1. 아이디어의 시작

저번 학기에 수학프로그래밍 수업에서는 내가 좋아하고 취미로 하고 있는 음악에 활용할 수 있는 방법을 배운 프로그래밍 지식을 통해 만들고자 했고 그렇게 만들었던 작곡 프로그램은 굉장히 유용하게 쓸 수 있었다. 그래서 이번에도 작곡 생활에 쓸 수 있는 프로그램을 만들되 더욱 많이 배운 지식을 어디에 활용해야 효과적일지를 생각해보던 도중 샘플링에 써보면 어떨까 생각했다. 샘플링이란 원래 있던 곡의 일부를 요소로 가지고 와 노래에 리믹스, 즉 재활용하는 방법이다. 이는 실제로 검증된 드럼 사운드나 베이스 사운드 등을 사용하고 싶을 때 가장 많이 채택하는 방식으로 실제 업계에서도 너무나 흔하게 볼 수 있는 방법이기도 하다. 다만 곡을 추출하고 분리하는 과정에서 음원 손실의 위험이 너무나도 크고 분리하는 프로그램 또한 가격이 높다. 그래서 내가 직접 AI를 이용한 프로그램을 만들면 돈도 아끼며 내가 원하는 결과물을 낼 수 있고 퀄리티 또한 더 높일 수 있지 않을까라는 생각으로 이 프로젝트를 시작하게 되었다.

1. 베이직 프로그램 설정 (base.py)

검색 결과, 음악 스케일을 만들어주던 도구가 이미 있었던 것처럼 음악의 각 악기 (이를 스템이라고 부른다) 를 분리해주는 도구가 이미 파이썬에 존재했다. 이를 spleeter 라고 하는데 이를 이용해서 처음에는 노래를 입력하면 스템을 분리해 각 파일에 저장해주는 가장 간단한 프로그램을 만들었다. 여기에서는 가장 자세한 분리를 하기 위해서 5-스템 모델을 사용하여 보컬, 드럼, 베이스, 피아노, 그리고 기타 악기 5가지로 분리할 수 있었다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

이를 실행하면 총 5개의 분리된 스템 파일이 나오게 된다.

1. 편의성 추가 (visual.py)

내가 실제로도 사용하기 위해서 만드는 프로그램인 만큼 가시적으로 보이는 정보가 간략하면서도 핵심적으로 모든 파일에 표시되었음 하였다. 이를 위해서 파일에 내가 샘플을 사용하기 위해 가장 중요하게 인지하고 있어야 하는 원본의 템포와 키를 추출 파일에 표시되도록 프로그램을 수정하였다. 원본 파일의 키와 템포를 알기 위해서는 librosa 라이브러리를 이용해야 했기 때문에 spleeter와 함께 설치해주었다. 추가 과정에서 노래 제목도 추가해주도록 하였다.

텍스트, 스크린샷, 폰트, 문서이(가) 표시된 사진

자동 생성된 설명

이제 파일은 제목\_스템\_템포\_키의 형식으로 저장이 될 수 있다.

1. 간단한 편집 기능 추가 (shifter.py)

샘플링 기법의 가장 큰 특징은 이를 임의대로 편집해서 원곡의 요소와 같은 요소를 쓰면서도 다른 느낌을 줄 수 있다는 점이다. 나 또한 이런 샘플링의 특징을 살리기 위해 원곡 템포와 다르게 내가 조정할 수 있도록 프로그램을 고쳐 보기로 했다. 내가 입력하는 키와 템포에 따라 샘플이 미리 편집되어 나오도록 코드를 작성하였다. Time stretching 과 pitch shifting 을 통해서 이를 할 수 있다.

텍스트, 폰트, 스크린샷, 친필이(가) 표시된 사진

자동 생성된 설명

이제 내가 지정한 값에 따라 파일은 제목\_스템\_템포\_키의 형식으로 저장이 된다.

1. 오디오 효과 추가 (effect.py)

추출한 스템들의 퀄리티는 원본에 비해서 손상이 되어 있기 때문에 낮아질 수 밖에 없다. 하지만 이를 직접 활용하기 위해서는 최대한 좋은 퀄리티를 유지하는 것이 중요하고 이는 리버브, 로우 패스 필터, 하이 패스 필터 등의 효과로 스템 사운드의 단점을 가리고 장점을 극대화할 수 있다. 직접 하는 방법도 있지만 나의 초기 목표는 최대한 편의성이 극대화 된 프로그램을 만드는 것이었기 때문에 이 또한 미리 프로그램에서 추가할 수 있도록 만들었다.

텍스트, 스크린샷, 폰트, 친필이(가) 표시된 사진

자동 생성된 설명

임의의 사운드 이펙트를 걸 수 있도록 하였고, 리버브, 로우 패스 필터, 하이 패스 필터, 그리고 볼륨을 줄일 수 있는 기능을 넣었다. 이제 이들 중 하나를 골라서 실행을 시킨다면 결과 파일은 제목\_스템\_템포\_키\_효과의 형식으로 저장이 된다.

1. 퀄리티를 위하여 (denoised.py)

효과만 들어간 스템들의 퀄리티는 여전히 만족스럽지 못했다. 따라서 추출된 스템들의 퀄리티를 높일 방법을 찾다가 AI 리마스터링 기능이 있는 고급 AI 모델인 Demucs를 찾게 되었다. 다만 이 프로그램을 돌리기 위해서는 충분한 GPU 기능이 받쳐줘야 했는데 아쉽게도 내가 이용할 수 있는 환경에서는 적합하지 않았다. 이후에 가능한 환경이 주어지면 사용해보기 위해서 demucs.py 를 만들어 뒀지만 예비 프로그램일 뿐이지 당장 사용하기 위해서는 다른 기능이 필요하였다. 따라서 방법을 생각하던 중 스템들이 원본에 비해서 대체로 머디하다는 것, 즉 소리가 둔해졌다는 것을 느꼈고 이를 해결하기 위해서 다른 음역대의 소리를 완전히 걸러줄 수 있는 노이즈 제거 기능을 넣으면 좋을 거 같다고 생각하게 되었다. 이를 위해 noisereduce 라이브러리를 사용하였고, 이를 하면서 너무 많은 소리가 걸러지지 않도록 실시간으로 효과가 입혀진 파일을 들으며 노이즈 제거 정도를 정할 수 있으면 좋겠다는 생각도 하여서 그 기능들 또한 추가했다.

텍스트, 스크린샷, 폰트, 문서이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트, 문서이(가) 표시된 사진

자동 생성된 설명

이를 통해 실시간으로 음악을 들어보면서 노이즈 제거 정도를 low, medium, high 중 정할 수 있게 되었고 최종 선택을 하는 강도로만 파일을 뽑을 수 있게 되었다. 파일 형식은 제목\_스템\_denoised\_강도로 저장이 되도록 하였다.

1. 범용성 확장 (lotsoffiles.py)

한 번에 여러 곡을 처리하면 프로그램의 범용성이 더 좋을 거 같아 단일파일과 디렉토리를 전부 커버할 수 있는 코드를 넣었고 이 프로그램을 돌릴 때 너무 많은 결과물이 나오면 정리를 하는 것도 힘들고 가시적으로도 불편할 것 같아 노래별로 압축파일로 묶여서 나오도록 기능을 추가하였다.

텍스트, 스크린샷, 폰트, 문서이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 폰트, 친필이(가) 표시된 사진

자동 생성된 설명

이 프로그램을 이용하면 각 노래별로 결과 압축 파일이 생성되며, 각 파일에 대해 선택된 노이즈 제거 강도를 적용한 결과물을 포함하게 된다.

1. 결과

이후에도 더 많은 편의성 기능을 추가해주기 위해 구글 드라이브 연동 및 공유 링크 자동 생성 등의 기능을 더해봤지만 이미 노래별로 스템 압축 파일을 만들어 뽑아낸 것만으로도 충분히 일이 줄어든 거 같고 오히려 프로그램을 돌릴 때 범용성을 줄이는 것 같아 결론적으로 추가하지 않았다. 이번에 프로젝트를 진행하면서 가장 크게 느낀 점은 AI에 기반한 여러 도구들이 상상 이상의 편의성을 가져다 줄 수 있다는 것이다. 저번 프로젝트는 창의성을 수치화 시킬 수 있는가에 대한 것이어서 사실 만족스러운 결과물을 얻기 힘들었지만 이번 프로젝트는 어찌보면 수치화 된 결과물을 임의로 변경하여 사용할 수 있는가에 관한 것이었기 때문에 생각 이상으로 결과물이 좋게 나온 것 같다. 실제로 이번에 만든 프로그램은 외부에 돈을 주고 팔고 있는 AI 프로그램들과 많이 다르지 않다는 것을 느낄 수 있었고 프로그래밍의 활용도에 대해 감탄하게 된 계기였다.

1. 추가적인 궁금증

앞에서 demucs 를 활용한 프로그램을 만들었지만 컴퓨터의 한계로 사용해보지 못하였다. 그렇다면 내가 사용한 spleeter와 고급 프로그램인 demucs 사이에, 즉 spleeter보다는 뛰어나면서도 컴퓨터가 감당할 수 있는 도구가 있을지 더 자세하게 찾아보았다. 그 결과 spleeter 만큼 사용하기 편리하지는 않지만 코딩에 대해서 지식이 좀 있다면 더 좋은 퀄리티로 활용할 수 있는 open-unmix라는 도구가 있다는 것을 알게 되었고 spleeter와의 퀄리티를 비교해 보고 싶어서 간단한 스템 스플리터를 만들게 되었다.

텍스트, 스크린샷, 폰트, 친필이(가) 표시된 사진

자동 생성된 설명

결과적으로 open-unmix를 활용한 스템들의 품질이 원본적으로는 조금 더 좋다고 느껴지긴 했지만 이 부분은 사람들마다 차이가 있는 부분이기도 하고 너무 미세하다는 생각 또한 들었다. 또한 spleeter에 오디오 효과들을 더한 스템들은 충분히 활용이 가능한 정도였고 굳이 spleeter 로 만든 프로그램은 open-unmix로 다시 만들 필요성은 느끼지 못하여 프로젝트를 마치게 되었다.