

K - Digital Training

부산대학교 AI 양성과정



음원 추천 시스템

B TEAM

남예린, 박동찬, 박경근, 최재호, 홍혜성

멘토 : 윤병수



고용노동부



부산대학교
PUSAN NATIONAL UNIVERSITY

목차

- 01. 프로젝트 개요
- 02. 프로젝트 팀 구성 및 역할
- 03. 프로젝트 수행 절차 및 방법
- 04. 프로젝트 수행 결과
- 05. 자체 평가 의견

01 프로젝트 개요

▶ 프로젝트 개요



관련 주제 탐색 결과 기존에 활용할 수 있는 레퍼런스 및 데이터셋이 모두 해외 음원으로 작성되어 있기 때문에 **한국 음원을 활용**한 분석과정을 경험하는 것에 의의.

목차

01. 프로젝트 개요

02. 프로젝트 팀 구성 및 역할

03. 프로젝트 수행 절차 및 방법

04. 프로젝트 수행 결과

05. 자체 평가 의견

02 프로젝트 팀 구성 및 역할

▶ [프로젝트 팀 구성 및 담당 업무]

훈련생	역할	담당 업무
박동찬	팀장	<ul style="list-style-type: none">▶ 딥러닝 분석, 설계 및 통합 작업▶ Django 웹 구축, 서비스 설계, 발표 자료 준비
남예린	팀원	<ul style="list-style-type: none">▶ 딥 러닝 부분 모델링 작업, 정확도 측정▶ 시뮬레이션 담당
최재호	팀원	<ul style="list-style-type: none">▶ Raw 데이터 분석, CNN 모델링 수립 (장르 식별)▶ 유사도 분석
홍혜성	팀원	<ul style="list-style-type: none">▶ 딥 러닝 모델 수립(featuring 특성에 따른 학습)▶ 모델 평가
박경근	팀원	<ul style="list-style-type: none">▶ 모델링 작업에 사용되어질 Wav 음원 수집 및 편집

목차

01. 프로젝트 개요

02. 프로젝트 팀 구성 및 역할

03. 프로젝트 수행 절차 및 방법

04. 프로젝트 수행 결과

05. 자체 평가 의견

02 프로젝트 팀 구성 및 역할

▶ 사용한 도구 및 라이브러리



django



03 프로젝트 수행 절차 및 방법

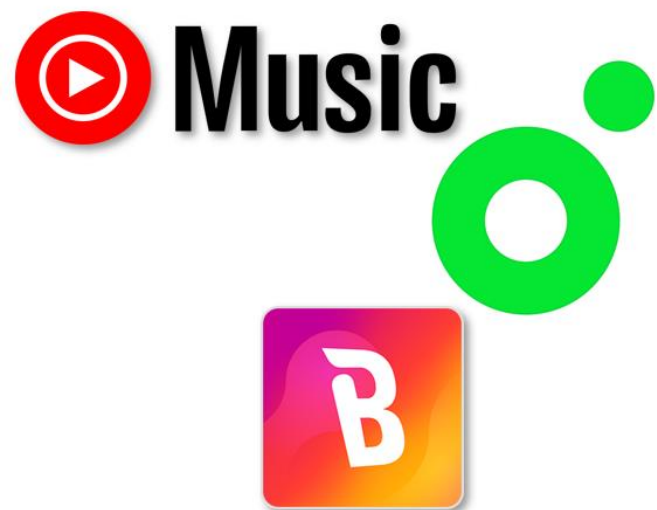
▶ 프로젝트 수행 절차

구분	기간	활동	비고
사전 기획	▶ 7.24(토)~9.24(금)	▶ 프로젝트 기획 및 주제 선정 ▶ 세부 계획 수립	▶ 아이디어 선정 ▶ 계획표 수립
데이터 수집	▶ 9.27(월)~10.01(금)	▶ 필요 데이터 및 수집 절차 정의 ▶ 외부 데이터 수집	▶ 데이터 수집 절차 검수
데이터 전처리	▶ 10.05(화)~10.12(화)	▶ 데이터 정제 및 정규화	
모델링	▶ 10.12(화)~10.21(목)	▶ 모형 구현	▶ 팀별 중간보고 실시
서비스 구축	▶ 10.29(금)~12.10(금)	▶ 웹 서비스 시스템 설계 ▶ 플랫폼 구현	▶ 최적화, 오류 수정
총 개발기간	▶ 7.24(토)~12.10(금)	-	-

03 프로젝트 수행 절차 및 방법

프로젝트 사전 기획

▶ 주제 선정



음원 추천 시스템



해양 쓰레기 탐지
시스템

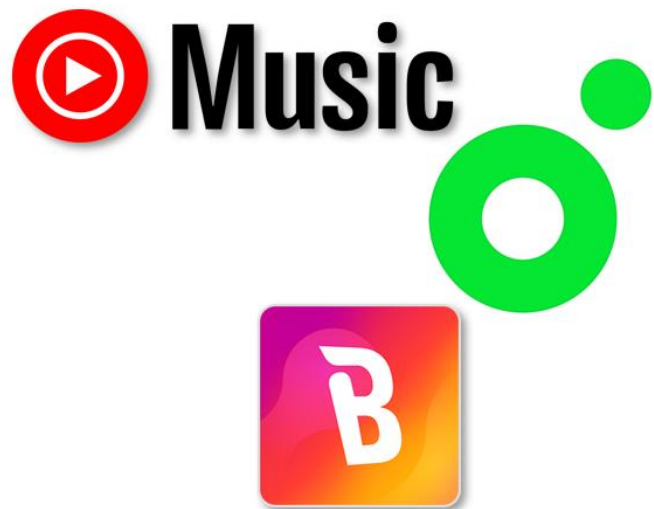


선박 입출항 스케줄링
시스템

03 프로젝트 수행 절차 및 방법

프로젝트 사전 기획

▶ 주제 선정 이유



음원 추천 시스템

1. 사업성 측면이 아닌 **학습**에 의의(훈련기간 동안 학습한 것을 최대한 활용할 수 있는 주제라고 판단)
2. 주제 선정 과정 중 관련 주제 탐색 결과 기존에 활용할 수 있는 레퍼런스 및 데이터셋이 모두 해외 음원으로 작성되어 있기 때문에 **한국 음원을 활용**한 분석과정을 경험하는 것에 의의.
3. 초기 팀 구성이 “데이터 분석”에 초점을 맞춘 것으로 시작되었기 때문에 **데이터 분석**과 관련된 보다 다양한 경험을 위해 음원 추천 시스템을 선택.

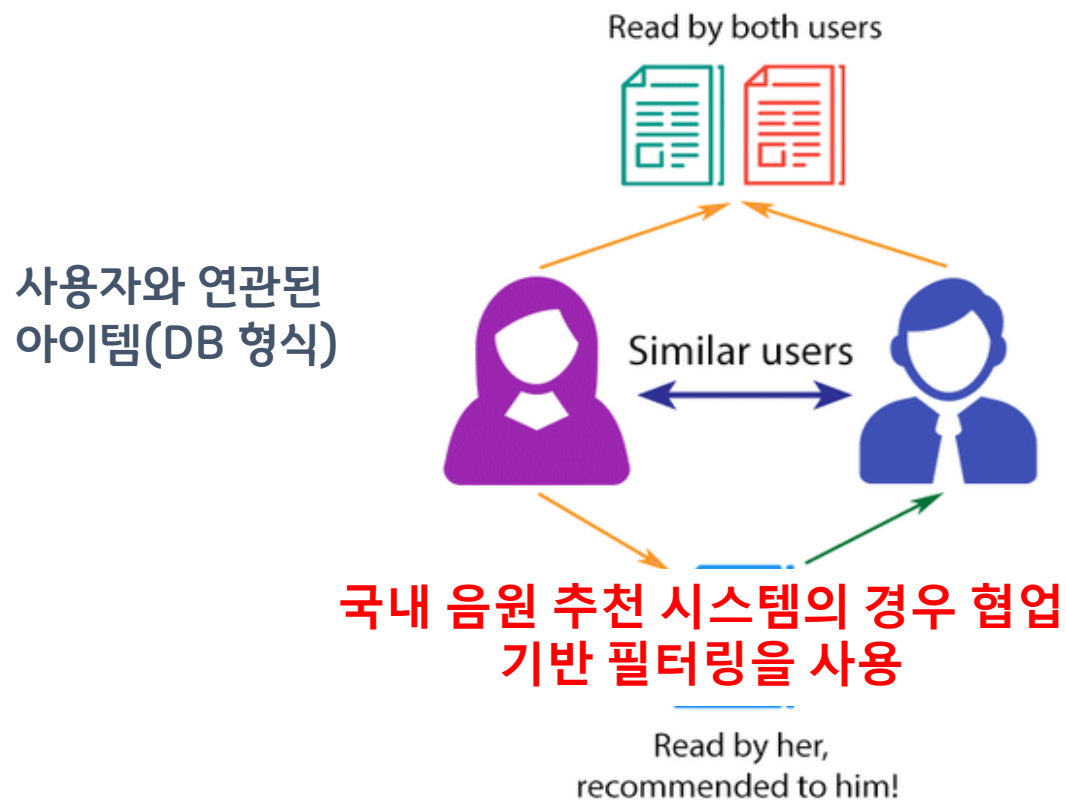
03 프로젝트 수행 절차 및 방법

프로젝트 사전 기획

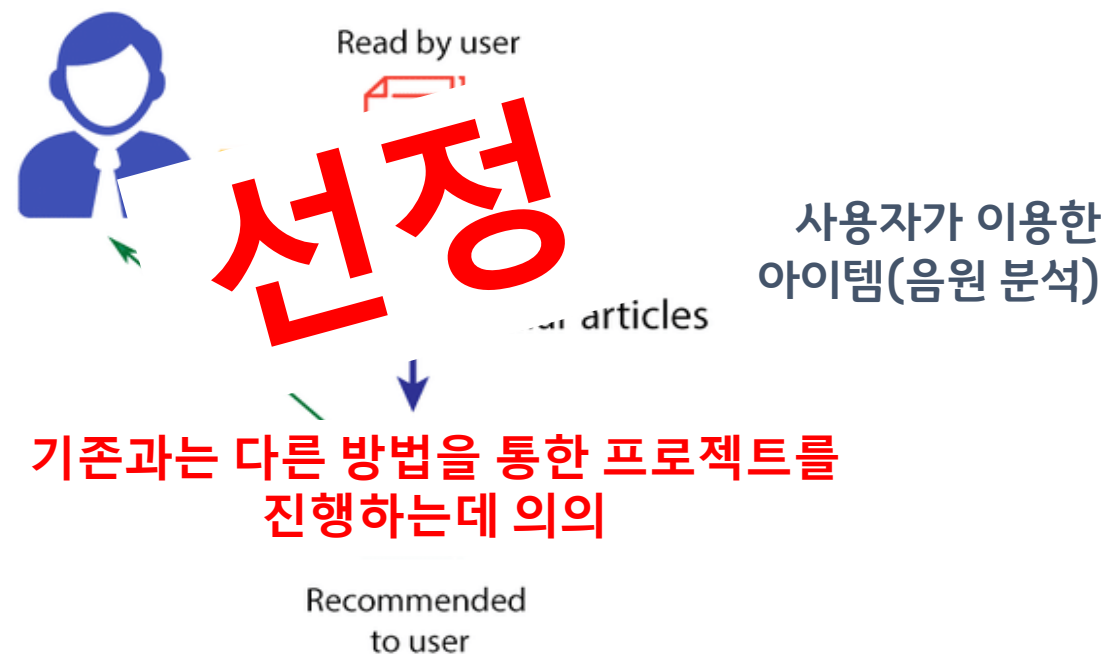
▶ 주제 선정 - 심층 탐색

음원 추천 시스템

협업 기반 필터링



컨텐츠 기반 필터링



03 프로젝트 수행 절차 및 방법

프로젝트 사전 기획

▶ 주제 선정 – 콘텐츠 기반 필터링 탐구



▶ 주제 선정 – 콘텐츠 기반 필터링 탐구

◆ 특성 추출 2가지의 방법

1. plot 차트로 표현되는 특성 값을 분리하여 numpy를 이용하여 2차원 Data로 추출 후 모델링에 사용
 - 장점 – 데이터 값의 무결성을 해치지 않고 그대로 사용가능
 - 단점 – 연산에 많은 시간 소요 (30초 기준 wav 파일당 80MB의 처리 요구)
2. plot 차트 출력 PNG 등의 이미지 파일로 추출하여 CNN 등을 활용하여 이미지 학습에 사용
 - 장점 – 연산에 있어 표면적인 데이터 처리량 감소
 - 단점 – 이미지 파일을 사용함에 있어 데이터 손실 가능성 존재

▶ 주제 선정 - 세부 계획 수립

◆ 데이터 분석 목표

- 음원 분석 결과 데이터 2가지 획득(numpy array, plt plot)
- 수집된 데이터를 labeling 하여 학습 모델 생성
(chroma_stft, rmse, spectral_centroid, spectral_bandwidth, rolloff, zero_crossing_rate, mfcc 복합 모델링)

◆ 목표 및 의의

- 음원 분석을 통하여 멜론, vibe, shazam 과 유사한 알고리즘 구현
- 데이터 수집부터 정제, 알고리즘까지 자체 구현 목표
- 훈련기간 내 배운 학습도구인 python을 활용하여 구축
- Train / Test 음원의 유사성 측정 결과(genre) 도출

03 프로젝트 수행 절차 및 방법

프로젝트 데이터 수집

▶ 데이터 수집



TeraBox

USER971082EC Premium 한국어

첫 페이지

전송

내 파일

사진

영상

서류

뮤직

Bit 토렌트

기타

휴지통

빠른 접속

자주 사용하는 폴더에 끌어오기

업로드

폴더 새로 만들기

내 웹디스크 > 본프로젝트 음악 > 본프로젝트 음악

내 웹디스크 파일 검색

파일명	수정시간	유형	크기
힙합	2021-11-23 11:21	폴더	-
포크	2021-11-23 11:16	폴더	-
트로트	2021-11-23 11:09	폴더	-
인디	2021-11-23 11:03	폴더	-
알앤비	2021-11-23 10:57	폴더	-
발라드	2021-11-23 10:49	폴더	-
록	2021-11-23 10:41	폴더	-
댄스	2021-11-23 10:38	폴더	-

8항

APP 다운로드

32G/1024G

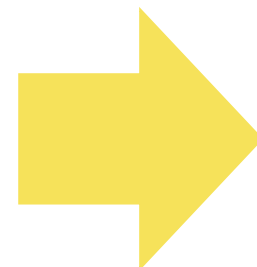
03 프로젝트 수행 절차 및 방법

프로젝트 데이터 전처리

▶ 데이터 전처리

The screenshot shows the TeraBox web interface. The user is logged in as USER971082EC with a Premium account. The interface displays a file directory structure. The left sidebar shows navigation options like '내 파일', '사진', '영상', '서류', '음악', '토렌트', '기타', '휴지통', and '빠른 접속'. The main area shows a directory tree: '내 웹디스크 > 본프로젝트 음악 > 본프로젝트 음악'. Below this, a table lists the contents of the '본프로젝트 음악' directory.

파일명	수정시간	유형	크기
합합	2021-11-23 11:21	폴더	-
포크	2021-11-23 11:16	폴더	-
트로트	2021-11-23 11:09	폴더	-
인디	2021-11-23 11:03	폴더	-
알앤비	2021-11-23 10:57	폴더	-
발라드	2021-11-23 10:49	폴더	-
록	2021-11-23 10:41	폴더	-
댄스	2021-11-23 10:38	폴더	-



◆ 수집 된 음원에서 특징 추출

chroma_stft, rmse, spectral_centroid, spectral_bandwidth, rolloff, zero_crossing_rate, mfcc

03 프로젝트 수행 절차 및 방법

프로젝트 데이터 전처리

▶ 데이터 전처리 단계

1. 아날로그 파형 분석
2. 음원데이터 추출 가능 특성 파악
3. Fourier 변환
4. Mfcc 생성
5. (번외학습)Mel-Spectrogram 생성 –

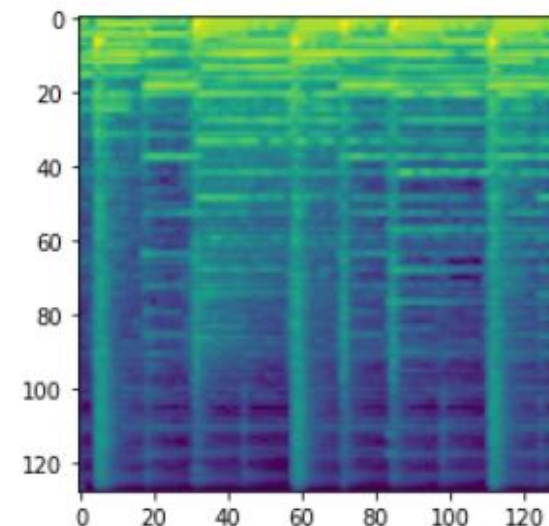
CNN 모델활용

푸리에 변환

- FFT(Fast Fourier Transform)
- STFT(Short-Time Fourier Transform)

스펙토그램

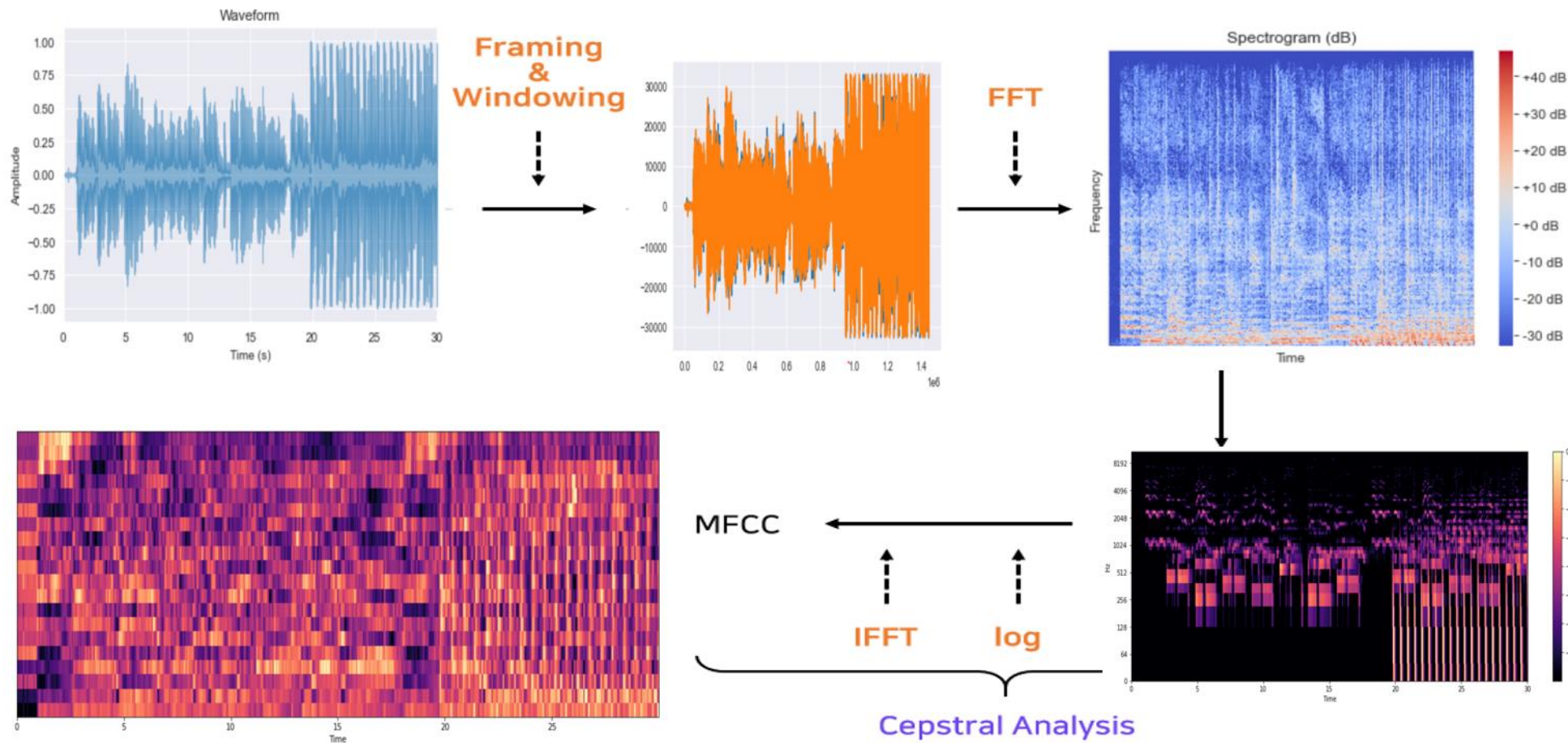
- X축: 시간
- Y축: 주파수
- Z축: 진폭



03 프로젝트 수행 절차 및 방법

프로젝트 데이터 수집

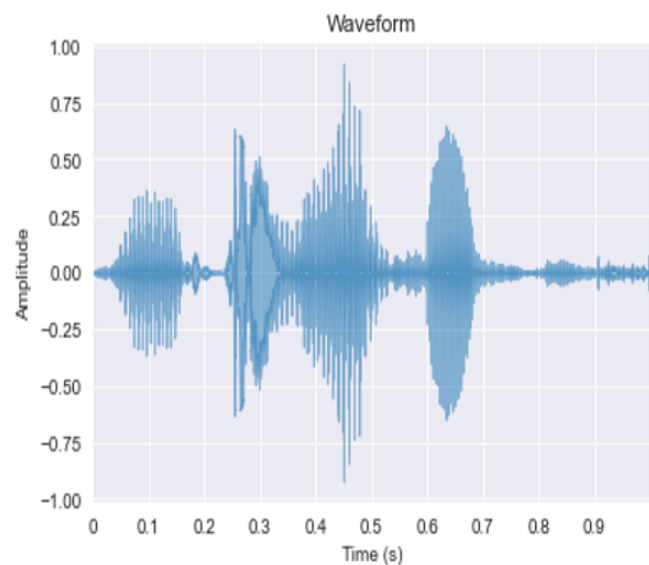
▶ 데이터 전처리 과정



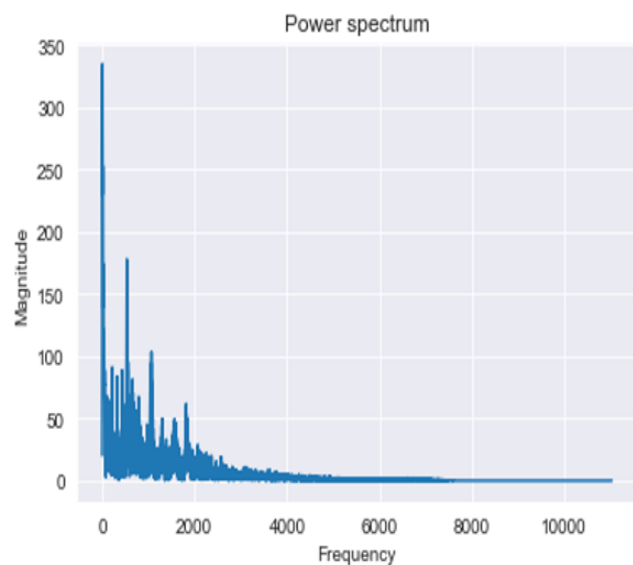
03 프로젝트 수행 절차 및 방법

프로젝트 데이터 전처리

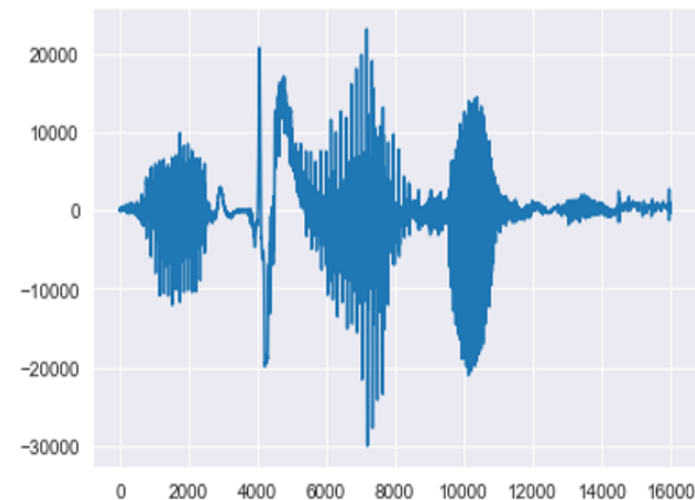
▶ 데이터 전처리 – 아날로그 파형 분석



Waveform



Power spectrum



Line2D

아날로그 파형에서 디지털 파형으로 변환 후 분석

03 프로젝트 수행 절차 및 방법

프로젝트 데이터 전처리

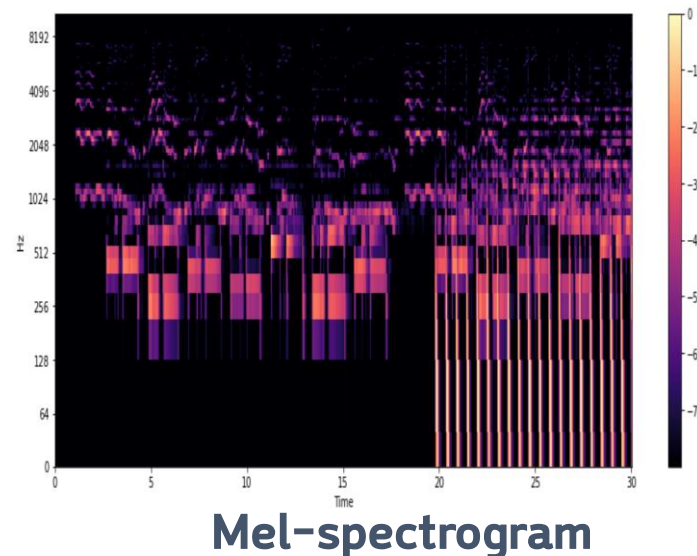
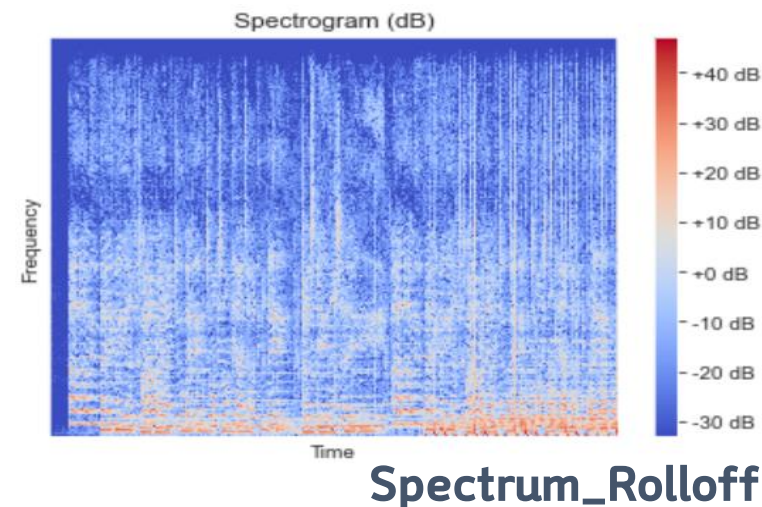
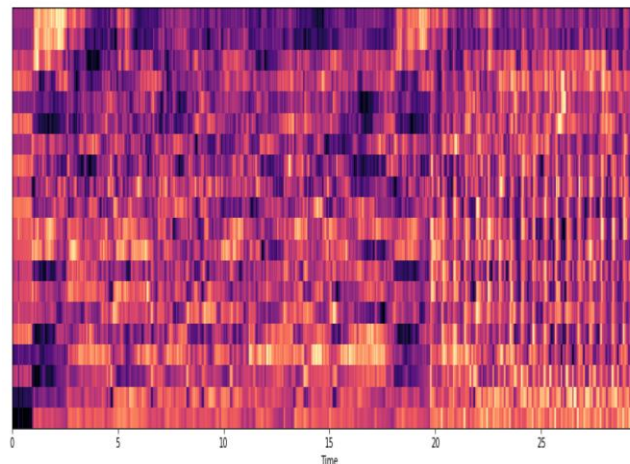
▶ 데이터 전처리 - 음원데이터 추출 가능 특성 파악

Data Set

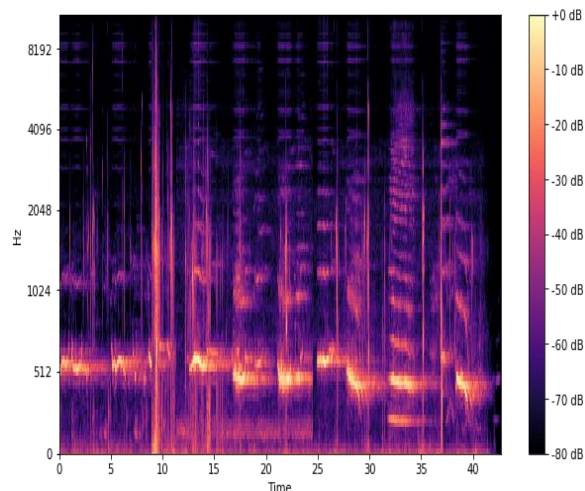
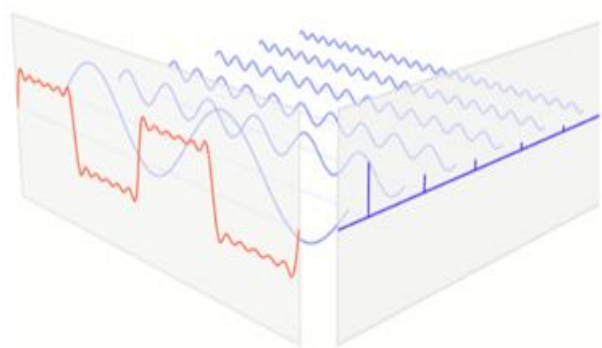
<input type="checkbox"/> 힙합	2021-11-23 11:21	물디	-
<input type="checkbox"/> 포크	2021-11-23 11:16	물디	-
<input type="checkbox"/> 트로트	2021-11-23 11:09	물디	-
<input type="checkbox"/> 인디	2021-11-23 11:03	물디	-
<input type="checkbox"/> 알앤비	2021-11-23 10:57	물디	-
<input type="checkbox"/> 발라드	2021-11-23 10:49	물디	-
<input type="checkbox"/> 록	2021-11-23 10:41	물디	-
<input type="checkbox"/> 댄스	2021-11-23 10:38	물디	-



mfcc



▶ 데이터 전처리 – 푸리에 변환



Audio Signal Processing (오디오 신호 처리)

◆ Fourier Transform (푸리에 변환)

- 개념

임의의 입력신호를 다양한 주파수를 갖는 주기함수들의 합으로 분해하여 표현하는 것

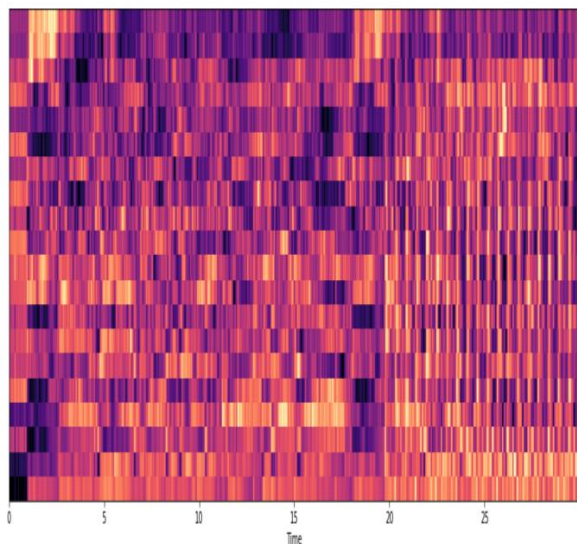
- 활용

1. 신호(ex.음성): time domain \rightarrow frequency domain
2. 영상처리: spatial domain \rightarrow frequency domain

03 프로젝트 수행 절차 및 방법

프로젝트 데이터 전처리

▶ 데이터 전처리 – Mfcc 생성



◆ MFCC

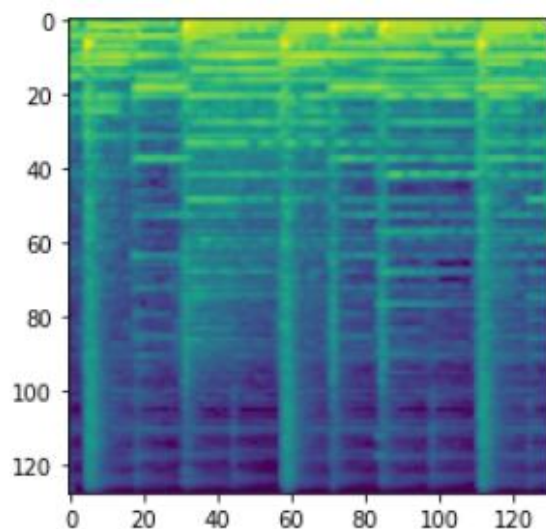
- 개념

바로 소리의 특징을 추출하는 기법. 입력된 소리 전체를 대상으로 하는 것이 아니라, 일정 구간으로 나누어 해당 구간에 대한 스펙트럼을 분석하여 특징을 추출하는 기법.

- 활용

구간별 스펙트럼을 분석하여 특징을 추출하기 때문에 곡의 유사도 관련한 모형에서 사용. 추천 시스템의 목적에 걸맞게 곡의 특성을 구간별로 추출하였기 때문에 높은 정확도를 보임.

▶ 데이터 전처리 – Mel-Spectrogram(변외학습)



◆ Mel-Spectrogram (멜 스펙트로그램)

- 개념

사람이 인식하는 주파수 중 예민하게 받아들이는 고주파를 Mel-scale 화 시켜 볼 수 있게 만든 지표.

- 활용

Mel-Spectrogram의 경우 주파수끼리 Correlate하기 때문에 데이터 셋이 한정적인 문제에서 mfcc 모형보다 더 좋은 성능을 보임. (CNN모형 한정)

03 프로젝트 수행 절차 및 방법

프로젝트 모델링

▶ 모델링

```
header = 'filename chroma_stft rmse spectral_centroid spectral_bandwidth rolloff zero_crossing_rate'
for i in range(1, 21):
    header += f' mfcc{i}'
header += ' label'
header = header.split()

file = open('data.csv', 'w', newline='', encoding='utf-8-sig')
with file:
    writer = csv.writer(file)
    writer.writerow(header)

genres = 'elec dance hip ball rnb'.split()
for g in genres:
    for filename in os.listdir(f'./tera/music/{g}'):
        filename1 = unicodedata.normalize('NFC', filename)
        songname = f'./tera/music/{g}/{filename1}'
        filename1 = filename1.replace(' ', '')
        y, sr = librosa.load(songname, mono=True, duration=30)
        chroma_stft = librosa.feature.chroma_stft(y=y, sr=sr)
        rmse = librosa.feature.rms(y=y)
        spec_cent = librosa.feature.spectral_centroid(y=y, sr=sr)
        spec_bw = librosa.feature.spectral_bandwidth(y=y, sr=sr)
        rolloff = librosa.feature.spectral_rolloff(y=y, sr=sr)
        zcr = librosa.feature.zero_crossing_rate(y)
        mfcc = librosa.feature.mfcc(y=y, sr=sr)
        tempo = librosa.beat.tempo(y=y, sr=sr)
        beat = librosa.beat.beat_track(y=y, sr=sr)

        to_append = f'{filename1} {np.mean(chroma_stft)} {np.mean(rmse)} {np.mean(spec_cent)} {np.mean(spec_bw)} {np.mean(rolloff)}'
        for e in mfcc:
            to_append += f' {np.mean(e)}'
        to_append += f' {g}'
        file = open('data.csv', 'a', newline='', encoding='utf-8-sig')
        with file:
            writer = csv.writer(file)
            writer.writerow(to_append.split())
```

음원 데이터 특성 추출

03 프로젝트 수행 절차 및 방법

프로젝트 모델링

▶ 모델링

	A	B	C	D	E	F	G	H	I	J
1	filename	chroma_stft	rmse	spectral_centroid	spectral_bandwidth	rolloff	zero_crossing_rate	mfcc1	mfcc2	mfcc3
2	10cm_이밤을빌려말해요.wav	0.27423048	0.086	1747.725457	2223.050028	3917.01	0.067089163	-236.793	120.1929	25.57938
3	2am_가까이있어서몰랐어.wav	0.234887198	0.079	2225.305773	2446.427761	4901.27	0.089869071	-256.683	81.97601	25.44086
4	2am_잘가라니.wav	0.236558393	0.111	1699.779874	1899.930217	3581.737	0.070154149	-229.941	110.4097	12.62597
5	CHEEZE(치즈)_이렇게좋아해본적이없어요.wav	0.254874587	0.084	1991.099524	2388.498538	4460.956	0.082143494	-215.511	92.17947	10.79704
6	IU(아이유)_ThroughtheNight(밤편지).wav	0.513428628	0.034	2470.875473	2461.29296	5311.768	0.111917691	-296.477	76.9678	1.227737

Standard Scaler



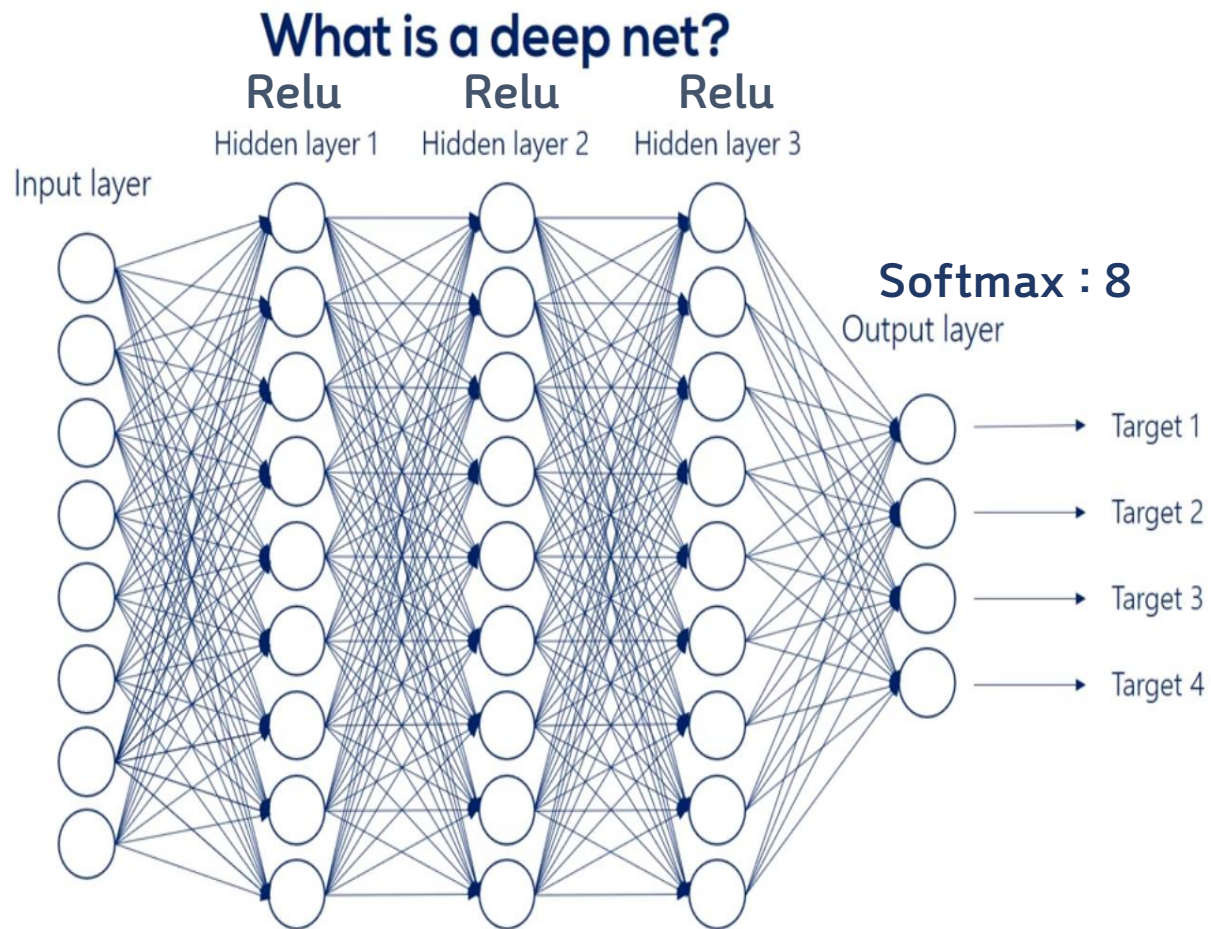
Normalize

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	0	0.542279	0.038184	0.598414	1.172161	0.751664	0.3621	0.481012	0.202745	0.390042	-1.38195	0.58405	0.032305	-0.43789	0.187465	-0.7401
3	1	1.816175	1.011811	0.838801	0.866773	0.922787	0.639094	0.046519	-0.91305	0.276852	-0.53473	-0.05329	0.369132	0.362432	1.92131	1.129336
4	2	1.282791	-2.81234	-0.98095	-0.58212	-0.83902	-0.94077	-3.10028	-0.39648	0.543562	0.170308	0.71877	0.91023	1.146078	0.316714	0.914531
5	3	-0.28259	0.756231	0.46404	0.094111	0.227379	1.063658	0.279124	-0.34743	-0.55358	1.183241	0.906099	1.642004	0.886347	-0.76378	0.058714

03 프로젝트 수행 절차 및 방법

프로젝트 모델링

▶ 모델링 – 하이퍼 파라미터 값 조정



◆ 하이퍼 파라미터 값 조정

Optimizer = Adam

Batch size = 128

**Loss function =
'sparse_categorical_crossentropy'**

Metrics = accuracy

Epochs = 100

03 프로젝트 수행 절차 및 방법

프로젝트 모델링

▶ 모델링 - 코드 예제

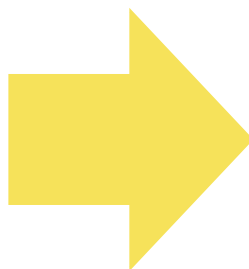
```
from keras import models
from keras import layers
```

```
model = models.Sequential()
model.add(layers.Dense(256, activation='relu', input_shape=(X_train.shape[1],)))
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(32, activation='relu'))
model.add(layers.Dense(16, activation='relu'))
model.add(layers.Dense(8, activation='softmax'))

model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

history = model.fit(X_train, y_train, epochs=50, batch_size=256)

test_loss, test_acc = model.evaluate(X_test, y_test)
print('test_acc: ', test_acc)
```

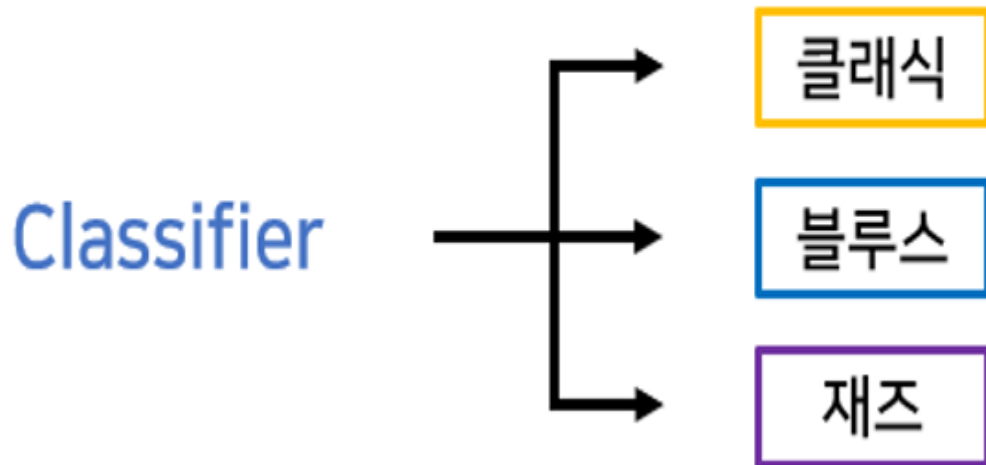


```
Epoch 47/50
2/2 [=====] - 0s 38ms/step - loss: 0.5374 - accuracy: 0.8503
00e+00
Epoch 48/50
2/2 [=====] - 0s 40ms/step - loss: 0.5621 - accuracy: 0.8153
00e+00
Epoch 49/50
2/2 [=====] - 0s 40ms/step - loss: 0.4861 - accuracy: 0.8583
00e+00
Epoch 50/50
2/2 [=====] - 0s 39ms/step - loss: 0.4540 - accuracy: 0.8726
```

03 프로젝트 수행 절차 및 방법

프로젝트 모델링

▶ 모델링 - CNN 활용 목적



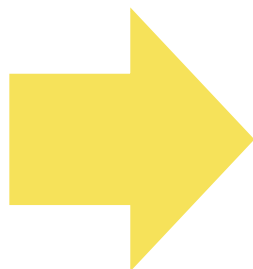
- 목적
기존 특성을 이용한 분류에서
보다 정확한 장르의 분류를 위해
- 문제점
CNN을 통해 학습된 데이터를
기존에 사용하던 모델의 특징 값
중 하나로 추가할 수 있는가?

03 프로젝트 수행 절차 및 방법

프로젝트 모델링

▶ 모델링 - CNN 활용 목적

```
def GenreModel(input_shape = (288,432,4),classes=5):  
    X_input = Input(input_shape)  
  
    X = Conv2D(8,kernel_size=(3,3),strides=(1,1))(X_input)  
    X = BatchNormalization(axis=3)(X)  
    X = Activation('relu')(X)  
    X = MaxPooling2D((2,2))(X)  
  
    X = Conv2D(16,kernel_size=(3,3),strides = (1,1))(X)  
    X = BatchNormalization(axis=3)(X)  
    X = Activation('relu')(X)  
    X = MaxPooling2D((2,2))(X)  
  
    X = Conv2D(32,kernel_size=(3,3),strides = (1,1))(X)  
    X = BatchNormalization(axis=3)(X)  
    X = Activation('relu')(X)  
    X = MaxPooling2D((2,2))(X)  
  
    X = Conv2D(64,kernel_size=(3,3),strides=(1,1))(X)  
    X = BatchNormalization(axis=-1)(X)  
    X = Activation('relu')(X)  
    X = MaxPooling2D((2,2))(X)  
  
    X = Conv2D(128,kernel_size=(3,3),strides=(1,1))(X)  
    X = BatchNormalization(axis=-1)(X)  
    X = Activation('relu')(X)  
    X = MaxPooling2D((2,2))(X)  
  
    X = Flatten()(X)
```



Epoch 66/70

4/4 [=====] - 25s 6s/step - loss: 0.0587 - accuracy: 0.9925 -
_accuracy: 0.2000 - val_get_f1: 0.2000

Epoch 67/70

4/4 [=====] - 24s 8s/step - loss: 0.0516 - accuracy: 0.9925 -
_accuracy: 0.2000 - val_get_f1: 0.2000

Epoch 68/70

4/4 [=====] - 25s 6s/step - loss: 0.0439 - accuracy: 0.9950 -
_accuracy: 0.2100 - val_get_f1: 0.2010

Epoch 69/70

4/4 [=====] - 26s 6s/step - loss: 0.0425 - accuracy: 0.9875 -
_accuracy: 0.2000 - val_get_f1: 0.2000

Epoch 70/70

4/4 [=====] - 26s 6s/step - loss: 0.0352 - accuracy: 0.9925 -

▶ 모델링 - CNN 활용 목적

◆ 문제점

CNN을 통해 학습된 데이터를 기존에 사용하던 모델의 특징 값 중 하나로 추가할 수 있는가?

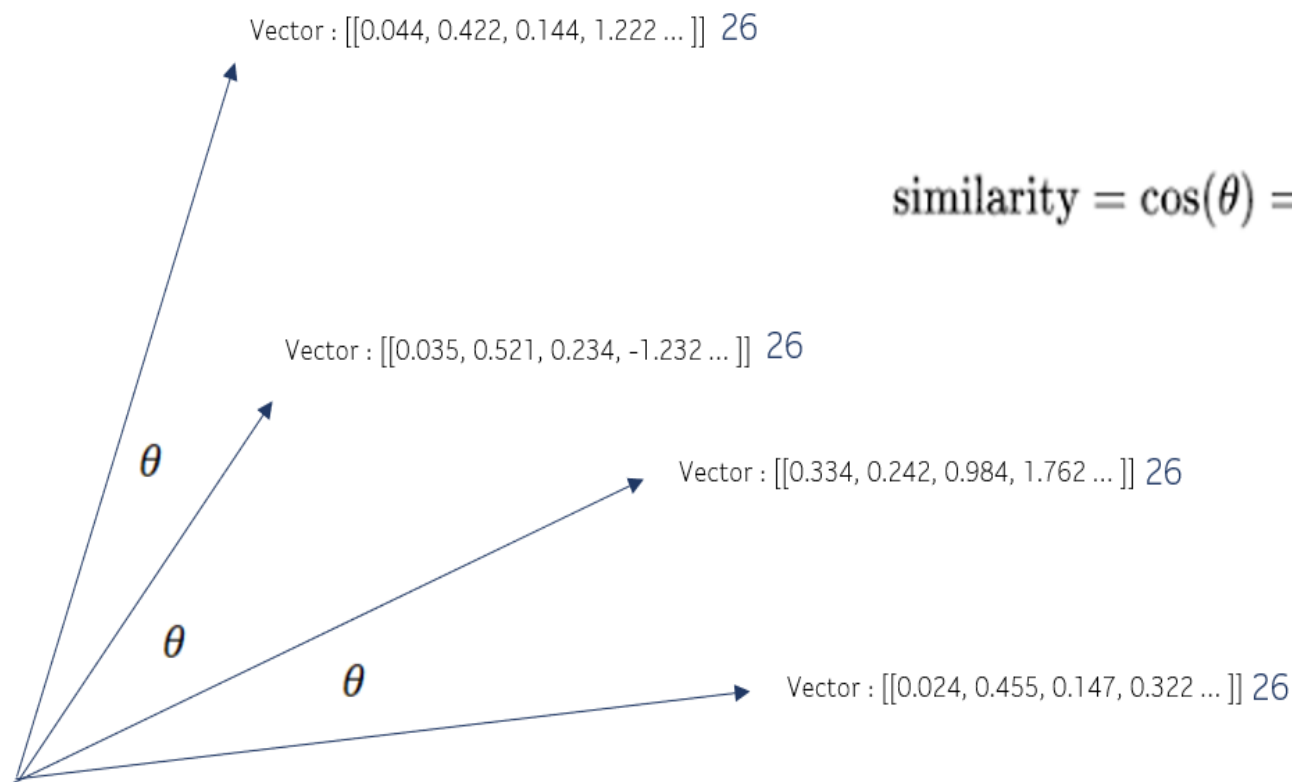
- 결과

400곡을 3초단위로 쪼갠 뒤 Mel-Spectrogram의 사진을 추출하여 CNN모델을 적용 후 학습 한 결과 8:2로 설정한 train set 에서 97%의 정확도를 보였으나, 하나의 특징 값으로 모델에 적용하기 위해서는 추후 연구가 더 필요함.

03 프로젝트 수행 절차 및 방법

프로젝트 모델링

▶ 모델링 - 유사도 분석



$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

각각의 음악 벡터 값들 사이의 코사인 유사도 확인. (26개 특징의 벡터 값)

03 프로젝트 수행 절차 및 방법

프로젝트 모델링

▶ 모델링 – 유사도 분석 및 추천

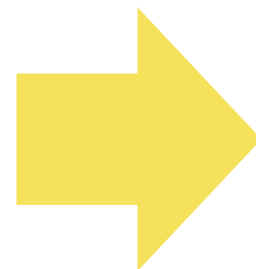
```
from sklearn.metrics.pairwise import cosine_similarity

similarity = cosine_similarity(df_30)

sim_df = pd.DataFrame(similarity, index=labels.index, columns=labels.index)

sim_df
```

filename	10cm_이 밤을빌려 말해 요.wav	2am_가 까이있어 서를랏 어.wav	2am_잘 가라 니.wav	CHEEZE(치 즈)_이렇게 좋아해본적 이없어 요.wav	IU(아이 유)_ThroughtheNight(밤 편지).wav	IU(아이 유)_비밀 의화 원.wav	IU(아이 유)_아이 와나의바 다.wav	M.O.M(MSG 워너비)_바 라만본 다.wav	M
filename									
10cm_이밤을빌려말해 요.wav	1.000000	0.255783	-0.032237	0.314266	-0.333137	-0.162645	0.333738	0.041394	
2am_가까이있어서를랏 어.wav	0.255783	1.000000	0.366534	-0.181952	0.018802	-0.426115	0.001105	0.109390	
2am_잘가라니.wav	-0.032237	0.366534	1.000000	-0.044117	-0.058063	0.202881	0.044799	0.245415	
CHEEZE(치즈)_이렇게 좋아해본적이없어요.wav	0.314266	-0.181952	-0.044117	1.000000	-0.111468	0.059155	0.505204	-0.306271	
IU(아이 유)_ThroughtheNight(밤 편지).wav	-0.333137	0.018802	-0.058063	-0.111468	1.000000	0.098344	-0.056278	-0.497783	
...
홍진영_사랑의배터 리.wav	-0.300388	0.033720	-0.295411	0.111850	0.344388	-0.171264	0.074542	-0.490641	
홍진영_사랑이좋아.wav	-0.360893	-0.644299	-0.357088	0.024760	0.068972	0.260524	0.048112	-0.177695	
홍진영_산다는건.wav	0.185473	0.027868	0.178099	-0.153253	-0.100436	0.156004	0.043754	0.357161	
홍진영_오늘밤에.wav	-0.305284	0.261596	-0.283289	-0.064208	0.346210	-0.320227	-0.041086	-0.335269	
홍진영_잘가라.wav	0.230113	-0.032890	-0.028268	0.612346	-0.404304	-0.093194	0.180742	-0.086824	



추출된 특성을 통해
유사도 분석을 진행

▶ 서비스 구축

django

◆ django

Python 한 가지만 사용하였기 때문에 이를 구현 할 웹 프레임 워크를 장고로 선택해 구현함

- 활용

모바일과 관련된 학습은 훈련기간 동안 배우지 않았기 때문에 웹 서비스로 구현 함.

웹 구현 시, 2개의 페이지로 구현하여, 입력 페이지와 출력 페이지로 구현 함.

03 프로젝트 수행 결과

▶ 서비스 로직

Test 음원 분해

```
def upload_file(req):
    if req.method == 'POST':
        if req.FILES['my_file'] is not None:
            filename = req.FILES['my_file'].name
            upload_file_loc = "media/upload/" + filename
            print("파일 네임 : ", filename)
            print("로케이션 : ", upload_file_loc)
            # if not filename.endswith('.wav'):
            #     return redirect(req, 'main.html', {'msg': '파일은 wav 형식만 가능합니다.'})
            # # if file name is empty return error:
            # if req.FILES.get('filepath') is None:
            #     return redirect(req, 'main.html', {'msg': '파일을 선택하세요.'})

            with open(os.path.abspath('media/upload/' + req.FILES['my_file'].name), 'wb+') as dest:
                for chunk in req.FILES['my_file'].chunks():
                    dest.write(chunk)

            oldmodelreturn = oldmodel[filename]

            print(oldmodelreturn)

            model = tf.keras.models.load_model('static/model.h5')
            model.summary()
            predictions = model.predict(oldmodelreturn)

            print("shape = ", predictions[0].shape)
            print("np.sum = ", np.sum(predictions[0]))
            print("argmax = ", np.argmax(predictions[0]))

            print('type = ', type(predictions[0]))
            pred_ped = predictions[0].tolist()
            pred_ped_1 = list(np.round(pred_ped, 2))
            # print(pred_ped_1)
            b = []
```

모델 통과

```
def oldmodel(myfilename):
    header = 'filename chroma_stft rmse spectral_centroid spectral_ba
    # file2 = pd.DataFrame()

    for i in range(1, 21):
        header += f' mfcc{i}'

    header += ' label'
    header = header.split()
    genres = 'Ballad Dance folk hiphop Indi RnB Rock troat'.split()

    filename1 = unicodedata.normalize('NFC', myfilename)
    songname = f'media/upload/{filename1}'

    filename1 = filename1.replace(' ', '')
    y, sr = librosa.load(songname, mono=True, duration=30)
    chroma_stft = librosa.feature.chroma_stft(y=y, sr=sr)
    rmse = librosa.feature.rms(y=y)
    spec_cent = librosa.feature.spectral_centroid(y=y, sr=sr)
    spec_bw = librosa.feature.spectral_bandwidth(y=y, sr=sr)
    rolloff = librosa.feature.spectral_rolloff(y=y, sr=sr)
    zcr = librosa.feature.zero_crossing_rate(y)
    mfcc = librosa.feature.mfcc(y=y, sr=sr)

    to_append = f'{filename1} {np.mean(chroma_stft)} {np.mean(rmse)}
    for e in mfcc:
        to_append += f' {np.mean(e)}'

    list_of_append = to_append.split(' ')
    # str to float
    list_of_append = list(map(float, list_of_append[1:27]))
```

목차

01. 프로젝트 개요

02. 프로젝트 팀 구성 및 역할

03. 프로젝트 수행 절차 및 방법

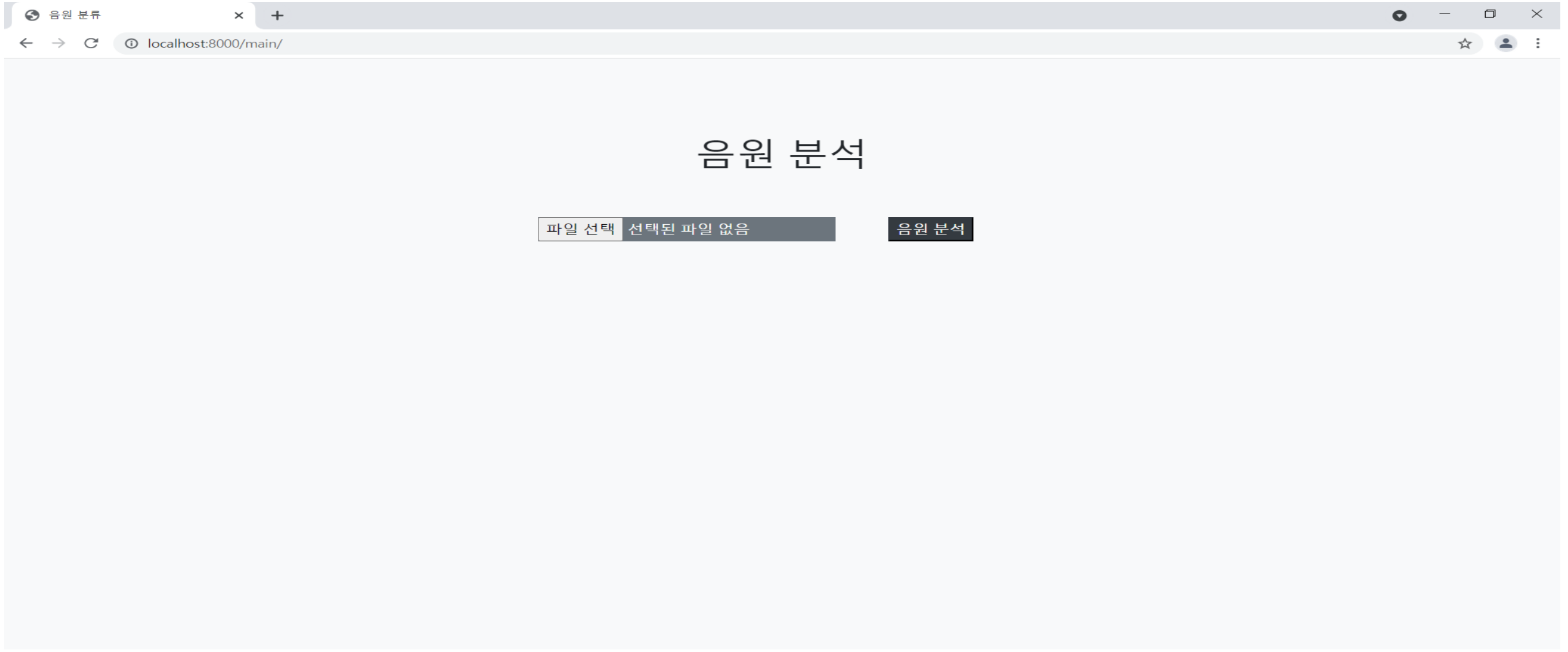
04. 프로젝트 수행 결과

05. 자체 평가 의견

04 프로젝트 수행 결과



음원 입력 화면



03 프로젝트 수행 절차 및 방법

▶ 장르 판독 및 추천 결과 (장르)

1. 음원 분류 결과

Ballad의 확률 : 81.0 %

Dance의 확률 : 0.0 %

Folk의 확률 : 19.0 %

Hiphop의 확률 : 0.0 %

Indi의 확률 : 0.0 %

RnB의 확률 : 0.0 %

Rock의 확률 : 0.0 %

troat의 확률 : 0.0 %

가장 유사한 장르 **Ballad : 81.0%**

[다시 확인해보기](#)

2. 음원 추천

이하 유사한 음원

업로드한 음원 : 2am_잘 가라니.wav

▶ 0:00 / 3:32 🔊 ⋮

1. 정동하_추억은 만남보다 이별에 남아.wav
▶ 0:00 / 4:27 🔊 ⋮
2. 김나영_The Youngest Day.wav
▶ 0:00 / 4:21 🔊 ⋮
3. 윤종신_Like it(종니).wav
▶ 0:00 / 5:28 🔊 ⋮
4. 성시경_거리에서.wav
▶ 0:00 / 4:37 🔊 ⋮
5. 경서예지 X 전건호_다정히 내 이름을 부르면.wav
▶ 0:00 / 4:23 🔊 ⋮

04 프로젝트 수행 결과

시연 동영상

※ 별도 첨부 가능

음원 분류

www.BANDICAM.com

localhost:8000/main/

음원 분석

파일 선택

선택된 파일 없음

1920x1020 - (0, 0), (1920, 1020)

00:00:00
0 bytes / 12.2GB

녹화 시작 / 중지

홈

일반

비디오

이미지

정보

화면 녹화

시작/정지 단축키
일시 정지 단축키
마우스 커서 표시
마우스 클릭 효과 추가
웹캠 오버레이 추가

Shift+F12
Ctrl+Shift+F12
설정

녹화 포맷 - MP4

비디오
H264 - NVIDIA® NVENC (VBR) (자동)
Full Size, 60.00fps, 100q

오디오
AAC - Advanced Audio Coding
48.0KHz, stereo, 192kbps

빠른 설정
설정

BANDICUT

컴퓨터+마이크 소리를 녹음하는 방법

03 프로젝트 수행 결과

프로젝트 수행 결과 - 음원 추천

▶ 모델링 - 추천 모델 결과(cos 유사도 활용)

```
labels = train_one[['노동요']]
# 코사인 시밀러리티 돌림
similarity = cosine_similarity(train_one)
# 어차피 변수 이름은 상관없음 각 행에 있는 노래들이 가지고 있는 벡터가 바라보는 방향이 중요하기 때문.
sim_df = pd.DataFrame(similarity, index=labels.index, columns=labels.index)
sim_df

# In[28]:

# 제일 마지막에 해야할 일
def find_similar_songs(name, n=5):
    series = sim_df[name].sort_values(ascending=False)
    series = series.drop(name)
    return series.head(n).to_frame()
print(find_similar_songs('카라멜 마끼아또'))
```

어떤 그림

열 추가

Teen Age Riot (Album Ver.)	0.755929
가까이 하기엔 너무 먼 당신	0.707107
Aphrodite	0.707107
Nothing In Common	0.666667
Hate Me!	0.666667

목차

01. 프로젝트 개요

02. 프로젝트 팀 구성 및 역할

03. 프로젝트 수행 절차 및 방법

04. 프로젝트 수행 결과

05. 자체 평가 의견

05 자체 평가 의견

▶ 의견

박동찬 :

1. Feature 추출 기준이 되는 경우의 수를 많이 테스트 해보지 못함
2. Train 과정에 있어 전처리 아쉬움 (Model 추출 전 Scaling 값의 효율적인 사용)
3. 데이터 정제 부분에 있어 하나도 버리는 데이터가 없다는 걸 간과하여 scaling 후처리 과정에서 또다시 호출하여 중복 처리를 하게 됨.

➔ (File I/O 횟수 증가 = 시공간 복잡도 상승 요인)

05 자체 평가 의견

▶ 의견

홍혜성 :

1. 알고리즘 효율을 끊임없이 고민해야 겠다고 생각 함.(함수와 클래스를 활용한 코드자체의 효율, 프로그램 계산 시간단축 효율)
2. 인공지능 기술 구현을 위해서는 도메인 지식이 효과적인 모델링에 상당한 도움을 줄 수 있을 것이라 생각했음.
3. 실제 결과물을 만들어내는 전체 과정을 통해 이론만으로 맞이하 수 없는 다양한 문제에 봉착한 것이 경험이 되었음.
4. 팀으로 움직였기 때문에 상호간에 부족분을 채워주며 작업할 수 있어서 수월했음.

05 자체 평가 의견

▶ 의견

남예린 :

1. CNN 모델 학습에서 교차검증도 상승을 위한 개선이 필요해 보인다.
2. 처음 목표였던 사용자 듣기이력 기반 음악추천 시스템을 만들기 위해서는 추후 연구가 더 필요하다고 생각된다.
3. 오디오 데이터 처리를 위한 파이썬 라이브러리가 존재해 효율적인 작업을 할 수 있었다.
4. 짧은 기간이었지만 팀원간 적절한 역할 분배와 협력으로 좋은 결과물을 낼 수 있었다.

05 자체 평가 의견

▶ 의견

최재호 :

1. 데이터를 수집할 때 가이드라인을 좀 더 명확히 하여 수집했으면 정확도 향상에 도움이 되었을 것이라 생각함.
2. 번외 학습을 진행하며 CNN알고리즘에 어떤 것이 있고, 어떤 문제에 이런 알고리즘을 적용해야 하는지 배우지를 못해 아쉬웠음.
3. 음원에 대한 도메인 지식이 있었다면, 전처리 과정에서 보다 다양한 분석이 가능했을 것.

The background is a solid blue gradient. Overlaid on this are numerous thin, white, curved lines that flow from the left side towards the right, creating a sense of movement and depth. These lines are more densely packed in some areas, forming soft, undulating shapes that resemble waves or smoke.

감사합니다