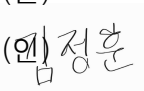


Capstone Design 과제 결과보고서

| | | | | | |
|--|--|-------------|-----------------------------------|--------|---------------------------|
| 유 형 | <input checked="" type="checkbox"/> 전공확장형 <input type="checkbox"/> 창업형 <input type="checkbox"/> 기업연계형 <input type="checkbox"/> 창조IP개발형 | | | | |
| 단과대학명 | 융합학부 | | 소속학과 명 | 인공지능전공 | |
| 참여학기 | <input checked="" type="checkbox"/> 1학기 <input type="checkbox"/> 2학기 | | <input type="checkbox"/> 1학기, 2학기 | | |
| 과제분야 | 보고서형 <input type="checkbox"/> 제 작 형 (H/W 형 <input type="checkbox"/> , S/W 형 <input checked="" type="checkbox"/>) | | 심화형 <input type="checkbox"/> | | |
| 팀 명 | AI DreamTeam | | 구성인원 | 총 2명 | |
| 지도교수 | 성 명 | 정 호 영 | | | |
| * 기업체 멘토 (기업연계형 필수) | 기업명 | - | 담당자명 | - | |
| 작품명 | 카메라를 통한 신체 인식, 분류 및 예측 모델 개발 | | | | |
| 과제수행 업무분담 | 구 분 | 학 번 | 소 속 | 이 름 | 담당역할 |
| | 팀 장 | 2017112066 | 인공지능학과 | 김정훈 | 자료 조사, 제품 설계, 코드 작성 |
| | 팀 원 | 2017113770 | 인공지능학과 | 강명철 | 자료 조사, 제품 설계, 발표 자료 제작 |
| | 팀 원 | | | | |
| | 팀 원 | | | | |
| | 팀 원 | | | | |
| | 팀 원 | | | | |
| 지원금 총 집행금액 | 세부항목 | 품목 및 내용(요약) | | 금 액(원) | 비 고 |
| | 재료비, 회의비 등 | 클라우드 서버 대여비 | | 3700 | 공통지원 |
| | 자문료 | - | | - | 별도지원 |
| | 최종결과 발표회 | - | | - | |
| | 합 계 | | | (원) | |
| 위와 같이, 경북대학교에서 수행한 캡스톤디자인 결과보고서를 제출합니다. | | | | | |
| 2022 . 06. 15. | | | | | |
| 지도교수 : 정호영 (인) | | | | | |
| 팀 장 : 김정훈 (인)  | | | | | |
| 단과대학장 귀하 | | | | | |

1. 목적 및 필요성

1.1 개발 목적

최근 딥러닝을 활용한 이미지 인식 기술의 발전으로, Google사의 Mediapipe나 Nvidia사의 BodyPoseNet과 같은 카메라를 통한 신체 인식 API들이 출시되었다. 특히 적은 연산량을 요구해 스마트폰이나 저사양 PC와 같은 환경에서도 인식된 데이터를 바탕으로 클래스 분류 작업을 수행하는 등의 작업을 수행할 수 있게 됨으로써 접근성이 크게 증가하였다. 제안서 작성 단계에서는 손동작을 분류하는데 초점을 두었으나, 신체 동작을 분류하는 것이 보다 범용성이 높다고 생각되었다. 최종적으로, 실생활에 적용 가능한 형태로, 사람의 신체 관절의 위치를 인식한 뒤 동작의 종류를 구분할 수 있게 하는 어플리케이션 형태의 신체 동작 분류 시스템을 제작하고자 한다.

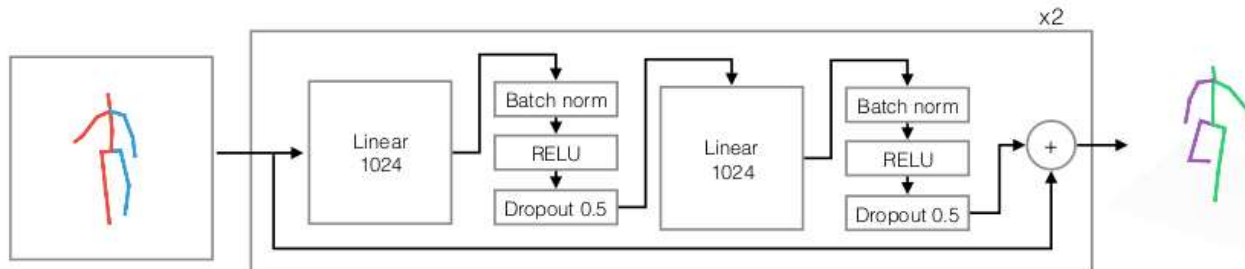
1.2 개발 필요성

신체 인식 및 동작 분류 시스템을 통해 다른 분야에 응용 가능성이 있다. 스포츠 선수와 플레이를 인식하고 경기 상황을 데이터로 수집하여 객관적인 경기 분석을 제공하는데 활용하거나, 헬스케어 등 환자모니터링 시스템, 낙상방지 시스템 등을 포함한 분야에 응용 가능하다. 또한 최근에는 인식된 신체 데이터를 바탕으로 사람의 움직임을 3D 공간으로 정확하게 투영하는 기술을 추가하여 현실 세계와 같은 메타버스를 구축하는데 활용될 수 있다.

2. 기술 현황

A simple yet effective baseline for 3d human pose estimation (Julieta Martinez. et al., 2017)

- ResNet 모델을 기반으로 한 CNN 학습을 통해 신체 동작을 인식함

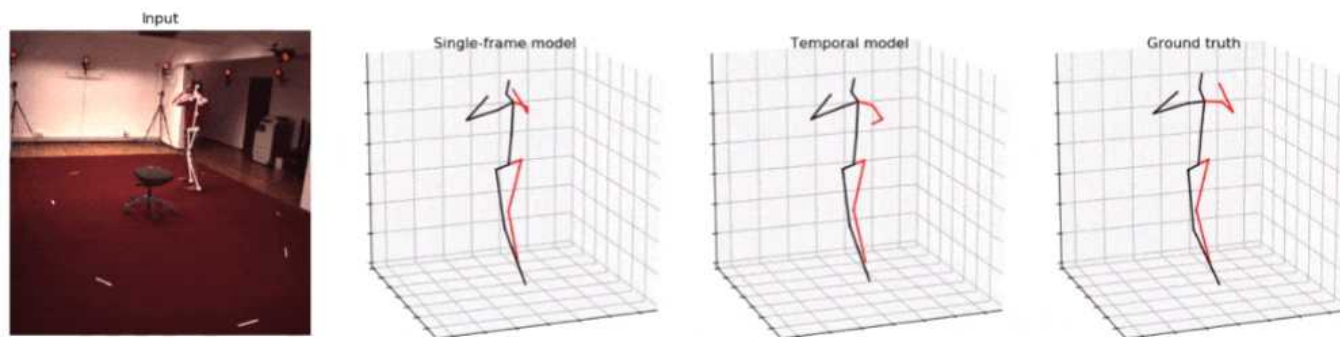


< 그림 1 > 출처: [1]

3D human pose estimation in video with temporal convolutions and semi-supervised training (Dario Pavllo. et al., 2019)

- TCN 모델을 기반으로 한 CNN 학습을 통해 신체 동작을 인식함

- 2D 추정을 3D 추정으로 변환시키는데 유효함, 따라서 2D 추정을 수행하는 시스템이 별도로 필요함



< 그림 2 > 출처: [2]

Mediapipe Pose (2020)

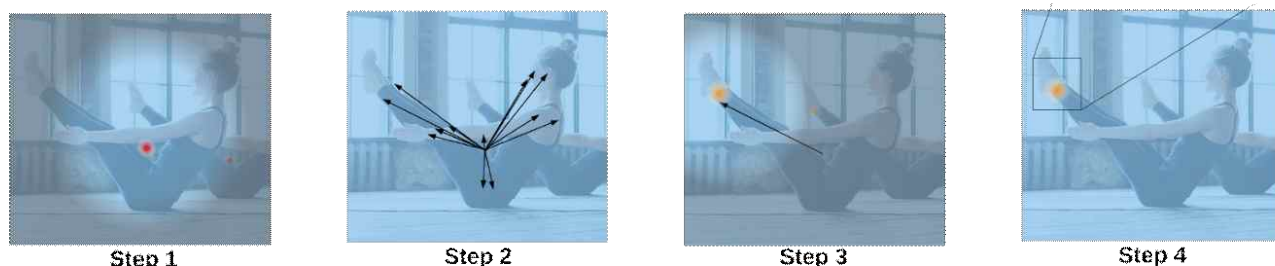
- Google에서 공개한 API로, BlazePose 모델을 사용하여 object segmentation을 수행한 이미지에서 관절 랜드마크를 추적하는 방법으로 신체 동작을 추정하는 모델을 포함



< 그림 3 > 출처: [3]

MoveNet (2021)

- tensorflow에서 공개한 API로, 이미지에서 사람의 키포인트를 찾고, 관절 랜드마크를 추적하는 방법을 통해 신체 동작을 추정하는 모델을 포함



< 그림 4 > 출처: [4]

조사결과를 바탕으로 과제 수행에 사용할 API로 Mediapipe와 MoveNet 둘 중 하나를 사용하기로 논의했고, Mediapipe는 아직 베타테스트 단계의 API인 점, 처리된 데이터에 별도의 처리과정이 더 필요하다는 점을 들어 최종적으로 MoveNet을 사용하기로 선정함.

3. 과제 해결 방안 및 과정

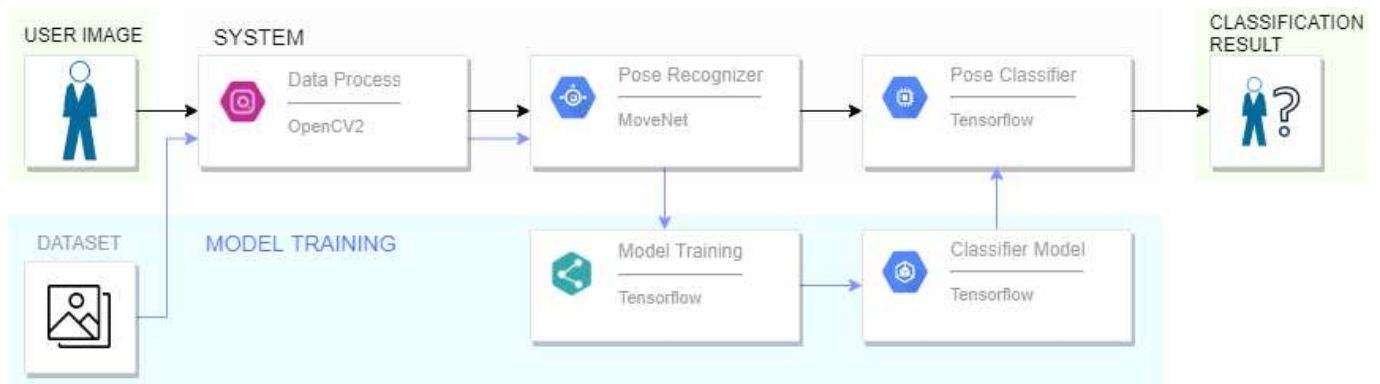
3.1 베이스라인 제작

개발 목적은 사람의 신체를 인식해서 분류 결과를 표시하는 것이므로, 각각을 입력과 출력으로 설정했다. 이 과정을 수행할 시스템에 적합한 베이스라인을 구성하고자 했다. 시스템은 크게 세 부분으로 구성되어있다.

| | Input | Output |
|-----------------|---------|----------|
| Baseline | 촬영 이미지 | 동작 분류 결과 |
| Data Process | 촬영 이미지 | 전처리 이미지 |
| Pose Recognizer | 전처리 이미지 | 벡터화 데이터 |
| Pose Classifier | 벡터화 데이터 | 동작 분류 결과 |

- Data Process: 카메라로 촬영된 영상을 시스템이 처리할 수 있도록 변환한다.
 - Pose Recognizer: 전처리된 이미지 데이터에서 사람을 인식하고, 각 관절의 위치를 분석하여 이를 벡터값으로 출력한다.
 - Pose Classifier: 벡터값을 입력으로 받아 학습된 모델을 통해 동작을 분류하고, 결과를 출력한다.
- 코드 작성은 python으로 진행하였고, Data Process와 Pose Recognizer는 공개된 API를 사용하고, Pose Classifier에 사용될 분류 모델은 tensorflow를 통해 직접 학습시켰다.

최종적으로 구성한 베이스라인은 다음 그림과 같다. Pose Classifier에 필요한 분류 모델을 제작하기 위해 Model Training 과정을 수행하고, 이를 바탕으로 System을 통해 실제 과제를 수행한다.



< 그림 5 >

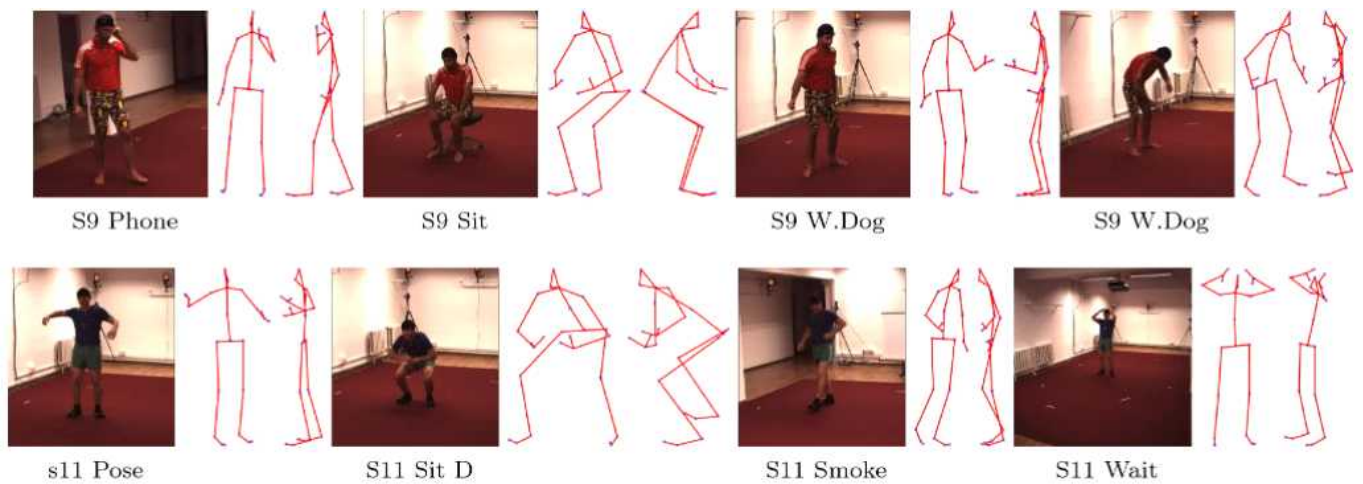
3.2 학습 모델 선정

Pose Recognizer 부분을 구현하기 위해, Google Research에서 공개된 API인 MoveNet을 사용하기로 했다. Pose Classifier 부분에 필요한 분류 모델은 Tensorflow를 통해 구현시켰다.

3.3 학습 데이터셋 선정

분류 모델 학습을 위해 사용할 이미지들을 모은 데이터셋을 수집했으며, 공개된 데이터셋들 중 이용 가능하고 학습에 적합한 데이터셋들을 선택했다.

| 데이터셋 | 장수 | 동작 분류 수 | 비고 |
|----------------------------|---------|---------|-------------|
| Human 3.6m <그림 6> | 3600000 | 16 | 11명의 배우가 촬영 |
| Penn Action Dataset <그림 7> | 164000 | 15 | 불특정 다수의 인물 |



< 그림 6 > 출처: [5]



< 그림 7 > 출처: [6]

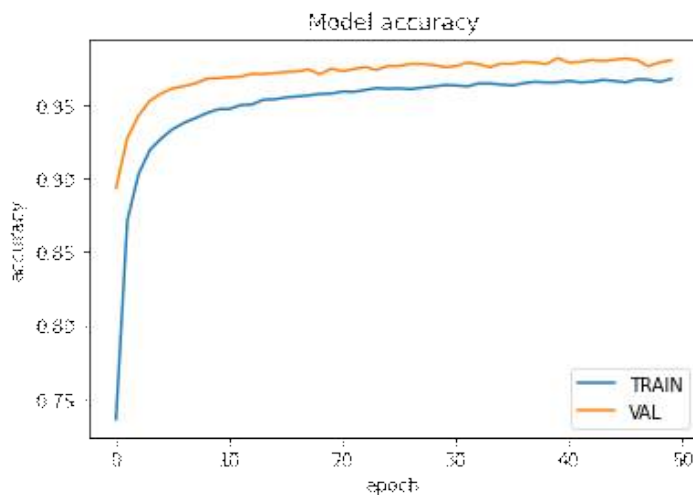
3.4 학습 모델 평가

수집한 데이터 셋에서 Train 데이터와 Test 데이터를 여러 조건으로 선별하여 학습을 진행했다. 시간 제약상 모든 데이터를 활용하지는 않았고, 데이터 외에는 다른 모든 조건은 동일하게 사용하였다. Overfitting 문제를 방지하기 위해, Train error가 개선되지 않는다면 학습을 중단하고 Test를 했다.

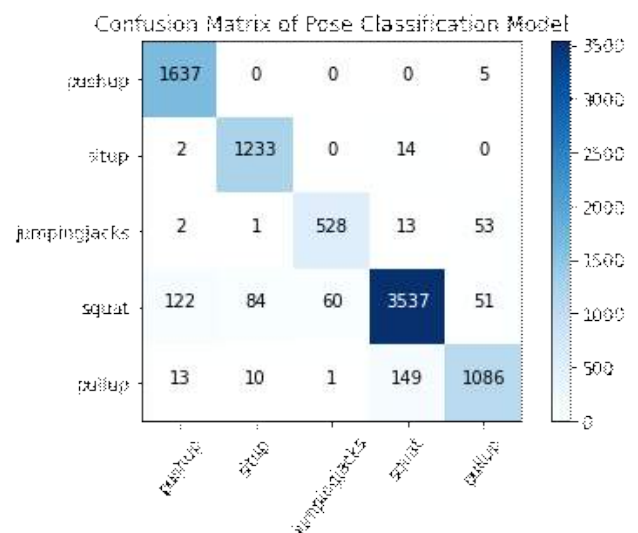
처음 Human 3.6m 데이터셋을 가지고 여러 방법으로 학습을 시켰을 때 오차율이 현저히 높게 나오는 문제가 계속되었다. TRIZ 기법 중 사전반대조치와 추출 원리를 이용하여, 실제 Test 환경에서는 여러 가지 변수가 나올 것을 감안해, 보다 여러 가지 배경과 촬영각도가 포함된 Penn Action 데이터를 사용하고, Label 수가 많아 분류가 잘 되지 않는다고 생각해 Label의 일부만 추출하여 학습시켰다.

| 분류 모델 | Train 데이터 (수) | Test 데이터 (수) | 동작 갯수 | Label | Train error | Val. error | Test error | 비고 |
|-------|---|---|-------|-------|-------------|------------|------------|-------|
| A | Human 3.6m 中 Subject01/Subact01 (29996) | Human 3.6m 中 Subject01/Subact01 (29996) | 15 | | 0.20 | 0.09 | 0.43 | |
| B | Human 3.6m 中 Subject01/Subact02 (45223) | Human 3.6m 中 Subject05/Subact02 (29996) | 15 | | | | 0.68 | |
| C | Penn Action 中 5개 동작의 80% (43991) | Penn Action 中 5개 동작의 20% (10239) | 5 | | 0.03 | 0.02 | 0.07 | 최종 선택 |

<그림 9>에 분류모델 C의 학습 과정에서 Train 정확도와 Validation 정확도를 매 epoch마다 나타내었다. Validation 데이터는 Train 데이터에서 무작위로 15% 추출해서 검증되었다. Overfitting 문제를 방지하기 위해 정확도 개선이 멈추는 50 epoch 정도에서 학습을 중단했다. <그림 10>에 학습된 모델의 Test 과정에서 분류의 정확도를 나타내었다. 가로축은 추측한 동작, 세로축은 실제 동작을 나타낸다. Test 정확도는 93%로 측정되었다.



< 그림 12 >, 분류 모델 C



< 그림 13 >, 분류 모델 C

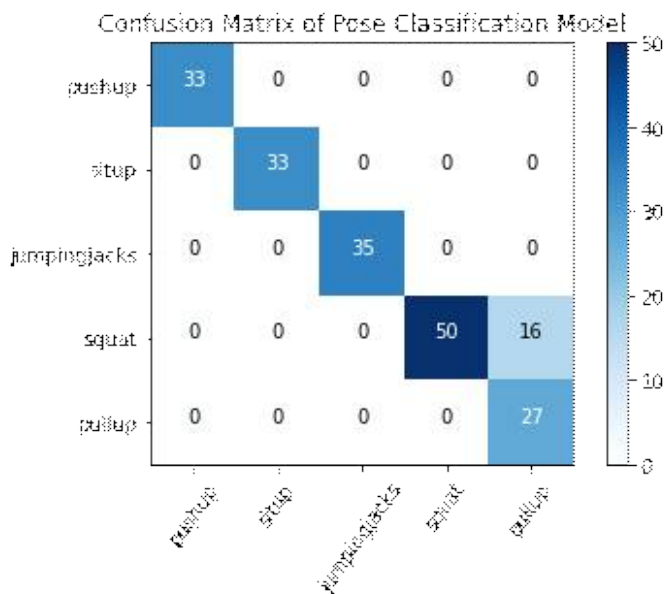
3.5 최종 결과물

최종적으로 오차율이 가장 낮았던 학습모델 C를 사용해 시스템을 시연했다. 실제 상황에서 시스템의 오차를 검증하기 위해, Test 데이터에 노트북 웹캠으로 직접 촬영한 243장의 5종류의 동작 모델을 구성하고 시스템을 시연하였다. <그림 11>에 촬영한 데이터의 일부를 나타내었다.

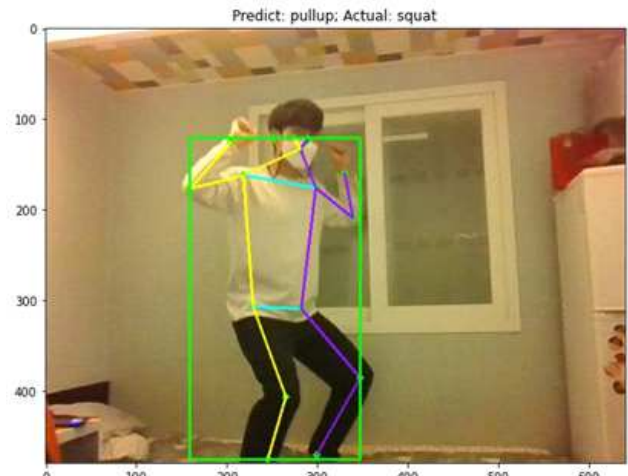


< 그림 11 >

<그림 12>을 통해 분류의 정확도를 나타내었다. 가로축은 추측한 동작, 세로축은 실제 동작을 나타낸다. 종류 중 squat를 제외한 모든 동작에서 정확한 분류를 수행했다. 결과 보안을 위해 <그림 13>과 같이 잘못 분류된 이미지를 수집했으며, 두 동작이 유사하기 때문이라고 추측했다.



< 그림 12 >



< 그림 13 >

4. 기업체 멘토 역할

-해당없음.

5. 기대 효과 및 활용 방안

기존의 모션 캡처 방식은 수십대의 광학 장비와 관성 센서등이 필요해 많은 비용이 드는 문제점이 있음. 딥러닝 기법을 활용한 시스템 설계를 통해 카메라만을 사용해서 저비용으로 대체할 수 있는 방법을 제시했다.

과제 설계에서 제시한 데이터셋 이외에 다른 데이터셋을 사용한다면 다른 여러 가지 분야에 최적화하여 사용이 가능할 것으로 예상된다. 기존 Penn Action 데이터셋과 같은 운동과 관련된 데이터를 학습시킨 후 특정 운동을 분류하고 시작과 끝을 체크하는 방법으로 동작 횟수를 체크하거나, 요가, 필라테스 데이터를 학습시켜서 사용자가 정확하게 동작을 수행하고 있는지, 얼마나 동작을 수행했는지 등을 피드백함으로써 실생활에 유용한 형태로 기능을 추가로 제작할 수 있을 것으로 예상된다.

또한 3D 동작을 추정하는 기술[2]에 본 과제의 출력 결과를 입력으로 사용할 수 있게 한다면, 향후 증강현실 및 가상현실 분야에서 메타버스 아바타를 구현할 때 적용시켜서 아바타의 모션을 생성하는 등의 활용 가능성이 있을 것으로 보인다.

6. 한계점 및 향후 개선사항

시스템 처리속도에 한계가 있어, 실시간으로 구동하는 목표는 달성하지 못했다. Jetson nano와 같은 AI처리 전용 하드웨어에 시스템을 이식하거나, 입출력 이미지의 해상도를 떨어트리거나, 인식 부분의 데이터 전처리를 정확도를 조금 희생해서 분류 부분으로 최대한 빠르게 입력시키는 등의 여러 가지 방법을 통해 실시간으로 동작하게 할 수 있을 것으로 보인다.

또한 여러 가지 학습 데이터를 구성하면서, 동작의 종류가 많으면 동작을 잘못 분류하여 분류기의 성능이 급격하게 떨어지는 문제점이 있었다. 이 문제점은 분류 모델을 학습시킬 때 사용한 데이터셋을 분석해서, 사람을 가리는 장애물이 있거나 여러 사람이 나오는 등, 배경이 단순하고, 장애물이 없고 한 사람만 있을 때 인식이 최적화되는 데이터 전처리 조건 자체를 벗어나는 데이터들을 제외시켜 보다 정제된 데이터를 통해 오차율을 줄여서 학습하는 방법으로 해결할 수 있을 것으로 보인다.

7. 참고문헌

- [1] A simple yet effective baseline for 3d human pose estimation (Julieta Martinez. et al., 2017)
- [2] 3D human pose estimation in video with temporal convolutions and semi-supervised training (Dario Pavllo. et al., 2019)
- [3] <https://ai.googleblog.com/2020/08/on-device-real-time-body-pose-tracking.html>
- [4]<https://blog.tensorflow.org/2021/05/next-generation-pose-detection-with-movenet-and-tensorflowjs.html>
- [5] <http://vision.imar.ro/human3.6m/description.php>
- [6] <http://dreamdragon.github.io/PennAction/>