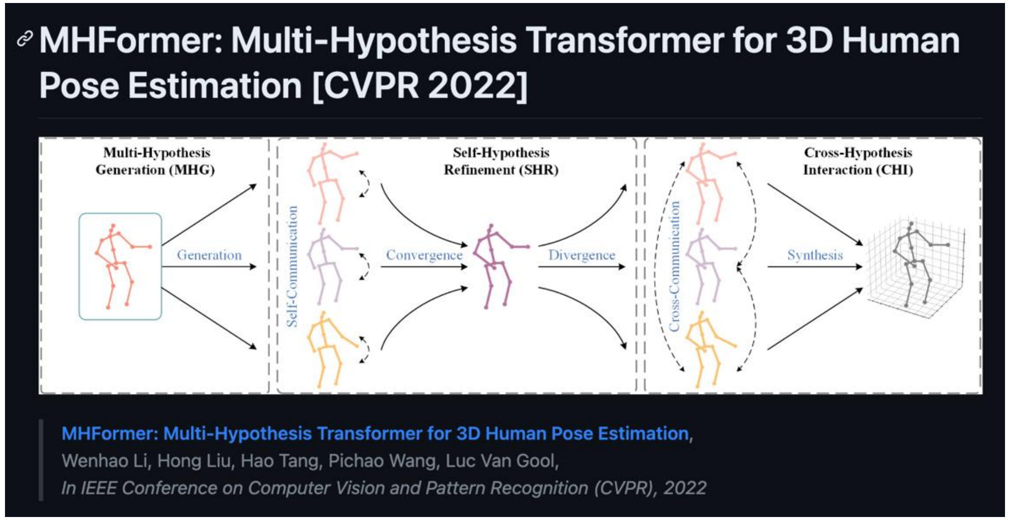
This repo is the official implementation for 3D Human Pose Estimation with Spatial and Temporal Transformers. The paper is accepted to ICCV 2021.

- 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

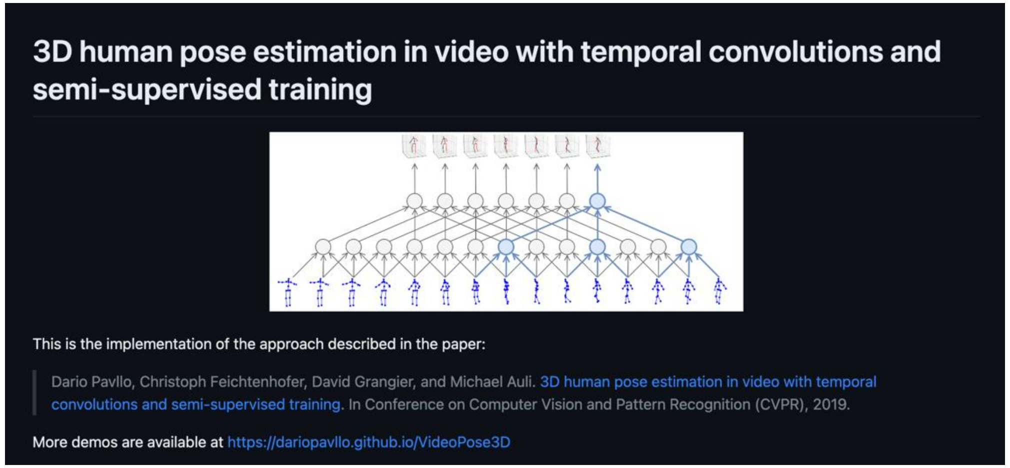
Encodes relationships between human body joints
PoseFormer
3D pose for the center frame
Captures the global dependencies across frames
Regression Head
Temporal Transformer Encoder
PE1
PE2
PE3
...
PE7
PE8
PE9
Temporal Position Embedding
Spatial Transformer
1
2
3
7
8
9
Input: 2D pose sequence (e.g., 9 frames)
Spatial Transformer Module
Encoded feature
Transformer Encoder
PE15
PE16
PE17
Spatial Position Embedding
Patch Embedding
Joint coordinate (x, y)
(x1,y1)
(x2,y2)
(x17,y17)
15
16
17
2D pose (e.g., 17 joints)



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

Multi-Hypothesis Generation (MHG)
Generation
Self-Communication
Self-Hypothesis Refinement (SHR)
Convergence
Divergence
Cross-Communication
Cross-Hypothesis Interaction (CHI)
Synthesis

> 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*



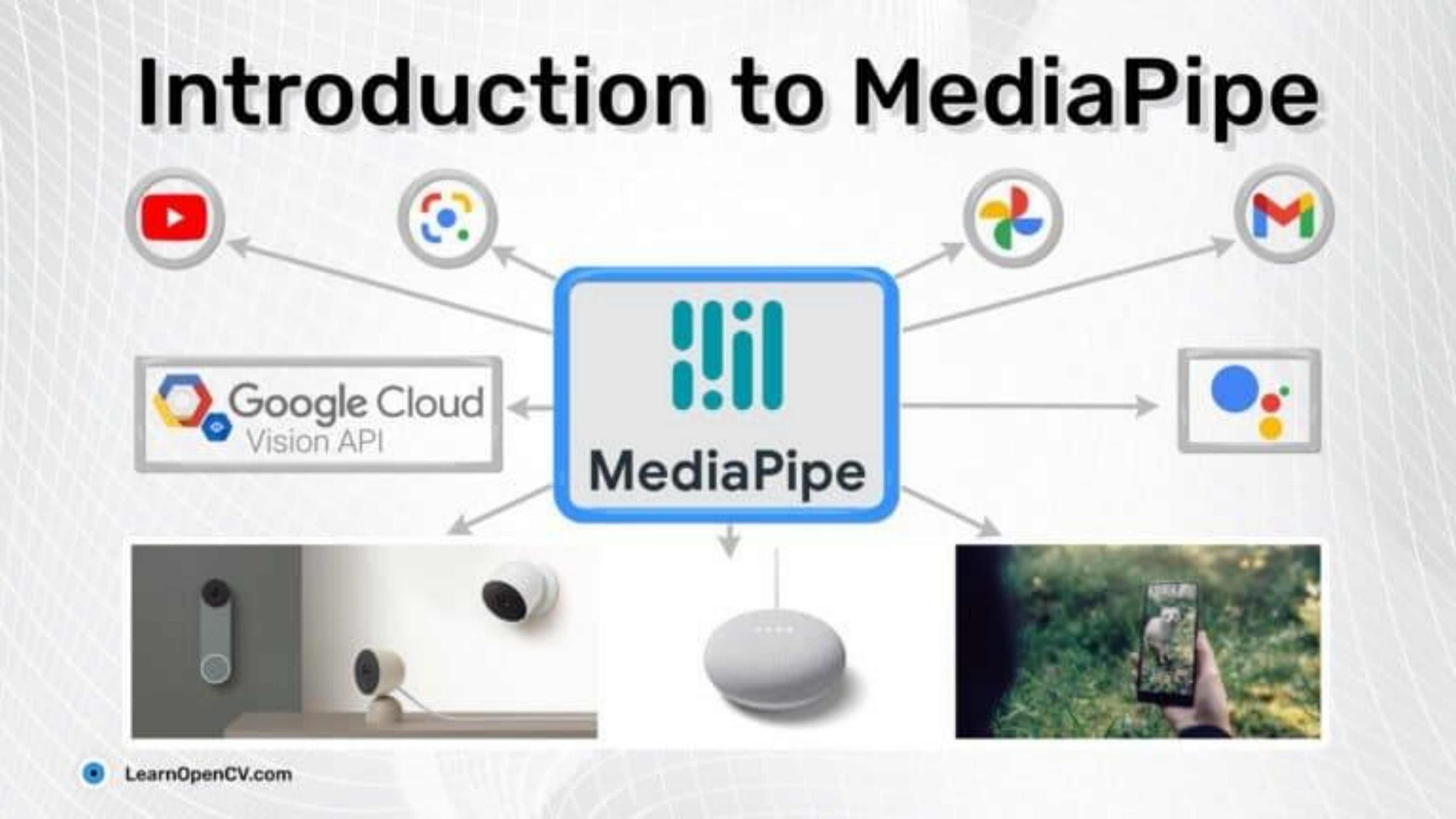
3D human pose estimation in video with temporal convolutions and semi-supervised training

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

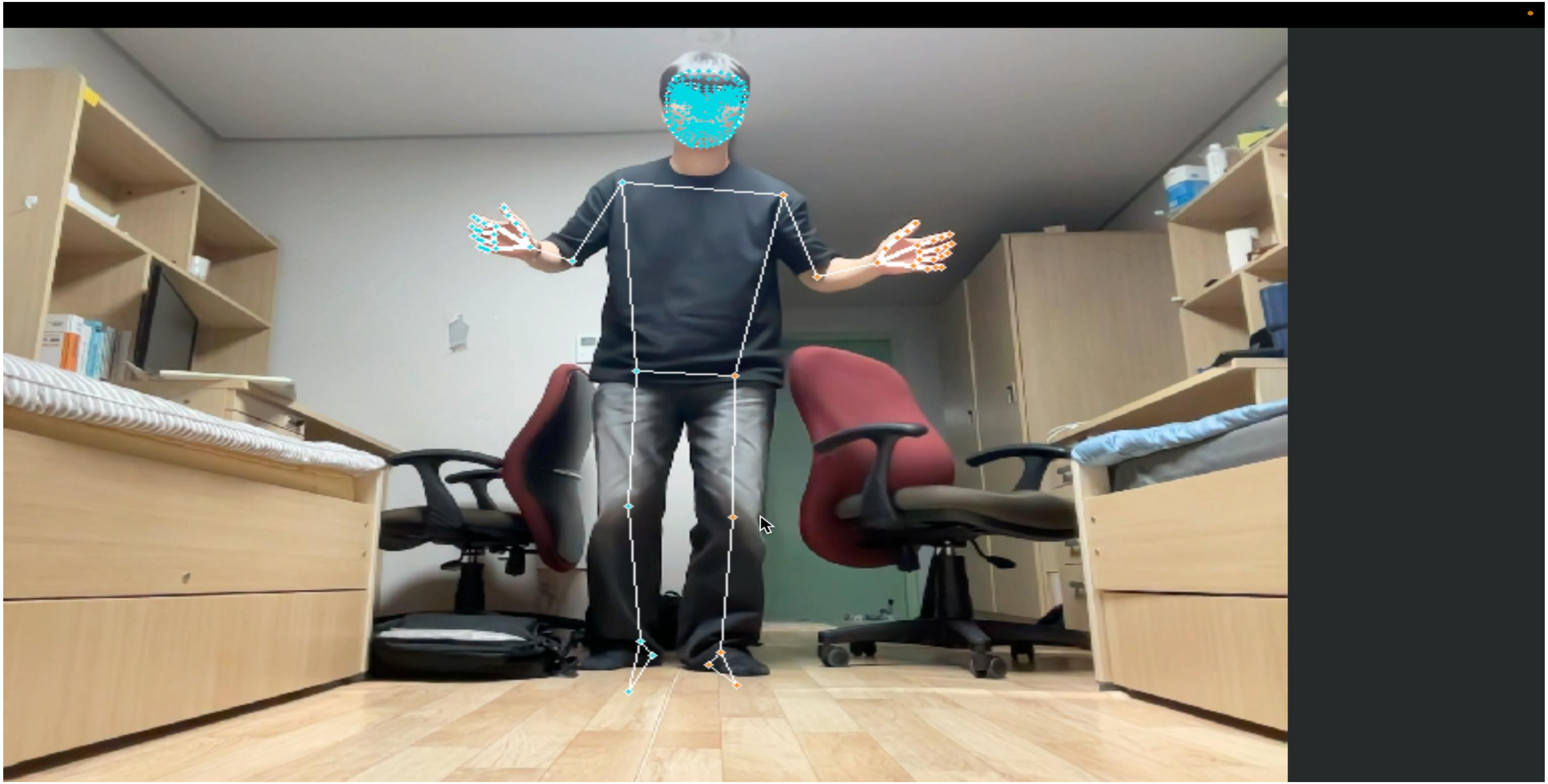
> Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3D human pose estimation in video with temporal convolutions and semi-supervised training. In Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

More demos are available at https://dariopavllo.github.io/VideoPose3D

2. MediaPipe

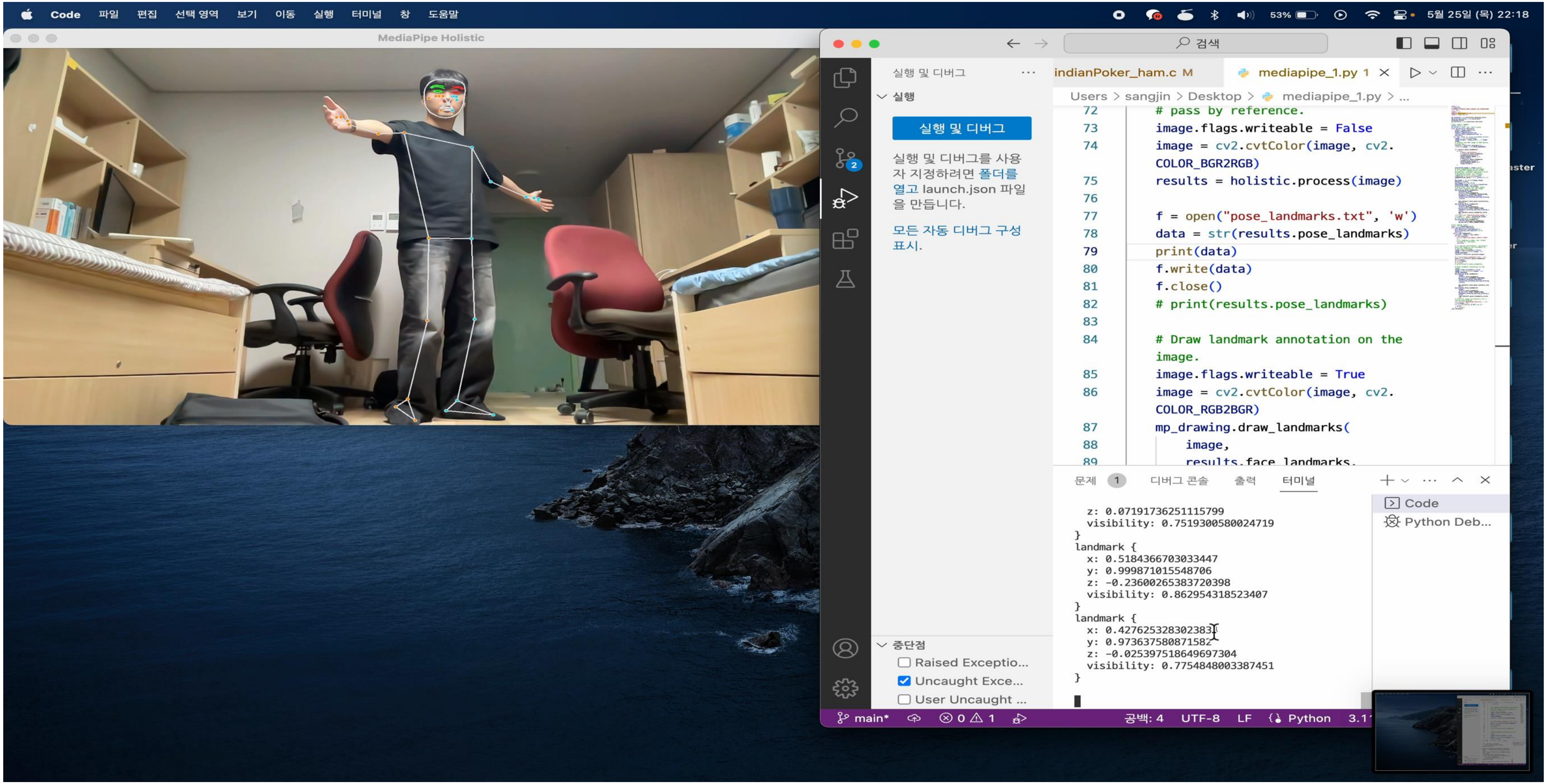

Introduction to MediaPipe
Google Cloud
Vision API
MediaPipe
LearnOpenCV.com

2. MediaPipe (Holistic) – 성능 확인



ⓒSaebyeol Yu. Saebyeol's PowerPoint

2. MediaPipe (Holistic) – 좌표 값 확인



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

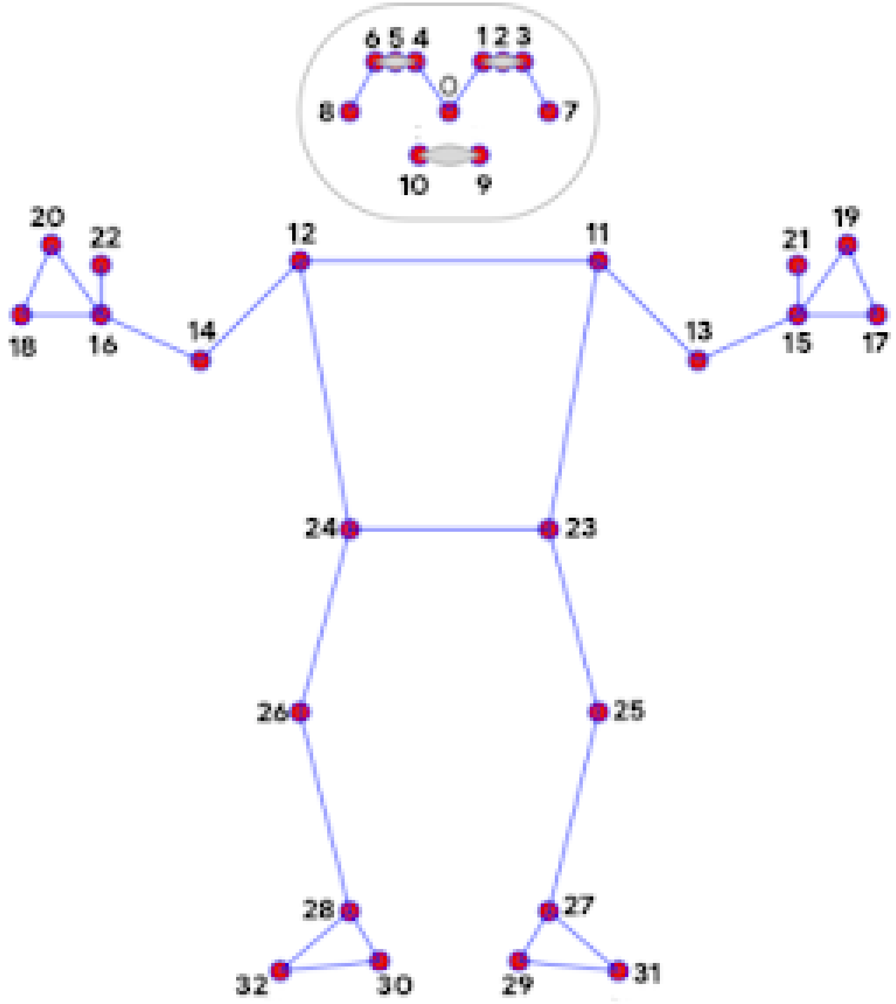
```python
    # pass by reference.
    image.flags.writeable = False
    image = cv2.cvtColor(image, cv2.
    COLOR_BGR2RGB)
    results = holistic.process(image)

    f = open("pose_landmarks.txt", 'w')
    data = str(results.pose_landmarks)
    print(data)
    f.write(data)
    f.close()
    # print(results.pose_landmarks)

    # Draw landmark annotation on the
    image.
    image.flags.writeable = True
    image = cv2.cvtColor(image, cv2.
    COLOR_RGB2BGR)
    mp_drawing.draw_landmarks(
        image,
        results.face_landmarks,
```

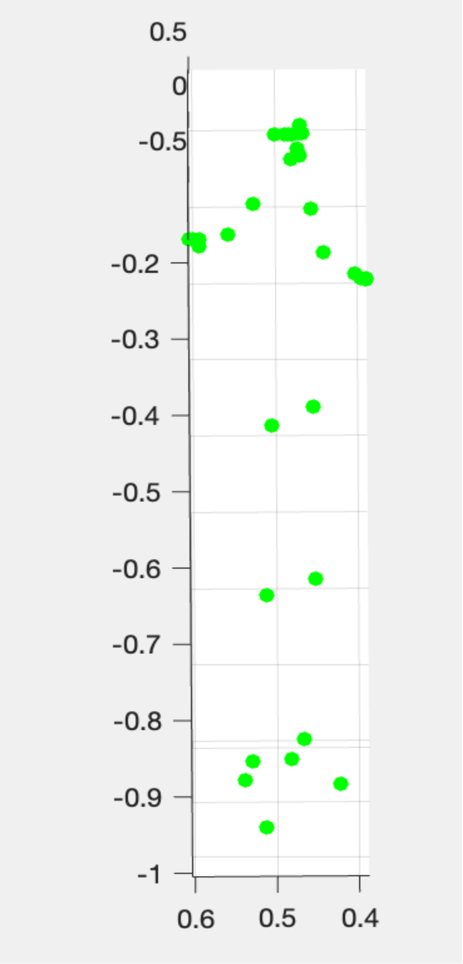
```
  z: 0.07191736251115799
  visibility: 0.7519300580024719
}
landmark {
  x: 0.5184366703033447
  y: 0.999871015548706
  z: -0.23600265383720398
  visibility: 0.862954318523407
}
landmark {
  x: 0.427625328302383
  y: 0.973637580871582
  z: -0.025397518649697304
  visibility: 0.7754848003387451
}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

2. MediaPipe (Holistic) – 좌표 값 확인

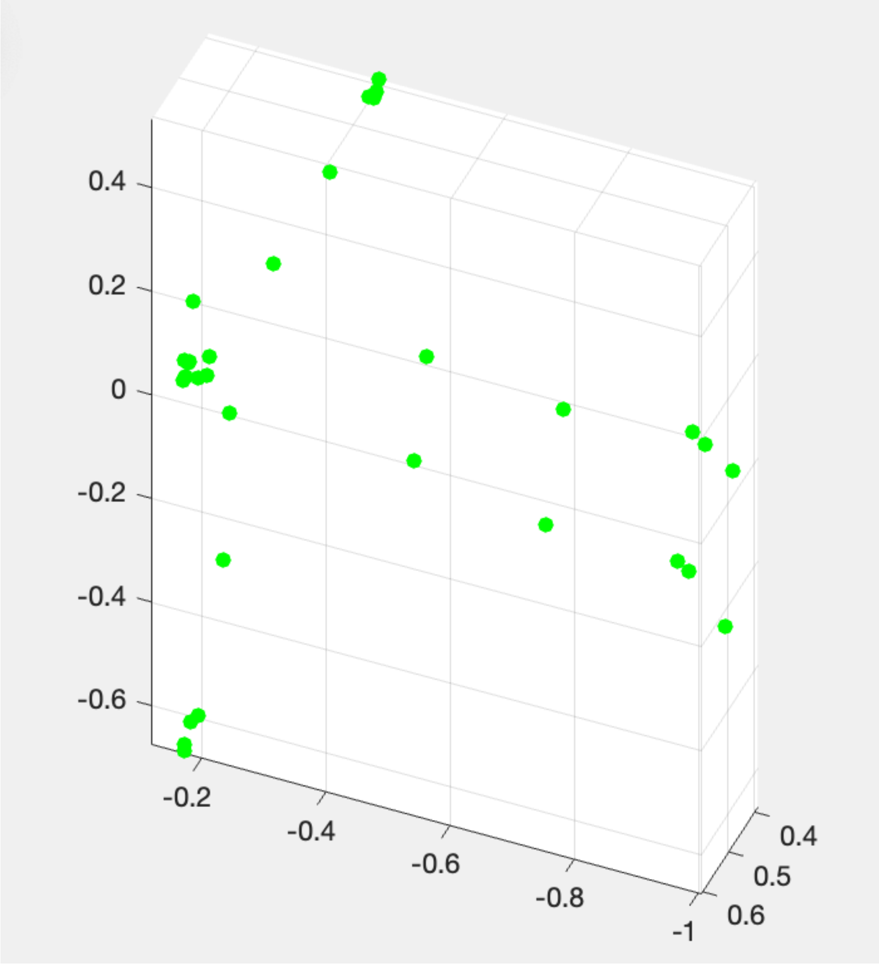


0. nose
1. left_eye_inner
2. left_eye
3. left_eye_outer
4. right_eye_inner
5. right_eye
6. right_eye_outer
7. left_ear
8. right_ear
9. mouth_left
10. mouth_right
11. left_shoulder
12. right_shoulder
13. left_elbow
14. right_elbow
15. left_wrist
16. right_wrist
17. left_pinky
18. right_pinky
19. left_index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

2. MediaPipe (Holistic) – 좌표 값 렌더링



정면



측면

결론

1

1명만 인식하는 mediapipe 모델의 단점

2

Human 3.6M 활용 모델 사용