

```

1 #*****
2 # 구글 음성인식 및 MelSpectrogram 이미지 파일 출력 + TTS 테스트
3 #*****
4
5 # pip install gTTS
6 # pip install playsound
7 # pip install PyObjC
8
9 # conda install matplotlib, pyaudio
10 # conda install -c conda-forge librosa
11 # D:\programWAnaconda30WenvsWvoicerecogWLibWsite-packagesWlibrosaWutilWdecorators.py 파일에서
12 #     numba.decorators를 numba.core.decorators로 변경 (안할 경우 오류 발생)
13
14 # pip install youtube_transcript_api
15
16 import speech_recognition as sr
17 import os, random
18 from time import ctime
19 import webbrowser
20
21 # TTS 관련
22 import playsound
23 from gtts import gTTS
24
25 # 음성인식 관련
26 import librosa, librosa.display
27 import matplotlib.pyplot as plt
28 import numpy as np
29
30 # voice search on youtube
31 from youtube_transcript_api import YouTubeTranscriptApi
32 import webbrowser, urllib, re
33 from urllib import request, parse
34
35 import cv2
36
37 DISPLAY_TIME_AMPLITUDE = True
38 DISPLAY_FREQUENCY_MAGNITUDE = True
39 DISPLAY_MELSPECTROGRAM = True
40 DISPLAY_LOGSPECTROGRAM = True
41 DISPLAY_MFCC = True
42
43 r1 = sr.Recognizer()
44
45 #*****
46 # 동일 파일이름의 다음 파일명 구하기
47 #*****
48 def get_next_filename(filenamePrefix, extension):
49     counter = 0
50     filename = filenamePrefix+'{'+'.' + extension
51     while os.path.isfile(filename.format(counter)):
52         counter += 1
53     newFilename = filename.format(counter)
54     return newFilename
55
56 #*****
57 # 음성을 입력받아 문자로 변환
58 #*****
59 def sentence_from_audio(direction='How can I help You?', soundFrom='MIC') :
60     #print('energy threshold:',sr.energyThreshold)
61
62     existingWavFileName = "wavFile.wav"
63     recognizedText = ''
64
65     with sr.Microphone() as source:
66         r1.adjust_for_ambient_noise(source,duration=1)
67         #cnt = 1
68         while True:
69             ### 마이크로부터 음성 입력
70             #print(ctime())
71             audioType = ''
72             if soundFrom == 'MIC':
73                 print(direction) # direction 출력
74                 #googleTTS(direction)

```

```

75
76     audio = r1.listen(source)                                # 마이크로부터 음성 입력
77
78     try :
79         recognizedText = r1.recognize_google(audio)          # 구글 WebSpeech 음성인식
80         print("you spoke: ", recognizedText)
81         audioType = 'ordinary'
82         recordedWavFileName = recognizedText + '.wav'        # 인식된 문장 이름의 wav파일명 생성
83         #return sentence
84     except sr.UnknownValueError:
85         print('Please say again')
86     except sr.RequestError as e:
87         print('오류:{}'.format(e))
88
89     ### 오디오 파일로부터 음성 입력 후 인식
90 else:
91     audio_file = sr.AudioFile(existingWavFileName)
92     with audio_file as sourceFile:
93         audio = r1.record(sourceFile)
94         recognizedText = r1.recognize_google(audio)
95         print("==> The Audio File Says: ", recognizedText)
96
97     if audioType == 'ordinary' :
98         # 음성을 wav파일로 저장
99         with open(recordedWavFileName,"wb") as fpWav:
100             fpWav.write(audio.get_wav_data())
101
102         # Time-Amplitude 축 상에서 파형 출력해보기
103         spokenWaveData, srVal = librosa.load(recordedWavFileName, sr=None) # 음성파일을 읽어서 spokenWaveData
104     # 다음줄의 waveplot함수를 위하여, float 타입
105     #print("wave Data")
106     #print(spokenWaveData)
107     plt.interactive(False)
108     # TIME-AMPLITUDE 데이터로 표시
109     if DISPLAY_TIME_AMPLITUDE :
110         librosa.display.waveplot(spokenWaveData, sr=srVal, x_axis='time', offset=0.0)
111         plt.title('Time-Amplitude')
112         plt.xlabel("Time")
113         plt.ylabel("Amplitude")
114         plt.show()
115
116     #-----
117     # MelSpectrogram
118     #-----
119     plt.figure()
120     #S = librosa.feature.melspectrogram(y=spokenWaveData, sr=srVal)
121     S = librosa.feature.melspectrogram(y=spokenWaveData, sr=srVal, n_mels=64)
122     S_dB = librosa.power_to_db(S, ref=np.max)
123     print("S_dB:", len(S_dB), "한줄의 크기: ", len(S_dB[0]))
124     print(S_dB)
125     if DISPLAY_MELSPECTROGRAM :
126         fig = plt.figure(figsize = [0.72, 0.72])
127         ax = fig.add_subplot(1,1,1)
128         ax.axes.get_xaxis().set_visible(False)
129         ax.axes.get_yaxis().set_visible(False)
130         ax.set_frame_on(False)
131         librosa.display.specshow(S_dB)
132         fileName = get_next_filename(recognizedText, 'jpg') # MelSpectrogram를
133         plt.savefig(fileName, dpi=400, bbox_inches='tight', pad_inches=0) # 이미지파일로 출력
134         plt.close('all')
135         del fileName, fig, ax, S, S_dB
136         #plt.colorbar()
137         #plt.show()
138
139     # fft
140     fftResult = np.fft.fft(spokenWaveData)
141
142     magnitude = np.abs(fftResult)
143     frequency = np.linspace(0, srVal, len(magnitude)) # Lineary Spcaed (배열시작값, 배열 끝값, 갯수)
144
145     # Frequency magnitude 상에서 파형 출력해보기
146     if DISPLAY_FREQUENCY_MAGNITUDE :
147         #left_frequency = frequency[:int(len(frequency)/2)]

```

```

148     #left_magnitude = magnitude[:int(len(magnitude)/2)]
149     #plt.plot(left_frequency, left_magnitude)
150
151     plt.plot(frequency, magnitude)
152     plt.title('MAGNITUDE-FREQ')
153     plt.xlabel("Frequency")
154     plt.ylabel("Magnitude")
155     plt.show()
156
157     # Spectrogram으로 변환
158     noOfFft = 2048
159     hopLength = 512
160     # Short Time Fourier Transform
161     stft = librosa.core.stft(spokenWaveData, hop_length = hopLength, n_fft = noOfFft)
162     spectrogram = np.abs(stft)
163     # Amplitude Spectrogram을 dB-scaled Spectrogram으로 변환
164     logSpectrogram = librosa.amplitude_to_db(spectrogram)
165     #cv2.imshow('opencv-logspectrogram', logSpectrogram)
166     print('LogSpectrogram', len(logSpectrogram), "한줄의 크기: ", len(logSpectrogram[0]))
167     print(logSpectrogram)
168
169     # Spectrogram 공간에서 강도로 표현
170     if DISPLAY_LOGSPECTROGRAM :
171         #librosa.display.specshow(pectrogram, sr=srVal, hop_length = hopLength)
172         librosa.display.specshow(logSpectrogram, sr=srVal, hop_length = hopLength)
173         plt.title('Log Spectrogram')
174         plt.xlabel("Time")
175         plt.ylabel("Frequency")
176         plt.colorbar()
177         plt.show()
178
179     # MFCC
180     #mfcc = librosa.feature.mfcc(spokenWaveData, n_fft=noOfFft, hop_length = hopLength, n_mfcc = 13)
181     mfcc = librosa.feature.mfcc(spokenWaveData, n_fft=noOfFft, hop_length = hopLength, n_mfcc = 40)
182     print("\nMFCC", len(mfcc), "한줄의 크기: ", len(mfcc[0]))
183     print(mfcc)
184     #cv2.imshow('opencv-mfcc', mfcc)
185     if DISPLAY_MFCC :
186         #librosa.display.specshow(pectrogram, sr=srVal, hop_length = hopLength)
187         librosa.display.specshow(mfcc, sr=srVal, hop_length = hopLength)
188         plt.title('MFCC')
189         plt.xlabel("Time")
190         plt.ylabel("MFCC")
191         plt.colorbar()
192         plt.show()
193
194     #cnt += 1
195
196     if spokenWaveData is not None:
197         del spokenWaveData
198     if srVal is not None:
199         del srVal
200
201     if 'bye-bye' in recognizedText:
202         print('Bye bye.....')
203         break
204     else :
205         response = respond(recognizedText)
206         print(response)
207
208 *****
209 # 구글 TTS를 이용하여, TEXT를 음성파일로 저장후 소리 출력
210 *****
211 def googleTTS(sentence, language='en') :
212     tts = gTTS(text=sentence, lang=language) # language로 설정한 언어와 문장으로 tts 인스턴스 생성
213
214     r = random.randint(1,1000000) # 파일명 난수로 임의로 설정
215     audioFile = 'audio_' +str(r) + '.mp3'
216
217     tts.save(audioFile) # 문장을 음성파일로 저장
218     playsound.playsound(audioFile) # 음성파일을 불러서 음성 출력
219
220     #print(sentence) # 문장 내용 출력 후 음성 파일은 지움
221     os.remove(audioFile)
222

```

```

223 #*****
224 # 사용자가 입력한 문장에 따라 몇가지만 반응하는 함수
225 #*****
226
227 def respond(sentence=''):
228     searchWords = ['search', 'find', 'i want to know', 'open web browser']
229     nameWords = ['your name', 'should i call']
230     positiveAnswerWords = ['you free', 'can i call you', 'is it okay']
231     # 영어학습과 관련하여 발생 가능한 문장
232     conversation = {}
233
234     if 'what time' in sentence:                # time 단어가 들어있는 경우
235         print(ctime())
236         googleTTS(ctime(), 'en')
237         return ''
238
239     for phrase in nameWords :
240         if phrase in sentence:                # 이름을 묻는 질문인 경우
241             txt = 'My name is AI Kim'
242             txtKor = '저는 AI 김입니다'
243             print(txt)
244             googleTTS(txtKor, 'ko')
245             return ''
246
247     for phrase in positiveAnswerWords :
248         if phrase in sentence:                # 긍정적 답을 하는게 좋은 질문인 경우
249             googleTTS('Sure, Feel free to say anything', 'en')
250             return ''
251
252     for phrase in searchWords :
253         if phrase in sentence:                # search 단어가 들어있는 경우, 웹브라우저로 검색결과 실행
254             search_data = sentence_from_audio('What do you want to search for?', 'MIC')
255             url = 'https://google.com/search?q=' + search_data
256             webbrowser.get().open(url)
257             str = 'Search Result for ' + search_data
258             print(str)
259             return ''
260
261     return 'I am ready.'
262
263 if __name__ == '__main__':
264     print("종료하려면 'bye bye'를 말하세요")
265     sentence_from_audio('How Can I Help You?', 'MIC')
266     print('Program Finished')
267

```