**Programming Assignment #1**

20231186 AI대학원 최유선

**과제 설명**

* Python의 wave 및 numpy 라이브러리를 활용해, WAV 오디오 파일을 처리하고 Saturation 적용 여부에 따른 변화 분석하기

**구현 내용**

*import* wave

*import* numpy *as* np

def *process\_file*(input\_file, output\_file, value, saturation=False):

*with* wave.open(input\_file, 'rb') *as* wav\_in:

params = wav\_in.getparams()

*# params (nchannels=1, sampwidth=2, framerate=8000, nframes=18800, comptype='NONE', compname='not compressed')*

frames = wav\_in.readframes(params.nframes) *# wav파일에서 프레임(18800)을 읽어 바이트(2) 형태로 반환 -> 37600*

*# 바이트 데이터를 numpy 배열로 변환*

dtype = np.int16

audio\_data = np.frombuffer(frames, dtype=dtype) *# 부호 있는 16비트 오디오*

*# audio1.min(): -6693 / audio1.max(): 23740*

*# audio2.min(): -49849 / audio2.max(): 102316*

*# 오디오 증폭*

audio\_data = (audio\_data.astype(np.float32) \* 5).astype(np.int32)

modified\_audio\_data = audio\_data + value *# 샘플 값에 4096 더하기*

*# saturation 연산*

*if* saturation:

modified\_audio\_data = np.clip(modified\_audio\_data, -2\*\*15, 2\*\*15-1).astype(np.int16)

*# 수정된 데이터를 바이트 형식으로 변환*

modified\_frames = modified\_audio\_data.astype(dtype).tobytes()

*with* wave.open(output\_file, 'wb') *as* wav\_out:

wav\_out.setparams(params)

wav\_out.writeframes(modified\_frames)

input\_file = './data/a.wav'

output\_file\_no\_saturation = './output/no\_saturation.wav' *# saturation 처리 안함*

output\_file\_saturation = './output/saturation.wav' *# saturation 처리*

value = 4096

process\_file(input\_file, output\_file=output\_file\_no\_saturation, value=value, saturation=False)

process\_file(input\_file, output\_file=output\_file\_saturation, value=value, saturation=True)

* **WAV 파일 읽기 및 변환**
  + Python 라이브러리 wave와 numpy를 사용하여, wav 파일에서 프레임을 읽어 바이트 형태로 반환
  + 바이트 데이터를 numpy의 int16 배열로 변환하여 샘플 값으로 활용
* **오디오 샘플 값 조정**
  + Saturation 효과를 보다 명확히 관찰하기 위해, 샘플 값을 5배 증가하고 int32로 변환 후 각 샘플 값에 4096을 추가하여 오디오 신호의 크기를 증폭
* **Saturation 적용 여부에 따른 처리**
  + Saturation이 True인 경우, clipping을 통해 int16의 유효 범위인 -32768~32767로 제한하고 다시 16비트 정수로 반환함
  + Saturation이 False인 경우,, overflow가 발생할 수 있도록 별도로 clipping을 적용하지 않음
* **WAV 파일 저장**
  + int16 배열을 다시 바이트 형식으로 변환하여 각각 wav 파일로 저장

**결과 분석**

**다채로움, 스크린샷, 레드, 예술이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.**

**8kHz원본 신호**

**물, 다채로움, 호박, 스크린샷이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다. 다채로움, 오렌지, 호박, 레드이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.**

**Saturation 적용 Saturation 미적용**

* 원본 신호에서 5배 증폭되었기 때문에 스펙트로그램 상에서 에너지가 증가한 걸 확인할 수 있음
* Saturation 적용한 결과는 원본과 비교적 비슷하게 들림
* Saturation 미적용한 결과는 distortion이 발생하여 들었을 때 음질이 좋지 않음