<채보 프로그램 구현>

201710854 김병찬

201710877 이동원

<1차> : 제안 발표 (주제 선정, 방법, 예상 결과)

* 주제 : 피아노 연주 녹음 파일을 읽고, 악보 노트를 찍는 프로그램 작성
* 예상 구조 : 음 – 반음 단위로 주파수 구간 분할 후 그래프 형식 timeline 생성

\*피아노 88 건반 음계 구현(7bit필요), 평균율

박자 – \*\*BPM 기반으로 시간 축 분할

미리 입력 받은 BPM값을 기준으로 설정

노트의 시간 기준으로 추정 및 설정

File input – 구간별 양자화(Quantize)

* 예상 방법 : 음 연주 -> Noise Cancelling -> 박자, bpm 측정, 악보 base 형성 -> 음정 note 변환 -> note 작성
* 예상 결과 : 텍스트이(가) 표시된 사진

  자동 생성된 설명

텍스트, 모니터, 실내, 디스플레이이(가) 표시된 사진

자동 생성된 설명

(출처 : https://www.youtube.com/watch?v=S\_xfcvMoCAA&t=290s 4:43)

(추가 설명)

\* 피아노 88건반을 기준, 88음 표현하기 위해서 (64 < 88 < 128) 7bit 이상 구간 개수 필요,

**평균율** 표준 음계(가온 도는 C4라고 한다)에서 **A4**(4옥타브 라)의 주파수 = **440.0Hz**

C4(가온 도/ 4옥타브 도) = 261.63Hz

평균율 : 1옥타브 차 = 1 : 2 , 한 옥타브 간격 내의 12음을 동일 간격 주파수 비로 분할

평균율에서 한 음은 반음 단위로 이전 음의 배, 약 1.05946배

A4의 주파수 ƒA4 = 440Hz

\*\* BPM : 1분간 Bit 수 -> Bit = 4분음표

즉, 1분간 4분음표를 넣은 개수 => 1 / (4분음표notetime)

<2차> : 알고리즘 및 응용 가능 분야 조사

* 알고리즘

1. 피아노 연주 녹음 파일을 프로그램에 입력

2. 반음 단위로 주파수 구간 분할텍스트이(가) 표시된 사진

자동 생성된 설명

3. 주파수 f, 시간 t축으로 설정

4. 주파수 f를 음계 구간에 대해 양자화 : 주파수 f값을 구하는 경우에서

- 피아노 88건반을 기준, 88음 표현하기 위해 (64 < 88 < 128) 7bits 이상 구간 개수 필요

- 평균율 표준 음계에서 A4(4옥타브 라)의 주파수 = 440Hz

- C4 = 261.63Hz

- 평균율 : 1옥타브 차 = 1 : 2, 한 옥타브 간격 내의 12음을 동일 간격 주파수 비로 분할

- 평균율에서 한 음은 반음 단위로 이전 음의 약 1.05946배

5. 시간 t는 BPM 값에 따라 구간 설정

- BPM 값을 계산하는 알고리즘을 설정한 후 적용

6. 반음 단위 양자화 구간의 값 Q = g(f)

- 양자화된 주파수 Q와 시간 t를 축으로 구간 분할 함수 생성

가로축 : t

세로축 : Q

- Q = 400Gz (A4음), t = 0 영점

- 한 Q값에서 이어지는 구간을 한 음 기준, note로 정의할 수 있다.

7. Ableton Live의 Draw Mode 와 유사 형태로 반음 단위 블록 형태로 구현 가능

텍스트, 모니터, 전자기기, 디스플레이이(가) 표시된 사진

자동 생성된 설명

* 결과물 활용 가능한 분야

1. 음이탈, 탈박 검출 알고리즘

2. 해당 data를 MIDI data로 변환 후 porting하는 알고리즘

3. 연주를 real time으로 들으며 작성

<3차> : 프로그래밍 구현

* 구현 과정

1. 오디오 파일을 읽은 뒤 Sampling rate, Audio length 확인

import numpy as np

import librosa

import matplotlib.pyplot as plt

import librosa.display

import IPython.display as ipd ## 음악 듣기용

## 메인 테스트 대상은 sample0006.wav / 동요 비행기를 120BRM 메트로놈에 맞춰 Ableton Live VST, MIDI Keyboard로 연주한 녹음 파일. 활용 음원 (추정) E\_Piano Basic.adg(Ableton Live Lite 기본 포함)

## 입력 시스템 활용 가능성.

filepath = "C:/Users/82108/Desktop/python/sound"

filename = "sample0006.wav"

## data : wav file 읽어온 data값. 2-5.Spectogram을 통해 시간, 주파수, 음압에 대한 정보가 있음을 알 수 있다.

## samplingrate : librosa.load(file, sr)에서 sr값은 파이르이 샘플링 주파수다. 이 값을 받아 samplingrate 변수에 저장.

data , samplingrate = librosa.load(filepath + filename, sr=48000)

## 2. 오디오 파일 이해하기

## 2-1. 기본값 확인

'''

print(data)

print(len(data)) ## len은 array형식인 data의 정보 개수

print('Sampling rate (Hz): %d' %samplingrate)

print('Audio length (seconds): %.2f' % (len(data) / samplingrate))

'''

2. Waveplot : 시간 영역 표현

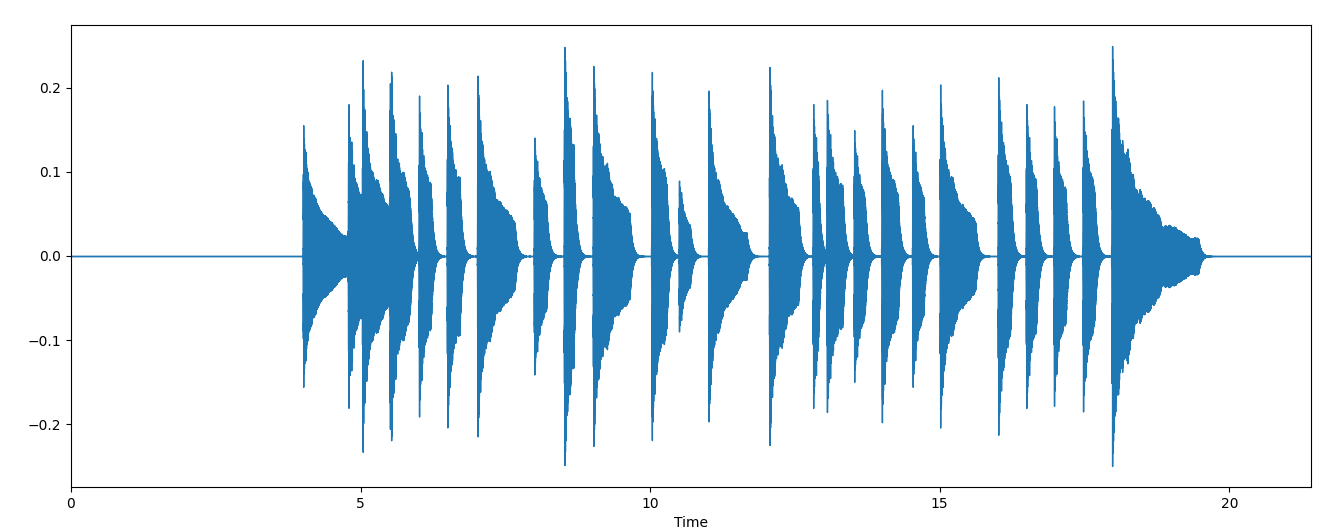
## 2-3. 2D 음파 그래프

'''

plt.figure(figsize =(16,6))

librosa.display.waveplot(y=data,sr=samplingrate)

plt.show()

3. 고속 푸리에 변환

## 2-4. Fourier Transform(푸리에 변환)

FTdata = np.abs(librosa.stft(data, n\_fft=2048, hop\_length=512)) #n\_fft : window size / 이 때, 음성의 길이를 얼마만큼으로 자를 것인가? 를 window라고 부른다.

'''

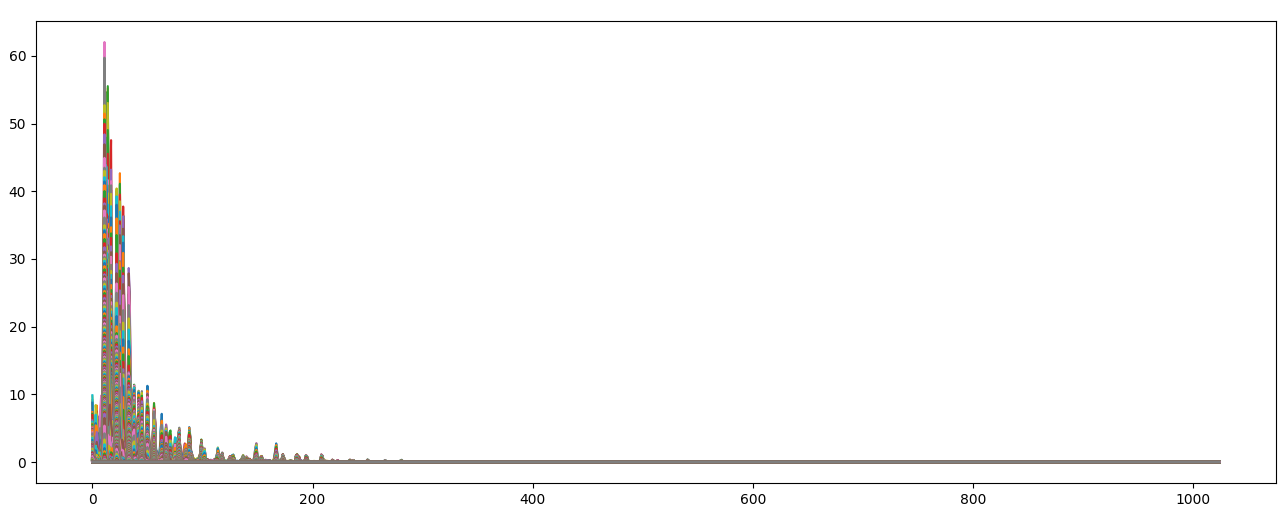
print(FTdata.shape)

plt.figure(figsize=(16,6))

plt.plot(FTdata)

plt.show()

'''



4. Spectogram 확인

## 2-5.Spectogram

FTDBdata = librosa.amplitude\_to\_db(FTdata, ref=np.max) #amplitude(진폭) -> DB(데시벨)로 바꿔라

'''

plt.figure(figsize=(16,6))

librosa.display.specshow(FTDBdata,sr=samplingrate, hop\_length=512, x\_axis='time', y\_axis='log')

plt.colorbar()

plt.show()

텍스트이(가) 표시된 사진

자동 생성된 설명

5. Mel Spectogram 확인

## 2-6. Mel Spectogram

Meldata = librosa.feature.melspectrogram(data, sr=samplingrate)

Mel\_DB = librosa.amplitude\_to\_db(Meldata, ref=np.max)

'''

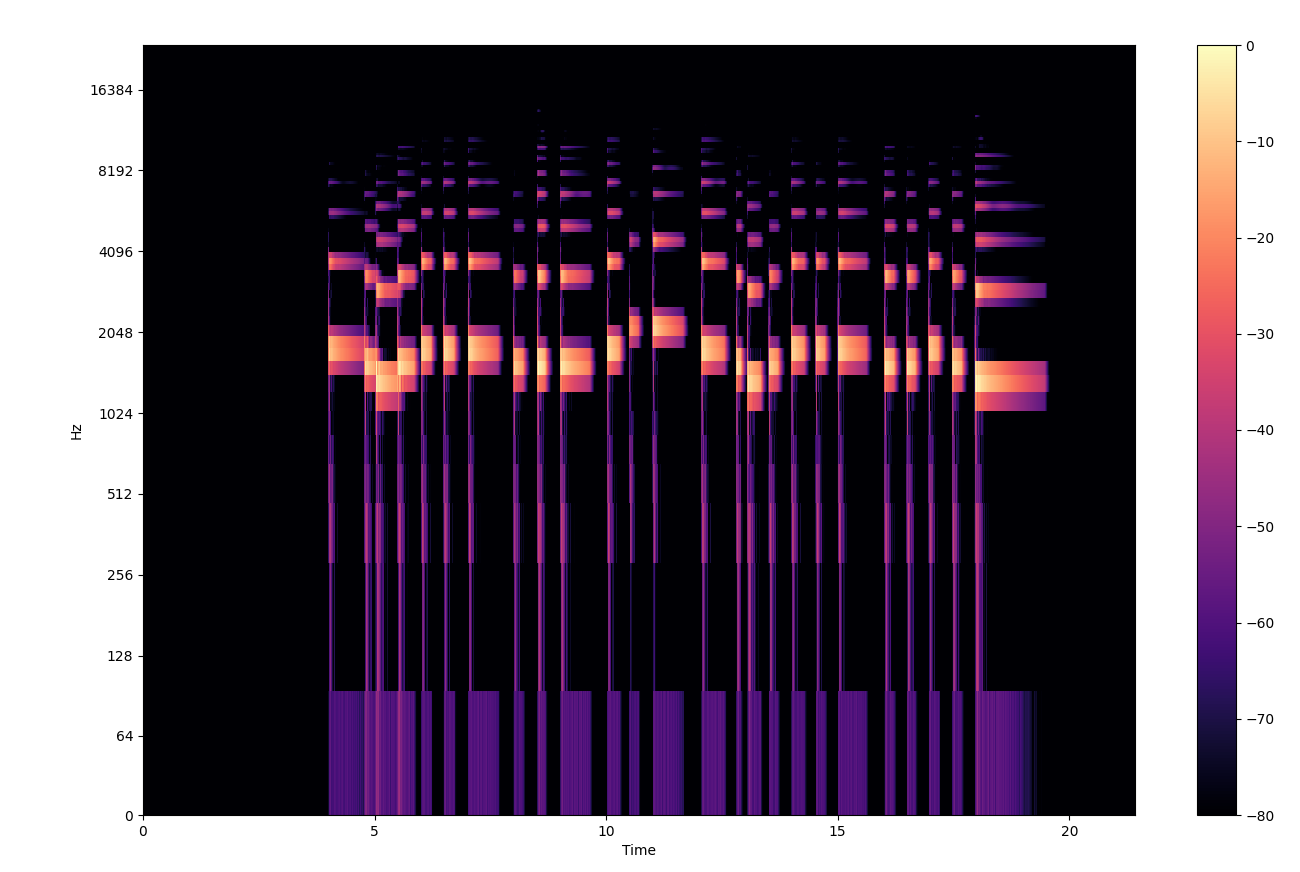
plt.figure(figsize=(16,10))

librosa.display.specshow(Mel\_DB, sr=samplingrate,hop\_length=512, x\_axis='time',y\_axis='log')

plt.colorbar()

plt.show()

'''



6. BPM 추정

텍스트이(가) 표시된 사진

자동 생성된 설명

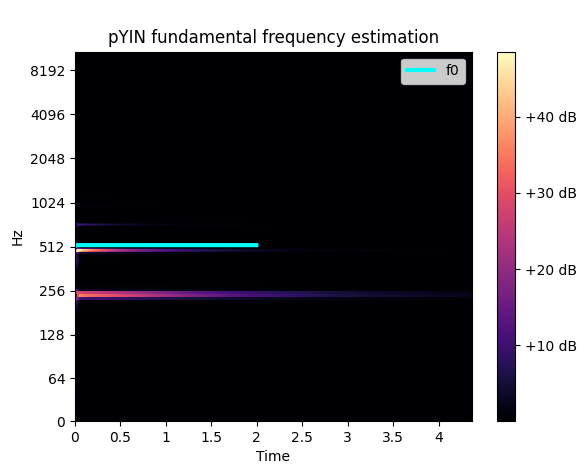
텍스트이(가) 표시된 사진

자동 생성된 설명

7. C4음 추출

텍스트, 모니터, 스크린샷, 화면이(가) 표시된 사진

자동 생성된 설명



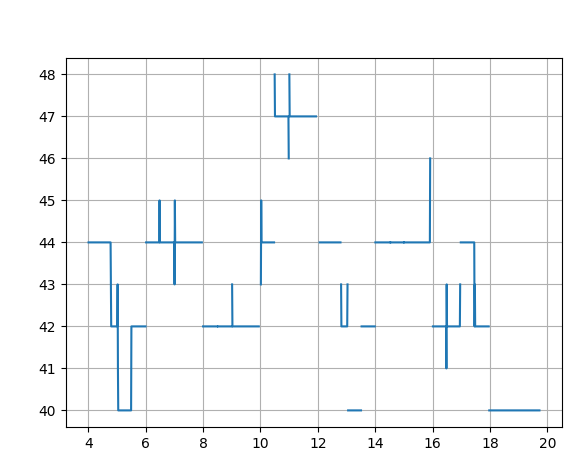
오류 : C5음으로 나타남

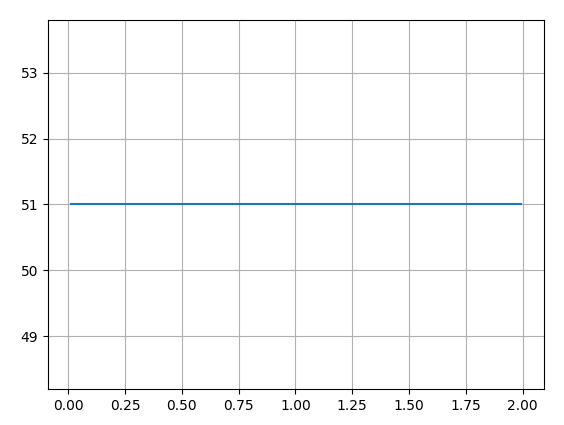
8. 정수화된 음값을 얻기 위한 과정

텍스트이(가) 표시된 사진

자동 생성된 설명 텍스트이(가) 표시된 사진

자동 생성된 설명





* 피드백 해 주신 내용

- 옥타브 위가 잡히는 오류 해결 및 오선보 적용

- 정확도 측정

<4차> : 피드백 받은 내용을 적용한 프로그래밍(최종 결과)

보고서 – 구현과정 file 참조