

학사학위 청구논문

Pose Estimation을 이용한
개인트레이너

(Using Pose Estimation for Personal Trainer)

2022년 11월 22일

숭실대학교 IT대학

AI융합학부

박 준 영

(속표지)

학사학위 청구논문

Pose Estimation을 이용한 개인트레이너

Using Pose Estimation for Personal Trainer

지도교수 : 노 동 건

이 논문을 학사학위 논문으로 제출함

2022년 11월 22일

숭실대학교 IT대학

AI융합학부

박 준 영

(인준서)

박준영의 학사학위 논문을 인준함

심사위원장 신동화 (인)

심 사 위 원 노동건 (인)

2022년 11월 22일

승실대학교 IT대학

감사의 글

논문이 완성되도록 도와주신 모든 분께 감사드립니다.

목 차

표 및 그림 목차	i
국문 초록	ii
I. 서론	1
II. Apple Vision과 Action Classifier 수행되는 구조	3
II-1. Apple Vision 과정	3
II-2. Action Classifier 과정	4
III. UI 및 데이터 처리	4
III-1. visionView 및 SummaryView	5
III-2. 체형에 따른 처리 & 자세에 따른 피드백	6
IV. 실행 결과	7
IV-1. VisionView	7
IV-2. SampleVisionView	8
IV-3. SummaryView	8
V. 결론	9
참고문헌	10

표 및 그림 목차

그림 1.1 작품 구조도.....	1
그림 2.1 Apple Vision이 인식하는 Pose Estimation의 관절.....	3
그림 2.2 Vision 및 Action Classifier의 process.....	3
그림 3.1 UI 구성	4
그림 3.2 다리 모양	6
그림 4.1 다리 체형에 따른 분리.....	7
그림 4.2 운동 중의 자세.....	7
그림 4.3 SampleVisionView 실행화면.....	8
그림 4.4 SumamryView.....	8

Pose Estimation을 이용한 개인 트레이너

AI융합학부 박 준 영

지도교수 노 동 건

Human Pose Estimation이란 사람의 관절이나 중요 신체 부위를 키포인트로 지정하여 사람의 자세를 예측하는 기술이다. Human Pose Estimation은 현재 여러 분야에서 사용되고 있으면 스포츠, VR/AR, 헬스케어 분야 등에서 활용되고 있다. 그중에 헬스케어 분야에서 Human Pose Estimation은 자세를 교정하는 헬스케어 앱이 연구되고 있다.

본 논문에서는 Human Pose Estimation에서 헬스케어 분야 중에서 자세 교정하는 앱에 대해 만들었다. Apple 사의 Vision framework에서 Human Pose Detection API를 이용하여 Pose를 추정한다. Vision framework의 경우 bottom up 방식을 사용하여 영상에 포함된 사람의 키포인트를 모두 추정하고, 키포인트간의 상관관계를 분석하여 포즈를 추정하는 방식을 사용하여 Real-time에 적용될 수 있어 사용되고 있다. 따라서 Video Session을 backgroundQue 에 실행하여 Real-time으로 영상 속 사람의 pose를 추정 나오는 각 joint의 location값이 MLmultiArray로 반환되는데 반환된 MLmultiArray는 coreML model인 action Classifier를 통해 예측값을 볼 수 있다. 스킵트를 수행하기 전 Human Pose Detction API를 통해서 나온 각 joint의 location 값을 일련의 함수를 통해서 그 사람에게 맞는 자세를 추천하고 추천해준 자세를 수행 중에는 미리 학습된 Action Classifier model을 이용하여 Predict 하여 잘못된 자세를 알려준다.

I. 서 론

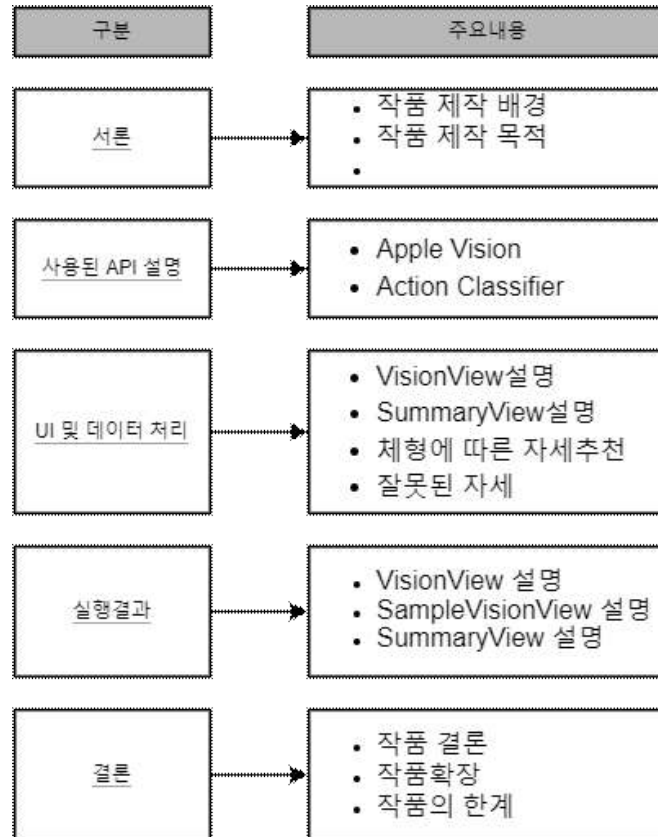


그림1.1 연구의 흐름도

본 연구의 전체적인 구조도는 그림 1.1과 같다. 먼저 서론에서는 본 작품을 만들게 된 배경과 목적에 관하여 서술한다. 그다음 사용된 API 설명에서는 본 작품에서 사용된 Vision과 Action Classifier에 대해 어떻게 작동되는지 자세히 서술한다. UI 및 데이터 처리에서는 각 View에 대한 설명 및 작동 및 작동하는데 필요한 class들 설명이 있으며 체형에 따른 자세를 어떻게 추천하는지와 잘못된 자세를 어떻게 확인하는지를 서술한다. 실행 결과에서는 각 View에서의 실행 중 화면 및 실행되면 각 components가 어떻게 변하는지를 설명한다. 결론에서는 작품의 결론, 작품을 어떻게 확장할 수 있는지, 현재 작품의 한계에 대해 간단히 서술한다.

최근 들어 Computer Vision을 이용한 다양한 서비스들이 제공되기 시작하였다. 얼굴 및 물체 인식, 동작 추적, 문자인식 등 다양한 기능들이 다양한 어플들을 통해 제공되고 있다. 그중에서도 모션 트래킹인 Human Pose

Estimation을 이용하여 사람의 관절이나 중요 신체 부위를 키포인트로 지정하여 사람의 자세를 예측한다. Human Pose Estimation은 현재 여러 분야에서 사용되고 있으면 스포츠, VR/AR, 헬스케어 분야 등에서 활용되고 있다. 그중에 헬스케어 분야에서 Human Pose Estimation은 자세를 교정하는 헬스케어 앱이 연구되고 있다. Human Pose Estimation이 자세 교정하는 헬스케어 서비스가 존재한다고 하지만 대부분 커다란 기계를 이용하여 휴대할 수 없으면 고액으로 누구나 이용이 매우 불편하다. 따라서 현재까지 나와 있는 옛지 디바이스중 우리가 쓰는 스마트폰을 이용하여 Vision을 실시간으로 수행하기에 성능이 충분하며 또한 다양한 API가 존재하여 사용하게 매우 유리하다. 또한 스마트폰 앱 중에 자세 교정 및 헬스 관련 앱이 많이 존재하지만, Pose Estimation을 이용하는 앱은 매우 단순하며 카운트해주는 정도이다. 따라서 카운트만 해주는 것이 아니라 개인 트레이너처럼 운동에 대해 피드백해주는 앱이 필요하고 생각하게 되었다. 이중의 IOS에서 실행이 가능한 앱을 제작하여 많은 운동 중 스쿼트의 자세와 운동 방법을 추천해주는 앱을 만들게 되었다. 이를 만드는 데에는 현재 사용 중인 APPLE 사의 iPhone을 이용할 건데 APPLE사에서 제공하는 API인 Vision Framework, coreML, createML을 이용하여 헬스케어 앱을 만든다.

서론에 이어 2장에서는 Apple 사의 Vision이 수행되는 구조와 coreML의 Action Classifier의 model에 관하여 기술하고 3장에서는 UI 및 데이터 처리에 대해 설명하며, 4장에서는 본 작품을 실행하여 볼 수 있는 결과에 대해 결과와 고찰을, 5장에서는 결론을 맺도록 한다.

II. Apple Vision과 Action Classifier 수행되는 구조

II-1. Apple Vision 과정

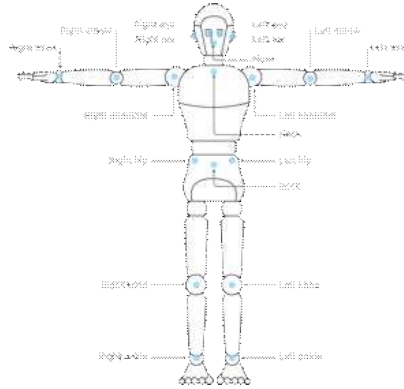


그림2.1 Apple Vision이 인식하는 Pose Estimation의 관절

Apple의 Vision Framework는 사진2의 process처럼 진행된다. 여기서 실시간일 경우에는 위의 사진처럼 device camera(AVSesseeion)를 통해서 frame을 받아오며 저장되어있는 동영상의 경우에는 AVAssetReader를 통해서 동영상을 프레임별로 불러온다. 불러온 frame은 Vision framework의 VNImageRequestHandler은 VNDetectHumanBodyPoseRequest를frame에서 처리하여 VNHumanBodyPoseObservation을 return 하는데 VNHumanBodyPoseObservaiton는 request를 처리한 결과이며 각 joint는 VNRecognizedPoint이며 Frame에서의 key는 JointName이고 value는 location이다. 이때 vision에서 detect 가능한 joints는 그림1에 있는 key point다. 또한 MLMultiArray를 내놓으면 이는 coreML에서 Action Classifier의 input이기도 하다.

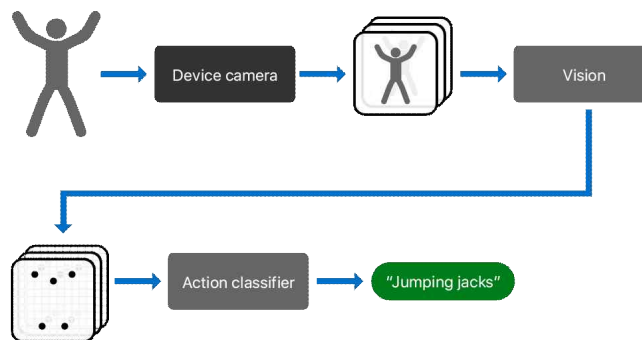


그림2.2 Vision 및 Action Classifier의 process

II-2. Action Classifier 과정

Apple의 coreML의 action Classifier는 createML을 통해서 만들 수 있다. Apple사에서 제공하는 Action Classifier는 데이터셋에 따라 layer가 자동으로 설정된다. 데이터셋은 training data와 validation data 그리고 testing data로 나뉘는데 Training data에 폴더별로 라벨링 하여 넣으면 validation data는 자동으로 나뉘어 수행한다. Training이 완료되고 output을 model로 사용한다. Model의 경우 학습할 때는 비디오 파일이지만 사용할 때는 VNHumanBodyPoseObservaiton의 결과에서 MLMultiArray를 추출하여 사용한다. MLMultiArray는 coreML에서 사용되는 다차원 배열로 숫자 값이 저장되어 있다. 크기는 프레임 X 3차원 X 라벨에 따른 수로 나타내어진다. 모델에 맞는 MLMultiArray는 ProcessingChain을 만들어주어 vision이 처리한 frame 들을 합쳐주어 만들어준다. 만들어진 MLMultiArray를 통해 Action Classifier는 prediction을 보여준다. Prediction은 label과 confidence를 포함하고 있다. 이를 통해서 Label을 확인할 수 있다.

III. UI 및 데이터 처리

III-1. VisionView 및 SummaryView



그림3.1 UI 구성

사진 3처럼 UI를 구성하였다. Tabbar에는 vision과 summary가 존재하며 vision에서는 실시간 확인과 동영상을 불러와 확인 하는 view가 존재한다. 실시간 view에는 videoCapture class에서 AVCaptureSession을 통해 카메라

에서 비디오를 입력받는다. 입력받은 frame은 AVCaptureVideoDataOutputSampleBufferDelegate를 통해 framePublisher를 통해 publish 된다. Publish 된 frame의 데이터 타입은 cm sample Buffer(모든 media type)이다. Publish 된 frame 들은 VideoProcessingChain에서 Subscribe한다. VideoProcessingChain에서는 frame을 image로 바꾸어주고 image에서 Pose라는 구조체를 바꾸어준다. Pose 구조체는 vision에서 찾은 joint 들을 landmark 리스트에 저장하고 connection이라고 하여 근접한 joint 들을 연결하고 또한 image 위에 그릴 수 있는 함수를 가진다. Pose에서 MLMultiArray들을 모아 합쳐 window를 만든다. 이때 window의 크기는 60(2초X 30프레임) X 3 X 21의 크기를 가진다. 이렇게 모은 window는 actionClassifier의 predictActionFromWindow를 통해 predict 값을 확인할 수 있다. 또한 image의 pose에서 checkSquat 구조체를 통해서 Pose의 값과 Predict 값을 가져와서 스크린 자세와 횡수를 센다.

동영상을 불러오는 view에서는 videoReader class를 이용하여 Photo Library에 있는 동영상을 불러온다. videoReaderProcessor를에서는 videoProcessingChain와 똑같은 역할을 한다.

자세가 틀렸을 때는 Recorder를 통해 녹화한다. 먼저 RenderSettings를 통해 fps, 동영상 사이즈, 동영상 이름과 디렉토리를 지정한다. Recorder는 RenderSettings를 불러와 AVAssetWriter와 AVAssetWriterInput을 설정한다. AVAssetWriter를 시작하면 PixelBufferAdapter는 동영상의 프레임에서 pixel buffer를 가져오고 이 buffer와 불러온 시간을 PixelBufferAdapter 추가한다. 이때 자세가 다시 올바르게 되면 finish를 하면 pixelBufferAdapter는 finishWriting을 통해서 동영상을 저장한다.

Summary View에서는 sqlite를 이용하여 day Table과 Summary Table을 생성하고 각각 id, day와 id, count, result, filename, day_id(foreign key day table id)이다. SQLite OMR을 이용하여 데이터의 입출력은 만들었다. SummaryView에서는 day Table row를 불러와 tableView에서 볼 수 있다. 이때 각 row의 cell을 누르면 summaryCellView에서는 summary table의 row를 불러온다. 이때 cell에는 count, result가 표시되고 누르면 filename을 가져와 AVPlayViewController를 불러와 동영상을 실행한다.

III-2. 체형에 따른 처리 & 자세에 따른 피드백



그림3.2 다리 모양

총 3가지의 무릎 체형이 존재한다. 이 체형들은 넓적다리뼈와 정강이의 각도에 의해서 정해진다. 이때 외반슬의 경우는 170도보다 작을 때이며 정상의 경우는 170도~180도 사이 그리고 내반슬은 180도 이상일 경우이다. 이는 checkSquat struct 에 함수로 구현되어있어 실시간으로 수행 중 운동 시작 4초 동안 확인하여 화면에 표시하여 준다. 이때 결과는 VisionView에서 typeLabel을 통해 알려준다.

이번 모델에서는 맞은 자세를 제외한 3가지(무릎 안쪽으로 말림, 허리 먼저 들림, 덜 앉음)의 틀린 자세를 학습하였다. 각 데이터셋은 100개로 각 2초짜리 동영상으로 약 60프레임이다. 실시간 및 동영상에서는 VideoProcessingCahin 및 VideoReaderProcessor에서 마지막에 checkSquat를 통해서 predict 값을 출력하며 VisonView의typeLabel에서 결과를 보여준다.

IV. 실행 결과

IV-1. VisionView

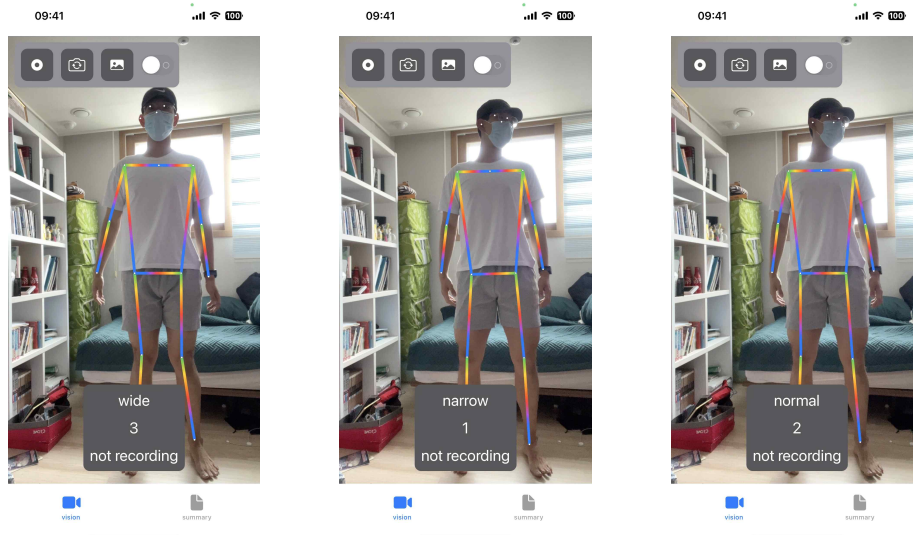


그림4.1 다리 체형에 따른 분리

위의 그림처럼 실행화면에서 위의 분석 시작(녹화) 버튼을 누르면 4초의 텀을 두고 다리의 체형을 분석한다. 위의 스위치 버튼은 시작 버튼을 누르면 녹화되는데 이때 관절표현이 같이 위에 저장되느냐 아니냐를 정한다. 아직 4초가 지나지 않아서 not recording 상태이다.

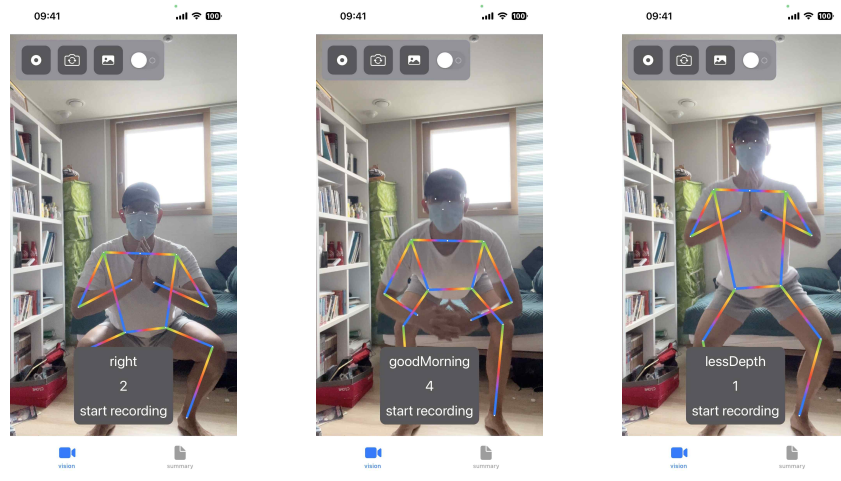


그림4.2 운동 중의 자세

운동을 수행하기 시작하면 위와 같이 현재 자세가 어떤지 나온다. 또한 start recording 상태에서는 실행하는 장면을 녹화하여 사진첩에 저장한다.

IV-2. SampleVisionView

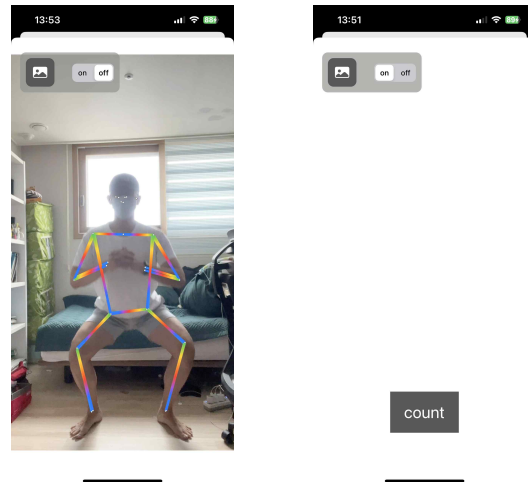


그림 4.3 SampleVisionView 실행화면

sampleVisionView에서는 위의 사진 버튼을 누르면 사진첩에서 동영상을 불러올 수 있다. 또한 동영상을 불러와서 실행할 때 운동 횟수를 세 줄뿐 아니라 자세 상태도 나타낸다. 위의 on-off toggle을 이용하여 라벨들을 볼 수도 있고 지울 수도 있다.

IV-3. SummaryView

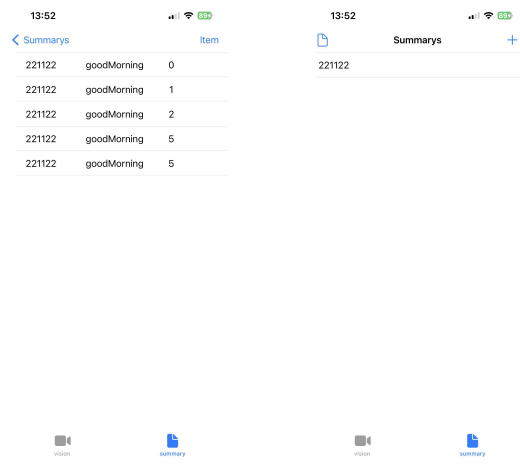


그림 4.4 SumamryView

SummaryView에서는 처음 tab bar에서 Summary를 누르면 나온다. 이때 첫 화면에서는 날짜가 나오며 날짜는 TableView의 형태로 보여주며 운동을 수행한 날짜를 보여준다. 이 날짜 Cell을 누르면 오른쪽과 같이 날짜, 자세, 횟수가 나온다. 횟수는 내려갔다 올라오는 것을 기준으로 하여 1회가 완료되지 않았으면 0회로 기록된다. 자세 Cell을 누르면 동영상을 통해 자신의 자세 왜 틀렸는지를 알 수 있다.

V. 결 론

본 작품에서는 Pose Estimation을 이용하여 헬스 케어 앱을 만들었다. 그중에 스쿼트를 특정하여 만들었으며 실시간과 촬영된 동영상을 이용하여 자신에게 맞는 자세를 추천받고 또한 운동을 수행하면서 자신의 자세를 피드백 받을 수 있게 하였다. 또한 피드백을 녹화하여 자신이 왜 잘못하였는지 확인할 수 있게 하였다. 현재의 작품으로서는 체형에 맞는 자세를 추천하여 다음에 발생할 수 있는 부상을 예방하고 또한 자세를 수행 중에 잘못된 자세를 지적해줌으로써 부상 및 타겟 근육에 대해 운동의 이해도를 증가시켜준다. 또한 현재는 apple 사의 IOS 운영체제하에서만 작동하지만 추후 open source의 Pose Estimation model을 이용하여 다양한 기기에서도 사용하게 하도록 고안할 것이며 스쿼트뿐만 아니라 다른 운동에서도 응용될 수 있게 적용할 것이다. 또한 현재는 2D Pose Estimation을 모델로 하여 촬영 각도에 따른 분석을 따로 하여야 하고 또한 다리의 각도가 제대로 나오지 않는 경우가 많아 오류가 생긴다. 하지만 3D Pose Estimation 모델을 사용하여 자세 분석률을 높이고 다양한 각도에서도 촬영하여 사용할 수 있게 한다.

참고문헌

- [1] Apple Developer Documentation, Vision
- [2] Apple Developer Documentation, CreateML
- [3] Apple Developer Documentation, coreML
- [4]<https://gist.github.com/yusuke024/b5cd3909d9d7f9e919291491f6b381f0>video Recoder
- [5] <https://github.com/stephencelis/SQLite.swift/> summaryDataBase sqlite ORM
- [6] Lorenzetti et al, “How to squat? Effects of various stance widths, foot placement angles and level of experience on knee, hip and trunk motion and loading”, BMC sports Science Medicine and Rehabilitation, vol.10, 10–14, July 17 2018