

잔소리하는 AI 헬스트레이너

Fox Fitness

13기 A반 4조

배재빈 | 김석우 | 김유민 | 이다경 | 박미소

< 목차 >

1.프로젝트 개요	4
1. 추진배경 및 프로젝트 소개	5
2. 기존 서비스 및 차별점.....	6
2. 프로젝트 상세 내용.....	8
1. 서비스 구조도.....	9
2. 구현 범위 및 방법.....	10
3. 활용 기술	
3-1. 골격 인식 - OpenPose	11
3-2. 동작 정확도 측정	14
3-3. 음성 합성 – TTS(Text-To-Speech)	16
4. Web 서비스	
3. 결론.....	21
1. 기대 효과 및 활용 분야.....	22
2. 한계점 및 개선사항	24
4. 참고 문헌	25

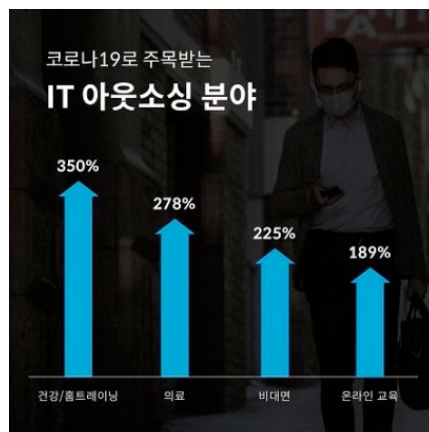
I. 프로젝트 개요

I. 프로젝트 개요

1. 추진배경 및 프로젝트 소개

□ 추진배경

최근 코로나 19로 인해 혼자하는 운동법에 대한 콘텐츠가 많이 생성되고 소비되며 헬스장 대신 집에서 하는 홈 트레이닝 시장이 빠르게 성장하고 있다. 바쁜 현대인에게 거리가 멀고 값비싼 헬스장과 같은 체육시설보다 시간과 장소의 제약을 받지 않는 홈 트레이닝이 각광받고 있다.



[그림 1] IT 아웃소싱 시장 성장률



[그림 2]체육시설 이용하지 않는 원인

하지만 홈 트레이닝에는 몇 가지 문제점이 나타난다. 예를 들어, 영상을 보고 혼자 따라해야 하는 홈트레이닝의 특성상 잘못된 자세로 운동하는 경우가 많다. 잘못된 자세로 운동을 지속할 시 신체 불균형과 부상을 초래할 수 있다. 또한 전문 트레이너의 직접적인 코칭과 독려가 없어 목표와 계획 수행이 어렵다. 이러한 문제점에서 착안하여 인공지능(AI)을 통한 개인 트레이너 기술이 필요하다고 생각하였고 인공지능(AI)을 통한 개인 헬스 트레이너 'Fox fitness' 프로젝트를 추진하게 되었다.

□ 프로젝트 소개

Fox fitness는 OpenPose기반 AI 기술을 활용하여 이용자의 실시간 관절 움직임을 검출하고 분석 후 올바른 운동 자세를 추천해주는 서비스이다. 운동 자세와 골격 인식을 인식하여 자세의 정확도를 분석해주고 틀린 동작에 대한 피드백을 해줌으로써 집에서 혼자 트레이닝을 하지만 트레이너가 옆에 있는 것과 같은 효과를 준다.

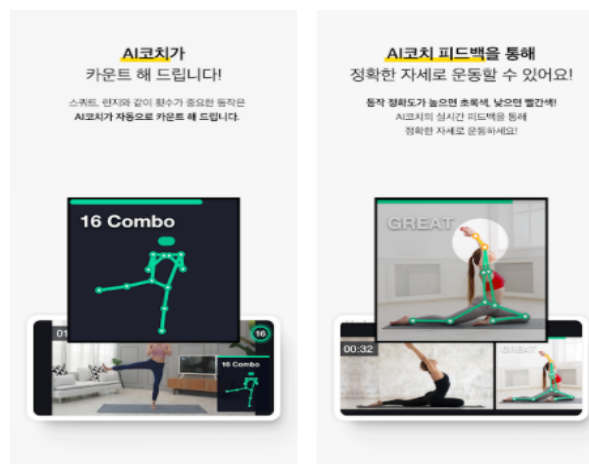
오픈소스인 openpose의 스켈레톤 인식 기술을 통하여 인체의 자세를 실시간으로 인식하고 이를 통해 특정행동을 인식하여 그에 대한 결과를 통해 사용자가 스스로 화면을 보고 정확한 운동자세를 취할 수 있도록 피드백해준다. 교정 피드백은 사용자가 멀리 있어도 소리로 본인의 자세를 피드백 받을 수 있도록 TTS API를 통해 구현하였고 세점과 코사인 공식을 이용해 원하는 관절의 각도를 추출하여 정확도 계산과 운동횟수를 카운트하였다.

2. 기존 서비스 및 차별점

□ 최근 기술 동향

1) 카카오VX ‘스마트홈트’

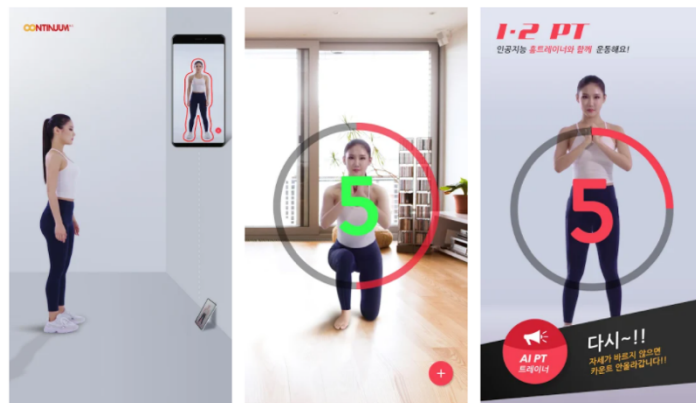
'스마트홈트'는 체계적인 피트니스 커리큘럼에 인공지능 코칭을 접목한 홈트레이닝 앱이다. 앱을 켜 채 화면을 앞에 두고 운동 동작을 따라 하면, 다양한 강사들의 정확한 동작을 학습한 AI가 카메라에 비친 자신의 동작이 얼마나 정확한지 판단해 색깔로 표시하고 조언해준다. 운동이 끝나면 신체 부위별 운동 시간과 소모 칼로리, 동작별 정확도 등이 자동으로 분석하여 사용자에게 정보를 제공한다. 하지만 실제로 서비스를 사용해본 결과 구체적인 자세에 대한 교정 피드백이 없었던 점이 아쉬웠다.



[그림 3] 스마트홈트

2) 컨티넘 '12PT(원투피티)'

스마트폰 카메라가 사용자의 운동 자세, 패턴, 관절의 움직임 등을 분석해 올바른 자세, 운동 횟수 등을 알려주는 앱이다. 운동의 강도를 '이지, 노멀, 하드'로 구성한 루틴을 기본으로 추천하는 동시에 사용자가 원하는 운동의 횟수와 순서도 직접 편집할 수 있어 사용자가 본인에게 알맞은 운동의 횟수와 종류를 직접 설정할 수 있다. 하지만 운동자 자세에 대한 실시간 분석이 없어 정확도 확인은 불가하였다.



[그림 4] 12PT

3) GymT '875 스쿼트'

'875 스쿼트'는 코치 캐릭터를 선택할 수 있는 등 운동을 게임처럼 즐길 수 있게 만들어진 서비스이다. 포즈 정확도 분석을 통해 동작마다 간단한 음성 피드백을 지원하고 코치가 매일 사용자에게 적합한 일일 스쿼트 프로그램을 만들어 준다. 이 앱 또한 실제로 체험해보니 틀린 자세를 바른 자세로 인지하는 경우가 많아 정확도가 낮은 편이었다.

□ 차별점

위의 3가지 서비스 이외에도 여러 서비스를 조사해 본 결과, 포즈 정확도 또는 틀린 부위는 표시해주는 서비스는 많았으나 자세 교정에 대한 구체적인 피드백과 운동 독려가 없다는 점을 발견하였다. Fox Fitness는 여기에 초점을 맞춰 기존의 서비스 기능에 '구체적인 음성 피드백'을 추가해 차별점을 두었다.

음성 피드백 시스템은 다음과 같다. 사용자의 자세 인식을 한 후 각도를 통한 정확도

계산을 한다. 만약, 기존 각도 범위에 맞는 동작일 경우 동작 횟수를 음성으로 카운트하고 “아주 잘했어요! 이제 3회 남았어요 힘내요.” 와 같은 격려의 멘트를 해준다. 기존 각도 범위를 벗어난 경우에는 정상 범위에서 벗어난 각도를 인식 후 해당 자세에 대한 음성 피드백을 지원한다.

기존 PT 시스템은 단순히 영상 매칭으로 자세 확인과 웨어러블 디바이스와의 연동을 통한 자세 인식으로 운동 카운트를 하는 서비스와 달리 Fox Fitness는 딥러닝 분석을 통해 오프라인 PT에서 행해지는 운동 자세 교정을 정확히 수행하여 오프라인 PT의 효과를 제공해주며, 실시간 음성 피드백을 통해서 사용자의 운동 지속을 도와주는 서비스(정확도 분석, 독려 멘트)도 제공을 하여 운동 프로그램을 완수 할 수 있도록 도와주는 차별점을 가지고 있다.

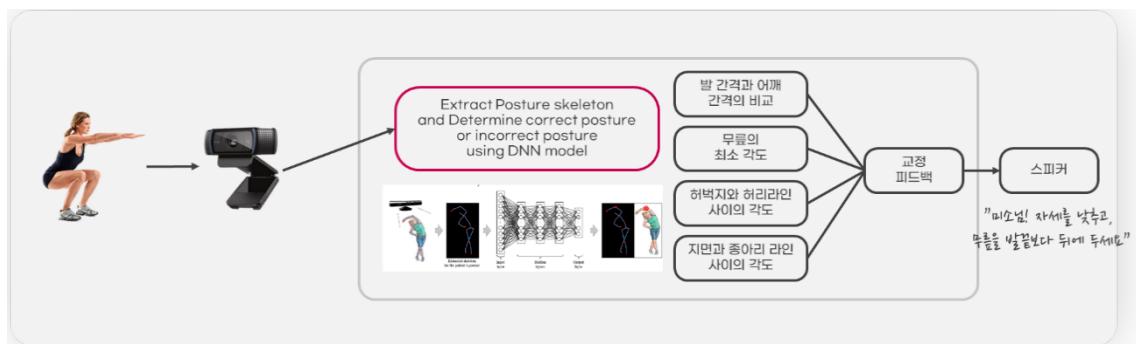
II. 프로젝트 상세 내용

II. 프로젝트 상세 내용

1. 서비스 구조도

□ 서비스 프로세스

운동 분석 전문가와 트레이너의 운동별로 정의된 Feature Component로 학습된 모델을 적용하여 사용자가 운동을 통해 실시간으로 생성된 스켈레톤 데이터를 기반으로 분석된 결과를 AI 스피커 혹은 화면으로 사용자에게 피드백 하는 기술 개발



[그림 5] fox fitness 구조도

사용자가 운동을 하면 웹캠은 이를 실시간으로 프로그램에 전송해 준다. 프로그램은 관절을 검출하는 모델을 활용하여 '발 간격과 어깨 간격, 무릎의 최소 각도, 허벅지와 허리라인 사이의 각도, 지면과 종아리 라인 사이의 각도 비교를 통해 자세 교정이 필요한 부분을 피드백 해준다. 교정 피드백은 TTS API를 통해 사용자가 멀리 있어도 소리로 본인의 자세를 피드백 받을 수 있다.

□ 사용자 프로세스

1) 사용자 인식

사용자가 설정한 프레임 안에 들어오면 화면의 어디에 어떤 각도로 서라는 안내 음성이 나온다.

2) 사용자의 위치 및 각도 조절

사용자가 정확한 위치와 각도로 설 수 있도록 안내하고 원하는 위치에 섰을 때 프로

그램을 시작한다.

```
if leg_x_point >= 100 and leg_x_point <= 220 and leg_y_point >=160 :
    location_judge = True
    leg_angle = angle_between(l_leg_x_point, leg_y_point, leg_x_point, leg_y_point, l_leg_x_point, l_leg_y_point) # 두 다리의 각도
    if leg_angle >= 25 and leg_angle <= 70 :
        angle_judge = True
        leg_gap = abs(abs(leg_x_point - l_leg_x_point) - abs(shoulder_x_point - l_shoulder_x_point)) # 발의 간격과 어깨간격 확인
        if leg_gap <= 20 or shoulder_y_point > 85:
            gap_judge = True
            if start_delay == 0: # 조건이 맞춰졌을 때 단 한번 실행
                audio('./audio/스타트.wav')
                start_delay = 1
```

위는 사용자가 정확한 자세로 섰을 때(두 다리의 각도와 발의 간격과 어깨 간격이 맞는 경우) 운동프로그램이 시작되도록 설정한 코드이다.

3) 운동 수행

운동에 대한 안내가 나오고 사용자가 운동 자세를 취하도록 안내한다.

4) 운동 동작 완료 시 피드백

운동 동작 1회 왕복 마다 운동 횟수와 정확도를 화면에 표시해 주고 자세가 틀렸을 경우 해당 자세에 대한 음성 피드백을 제공한다.

ex) 열하나(운동 횟수 안내) 잘했어요! (격려 멘트)

2. 구현 범위 및 기능

□ 구현 범위

- 인공지능(AI) 트레이너의 지시를 통해 사용자의 잘못된 운동 자세를 교정해 주고 운동의 효과를 높여주는 피트니스 PT 서비스
- 영상에서 사용자의 영역을 감지하고, 사용자의 스켈레톤 데이터 추출하는 영상 인식 기술
- 스켈레톤 데이터를 기반으로 운동 구성 Component를 정의하고 관절 각도를 추출하여 사용자의 자세에 대해 정확한 피드백을 줄 수 있는 기술
- AI 스피커 연동 음성 피드백 서비스 개발의 위한 인터페이스 개발

□ 기능

홈트레이닝에 나타나는 문제점 중 보완 가능한 사항을 인공지능 트레이너의 기능으로 추가하였다.

- 사용자의 운동 자세 인식(Open pose)을 통한 자세 정확도 분석
- 잘못된 자세와 운동 정보에 대한 실시간 음성 피드백
- 운동 완료 시 최종 정확도 및 소모 칼로리 등 운동 정보 제공

3. 활용기술

3-1 골격 인식 – OpenPose



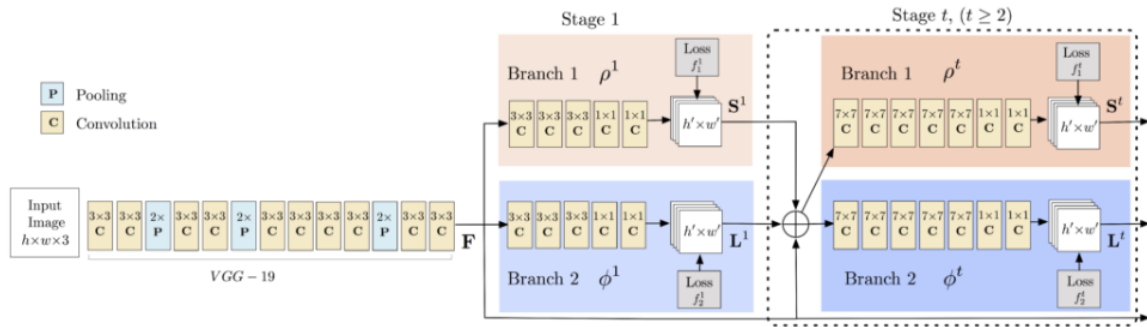
[그림 6] OpenPose 로고

1) 개요

OpenPose는 CNN 기반의 딥러닝 기술로 이미지 혹은 영상을 input으로 하여 사람의 움직임을 감지한다. Caffe와 OpenCV를 기반으로 구성된 손, 얼굴 등을 포함한 몸의 움직임을 추적해주는 API이다.

OpenPose는 센서 없이 일반 카메라로 이미지와 동영상을 추출할 수 있다. 이미지 내 관절을 먼저 파악하고 자세를 추정하는 Bottom-up 방식이며, 추정한 자세로 사람 신체부위의 좌표값을 추출한다는 특징이 있다.

2) 작동원리



[그림 7] OpenPose CNN 구조

- 1) 좌측단의 입력 데이터를 VGGNet-19를 통해 수행된 Output 데이터의 feature를 강조한 상태로 출력한다.
- 2) 출력된 output을 2번의 Branch를 통해 전파를 수행하게 되는데, 첫번째 분기점에서는 전반적인 신체 부위의 특징에 사용된다.
- 3) 반복되는 Stage에 따라 가지를 거쳐서 confidence map과 affinity field를 구한다. 여기서 confidence map이란 인간의 각 부위 간 유사성에 기반하여 2D vector fields를 예측하는 것이다.. 이는 인간의 관절 구조 등을 찾는 데 사용된다. 또한 affinity field는 추출된 관절 구조가 어떤 객체의 것인가에 대해 알아내는데 사용된다.
- 4) 연산된 confidence map과 affinity field로 영상 혹은 이미지에 존재하는 사람들의 keypoint를 예측하고 생성한다. 이러한 1번부터 3번까지의 과정을 반복하여 학습하면 OpenPose가 완성된다.

아래는 기학습되어 있는 모델을 OpenCV로 로드하여 구현한 결과이다.

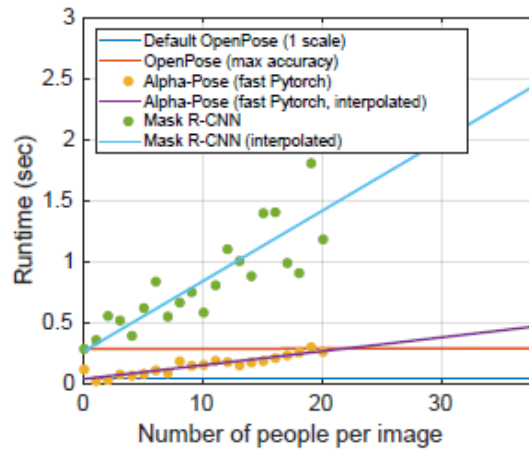


[그림 8] OpenPose 예시

3) 모델 생성

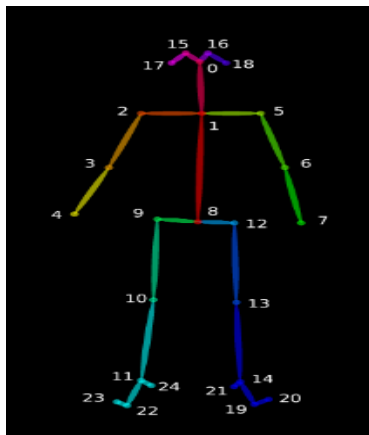
3-1) 모델 선정 이유

Multi-person pose estimation model 중에서 가장 유명한 모델이 OpenPose와 Alphapose 그리고 Mask R-CNN이 있다. 그 중에서 아래 [그림]을 보면 OpenPose모델은 하나의 이미지에서 사람의 수가 늘어나도 실행속도가 낮아지지 않는 것을 알 수 있다. Frame안에 많은 사람들이 들어와도 빠른 속도로 감지 할 수 있는 OpenPose모델을 최종적으로 선택하였다.



[그림 9] Multi-person pose estimation model 간 성능 비교

3-2) Body Keypoints 추출



[그림 10] Skeleton 출력 예시

0 :Nose	1 :Neck	2 :RShoulder	3 :RElbow	4 :RWrist
5 :LShoulder	6 :LElbow	7 :LWrist	8 :MidHip	9 :RHip
10 :RKnee	11 :Rankle	12 :LHip	13 :LKnee	14 :LAnkle
15 :REye	16 :LEye	17 :REar	18 :LEar	19 :LBigToe
20 :LSmallToe	21 :LHeel	22 :RBigToe	23 :RSmallToe	24 :RHeel

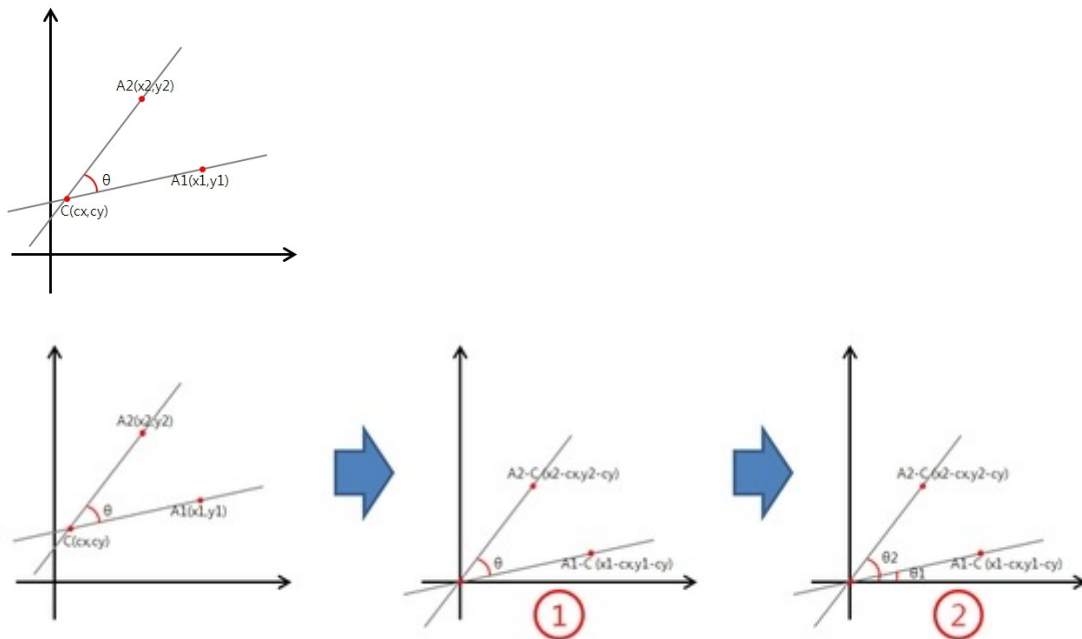
[그림 11] 신체부위 인덱스

Openpose의 24개의 keypoint중 운동 자세 측정에 필수적이지 않다고 생각되는 발과 관련된 6개의 포인트를 제외하고 18개의 keypoint를 선택하였다.

3-2 정확도 측정

1) 각도 추출

운동 자세에 대한 정확한 피드백을 주기 위해 관절 각도 계산이 필요하다. 두 직선 사이의 각도(사이각)를 계산하기 위해 세 점의 좌표를 받고 arc tangent를 이용해서 계산하였다.



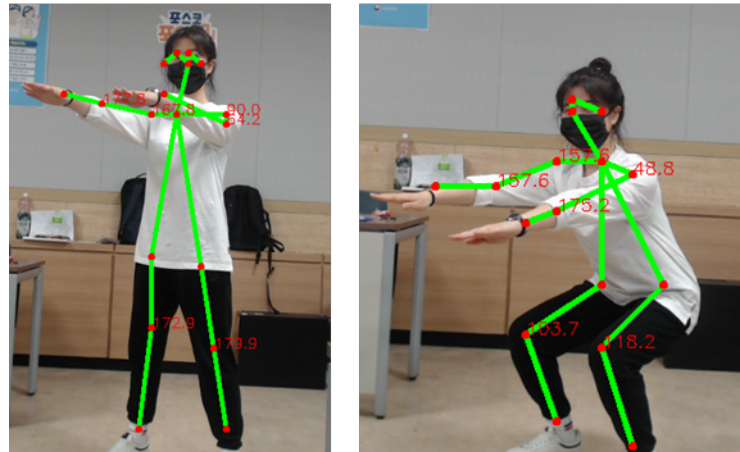
$$\angle = \theta_2 - \theta_1$$

$$\angle = \text{atan}((y_2 - cy)/(x_2 - cx)) - \text{atan}((y_1 - cy)/(x_1 - cx))$$

위 공식을 통해 사이각을 구할 수 있다.

```
def angle_between(x1, y1, x2, y2, x3, y3): # 각도 구하는 함수
    deg1 = (360 + degrees(atan2(x1 - x2, y1 - y2))) % 360
    deg2 = (360 + degrees(atan2(x3 - x2, y3 - y2))) % 360
    return deg2 - deg1 if deg1 <= deg2 else 360 - (deg1 - deg2)
```

위는 세 점의 좌표를 받아 원하는 각도를 구하는 코드이다.



[그림 12] 실시간 각도 추출

위는 웹캠영상에서 실시간으로 각도를 계산하여 표시한 결과이다.

2) 운동(스쿼트) 횟수 카운트

무릎각도가 일정 각도 이하가 되면 스쿼트를 한 것으로 인정

각도 설정

웹캠을 통한 영상에서 사용자의 각도는 45°로 들어온다. 45° 각도에서 촬영된 영상 36개의 데이터셋을 분석해본 결과 해당 각도에서 무릎의 실제 각도가 90° 일 때, 114.5°로 측정되는 것을 확인하였다. 따라서 기준 각도를 114.5°로 설정하였다.

```
else :
    squat_pos = 0 # 각도가 114.5도 이하일 때 squat_pos 인정
    if prev_squat_pos - squat_pos == 1: # 일어날 때 count 1추가
        count_of_squats += 1
        if count_of_squats == count_of_audio :
            audio(f"./audio/{count_of_squats}.wav")
            count_of_audio += 1
    prev_squat_pos = squat_pos # 이전 프레임의 자세값 입력
```

위는 무릎각도가 114.5도 이하일 때 스쿼트 자세를 인정하고 일어날 때 스쿼트 횟수를

1회 추가하는 코드이다.

3) 정확도 측정

스쿼트는 동작 정확도를 계산하기 위해 논문을 참고하여 세가지 주요 지표를 선정하였다. 스쿼트는 하체 운동이기 때문에 하체 동작에 주요한 각도를 종합하여 정확도를 구하였다

① 등 라인과 허벅지 라인의 각도

② 무릎의 각도

③ 지면과 종아리 라인의 각도

각도 기준을 선정하기 위해 36개의 정확한 스쿼트 자세 영상 데이터를 분석하여 기준 각도를 추출하였다. 각 관절 각도가 기준 각도의 오차범위를 넘어갈 때 해당 각도의 차이만큼 정확도가 감소한다. 정확도를 구하는 방법은 다음과 같다.

1) 하체 주요 3가지 각도를 기준 각도와 비교해 그 차이만큼 정확도 점수를 뺀다.

각 각도의 scale을 맞추기 위해 가중치를 부가하여 각도 점수를 계산하였다.

2) 무릎 각도에 해당하는 점수를 40%, 등 라인과 허벅지 라인의 각도에 해당하는 점수를 30%, 지면과 종아리 라인의 각도에 해당하는 점수를 30% 반영하여 최종 정확도 점수를 구한다.

```
if knee_theta <= 135.5 :  
    if knee_theta <= 114.5 and knee_point <= 40 and waist_theta >= 80 :  
        squat_pos = 1  
        knee_theta_score = 100-(2*abs(100-knee_theta))  
        knee_poin_score = 100-(4*abs(30-knee_point))  
        waist_theta_score = 100-(1*abs(100-waist_theta))  
        squat_accuracy = (knee_theta_score*0.4) + (knee_poin_score * 0.3) + (waist_theta_score * 0.3)
```

위는 각 관절 각도를 이용해 스쿼트 동작 정확도를 구하는 코드이다.

3-3 음성 합성 TTS (Text-To-Speech)

텍스트 입력을 오디오 데이터로 변환하는 프로세스를 합성이라고 하며 합성 출력을 합성 음성이라고 한다. Text-to-Speech는 원시 텍스트나 SSML 형식의 데이터와 같은 두 유형의

입력을 사용한다.

Google API – Text-to-Speech 사용



[그림 13] Google TTS

1) 개요

Google의 AI 기술을 기반으로 한 API 를 사용하여 텍스트를 자연스러운 음성으로 변환한다. DeepMind의 음성 합성 전문 기술을 기반으로 제작되어 인간과 흡사한 수준의 고품질 음성을 제공한다. 40개가 넘는 언어 및 방언을 지원하는 220여 개의 음성 조합 중에서 선택할 수 있다.

2) 주요 특징

- 90여개의 WaveNet 음성을 활용하여 사람 목소리와의 격차를 대폭 축소하는 음성을 생성한다.
- 선택한 음성의 높낮이를 기본 출력보다 최대 20반음까지 높거나 낮게 맞춤설정할 수 있으며 말하기 속도를 일반 속도보다 4배 빠르거나 느리게 조정할 수 있다.

3) 기술 적용

```
pip install --upgrade google-cloud-texttospeech
```

위의 코드로 우분투에서 간단하게 설치가 가능하다.

```

"""Synthesizes speech from the input string of text or ssml.

Note: ssml must be well-formed according to:
    https://www.w3.org/TR/speech-synthesis/
"""
from google.cloud import texttospeech

# Instantiates a client
client = texttospeech.TextToSpeechClient()

# Set the text input to be synthesized
synthesis_input = texttospeech.SynthesisInput(text="Hello, World!")

# Build the voice request, select the language code ("en-US") and the ssml
# voice gender ("neutral")
voice = texttospeech.VoiceSelectionParams(
    language_code="en-US", ssml_gender=texttospeech.SsmlVoiceGender.NEUTRAL
)

# Select the type of audio file you want returned
audio_config = texttospeech.AudioConfig(
    audio_encoding=texttospeech.AudioEncoding.MP3
)

# Perform the text-to-speech request on the text input with the selected
# voice parameters and audio file type
response = client.synthesize_speech(
    input=synthesis_input, voice=voice, audio_config=audio_config
)

# The response's audio_content is binary.
with open("output.mp3", "wb") as out:
    # Write the response to the output file.
    out.write(response.audio_content)
    print('Audio content written to file "output.mp3"')

```

위 코드를 사용하여 Speech-to-Text API로 synthesize 변환 요청을 보낸다.

```
def synthesize_text(text):
    """Synthesizes speech from the input string of text."""
    from google.cloud import texttospeech

    client = texttospeech.TextToSpeechClient()

    input_text = texttospeech.SynthesisInput(text=text)

    # Note: the voice can also be specified by name.
    # Names of voices can be retrieved with client.list_voices().
    voice = texttospeech.VoiceSelectionParams(
        language_code="en-US",
        name="en-US-Standard-C",
        ssml_gender=texttospeech.SsmlVoiceGender.FEMALE,
    )

    audio_config = texttospeech.AudioConfig(
        audio_encoding=texttospeech.AudioEncoding.MP3
    )

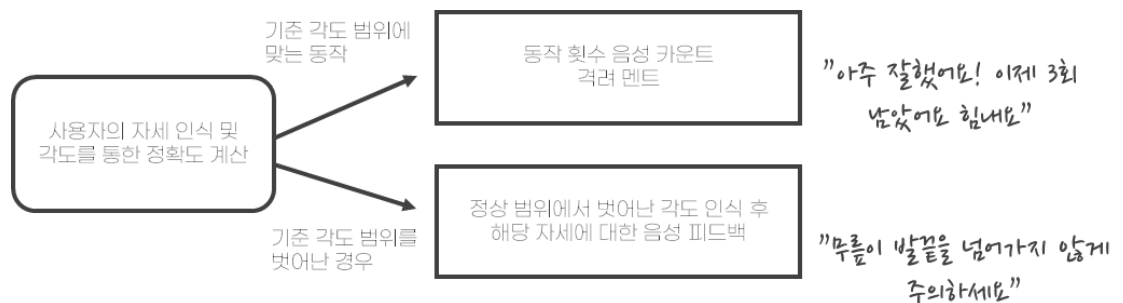
    response = client.synthesize_speech(
        request={"input": input_text, "voice": voice, "audio_config": audio_config}
    )

    # The response's audio_content is binary.
    with open("output.mp3", "wb") as out:
        out.write(response.audio_content)
        print('Audio content written to file "output.mp3"')
```

위는 문자열을 오디오 데이터로 변환하는 코드이다. 고유한 음성을 선택하거나 출력의 높낮이, 볼륨, 말하기 속도, 샘플링 레이트를 조정하는 등 다양한 방식으로 음성 합성 출력을 구성할 수 있다.

4) 음성 피드백 시스템

동작인식과 관절각도 비교를 통해 틀린 자세에 대한 음성 피드백 기능을 제공하고자 한다.



[그림 14] 음성 피드백

사용자의 자세 인식 및 각도를 통한 정확도 계산을 한 뒤 기준 각도 범위에 맞는 동

작을 할 경우 동작 횟수를 음성으로 알려주고 격려 멘트를 출력한다. 기존 각도 범위를 벗어난 경우 정상 범위에서 벗어난 각도 인식 후 해당 자세에 대해 음성 피드백을 준다.

```
elif knee_theta > 114.5 and feedback_delay >=20 :
    audio('./audio/영당이.wav')
    feedback_delay = 0
elif waist_theta < 80 and feedback_delay >=20 :
    audio('./audio/허리.wav')
    feedback_delay = 0
elif knee_point > 40 and feedback_delay >=20 :
    audio('./audio/무릎.wav')
    feedback_delay = 0
else :
    squat_pos = 0 # 각도가 114.5도 이하일 때 squat_pos 인정
if prev_squat_pos - squat_pos == 1: # 일어날 때 count 1추가
    count_of_squats +=1
    if count_of_squats == count_of_audio :
        audio(f"./audio/{count_of_squats}.wav")
```

위는 각 관절의 각도가 기준 범위를 벗어날 경우 해당 자세에 대한 음성 피드백을 출력하도록 설정한 코드이다. 스쿼트 횟수 또한 음성으로 안내해준다.

Ⅲ. 결론

Ⅲ. 결론

1. 기대 효과 및 활용 분야

□ 기대 효과

1) 스마트 헬스케어 시장의 활성화

지역 내 헬스센터와 연계하여 1:1 온라인 방문 코칭 서비스를 통해서 사용자에게는 운동을 자유롭게 할 수 있는 시간의 자유를 제공하며, 헬스센터에는 기존 고객에 대해 운동의 효율성을 제공함으로써 헬스(피트니스) 시장에 대한 사업화와 신시장 창출이 가능하다.

2) 타 산업 분야와의 융복합

영상 인식 기반의 인공지능 원천 기술을 타 산업 분야(재활 의료 산업, 스마트 공장 산업 등)에 활용함으로써 고부가가치 서비스 제공이 가능하여 관련 산업 활성화 및 매출 증대를 기대할 수 있다. 또한, 영상 분석 기반의 인공지능 서비스가 다양한 형태의 IoT 제품 기기(스마트 TV 등)에 탑재되어 융·복합 제품의 기능 향상 및 추가 서비스 발굴에 따른 수익 모델 확보가 가능하다.

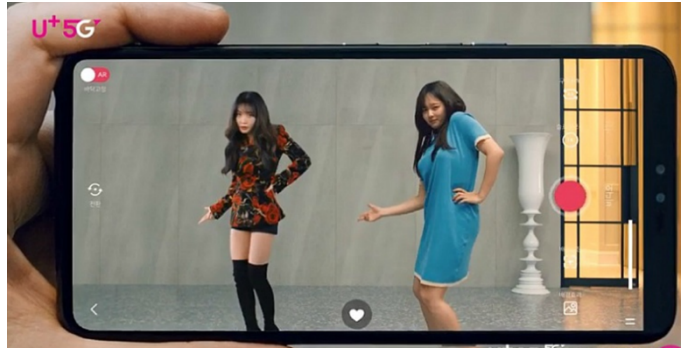
3) 경제적 기대효과

인공지능 PT 시스템 제작/판매 관련 판로 개척을 통해 생산 및 제품 품질 담당자 채용 등 일자리 창출을 기대할 수 있다.

□ 활용 분야

1) AI 댄스 트레이너

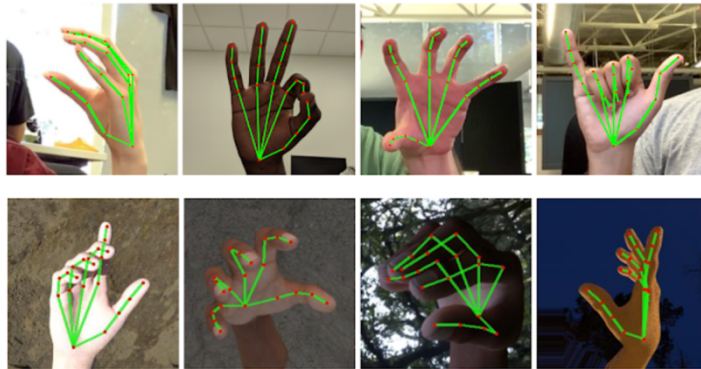
댄스 트레이너의 움직임을 스켈레톤으로 추출해 움직이는 춤추는 스켈레톤 영상 생성한다. 영상을 보고 사용자가 따라할 수 있도록 해 춤 동작에 대한 피드백 제공 가능하다.



[그림 **] 가수가 옆에서 직접 춤을 추는 듯한 AR컨텐츠

2) 수화 번역

청각장애인들을 위한 실시간 수화 번역 서비스 제공한다. CNN을 이용하여 특징을 추출하는 OpenPose 라이브러리와 RNN을 사용한다. 대화에 어려움을 느끼는 장애인 뿐만 아니라 일들과 대화를 함께하는 비장애인들에게도 유용하게 작용되어 장애인과 비장애인 사이의 대화의 벽을 허물 수 있는 것으로 기대된다.



3) 아동 학대 감시 모델 구현

기존의 Haar 기법을 사용하여 성인과 아동을 분류하려면 정면 영상이 필요하다. OpenPose를 사용하면 정면과 측면에 구애받지 않고 성인과 아동을 분류할 수 있어서 추가적 설치 없이 현재 설치된 CCTV를 활용해 아동이 학대를 당할 때 성인과 아동의 자세의 특성을 적용한다. 이를 이용하면 실시간으로 아동학대가 발생하고 있는지 모니터링하여 빠른 대처가 가능할 것이다.



[그림 **] 영상 이미지를 OpenPose를 사용하여
관절을 추출한 결과

2. 한계점 및 개선사항

1) 고해상도 속도 저하 문제

고해상도로 영상을 송출하면 딜레이 현상이 지속적으로 발생하였다. 이러한 문제는 Open CV CUDA 연동과 GPU사용으로 속도를 개선을 시도하였다. 우선, 영상 해상도를 640 x 480 에서 320 x 240 으로 낮추었다. 그리고 운동 시작 전 사용자가 자세를 잡을 때만 골격을 표시하고 운동 할 때는 골격을 나타내지 않도록 하였다. 그 결과, 12 FPS에서 18 FPS 로 속도 문제 개선할 수 있었다.

2) 음성 합성 문제

실시간으로 음성 피드백을 넣었을 때 속도 지연과 관련된 문제가 발생하였다. API를 사용해 실시간으로 음성을 출력하는 경우, 음성을 합성하고 인코딩 후 재생하는 과정으로 인해 출력 지연 문제 발생한 것이다.

이러한 문제는 미리 정해진 상황에 맞는 음성을 mp3 파일로 만들어 두는 방식으로 속도 문제를 개선시켰다.

IV. 참고 문헌

IV. 참고문헌

1. Zhe Cao, Student Member, IEEE, OpenPose : Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields
2. Steven Chen, Richard R.Yang, Pose Trainer: Correcting Exercise Posture using Pose Estimation, 2020. 06
3. 최동규, 장종욱, OpenPose와 심도카메라를 활용한 운동 및 재활 동작 관절각도 연산 방법 연구, 한국정보통신학회 종합학술대회 논문집 23(2), 2019.10, 14-17(4 pages), 2019.10
4. 한국전자통신연구원, 과제번호 R2014040035, 생체역학적용 K-POP 댄스 안무 검색 및 자세 정확성 분석 기술 개발, 2017. 06. 23
5. Marco C. Uchida, Renato Carvalho², Identification of muscle fatigue by tracking facial expressions, 2018. 12
6. 중소벤처기업부 Ministry of SMEs and Startups, 딥러닝을 활용한 운동 자세 분석 기술 및 인공지능 PT(Personal Training) 시스템 개발, 2018. 11
7. 장재호, 지준환, 윤태진, Development of exercise posture training system using deep learning for human posture recognition, 2020. 07
8. Travis Chan, tkeady⁵, Thomas Keady, FormCheck: Improve Your Strength Training Form
9. 김형균, 홍호표, 김용호, 딥러닝 기반 포즈인식을 이용한 체력측정 시스템, Journal of Digital Convergence Vol. 18. No. 12, pp. 97-103, 2020.12
10. 문화체육관광 연구개발사업, 해외 K-POP 커버댄스 영상의 한류 확산을 위한 AI 댄스 모션 피드백 기술 및 하이퀄리티 동영상 모바일 편집 앱 서비스 개발, 2019.08