Pattern Recognition - 프로젝트 최종 보고서

과 제 명	카메라를 통한 사람의 3차원 자세 추정 및 예측		
학 번	2017112066	성 명	김정훈
소속학과	경북대학교 융합학부 인공지능학과		
연구기간	2022. 05. 03. ~ 06. 12.		

1. 연구의 목적 및 필요성

1) 연구의 목적

- 본 연구의 목적은 카메라로 촬영된 디지털 이미지 (source)를 입력 받아서, 이미지에 존재하는 사람 한 명의 자세 (target)을 분석해서 3차원상에 나타낼 수 있는 방법을 제시함
- 별도의 추가 장비 없이 카메라만을 이용해, 머신 러닝 기법을 활용하여 자세를 관절 단위의 위치로 추적할 수 있음

2) 연구의 필요성

- 포즈(자세) 추정은 사람의 위치나 방향을 감지하는게 목표로, 컴퓨터 비전 분야에서 일반적인 문제로, 손, 머리, 팔꿈치 등과 같은 특정 키포인트(관절)의 위치를 예측해서 수행함
- 기존의 모션 캡쳐 방식은 배우와 수십대의 광학 장비와 관성 센서등이 필요해 많은 비용이 드는 문제점이 있음. 이를 머신 러닝 기법을 활용해, 평면 RGB 또는 RGB-D 이미지를 사용해서 저비용으로 대체할 수 있는 방법을 제시함
- 향후 증강현실 및 가상현실 분야에서 메타버스 아바타를 구현할 때 본 연구의 포즈 추정을 적용시켜서 아바타의 모션을 생성하는 등의 활용 가능성이 있음

2. 선행 연구 및 기술

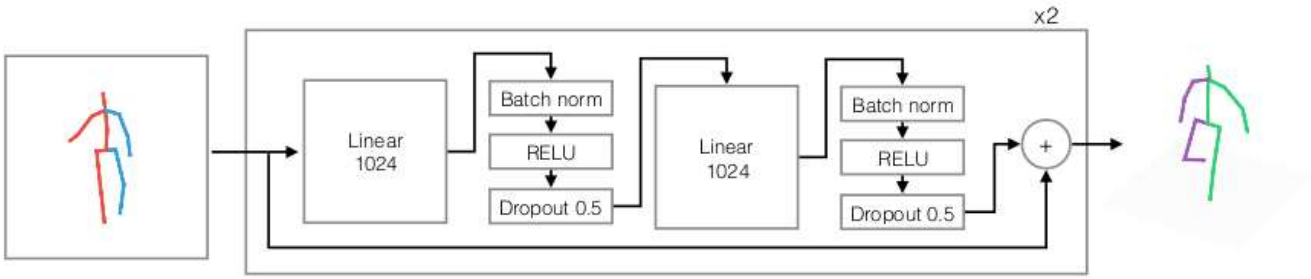
1) 기존 방법 분석

- 3D포즈를 추정하는 접근 방법은 두가지가 있는데, (1)2D포즈를 추정하고 3D포즈로 재건(reconstruct)하는 방법과 (2)바로 3D포즈로 회귀(regress)하는 방법이 있음
- (2)방법은 이미지의 깊이(Depth)정보가 필요하므로, 아닌 경우에 대해서도 적용할 수 있도록 범용성을 고려해 (1)의 접근방법을 사용

1) 선행 연구

1.1 A simple yet effective baseline for 3d human pose estimation[1]

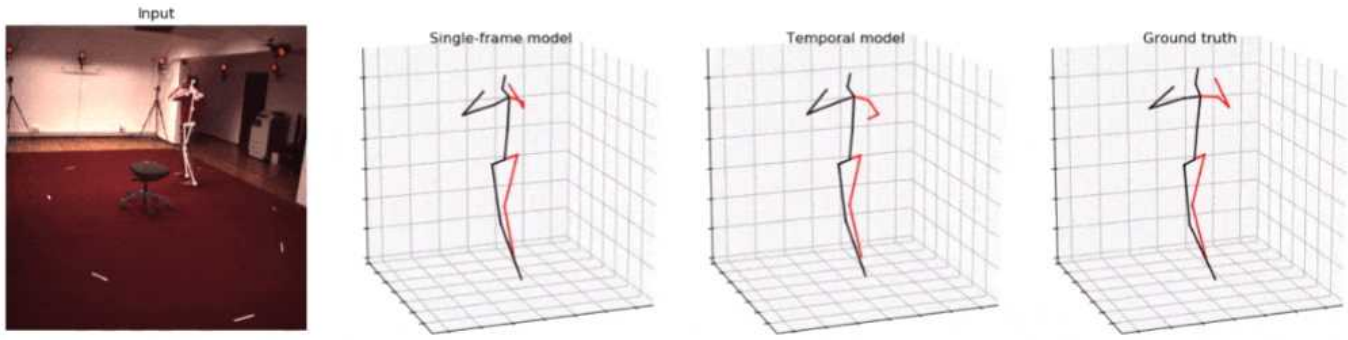
- 모델은 residual connections과 fully-connected network로 구성됨.
- 2D 이미지를 넣으면 regression loss를 이용해 3D포즈를 예측함, MPJPE 오차가 63mm으로 괜찮은 성과를 보이고 데이터 전처리를 통해 실시간 추정도 가능함.



< 그림 1 > 출처: [1]

1.2 3D human pose estimation in video with temporal convolutions and semi-supervised training[2]

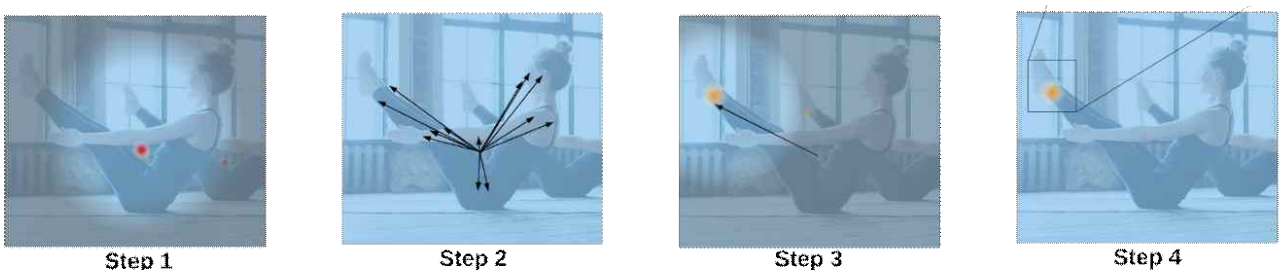
- 비디오의 2차원 키포인트에 대해 dilated temporal convolution기반의 fully convolutional model을 적용
- 레이블이 지정되지 않은 비디오 데이터를 활용하는 semi-supervised 학습 방법 사용
- 실제 코드 구현에서는 python detectron 패키지를 사용해 2D 추정된 결과를 입력받아서 추정에 사용함



< 그림 2 > 출처: [2]

1.3 MoveNet: Ultra fast and accurate pose detection model[3]

- tensorflow에서 2021년 공개된 포즈 추정 모델
- 이미지에서 사람의 키포인트를 찾고, 각 관절의 위치를 추적하는 방법으로 추정함



< 그림 3 > 출처: [3]

2) 학습 계획

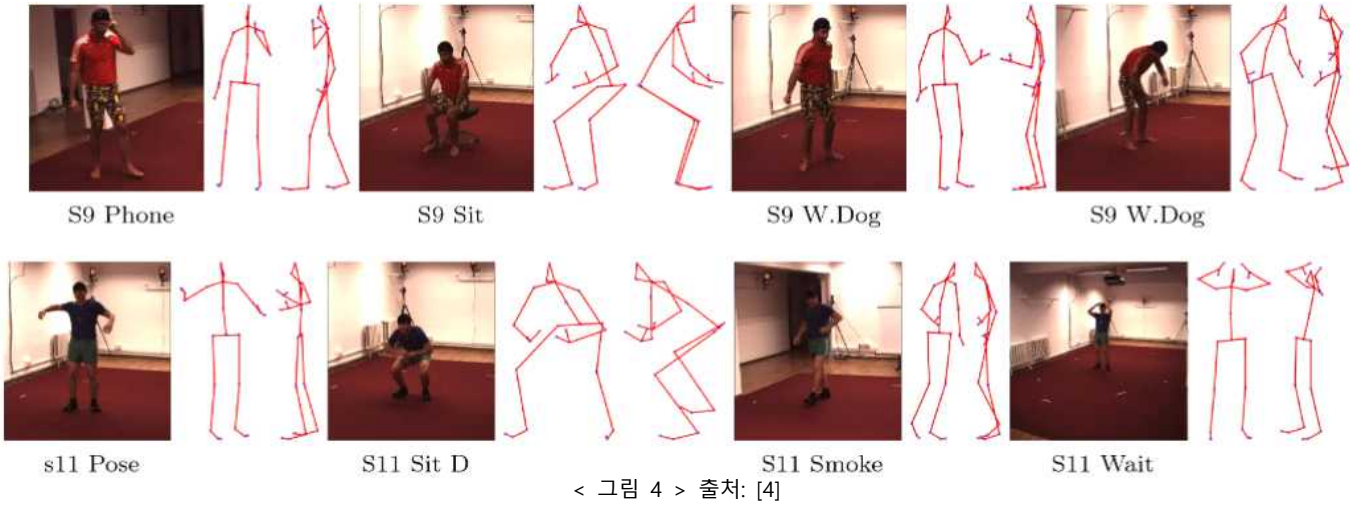
- input sequence로 2.1.1, 2.1.2, 2.1.3의 방법을 통해 변환된 좌표를 입력시켜 성능을 비교
- simple yet effective baseline의 input image로 카메라를 통해 촬영된 이미지를 사용, 개발 과정에서는 노트북 웹캠 등 접근성 높은 장비를 사용함
- 예측 결과를 시각화해서 분석하기 쉽게 함

3. 연구 내용 및 방법

1) 학습 데이터셋

1.1 Human 3.6m

- 11명의 배우가 촬영한 360만장의 포즈 및 대응하는 이미지로 구성되어있음. 17종의 동작으로 분류되어있음[4]
- 이미지를 입력으로 사용해서 3D포즈를 추정하고, 대응하는 Ground Truth와 오차율을 비교해보는 방법으로 모델을 학습시킬 계획임.



< 그림 4 > 출처: [4]

1.2 Penn Action Dataset

- 불특정 인물들을 촬영한 2326개의 비디오로 구성됨. 15종의 동작으로 분류되어있음[5]

< 그림 5 > 출처: [5]

3) 학습 방법

- 선행 연구에서 제시했던 방법 중 MoveNet(2.1.3)을 이용해 동작 분류기를 학습시킴
- 데이터셋은 각각 MoveNet을 통해 변환되고, 변환된 데이터는 모델 학습과 평가에 사용됨

4) 평가 지표

- 다양한 조건의 test 데이터셋들에 대해, 학습된 모델을 적용시킴
- 모델의 classification의 결과를 confussion matrix를 사용하여 오차율 등을 분석함

5) 코드 상세

- 작성한 코드는 아래표와 같이 구성됨

파일(폴더)명	실행환경	목적	비고
examples/ 및 하위 파일	python 3.9	이미지 전처리를 위한 MoveNet 모델의 동작	
01processs.ipynb	jupyter notebook (python 3.9)	학습 이미지 전처리	
01processs_iter.ipynb			메모리 한계로 분할 실행
01processs_iter2.ipynb			메모리 한계로 분할 실행
02train.ipynb		모델 학습 및 평가	구성 C(4.2.1)
02train_evaliter.ipynb		모델 학습 및 평가	구성 D(4.2.1)
cam2img.py	python 3.9	실사용 조건의 이미지 촬영 및 생성	노트북 웹캠으로 촬영
VideoPose3DTest.ipynb	GoogleColab (python 3.7)	3D 좌표 추정	리눅스 환경에서 동작

4. 결과 분석

1) 모델 평가 결과

- overfitting 문제를 방지하기 위해 적절한 epoch를 선택함
- 학습시킨 모델을 각각의 test 데이터셋에 대해 평가함

2) 오차율 분석

2.1 데이터셋 구성별 오차율 분석

- 다음 표와 같은 네가지 구성으로 모델을 학습시키고 평가함
- 모든 구성은 batch size 32, epoch 50의 동일한 조건 하에 학습

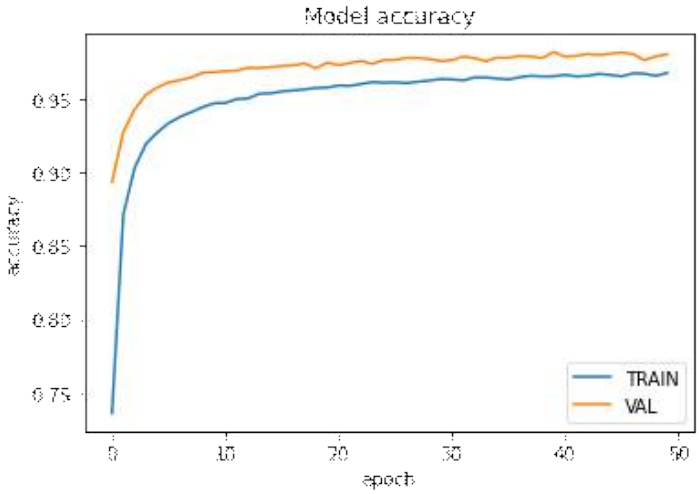
구성	Train 데이터 (수)	Test 데이터 (수)	동작 Label 갯수	Train error	Val. error	Test error	비고
A	Human 3.6m 中 Subject01/Subact02 (45223)	Human 3.6m 中 Subject01/Subact01 (29996)	15	0.20	0.09	0.43	
B		Human 3.6m 中 Subject05/Subact02 (29996)	15			0.68	

C	Penn Action 中 5개 동작의 80% (43991)	Penn Action 中 5개 동작의 20% (10239)	5	0.03	0.02	0.07	
D		직접 촬영한 이미지 (243)	5			0.08	구성 C와 동일, 실사용 조건 평가

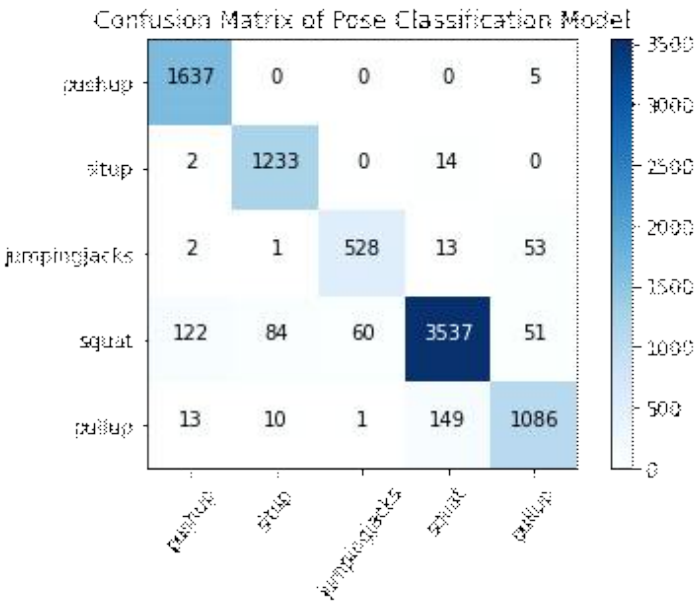
- A와 B는 현격히 높은 오차율을 보여 사용하지 않음
- C는 비교적 적은 오차율을 보여, 실사용 조건으로 구성한 D의 평가에 사용함

2.2 Label별 오차율 분석

- 가장 좋은 성능을 보인 구성 C에 대해서 평가함
- 50 epoch로 학습, overfitting을 방지하기 위해 임의로 epoch 선택함

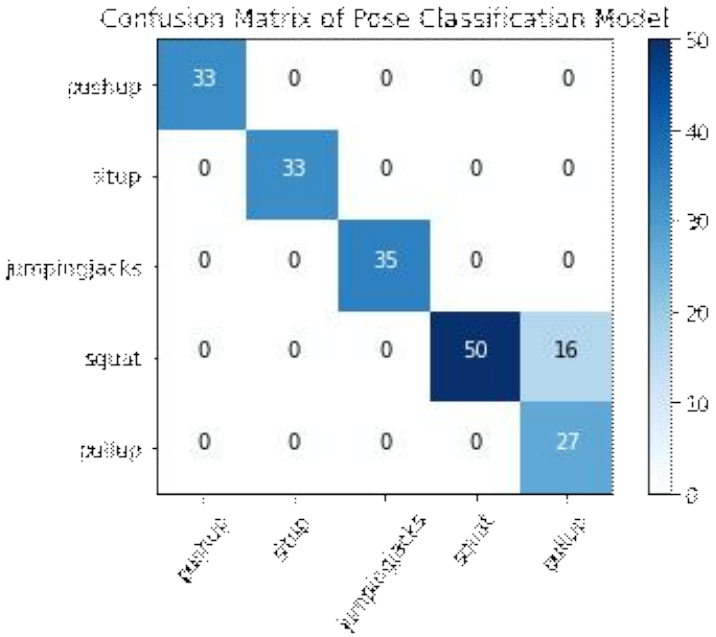


- 10239장의 test 이미지들에 대해 classification을 수행함
- 표에서 행은 추측한 이미지의 동작, 열은 실제 이미지의 동작을 나타냄

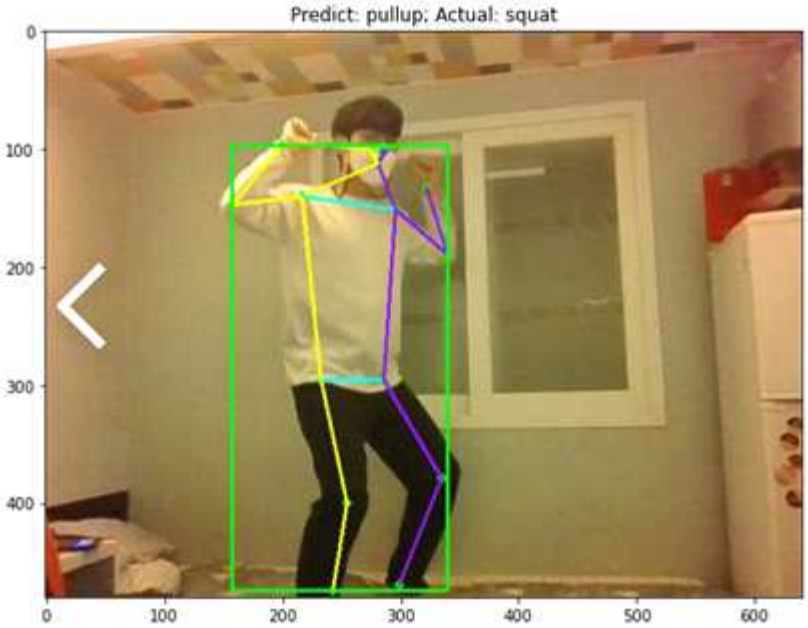


2.3 모델 적용 결과 분석(D)

- 구성 C로 학습된 모델을 실사용 평가를 위해 구성 D로 평가
- 243장의 test 이미지들에 대해 classification을 수행함
- 표에서 열은 추측한 이미지의 동작, 행은 실제 이미지의 동작을 나타냄

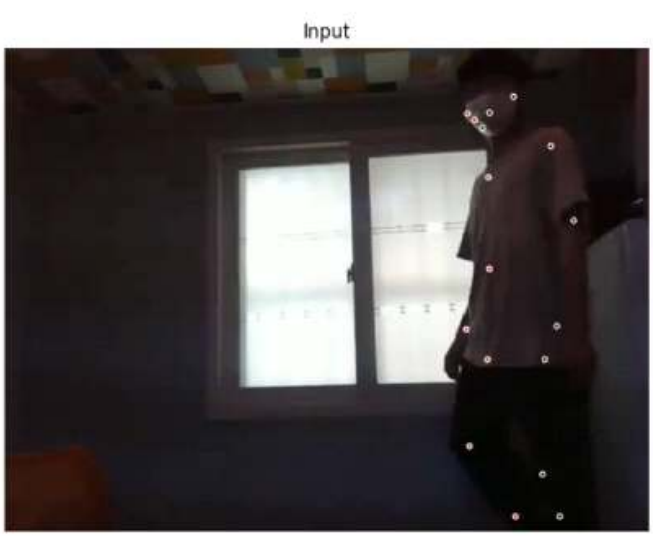


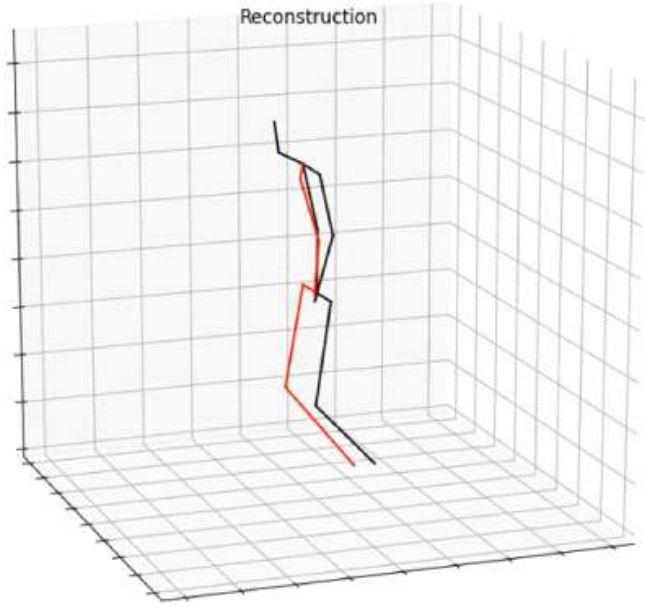
- squat를 제외하고 모든 classification을 정확하게 예측한 결과를 보여줌
- 성능 개선을 위해 다음 그림과 같이 잘못 분류한 이미지의 예제를 분석함
- 그림은 squat 이미지를 pullup으로 잘못 분류한 경우를 보여줌



2.4 3D 좌표 추정 결과

- [8], [9]을 통해 동영상 원본과 추정 결과를 각각 시연





3) 결과 의의와 한계

3.1 의의

- 학습된 분류 모델이 배경이 포함된 실생활 환경(in wild)에서도 동작하는 것을 검증함
- RGB 이미지만을 통해 3D 좌표로 변환시키는 과정을 수행해서 출력결과를 바탕으로 새로운 동작 데이터셋을 구성하거나, 3D CG작업 등에 활용 가능성 제시

3.2 한계

- 동작의 종류가 많아질수록 성능에 한계를 보임
- 3D 추정을 동작 분류에 적용하지는 못했음

4) 개선 방안

- penn action 데이터셋에 장애물에 사람이 가리거나, 다수의 인물이 나오는 등 인식 방해 요소가 많아 데이터셋을 더 선별한다면 높은 정확도를 기록할 수 있을 것으로 보임
- 현재 웹캠으로 비디오를 입력받고 MoveNet으로 데이터를 전처리하는데 지연이 있는데, 처리속도를 개선시킨다면 실시간으로 동작 분류도 가능할 것으로 보임

참고 문헌

[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)

참고 자료

[3]https://blog.tensorflow.org/2021/05/next-generation-pose-detection-with-movenet-and-tensorflowjs.html

[4] http://vision.imar.ro/human3.6m/description.php

[5] http://dreamdragon.github.io/PennAction/

[6] https://ko.wikipedia.org/wiki/%EB%AA%A8%EC%85%98_%EC%BA%A1%EC%B2%98

[7] https://link.springer.com/article/10.1007/s11263-018-1118-y/figures/6

외부 링크

[8] https://www.youtube.com/watch?v=qdvX3WLvBR4

[9] https://www.youtube.com/watch?v=kziOu3U3UGw