**AI Speech HW\_1**

전자통신공학과 2016707072 신민정

**Task :** 각 note의 fundamental frequency구하기

**Data**

녹색, 창문, 앉아있는, 모니터이(가) 표시된 사진

자동 생성된 설명

8개의 note로 이루어진 Audio data

Sampling rate = 1600KHz

**과제 흐름&Code 설명**

*교수님께서 제공해주신 Parameter, Baseline에 기반하여 코드를 구성하였습니다. C언어로 coding하였고, 추가 설명과 파형 또는 그래프가 필요한 부분은 python으로 구현하였습니다.*

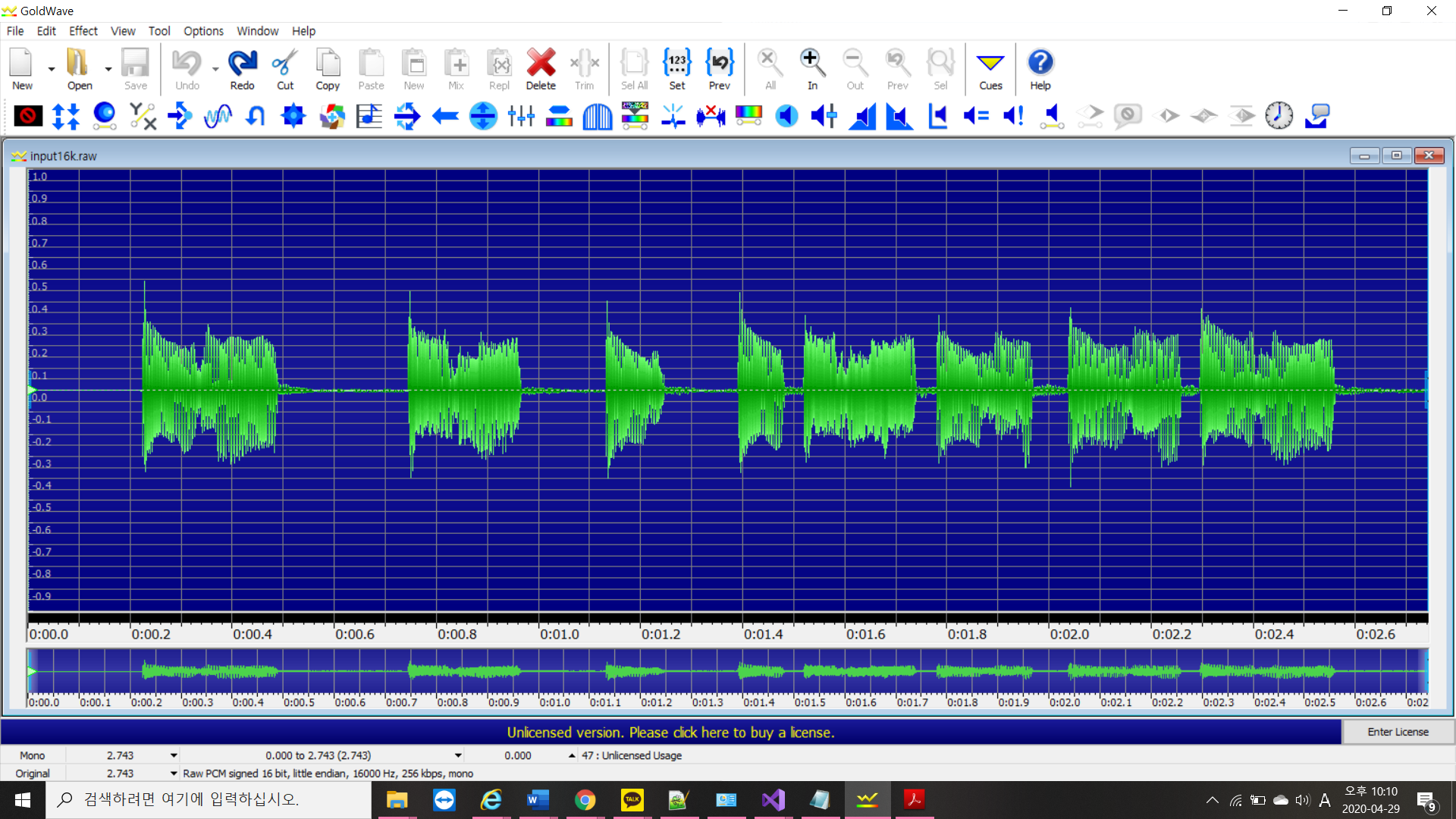
**DataLoad&Skip**

File pointer와  로 file을 읽어옵니다.

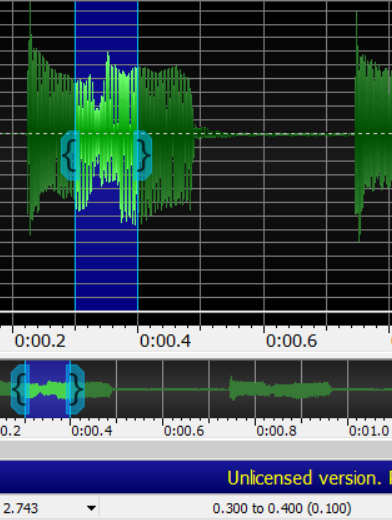
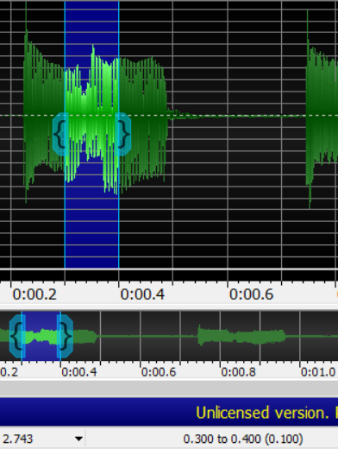
각 x[idx]는 16bit=2byte data(short)입니다.

Sampling freq를 16KHz, 0.1초 씩 sampling을 하면 1초에 1600 sample을 얻을 수 있습니다.

Gold wave프로그램으로 wave를 확인합니다. Wave form을 보면 값이 있는 영역과 값이 없는 영역이 구분되어있습니다. 소리가 있는 파형에서 1600개의 sample을 가져와야합니다.



8개의 note의 wave form을 뽑기 위한 skip배열을 지정합니다.

1 note ->  2 note -> 회로, 컴퓨터, 기타이(가) 표시된 사진

자동 생성된 설명

3 note -> 컴퓨터이(가) 표시된 사진

자동 생성된 설명 4 note -. 회로, 컴퓨터이(가) 표시된 사진

자동 생성된 설명

5 note -> 컴퓨터, 회로이(가) 표시된 사진

자동 생성된 설명 6 note -> 회로, 컴퓨터이(가) 표시된 사진

자동 생성된 설명

7 note ->회로, 컴퓨터이(가) 표시된 사진

자동 생성된 설명 8 note ->회로이(가) 표시된 사진

자동 생성된 설명

녹색, 테이블, 남자, 하얀색이(가) 표시된 사진

자동 생성된 설명

*<Python>*

Python의 Librosa를 이용하여 data를 load하고 waveform을 plt해보았습니다.

Librosa는 wave형식(.wav)만 불러올 수 있기 때문에, golde wave로 파일형식을 변경해 주었습니다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

Load할 때, sample rate를 지정해 주어야 하기 때문에 １６000으로 지정해주었습니다.(default ：sr = 22050)

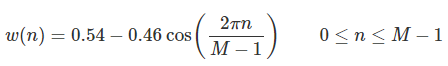
**Windowing**

**스크린샷이(가) 표시된 사진

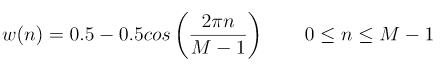
자동 생성된 설명**

Hamming window와 Hanning window를 구현하였습니다.

*Hamming window function*



*Hanning window function*



skip으로 뽑은 signal에 Window function을 window size만큼 곱해주며 windowing을 합니다.

*<Python>*

Scipy.sgnal의 get\_window함수를 사용하여 window함수와 sigal\*window를 plt해보았습니다.

get\_window에서는 따로 구현 없이 window의 이름을 input으로 입력하여 window fuction을 얻습니다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

Hanning window function입니다.

원신호가

앉아있는, 물, 줄지은, 하얀색이(가) 표시된 사진

자동 생성된 설명

이 신호라면 Hanning windowing을 거친 신호는

 (audio\_fame ：값이 있는 부분만 자른 wave)

앉아있는, 해변, 타기이(가) 표시된 사진

자동 생성된 설명

이렇게 구현됩니다.

Windowing은 신호의 양 끝을 zero에 가깝게 smooth하게 만들어 줄 뿐만 아니라, main-lobe의 width와 side-lobe의 레벨의 Trade-off 를 제어해 줍니다.. 그리고 깁스 현상을 방지해 줍니다.

**DFT**

**스크린샷이(가) 표시된 사진

자동 생성된 설명**

푸리에 변환(Fourier transform)은 임의의 입력 신호를 다양한 주파수를 갖는 주기함수(복수 지수함수)들의 합으로 분해하여 표현하는 것 입니다. 그리고 각 주기함수들의 진폭을 구하는 과정을 퓨리에 변환이라고 합니다.

sampling으로 들어온 데이터는 바로 시간의 간격에 따른 소리의 amplitude의 discrete한 데이터이기 때문에, 푸리에 변환식을 Discrete한 영역으로 생각해본 것이 DFT(Discrete Fourier transfom)입니다. Digital 처리를 위해 time과 frequency domain에서 모두 sampling되어야 하므로 DFT를 사용합니다.

Real part(cos)와 imaginary part(sin)으로 나누어 계산하였습니다.

DFT 공식

스크린샷이(가) 표시된 사진

자동 생성된 설명(출처:강의자료)

<Python>

오일러 공식에 의해 C에서 구현한 cos+jsin을 지수함수 형태로 바꾸어 함수를 구성할 수 있습니다.

python으로는 windowing이 포함되지 않은 DFT만 구현해 보았습니다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

**Peak magnitude**

**검은색, 화면, 컴퓨터, 방이(가) 표시된 사진

자동 생성된 설명**

f0(fundamental frequency)를 찾기 위해, 배열의 크기 비교를 하며 신호 중 가장 magnitude가 큰 신호를 찾습니다.

**Convert Hz**

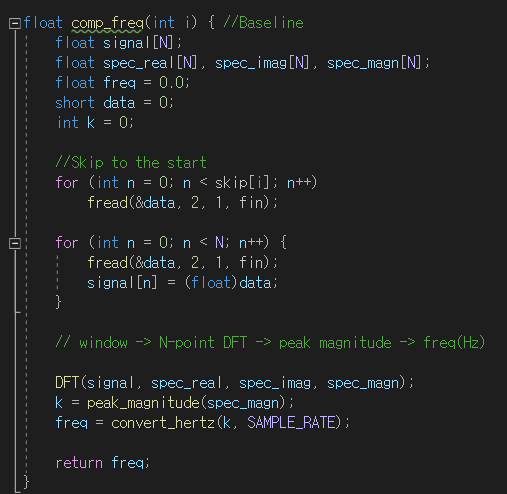
**그리기이(가) 표시된 사진

자동 생성된 설명**

N-pint DFT의 단위는 k(index)이므로 Hz단위로 바꾸어 주어야 합니다.

Hz = sample rate/N

**Comp freq**

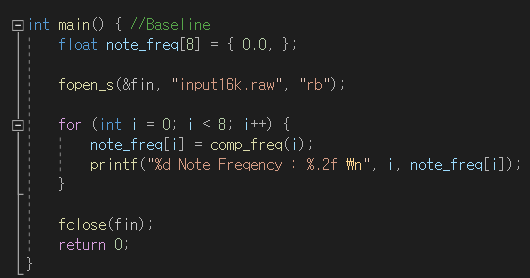
****

Spectrum의 real, imaginary part를 나누어 불러오고,

첫번째 for문으로 note를 1개씩 가지고 옵니다.

Read data for문으로 1600개의 sample을 불러옵니다. 그 후 window를 하고, N-point dft를 한 후 f0를 찾아 Hz단위로 바꾸어줍니다.

**Main**

****

위에서 구현한 함수들로, 8개의 주파수를 출력해보는 main함수입니다.

**추가 (STFT->Spectogram)**

시간의 변화에 따른 signal의 특징을 보기위해 STFT를 사용합니다. STFT를 하면 시간에 따른 주파수 특성이 포함되기 때문에 Spectrogram을 그릴 수 있습니다. Speech task에 많이 활용되는 tacotron, wave-Net등 많은 model에서 input으로 spectrogram을 많이 사용합니다. 주어진 data를 가지고 Mel-spectorgram과 MFCC를 plt해보았습니다.

**스크린샷이(가) 표시된 사진

자동 생성된 설명**

**스크린샷이(가) 표시된 사진

자동 생성된 설명**

**결과**

**텍스트이(가) 표시된 사진

자동 생성된 설명 Hammming**

**텍스트이(가) 표시된 사진

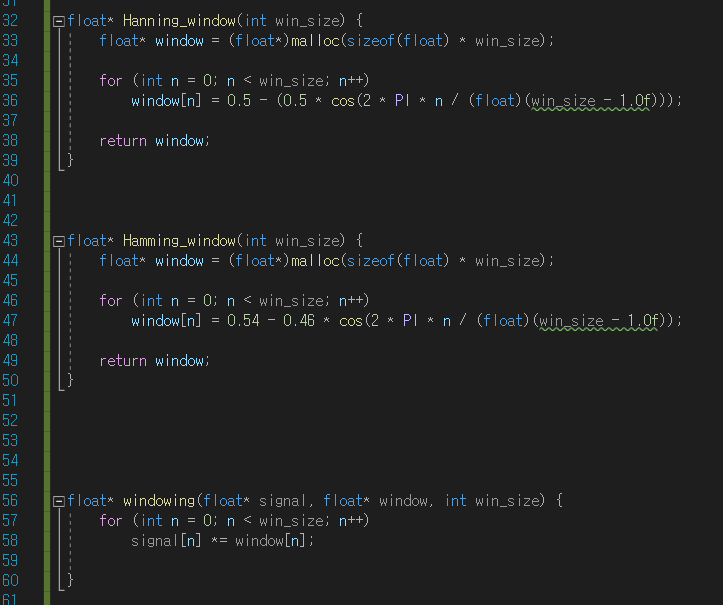
자동 생성된 설명Hanning**

|  |  |  |
| --- | --- | --- |
|  | **Found. Freq(Hz)** | **Note** |
| **1** | 340349.23 | 파 |
| **2** | 420 | 솔# |
| **3** | 350349.23 | 파 |
| **4** | 350349.23 | 파 |
| **5** | 470466.16 | 라# |
| **6** | 350349.23 | 파 |
| **7** | 310311.13 | 레# |
| **8** | 350349.23 | 파 |

**전체 소스코드**

스크린샷, 컴퓨터이(가) 표시된 사진

자동 생성된 설명



검은색, 앉아있는, 전화, 컴퓨터이(가) 표시된 사진

자동 생성된 설명

앉아있는, 테이블, 검은색, 전화이(가) 표시된 사진

자동 생성된 설명

노트북, 컴퓨터이(가) 표시된 사진

자동 생성된 설명