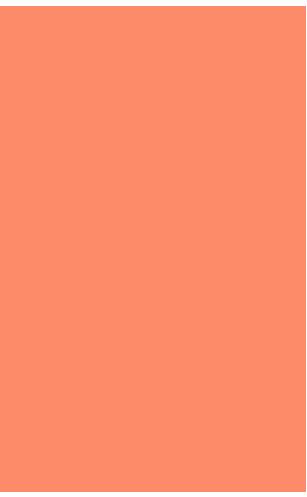


YOLOv5 및 MediaPipe를 이용한 실시간 3대 운동 AI 자세 교정 서비스에 대한 연구

A Study on Real Time the big three exercises AI posture correction service
Using YOLOv5 and MediaPipe

컴퓨터공학과 3학년 고영민

2023.10.04



_INDEX

I . 논문 진척사항

II . 개발 진척사항



| . 논문 진척사항



I. 논문 진척사항

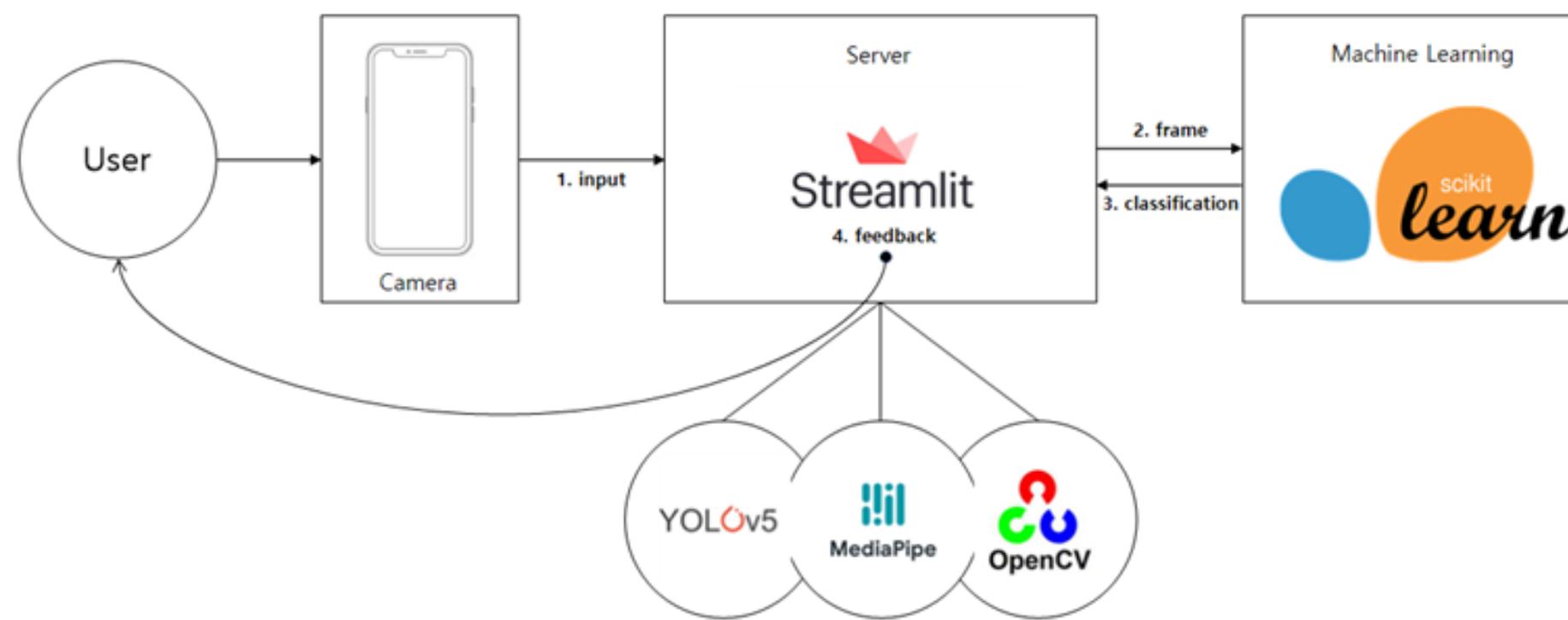


그림 4. 시스템 아키텍처

Fig. 4. System Architecture

운동을 수행 중인 사람 객체 탐지 후 관절 좌표 추정

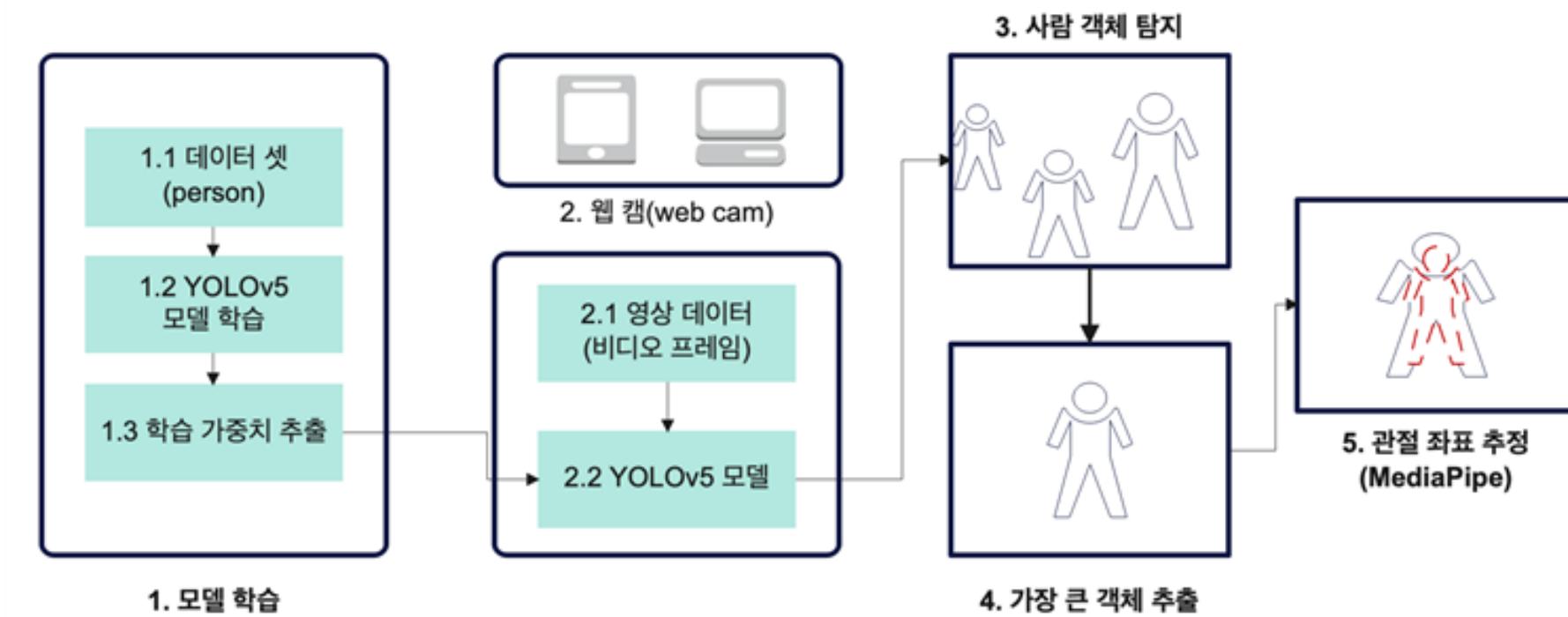


그림 5. 운동을 수행 중인 사람 객체 탐지 후 관절 좌표 추정

Fig. 5. Estimating joint coordinates after detecting a person object exercising

I. 논문 진척사항

YOLOv5 및 MediaPipe를 이용한
실시간 3대 운동 AI 자세 교정 서비스에 대한 연구

고영민

A Study on Real-Time the big three exercises AI posture correction service
Using YOLOv5 and MediaPipe

Yeong-Min Ko

요약

Abstract

I. 서론

2022년 문화체육관광부의 국민 생활체육 조사의 결과[1]를 보면 조사대상 중 61.2%는 주 1회 이상 꾸준히 생활체육을 즐겨하고 있으며, 그중에서 보디빌딩은 12.8%로 걷기 다음인 2위로 상당히 높은 비율을 차지하고 있다(그림 1). 보디빌딩의 핵심 운동은 3대 운동인 '벤치 프레스', '스쿼트', '데드리프트'가 있다. 하지만 3대 운동은 주로 고증량을 다루는 운동이므로 잘못된 자세로 운동을 수행하는 경우, 큰 부상을 초래할 위험이 크다. 그러나 대다수는 헬스장에 설치되어 있는 거울에만 의존하여 스스로 자세를 파악하며 운동을 수행한다. 하지만 전문적인 트레이너가 자세를 봐주면서 실시간으로 피드백을 하는 경우가 아닌 이상 스스로 자세의 옳고 그름을 판단하기 어렵다.

최근 옮지 않은 자세로 웨이트 트레이닝을 수행하여 부상을 초래하는 문제를 해결하기 위해 MediaPipe나 OpenPose와 같은 자세 추정 프레임워크를 활용한 연구가 나타나고 있다. 이러한 문제를 해결하기 위한 방법에는 MediaPipe를 통해 골격 정보를 활용해 관절 점의 각도를 계산하여 기준 범위를 정해 옳고 그름을 판단하는 방법[2], MoveNet를 통해 원본군(사용자)의 골격 정보와 대조군(전문 트레이너 영상)의 골격 정보 얻고 딥러닝을 통해 원본군과 대조군을 비교하여 각 관절별로 옳고 그름을 판단하는 방법[3], 정적인 이미지 파일을 또는 미리 촬영된 비디오 파일을 업로드하여 자세를 판단하는 방법이 있다.[4][5] 또한 국내에 등록된 실시간 운동 자세 교정 서비스로는 'GymMate'와 같은 서비스가 있다.[6] 'GymMate'는 스마트폰을 고정하여 수행할 운동을 선택한 다음 자세의 옳고 그름을 판단한다. 하지만 이 서비스 역시 단순히 Good이나 Bad라는 문구만을 사용자에게 전달할 뿐이다.

기존 연구들은 단순히 전문가와 사용자의 골격 정보를 비교하거나 골격 정보를 활용하여 각 관절의 각도를 구한 값만을 활용하여 자세의 옳고 그름을 판단하여 정확도가 낮다는 한계가 있다. 또한 일부 연구에서는 정적 이미지나 촬영된 비디오 파일에 대해서만 피드백하는 방식으로 실제로 부상 위험이 가장 높은 운동을 수행하는 동안에는 실시간으로 자세에 대한 피드백 할 수 없다는 한계가 있으며 실시간으로 피드백을 제공하는 서비스 역시 단순히 옳고 그름만 판단하여 운동을 수행할 때 정확히 어떤 자세가 문제인지 알려주지 못한다는 한계점이 있다.

본 논문에서는 MediaPipe의 스켈레톤 인식 기술을 사용하여 관절점을 추출하고 이를 기반으로 관절의 사이각을 구하여 데이터를 수집한다. 수집된 데이터를 기반으로 머신러닝 모델을 구축한 후, 음성합성(TTS) 기술을 활용해 실시간으로 운동 자세에 대해 세부적인 피드

백을 제공함으로써 올바른 운동 자세를 취할 수 있도록 도와주는 서비스를 구현하였다.

또한 본 논문에서는 MediaPipe를 통해 추출한 관절 좌표 데이터 중에서 각 운동별로 사용되는 주요 관절만을 사용할 뿐만 아니라 주요 관절 간의 각도를 계산한 값을 후처리하여 지도학습을 진행함으로써 모델의 정확도를 높이는 것을 목표로 한다.

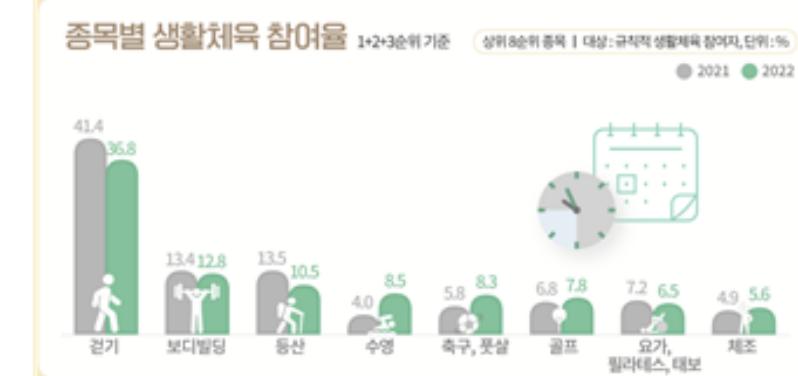


그림 1. 종목별 생활체육 참여율 통계

Fig. 1. Statistics of participation rate in daily sports by sport

II. 배경 지식

2.1 YOLOv5

YOLOv5(You Only Look Once version 5)는 컴퓨터 비전 및 객체 감지 분야에서 중요한 역할을 하는 딥러닝 기반의 객체 감지 알고리즘이다. YOLOv5는 YOLO 시리즈 중 비교적 최근에 제안된 5번째 버전으로, 빠른 속도와 높은 정확도를 제공하여 다양한 응용 분야에서 사용되고 있다. 이 알고리즘의 주요 특징 중 하나는 실시간 객체 감지를 가능하게 하는 높은 처리 속도로 인해 실시간 비디오 스트리밍에서 실시간 객체 검출을 수행하는 데 매우 유용하다. 그림 2는 YOLOv5의 모델별 성능을 나타낸 것이다.[7] 본 논문에서 관절점을 추출하기 위해 사용하는 MediaPipe의 경우 단일 인물에 대해서만 자세 추정을 할 수 있는데 운동을 수행하는 상황에서는 다른 인물이 카메라에 촬영될 수 있어 정확한 피드백을 제공하지 못할 수 있다. 이러한 점을 극복하기 위해 YOLOv5를 통해 운동을 수행 중인 객체를 탐지하여 해당 객체에 대해서만 자세 추정을 하도록 하는 운동을 수행 중인 사람 객체의 관절을 추정하는 알고리즘을 구현하였다. 그리고 실시간 영상처리를 위하여 연산 과정이 간단하고 가벼운 모델인 YOLOv5s 모델을 사용하였다.

I. 논문 진척사항

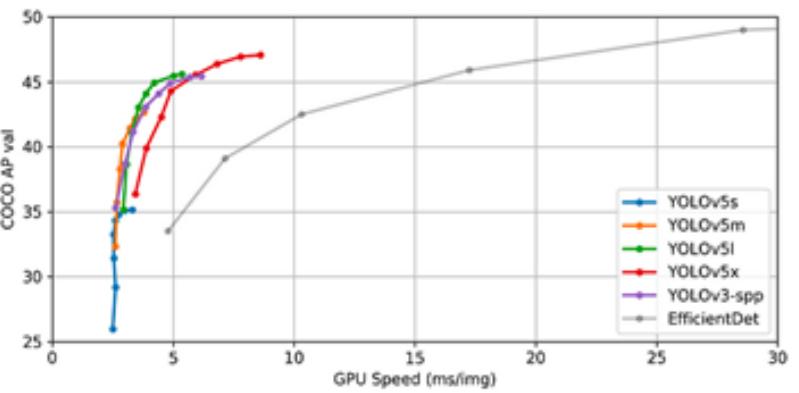


그림 2. YOLOv5 모델별 성능

Fig. 2. Performance of YOLOv5 models

2.2 미디어파이프(MediaPipe)

미디어파이프(MediaPipe)는 구글(Google)에서 개발한 오픈소스 프레임워크로, 비전 및 머신러닝을 기반으로 한 컴퓨터 비전 응용 프로그램을 개발하는 데 사용되며 특히 실시간 데이터 스트리밍에서 작동하도록 설계되어 있다. 미디어파이프는 객체 감지, 자세 추정, 얼굴 감지, 손 추적 등과 같은 다양한 비전 작업을 지원한다.[8] 그중에서 자세 추정(Pose Estimation)을 활용하면 그림 3과 같이 33개의 관절 좌표 데이터를 추출할 수 있다. 본 논문에서는 모든 관절의 좌표를 사용하지 않고 운동 종목별로 주요 관절의 좌표만을 사용한다. 모든 관절의 좌표를 사용하지 않은 이유는 얼굴 포인트(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)와 손가락을 나타내는 포인트(17, 18, 19, 20, 21, 22)는 운동 자세에 큰 영향을 끼치지 않으며 오히려 학습에 방해가 될 수 있기 때문이다.

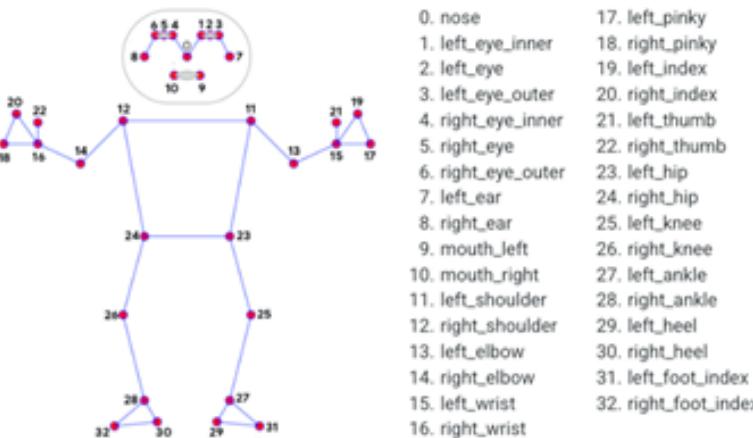


그림 3. MediaPipe Pose Landmark

Fig. 3. MediaPipe Pose Landmark

2.3 OpenCV

OpenCV(Open Source Computer Vision Library)는 컴퓨터 비전 및 이미지 처리 작업을 위한 오픈소스 라이브러리로 다양한 프로그래밍 언어에서 사용할 수 있다. OpenCV는 이미지 및 비디오 처리, 객체 감지, 얼굴 인식, 자동차 번호판 인식, 캘리브레이션 등 다양한 컴

퓨터 비전 작업을 지원하며, 컴퓨터 비전 연구 및 응용 프로그램 개발에 널리 사용된다. 본 논문에서는 파이썬 프로그래밍 언어를 통해 OpenCV 라이브러리를 사용한다.

2.4 Streamlit

Streamlit은 데이터 과학 및 웹 애플리케이션 개발을 위한 파이썬 기반의 오픈소스 라이브러리이다. Streamlit을 사용하면 데이터 시각화, 웹 애플리케이션, 대시보드 등을 빠르고 간편하게 개발할 수 있으며, 데이터 과학자 및 엔지니어들이 데이터를 시각화하고 공유하는 데 매우 유용한 도구이다. 본 논문에서는 Streamlit을 통해 실시간으로 이미지 처리를 수행함으로써, 운동 자세에 대한 피드백을 사용자에게 실시간으로 제공한다.

2.5 사이킷런(Scikit Learn)

III. 제안하는 실시간 3대 운동 AI 자세 교정 서비스

3.1 시스템 아키텍처

본 논문에서 제안하는 시스템 구성도는 그림 4와 같다. 제안하는 실시간 3대 운동 AI 자세 교정 서비스는 Streamlit을 이용하여 서버를 구축한다. 먼저, 사용자의 모바일 카메라로 촬영된 모든 프레임을 OpenCV를 통해 실시간 영상으로 처리하여 서버에 전송한다(그림 4, 1. input). 그리고 서버에서는 전달받은 영상의 모든 프레임에 대해서 YOLOv5를 이용해 사람 객체를 탐지하는 모델을 통해 바운딩 박스가 가장 큰 객체를 실제 운동 중인 객체로 판단하고, 가장 큰 바운딩 박스 내부에 있는 객체에 대해서 MediaPipe 프레임워크를 활용해 운동을 수행하는 사람의 관절 좌표 정보를 추정하여 각 운동별로 머신러닝 모델을 통해 자세 분류 결과를 서버에게 반환한다(그림 4, 2. frame & 3. classification). 마지막으로 서버에서는 운동 횟수를 카운트하고 잘못된 자세에 대해서는 피드백을 제공한다(그림 4, 4. feedback).

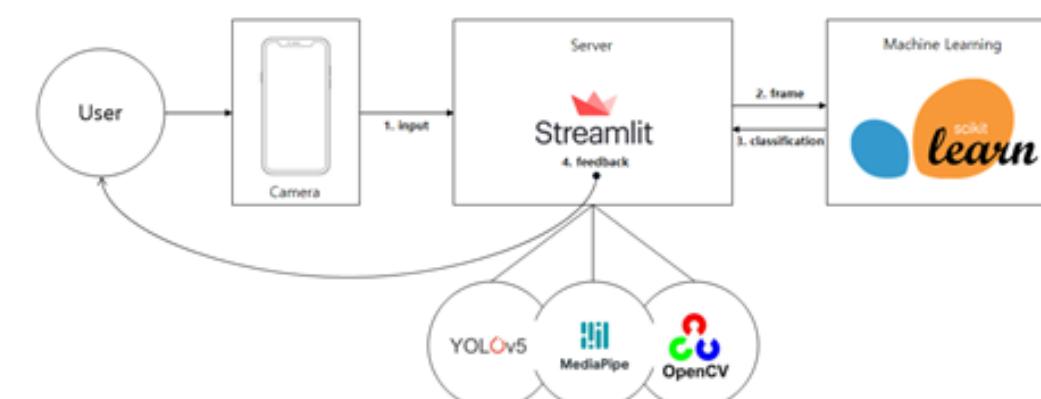


그림 4. 시스템 아키텍처

Fig. 4. System Architecture

3.2 운동을 수행 중인 사람 객체의 관절을 추정하는 알고리즘

MediaPipe는 모바일에서도 실시간 추론을 할 수 있는 매우 빠른 모델이지만 여러 사람이 아닌 오직 한 사람만 추적하여 관절 좌표를 추정할 수 있다. 하지만 헬스장과 같이 여러 사람이 함께 활동될 수 있는 환경에서는 자세 추정의 정확도가 낮아질 수 있으며 정확하지 않

I. 논문 진척사항

은 피드백을 제공할 위험이 있다. 따라서 본 논문에서는 이러한 한계를 극복하기 위하여 그림 5에 제시한 운동을 수행 중인 사람 객체의 관절을 추정하는 알고리즘을 제안한다.

본 논문에서 제안하는 운동을 수행 중인 사람 객체의 관절을 추정하는 알고리즘은 사람과 관련된 데이터 세트를 YOLOv5 모델로 학습시킨 후 그 가중치를 사용하여 사용자의 웹캠으로부터 입력된 영상 데이터 중 사람 객체를 탐지한다. 그리고 가장 큰 박스(bounding box)를 가진 객체를 실제 운동 중인 사람 객체라고 판단하고 해당 박스 내부에 있는 객체에 대해서 MediaPipe 프레임워크를 활용해 관절 좌표를 추정한다.

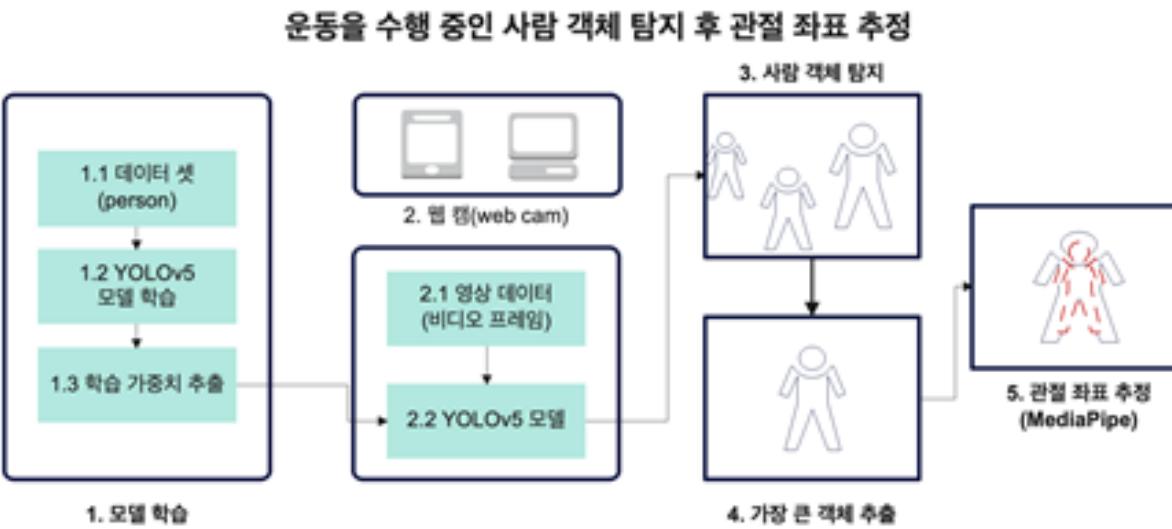


그림 5. 운동을 수행 중인 사람 객체 탐지 후 관절 좌표

Fig. 5. Estimating joint coordinates after detecting a person object exercising

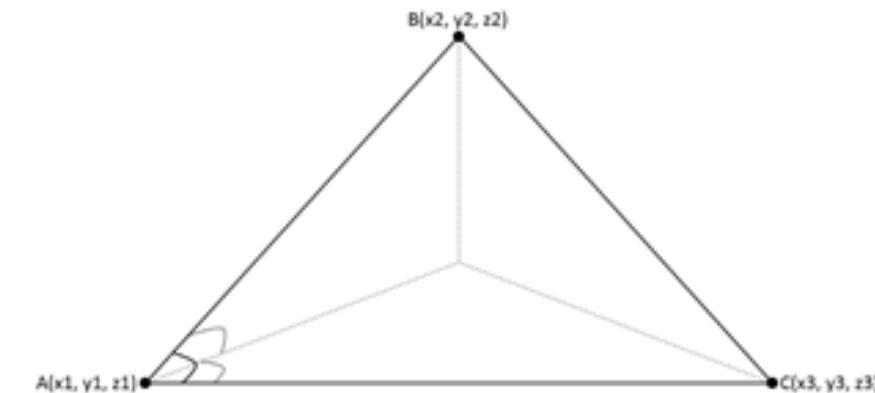
3.3 운동 횟수 세기 알고리즘

3.4 운동별 자세 추론 및 교정 알고리즘

3.4.1 관절 사잇각 계산 알고리즘

3.2절에서 추출된 관절 좌표는 각 관절점(landmark)에 대한 위치 정보($l = (x, y, z)$)만을 담고 있다. 관절점에 대한 위치 정보 자체는 무의미한 값이지만 이러한 정보를 통해 관절의 사잇각을 구하면 운동 자세 분석에서 유의미한 정보로 사용할 수 있다.

삼각함수를 활용하면 관절점의 위치 정보를 통해 관절 간 사잇각(joint angle)을 구할 수 있는데 그림 6은 관절 사잇각의 위치와 수식이다. 연산 방법은 x_1, y_1 과 x_1, z_1 또는 y_1, z_1 를 기준으로 세 점이 이루는 사잇각과 x_1, y_1 를 기준 원점 축에 대한 각도 2개를 구한 뒤 그 각을 빼는 방법이다.



$$\theta = (\arctan \frac{y2 - y1}{x2 - x1} - \arctan \frac{y3 - y1}{x3 - x1}) * \frac{180}{\pi}$$

그림 6. 관절 사잇각 계산 알고리즘

Fig 6. Algorithm for calculating joint angles

위의 알고리즘을 통해 왼쪽 팔꿈치의 각도를 구한다고 가정하면 그림 7과 같다. 변수 a, b, c에 각각 왼쪽 어깨, 왼쪽 팔꿈치, 왼쪽 손목 관절점을 대입하여 `np.arctan2()` 함수를 활용해 이들 간의 각도를 구하면 라디안으로 나타나는데 $180/\text{np.pi}$ 를 곱해 라디안을 다시 도 단위로 변환한다. 만약 각도가 180도가 넘으면 360에서 각도를 뺀 값을 반환한다.

```
# 計算角度
import math
def calculateAngle(a, b, c):
    a = np.array(a) # first
    b = np.array(b) # mid
    c = np.array(c) # end

    radians = np.arctan2(c[1]-b[1], c[0]-b[0]) - np.arctan2(a[1]-b[1], a[0]-b[0])
    angle = np.abs(radians*180.0 / np.pi)

    if angle > 180.0:
        angle = 360 - angle

    return angle

shoulder = [landmarks[sp_pose.PoseLandmark.LEFT_SHOULDER.value].x, landmarks[sp_pose.PoseLandmark.LEFT_SHOULDER.value].y]
elbow = [landmarks[sp_pose.PoseLandmark.LEFT_ELBOW.value].x, landmarks[sp_pose.PoseLandmark.LEFT_ELBOW.value].y]
wrist = [landmarks[sp_pose.PoseLandmark.LEFT_WRIST.value].x, landmarks[sp_pose.PoseLandmark.LEFT_WRIST.value].y]

calculateAngle(shoulder, elbow, wrist)
```

그림 7. 관절점으로부터 관절 사이각을 계산하는 파이썬 코드

Fig 7. Python Code for calculating joint angles from landmarks

아래 표는 각 운동별로 사용되는 관절점과 주요 관절의 사이각을 나타낸다.

I. 논문 진척사항

Angle	사용한 관절점(landmark) 번호
목((좌+우)/2)	좌: 11, 0, 23(left shoulder, nose, left hip) 우: 12, 0, 24(right shoulder, nose, right hip)
팔꿈치(좌 우)	좌: 11, 13, 15(left shoulder, left elbow, left wrist) 우: 12, 14, 16(right shoulder, right elbow, right wrist)
어깨(좌 우)	좌: 13, 11, 23(left elbow, left shoulder, left hip) 우: 14, 12, 24(right elbow, right shoulder, right hip)
골반(좌 우)	좌: 11, 23, 25(left shoulder, left hip, left knee) 우: 12, 24, 26(right shoulder, right hip, right knee)
무릎(좌 우)	좌: 23, 25, 27(left hip, left knee, left ankle) 우: 24, 26, 28(right hip, right knee, right ankle)
발목(좌 우)	좌: 25, 27, 29(left knee, left ankle, left heel) 우: 26, 28, 30(right knee, right ankle, right heel)

표 1. 주요 관절의 사잇각

Table 1. The angle of main joints

3.4.2 벤치프레스

상태	자세 분류	단축성 수축(up)	신장성 수축(down)
Correct	올바른 자세	b_correct_up	b_correct_down
Incorrect	허리가 과도한 아치 자세	b_excessive_arch_up	b_excessive_arch_down
	바를 너무 넓게 잡은 자세	b_arms_spread_up	b_arms_spread_down
	발이 지면에서 떨어진 자세	b_not_on_ground_up	b_not_on_ground_down

표 2. 벤치프레스 자세 분류

Table 2. Bench press posture classification

3.4.3 스쿼트

상태	자세 분류	단축성 수축(up)	신장성 수축(down)
Correct	올바른 자세	s_correct_up	s_correct_down
Incorrect	척추 중립이 아닌 자세	s_spine_neutral_up	s_spine_neutral_down
	무릎이 올풀 들어간 자세	s_caved_in_knees_up	s_caved_in_knees_down
	발을 너무 넓게 벌린 자세	s_knees_ahead_up	s_knees_ahead_down

표 3. 스쿼트 자세 분류

Table 3. Squat posture classification

3.4.4 데드리프트

상태	자세 분류	신장성 수축(up)	단축성 수축(down)
Correct	올바른 자세	d_correct_up	d_correct_down
	척추 중립이 아닌 자세	d_spine_neutral_up	d_spine_neutral_down
	바를 너무 넓게 잡은 자세	d_arms_spread_up	d_arms_spread_down
	바를 너무 좁게 잡은 자세	d_arms_narrow_up	d_arms_narrow_down

표 4. 데드리프트 자세 분류

Table 4. Deadlift posture classification

I. 논문 진척사항

IV. 실험

4.1 데이터 세트

- 4.1.1 운동을 수행 중인 사람 객체 탐지 모델 데이터 세트
- 4.1.2 운동 자세 분류 데이터 세트

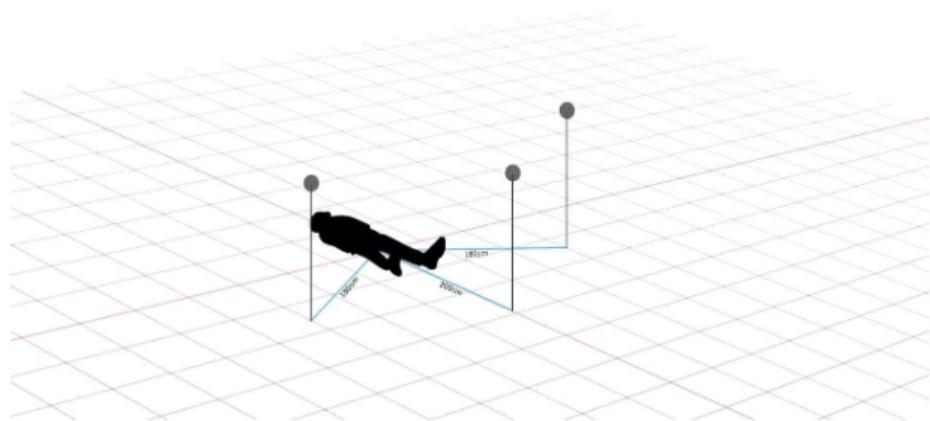


그림 8. 벤치프레스 - 촬영 거치대 쿼터뷰

Fig 8. Shooting stand position quater view for Bench Press

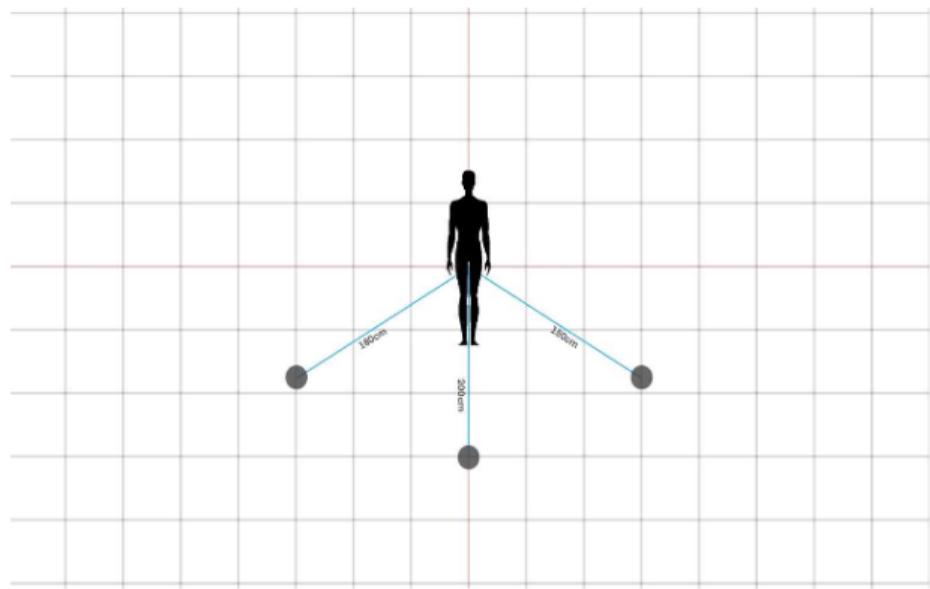


그림 9. 벤치프레스 - 촬영 거치대 위치 탑뷰

Fig 9. Shooting stand position top view for Bench Press

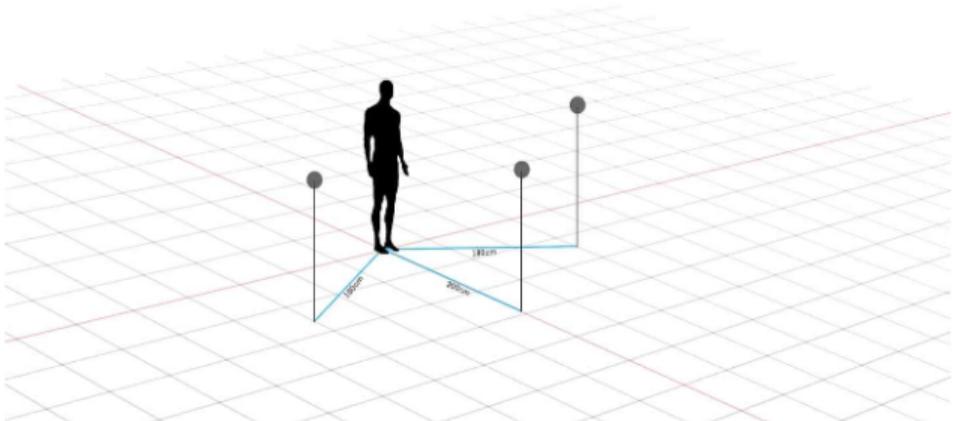


그림 10. 스쿼트 및 데드리프트 - 촬영 거치대 쿼터뷰

Fig 10. Shooting stand position quater view for Squat and Deadlift

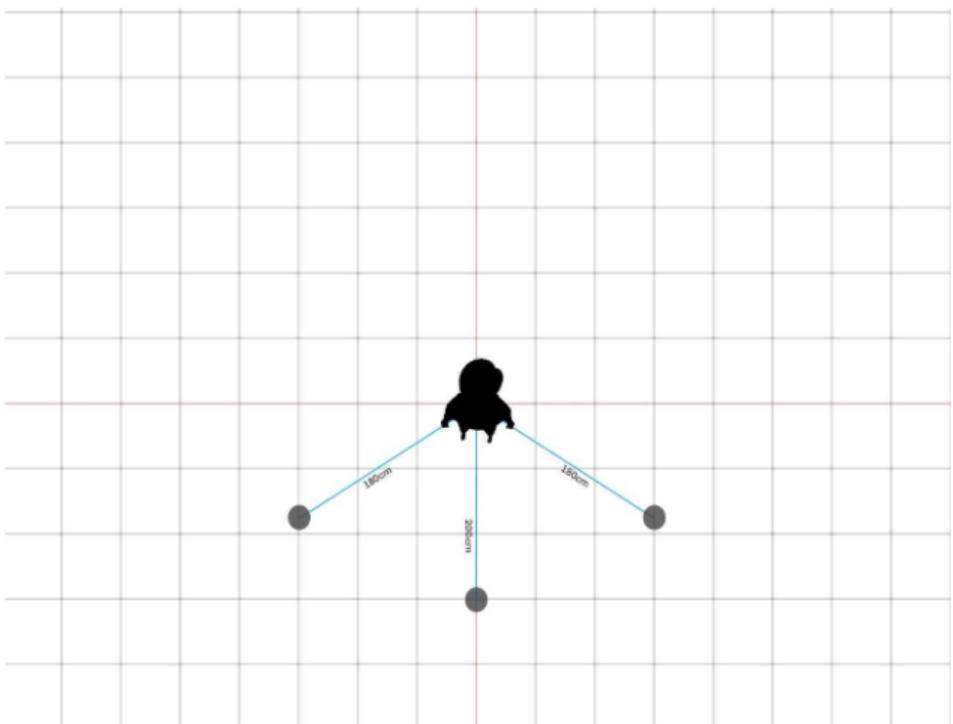


그림 11. 스쿼트 및 데드리프트 - 촬영 거치대 탑뷰

Fig 11. Shooting stand position top view for Squat Deadlift

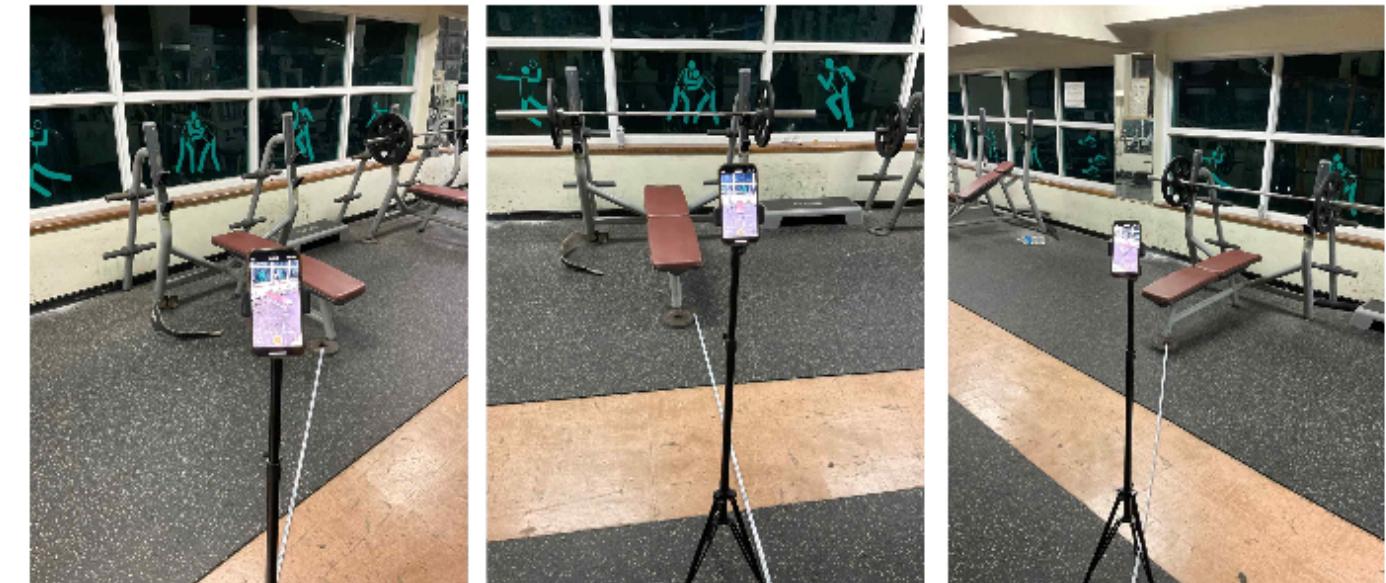


그림 12. 벤치프레스 촬영 예시

Fig 12. Shooting position for Bench Press

4.2 실험 환경

본 실험을 위한 서버는 Windows 11 Pro OS와 프로세서 13th Gen Intel(R) Core(TM) i5-13600KF 3.50GHz, 그래픽 카드 GeForce RTX 4080, 32GB RAM으로 구성된 PC를 사용하고, 클라이언트로 macOS 14와 m1 chip이 탑재된 맥북 m1과 아이폰 12 Pro를 사용하여 실험하였다. 그리고 서버에 사용된 파이썬은 Python 3.10 버전이 사용되었다.

4.3 성능 평가

- 4.3.1 평가 지표 설정
- 4.3.2 실험 결과

V. 결론

References

- [1] 2022 국민생활체육조사, [Internet], https://www.mst.go.kr/kor/s_policy/dept/deptView.jsp?pSeq=1691
- [2] 이용준, 김태영, "딥러닝 기반 포즈 인식 및 교정을 통한 효율적인 홈 트레이닝 시스템 개발", 2021.
- [3] 신윤지, 이현주, 김준희, 권다영, 이선애, 출윤진, 박지혜, 정자현, 이형석, 김준호, "딥러닝 기반 영상처리 기법 및 표준 운동 프로그램을 활용한 비대면 온라인 홈트레이닝 어플리케이션 연구", 2021.
- [4] S. Shirsath, P. Thakkar, B. Wolfson, "Using Human Pose Detection to Identify and Give Feedback on Workout Form."



II. 개발 진척사항



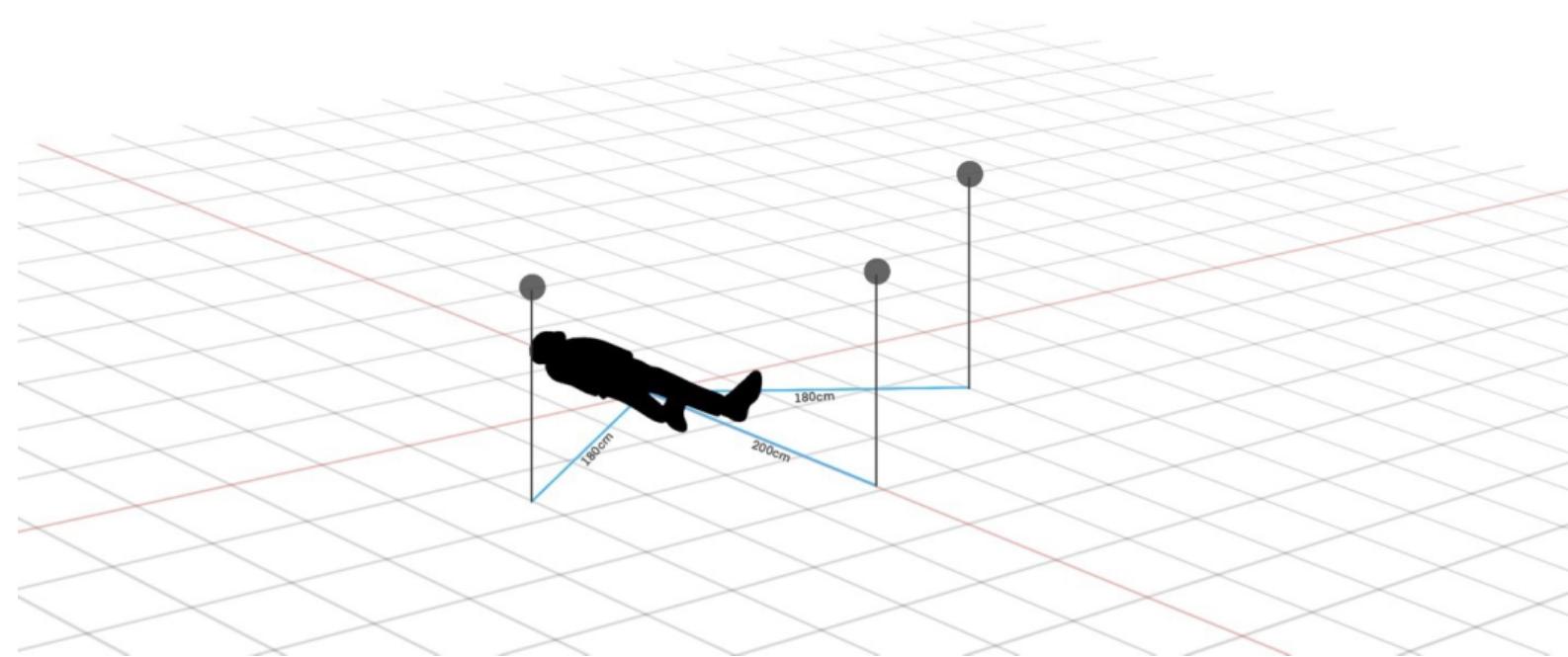
II. 개발 진척사항



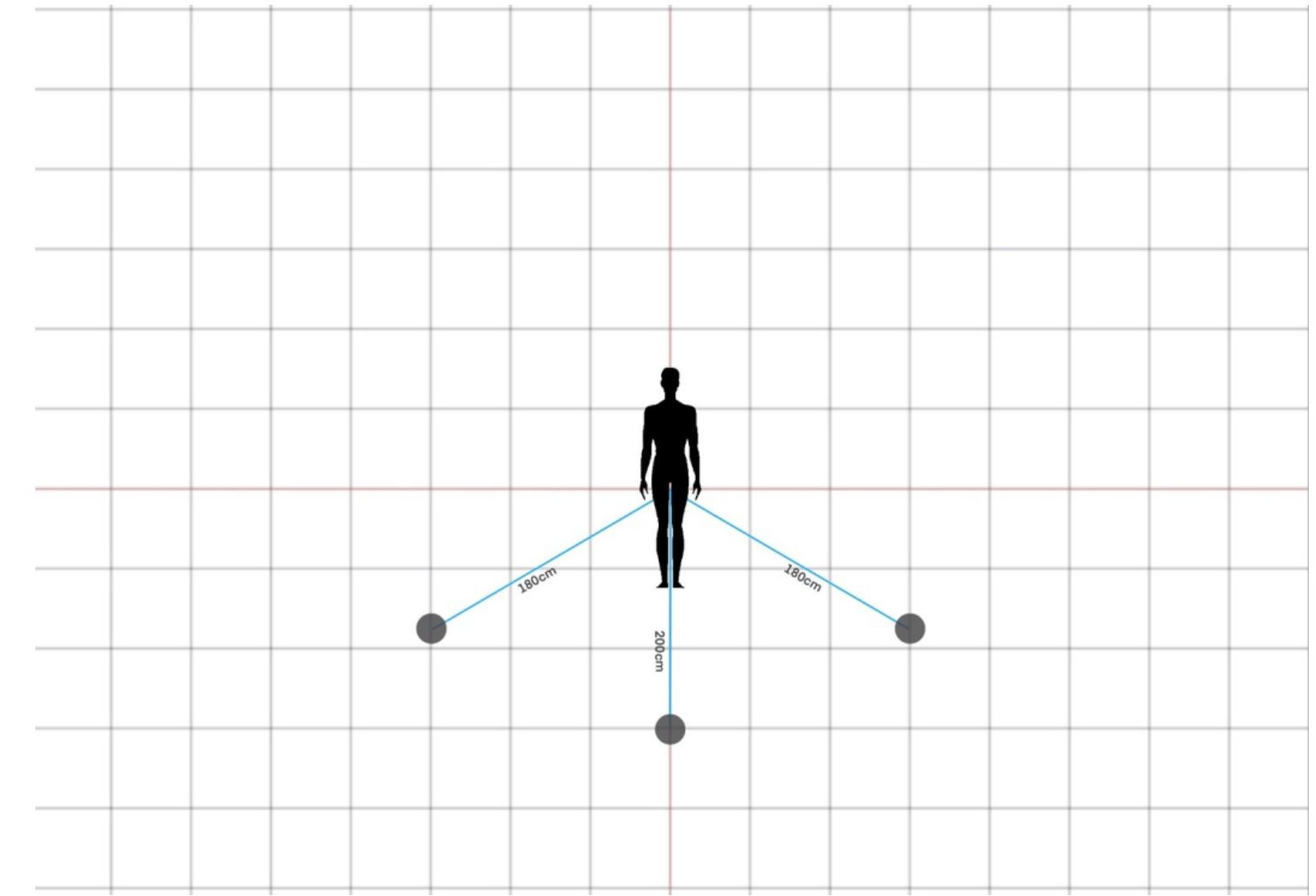
+



II. 개발 진척사항

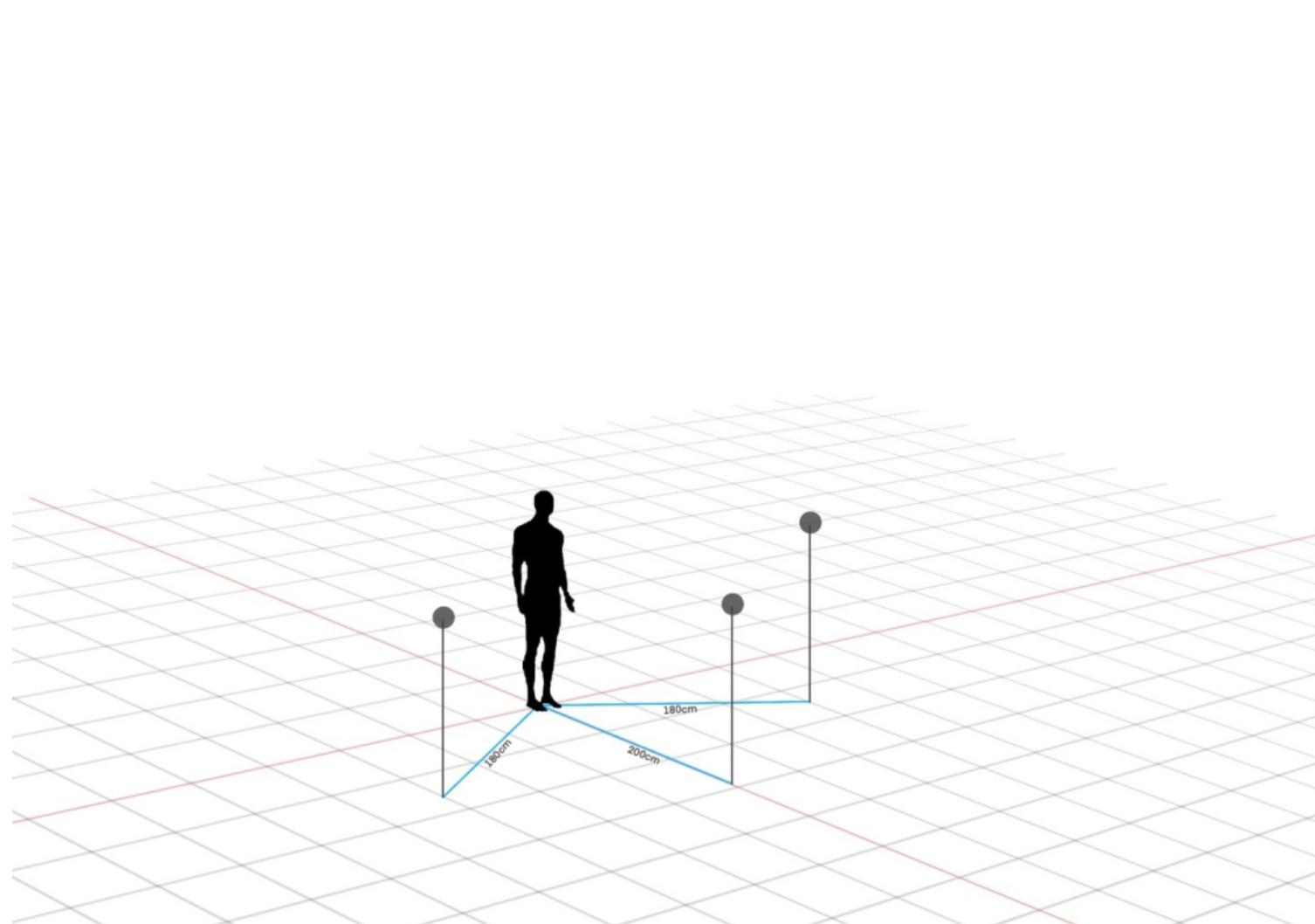


- 벤치프레스 탑뷰

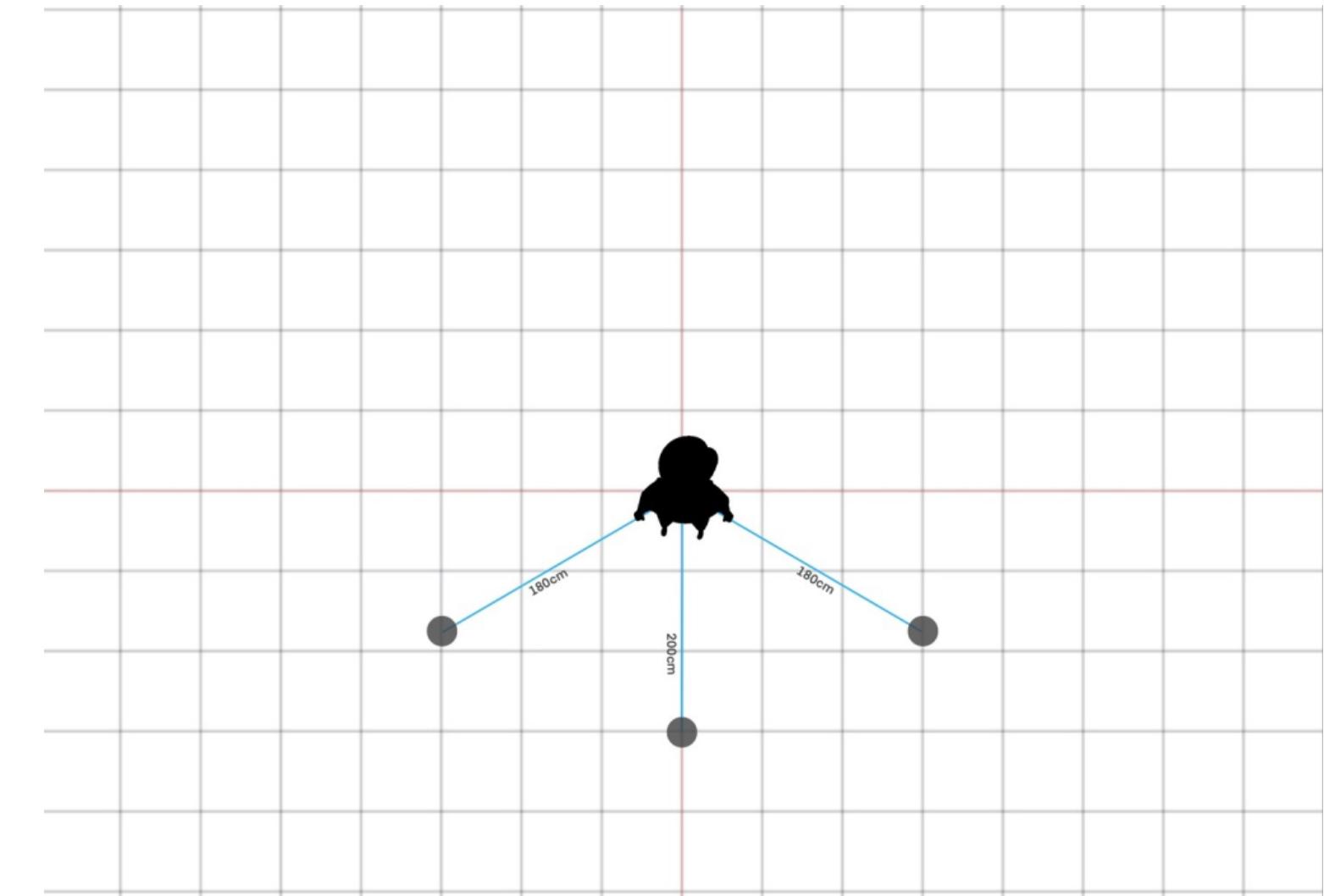


- 벤치프레스 쿼터뷰

II. 개발 진척사항

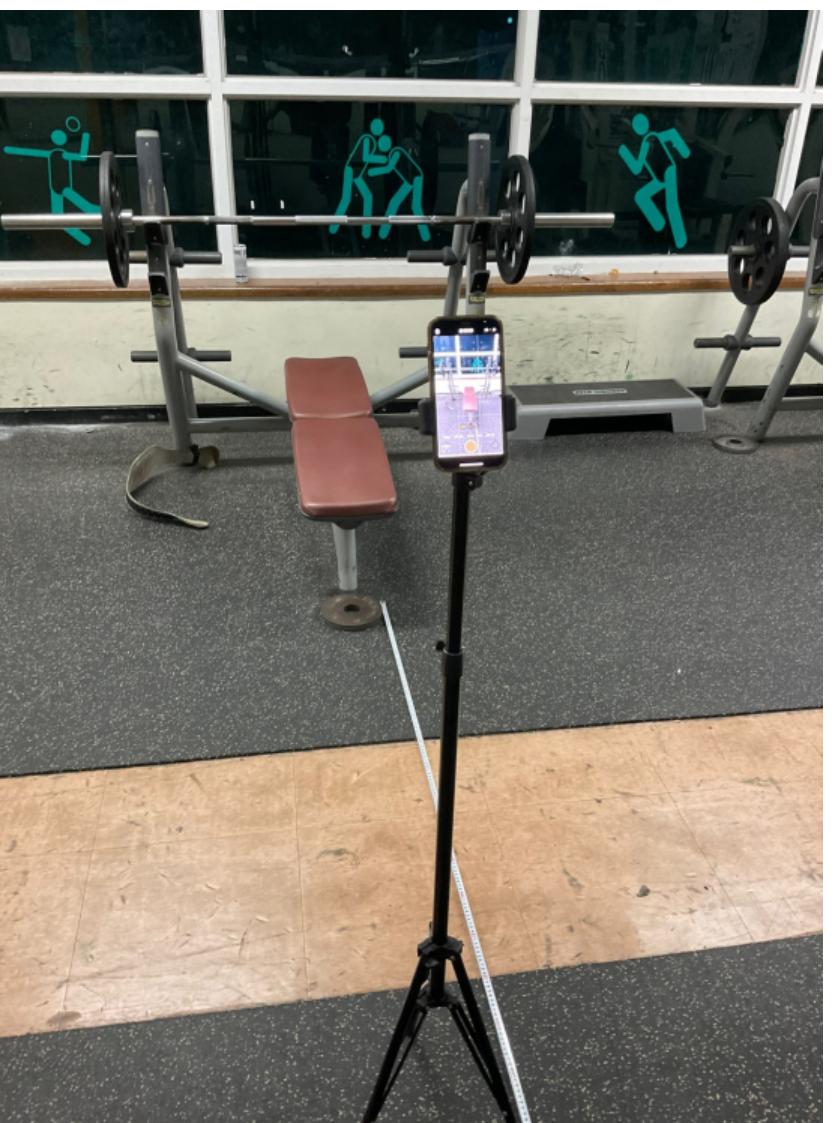
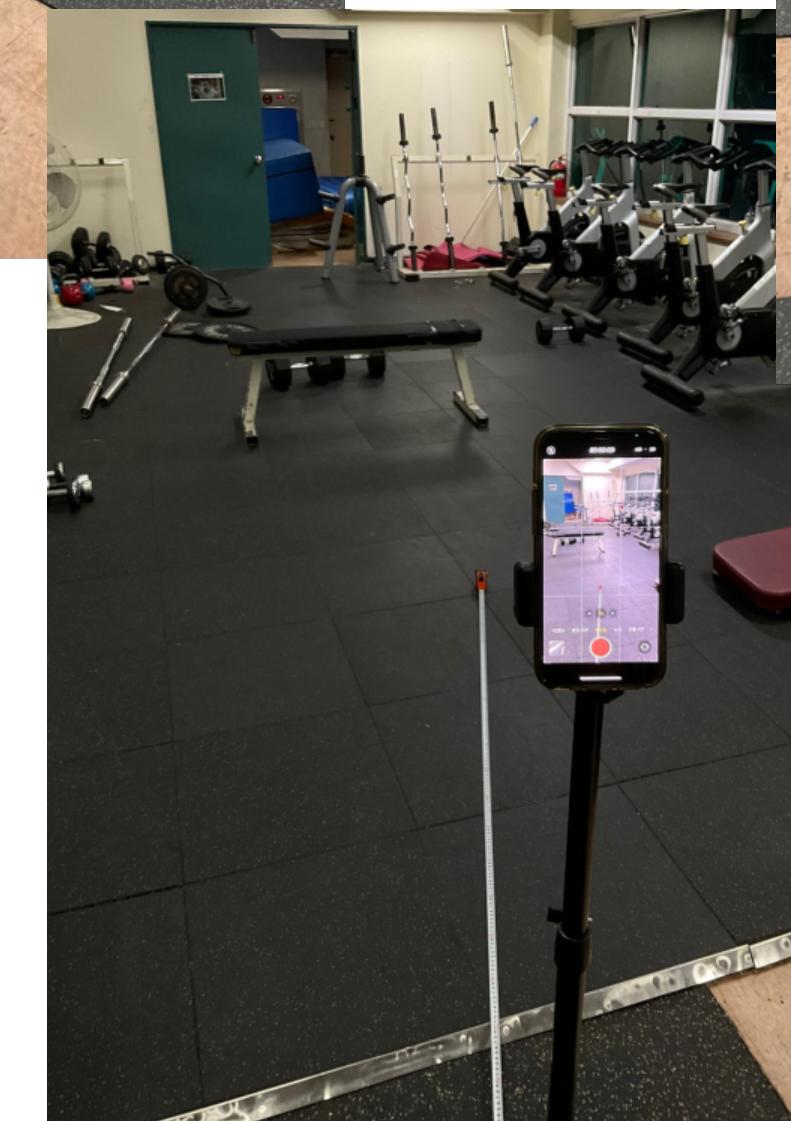
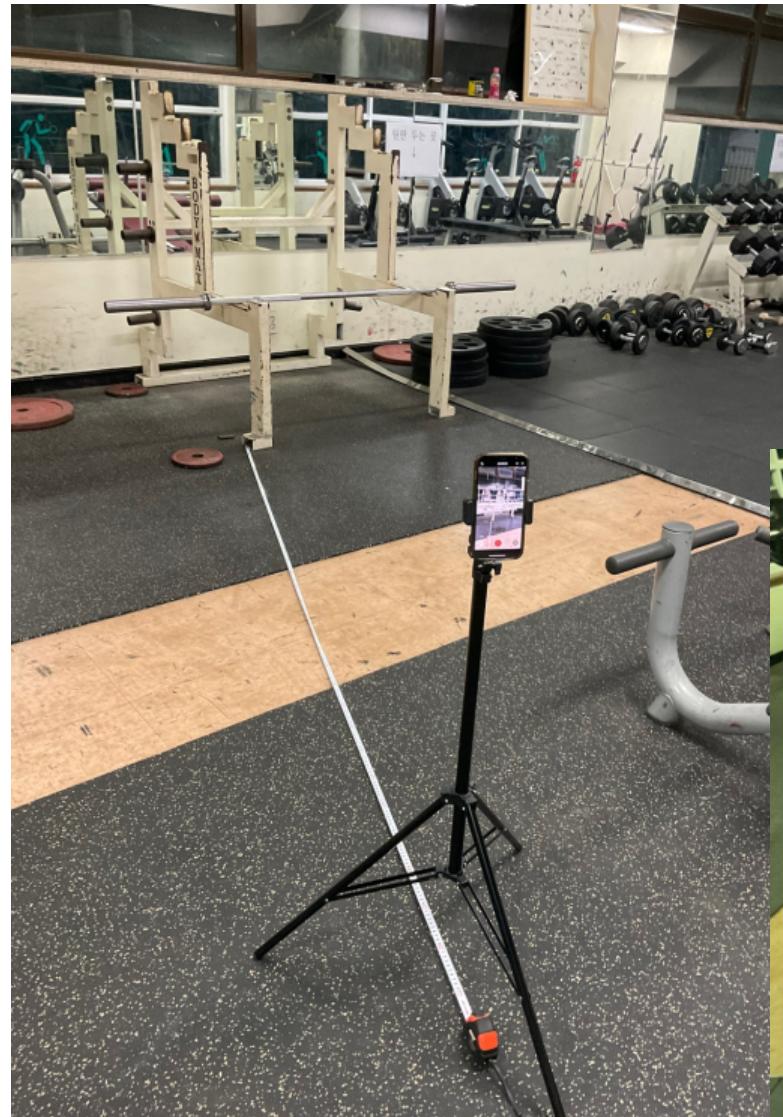


- 스쿼트 및 데드리프트 탑뷰

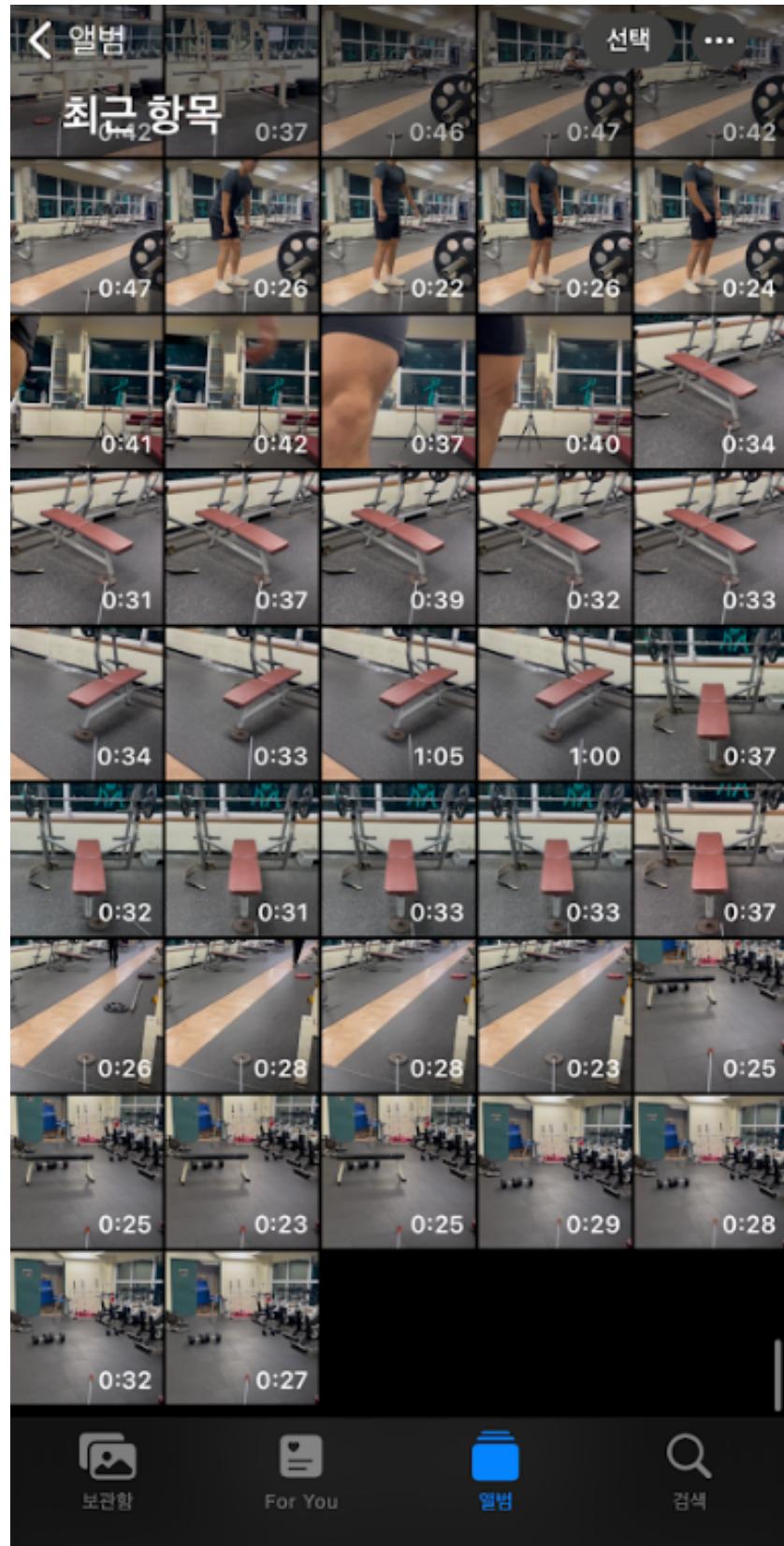


- 스쿼트 및 데드리프트 쿼터뷰

II. 개발 진척사항



II. 개발 진척사항



3.4.2 벤치프레스

상태	자세 분류	단축성 수축(up)	신장성 수축(down)
Correct	올바른 자세	b_correct_up	b_correct_down
	허리가 과도한 아치 자세	b_excessive_arch_up	b_excessive_arch_down
	바를 너무 넓게 잡은 자세	b_arms_spread_up	b_arms_spread_down
	발이 지면에서 떨어진 자세	b_not_on_ground_up	b_not_on_ground_down

3.4.3 스쿼트

상태	자세 분류	단축성 수축(up)	신장성 수축(down)
Correct	올바른 자세	s_correct_up	s_correct_down
	척추 중립이 아닌 자세	s_spine_neutral_up	s_spine_neutral_down
	무릎이 움푹 들어간 자세	s_caved_in_knees_up	s_caved_in_knees_down
	발을 너무 넓게 벌린 자세	s_knees_ahead_up	s_knees_ahead_down

3.4.4 데드리프트

상태	자세 분류	신장성 수축(up)	단축성 수축(down)
Correct	올바른 자세	d_correct_up	d_correct_down
	척추 중립이 아닌 자세	d_spine_neutral_up	d_spine_neutral_down
	바를 너무 넓게 잡은 자세	d_arms_spread_up	d_arms_spread_down
	바를 너무 좁게 잡은 자세	d_arms_narrow_up	d_arms_narrow_down

벤치프레스, 스쿼트, 데드리프트 각 자세별로
수축 ~ 이완 동작까지 운동을 10회씩 수행하여 총 44개 영상 수집
(맨몸, 빈바, 10kg 원판 사용)

II. 개발 진척사항

5_2. Train Machine Learning Classification Model

```
In [51]: from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler

from sklearn.linear_model import LogisticRegression, RidgeClassifier
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier

In [52]: pipelines = {
    'lr':make_pipeline(StandardScaler(), LogisticRegression()),
    'rc':make_pipeline(StandardScaler(), RidgeClassifier()),
    'rf':make_pipeline(StandardScaler(), RandomForestClassifier()),
    'gb':make_pipeline(StandardScaler(), GradientBoostingClassifier()),
}

In [54]: fit_models = {}
for algorithm, pipeline in pipelines.items():
    model = pipeline.fit(X_train, y_train)
    fit_models[algorithm] = model

In [55]: fit_models

Out[55]: {'lr': Pipeline(steps=[('standardscaler', StandardScaler()),
                             ('logisticregression', LogisticRegression())]),
          'rc': Pipeline(steps=[('standardscaler', StandardScaler()),
                             ('ridgeclassifier', RidgeClassifier())]),
          'rf': Pipeline(steps=[('standardscaler', StandardScaler()),
                             ('randomforestclassifier', RandomForestClassifier())]),
          'gb': Pipeline(steps=[('standardscaler', StandardScaler()),
                             ('gradientboostingclassifier', GradientBoostingClassifier())])}

In [56]: fit_models['rc'].predict(X_test)

Out[56]: array(['correct_bench', 'correct_bench', 'correct_bench', 'correct_bench'],
              dtype='<U15')
```

파이프라인을 통한 머신러닝 학습

- 로지스틱 회귀
- 릿지 분류
- 랜덤 포레스트
- 그레이디언트 부스팅 분류

* 파이프라인: 전처리의 각 단계, 모델 생성, 학습 등을 포함하는 여러 단계의 머신러닝 프로세스를 한 번에 처리할 수 있는 클래스

II. 개발 진척사항

5_3. Evaluate and Serialize Model

```
: paste cells below
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
import pickle

for algorithm, model in fit_models.items():
    yhat = model.predict(X_test)
    print(algorithm, accuracy_score(y_test.values, yhat),
          precision_score(y_test.values, yhat, average="binary", pos_label="correct_bench"),
          recall_score(y_test.values, yhat, average="binary", pos_label="correct_bench"))

lr 0.5 0.5 1.0
rc 0.5 0.5 1.0
rf 0.5 0.5 1.0
gb 0.5 0.5 1.0

yhat = fit_models['rf'].predict(X_test)

yhat[:]

array(['correct_bench', 'correct_bench', 'correct_bench', 'correct_bench'],
      dtype=object)

with open('benchpress.pkl', 'wb') as f:
    pickle.dump(fit_models['rf'], f)
```

5_4. Make Detections with Model

```
: with open('benchpress.pkl', 'rb') as f:
    model = pickle.load(f)

cap = cv2.VideoCapture(0)
counter = 0
current_stage = ''
```

평가 방법

- Accuracy(정확도)
- Precision(정밀도)
- Recall(재현율)
- F1-Score(Precision과 Recall의 조화평균)

YOLOv5 및 MediaPipe를 이용한 실시간 3대 운동 AI 트레이닝 서비스에 대한 연구

THANK YOU



컴퓨터공학과 3학년 고영민

2023.10.04