요구사항 분석

기능적

1. 회원관리
    - 회원가입
    - 로그인
    - 로그아웃
    - 수정
2. 댄스 튜토리얼 기능
    - 단계
        - 사용자의 모션 인식
        - 스탭 댄스
        - 전신 댄스
3. 검색 기능
    - 강사 검색 : 특정 강사가 춘 영상이 결과로
    - 춤 검색 : 해당 춤과 관련된 영상이 결과로
    - 장르 : 해당 장르와 관련된 영상이 결과로
4. URL 을 통한 유튜브 쇼츠 불러오기 기능
    - 이건 뭐 사용자가 올리고 싶은데로
5. 영상의 강사와 자신의 몸 동작을 보여주는 기능
    - 스켈레톤 on/off 기능 제공
6. 몸 동작의 일치율에 따른 정보를 출력하는 기능
    - 50% 밑 : BAD
    - 50% 이상 90% 밑 : GOOD
    - 그 이외 : PERFECT
7. 영상의 속도, 크기를 조절하는 기능
    - 속도
        - 0.5, 0.75, 1, 1.5, 2
    - 크기
        - 원래크기, 3 분의 2, 2 분의 1
8. 영상을 마이페이지에 저장하고 다운할 수 있는 기능
    - 저장 시 파일 뭐로 제공 해야하지
9. 강사 등록 기능
    - 어느 정도 역량이 있는 강사를 등록

비기능적

1. 사용자 보안
2. 페이지간 이동 속도는 0.5 초 이내이여야 한다.
3. 강사 신용도
4. 모션 인식 정확도

5. Java, Python 사용
6. 모션인식 : MMPose 확정

음악 : SPotify API

로그인 : Social Login API (Google, Kakao, Naver)

소스 : YouTube API

AI 테스트

Mmpose 모델 사용

https://github.com/open-mmlab/mmpose -> mmpose 학습 모델(RTMO, justDance) 프로젝트 이용 예정

설치가이드

https://mmpose.readthedocs.io/en/latest/installation.html

conda 환경 생성

```
conda create --name openmmlab python=3.8 -y
conda activate openmmlab
```

torch 버전이랑 cuda 등등 버전이 안맞으면 문제 발생

```
conda install pytorch==2.0.0 torchvision==0.15.0 torchaudio==2.0.0 pytorch-cuda=11.7 -c pytorch -c nvidia
```

```
pip install -U openmim
mim install mmengine
mim install "mmcv==2.0.1"
```

```
mim install "mmdet>=3.1.0"
```

```
git clone https://github.com/open-mmlab/mmpose.git
cd mmpose
pip install -r requirements.txt
pip install -v -e .
```

```
mim install "mmpose>=1.1.0"
```

```
mim download mmpose --config td-hm_hrnet-w48_8xb32-210e_coco-256x192  --dest .
```

```
python demo/inferencer_demo.py $IMAGE --pose2d rtmo --vis-out-dir vis_results
```

이 명령어를 통해 생성한 영상 결과

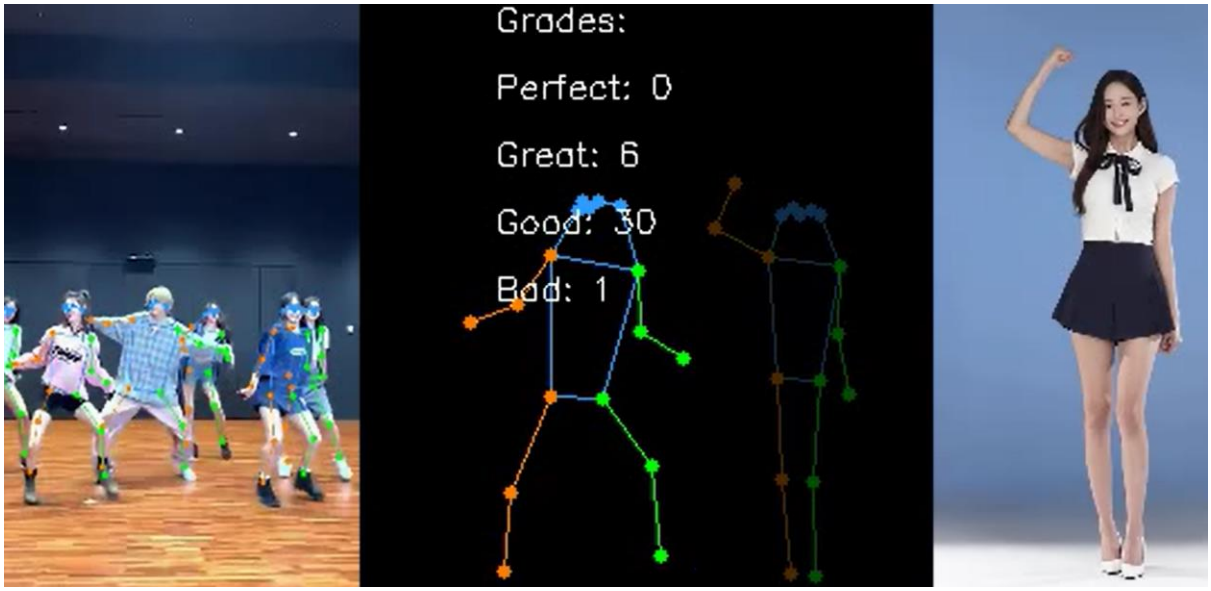
이 코드는 process_video.py 의 일부이다. 기존의 코드의 점수 였던걸 등급의 개수로 세분화 하는 코드로 수정을 하였다.

```
def generate_output_video(self, tch_video: str, stu_video: str,
                          output_file: str, tch_kpts: np.ndarray,
                          stu_kpts: np.ndarray, piece_info: dict) -> str:
        """Generate an output video with keypoints overlay."""

        tch_video_reader = mmcv.VideoReader(tch_video)
        stu_video_reader = mmcv.VideoReader(stu_video)
        for _ in range(piece_info['tch_start']):
            _ = next(tch_video_reader)
        for _ in range(piece_info['stu_start']):
            _ = next(stu_video_reader)

        grade_counts = {"Perfect": 0, "Great": 0, "Good": 0, "Bad": 0}
        video_writer = None
        for i in track_iter_progress(range(piece_info['length'])):
            tch_frame = mmcv.bgr2rgb(next(tch_video_reader))
            stu_frame = mmcv.bgr2rgb(next(stu_video_reader))
            tch_frame = resize_image_to_fixed_height(tch_frame, 300)
            stu_frame = resize_image_to_fixed_height(stu_frame, 300)

            stu_kpt = get_smoothed_kpt(stu_kpts, piece_info['stu_start'] + i,
5)
            tch_kpt = get_smoothed_kpt(tch_kpts, piece_info['tch_start'] + i,
5)

            # draw pose
            stu_kpt[..., 1] += (300 - 256)
            tch_kpt[..., 0] += (256 - 192)
            tch_kpt[..., 1] += (300 - 256)
            stu_inst = InstanceData(
                keypoints=stu_kpt[None, :, :2],
                keypoint_scores=stu_kpt[None, :, 2])
            tch_inst = InstanceData(
                keypoints=tch_kpt[None, :, :2],
                keypoint_scores=tch_kpt[None, :, 2])

            stu_out_img = self.visualizer._draw_instances_kpts(np.zeros((300,
256, 3)), stu_inst)
            tch_out_img = self.visualizer._draw_instances_kpts(np.zeros((300,
256, 3)), tch_inst)
            out_img = blend_images(stu_out_img, tch_out_img, blend_ratios=(1,
0.3))

            # draw grade
```

```
            score_frame = piece_info['similarity'][i]
            grade = score_to_grade(score_frame * 10)  # score_frame을 0-10
스케일로 변환
            grade_counts[grade] += 1

            self.visualizer.set_image(out_img)
            self.visualizer.draw_texts('Grades: ', (60, 30),
                                       font_sizes=15,
                                       colors=(255, 255, 255),
                                       vertical_alignments='bottom')
            y_offset = 60
            for g, count in grade_counts.items():
                self.visualizer.draw_texts(f'{g}: {count}', (60, y_offset),
                                           font_sizes=15,
                                           colors=(255, 255, 255),
                                           vertical_alignments='bottom')
                y_offset += 30
            out_img = self.visualizer.get_image()

            # concatenate
            concatenated_image = np.hstack((stu_frame, out_img, tch_frame))
            if video_writer is None:
                video_writer = cv2.VideoWriter(output_file,
                                               cv2.VideoWriter_fourcc(*'mp4v'),
                                               30,
                                               (concatenated_image.shape[1],
concatenated_image.shape[0]))
            video_writer.write(mmcv.rgb2bgr(concatenated_image))

        if video_writer is not None:
            video_writer.release()
        return output_file
```

python process_video.py ${TEACHER_VIDEO} ${STUDENT_VIDEO} –output-file ${원하는 파일경로와 이름}

이 명령어를 통해 아래와 같은 결과 생성



추가 사항으로

오디오와 비디오를 분리해서 합쳐야 할거 같음

conda install -c conda-forge ffmpeg

오디오 추출

ffmpeg -i input_video.mp4 -q:a 0 -map a output_audio.mp3

추출한 오디오와 결과 영상과 합치기

ffmpeg -i processed_video.mp4 -i audio.mp3 -c:v copy -c:a aac final_video.mp4