

1. Windowing Functions

o Rectangle Window

$$\mathcal{F}\{\Pi(t)\} = \text{sinc}(f)$$

o Triangular window

$$\mathcal{F}\{\Lambda(t)\} = \text{sinc}^2(f)$$

$$\mathcal{F}\{2\Lambda(2t)\} = \text{sinc}^2(f/2)$$

o Hamming window

$$\mathcal{F}\{w(t)\} = \int_{-\infty}^{\infty} w(t) e^{-j2\pi ft} dt$$

$$= \int_{-1/2}^{1/2} \cos^2(\pi t) e^{-j2\pi ft} dt$$

$$= \int_{-1/2}^{1/2} \frac{1}{4} (e^{j2\pi t} + e^{-j2\pi t} + 2) \cdot e^{-j2\pi ft} dt$$

$$= \int_{-1/2}^{1/2} \left(\frac{1}{4} e^{j2\pi(-f+1)t} + \frac{1}{4} e^{-j2\pi(f+1)t} + \frac{1}{2} e^{-j2\pi ft} \right) dt$$

$$= \left[\frac{1}{j8\pi(-f+1)} e^{j2\pi(-f+1)t} \right]_{-1/2}^{1/2} + \left[\frac{1}{-j8\pi(f+1)} e^{-j2\pi(f+1)t} \right]_{-1/2}^{1/2}$$

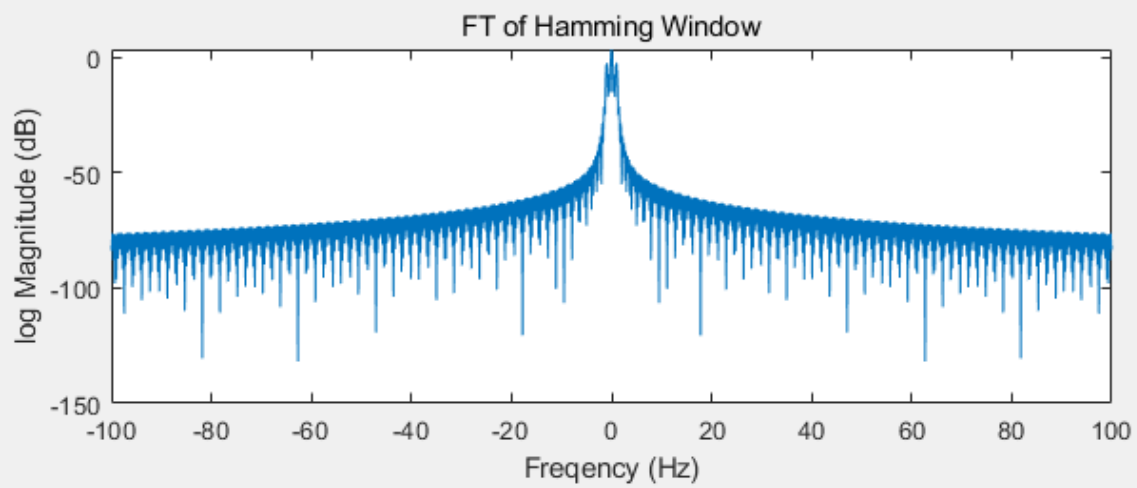
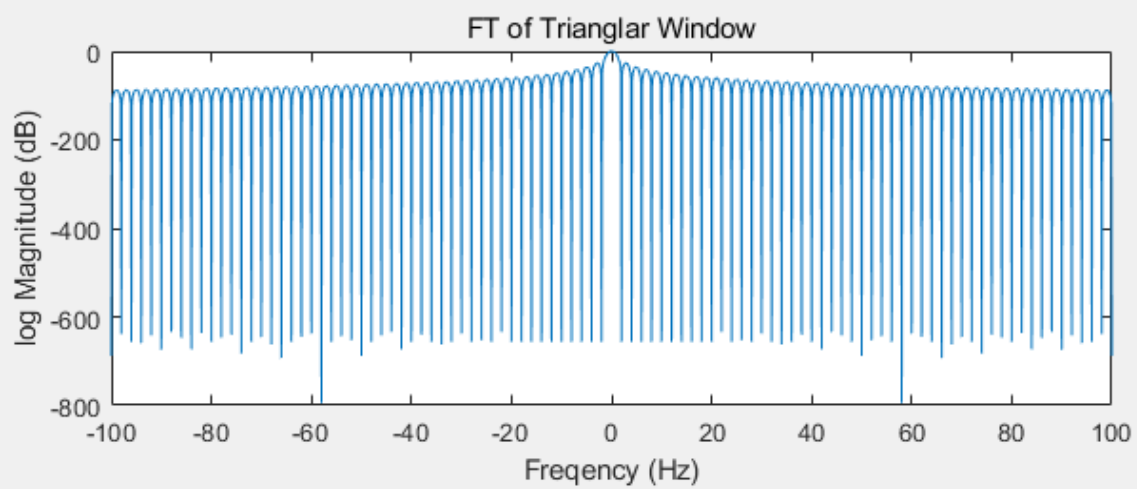
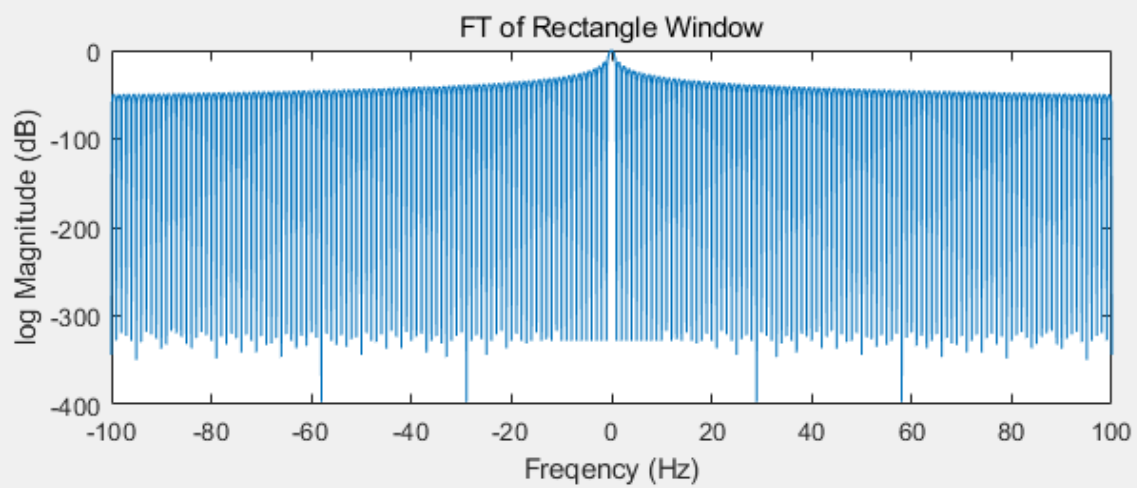
$$+ \left[\frac{1}{-j4\pi f} e^{-j2\pi ft} \right]_{-1/2}^{1/2}$$

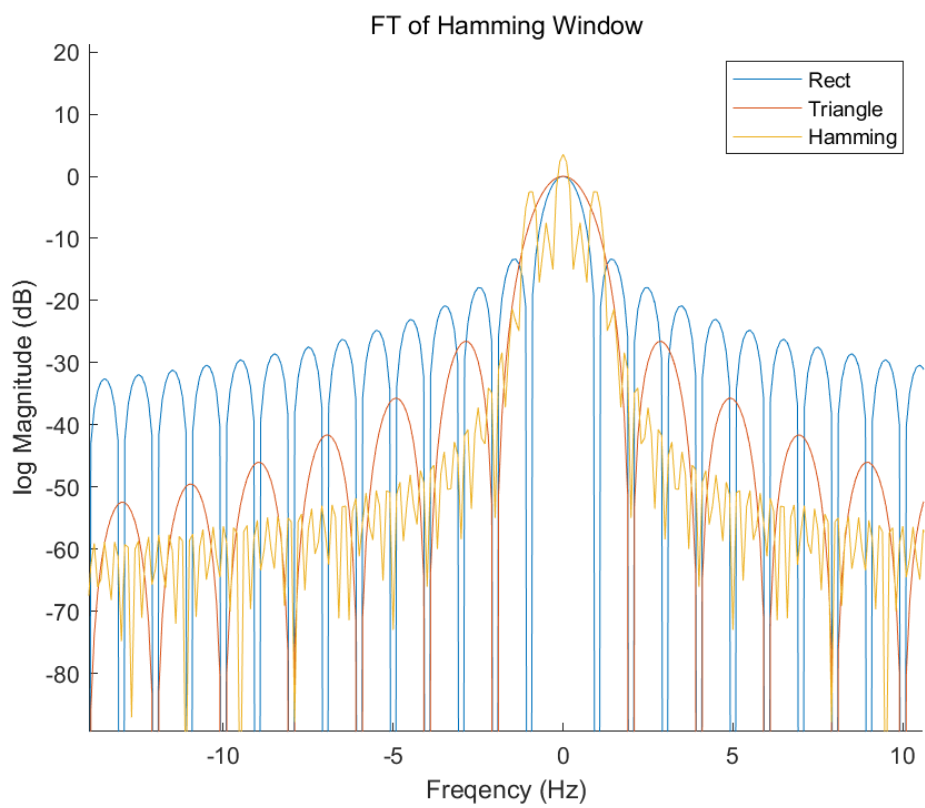
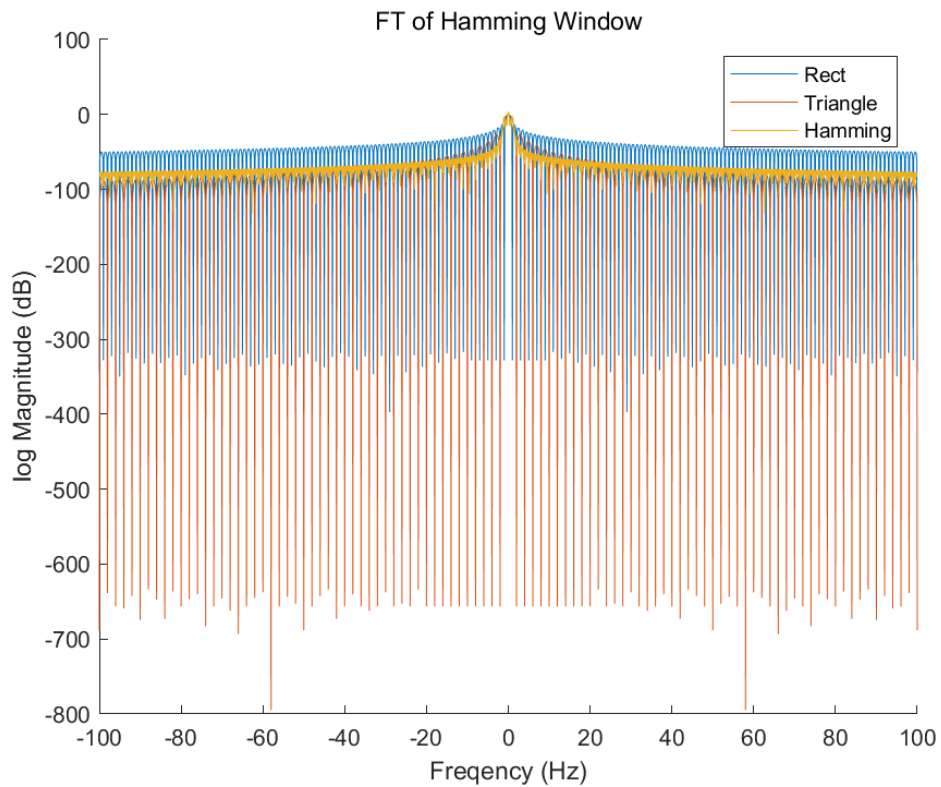
$$= \frac{1}{j8\pi(-f+1)} (e^{j\pi(-f+1)} - e^{-j\pi(-f+1)}) + \frac{1}{-j8\pi(f+1)} (e^{-j\pi(f+1)} - e^{j\pi(f+1)})$$

$$+ \frac{1}{-j4\pi f} (e^{-j\pi f} - e^{j\pi f})$$

$$= \frac{1}{4\pi(f-1)} \sin(\pi(f-1)) + \frac{1}{4\pi(f+1)} \sin(\pi(f+1))$$

$$+ \frac{1}{2\pi f} \sin(\pi f)$$

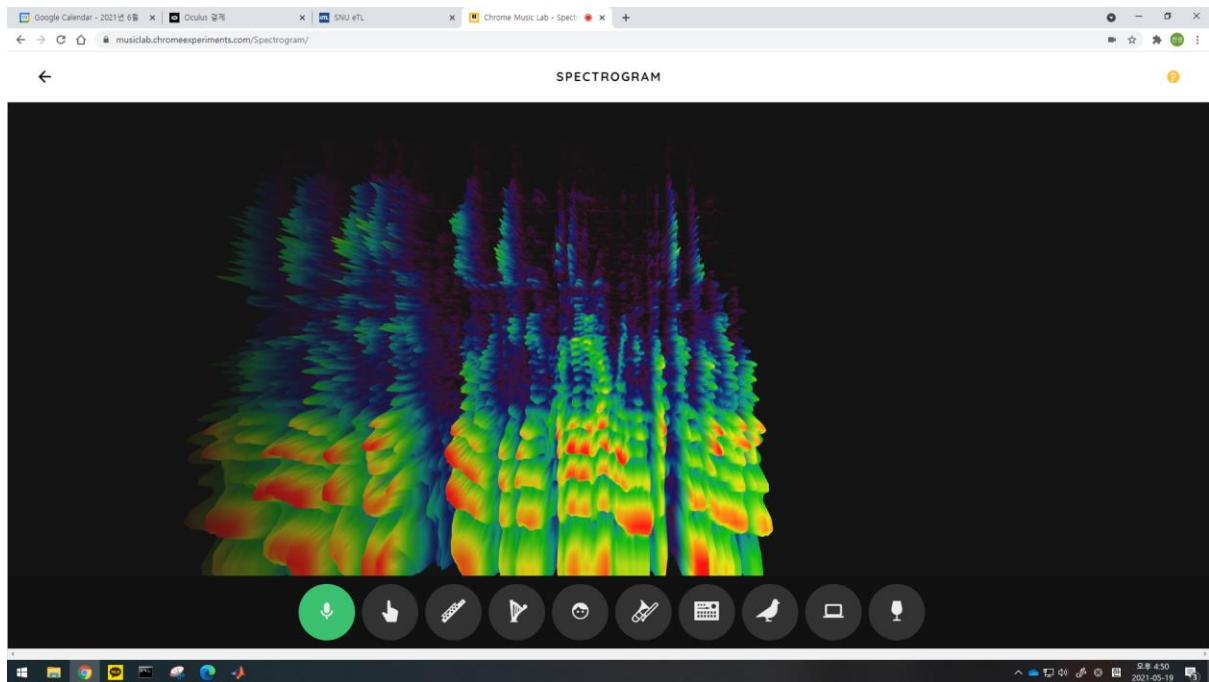




이상적인 Windowing Function은 $w(t)=1$ 이고, 이것의 푸리에 변환은 델타 함수이다.

위 함수들 중 델타 함수와 가장 가까운 모습을 한 것은 Hamming Function이다.

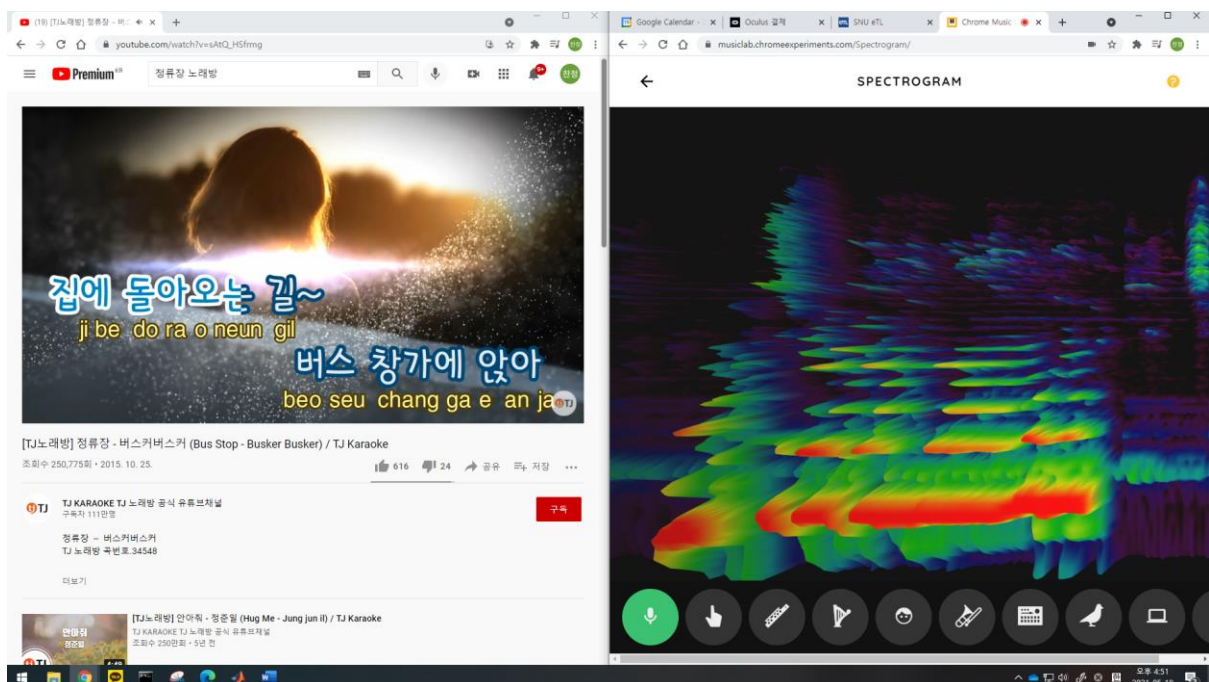
(a)



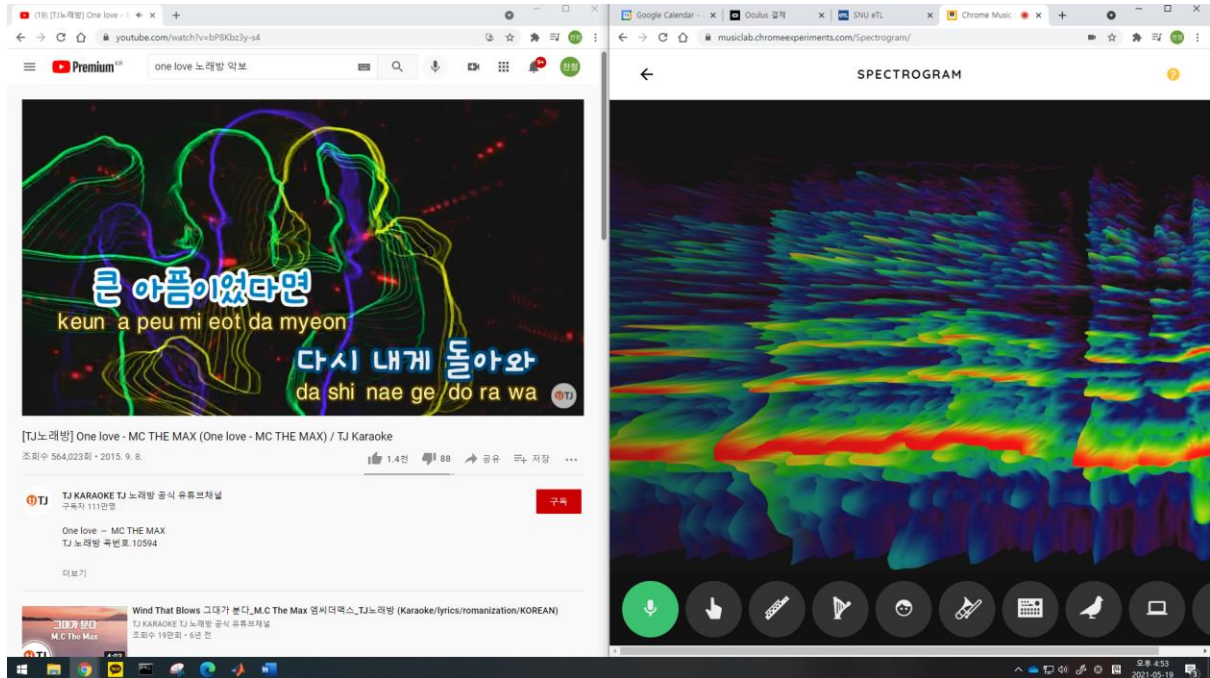
본인의 말소리를 Spectrogram으로 나타내보았다.

(b)

다른 음역대의 노래를 불러보았다.

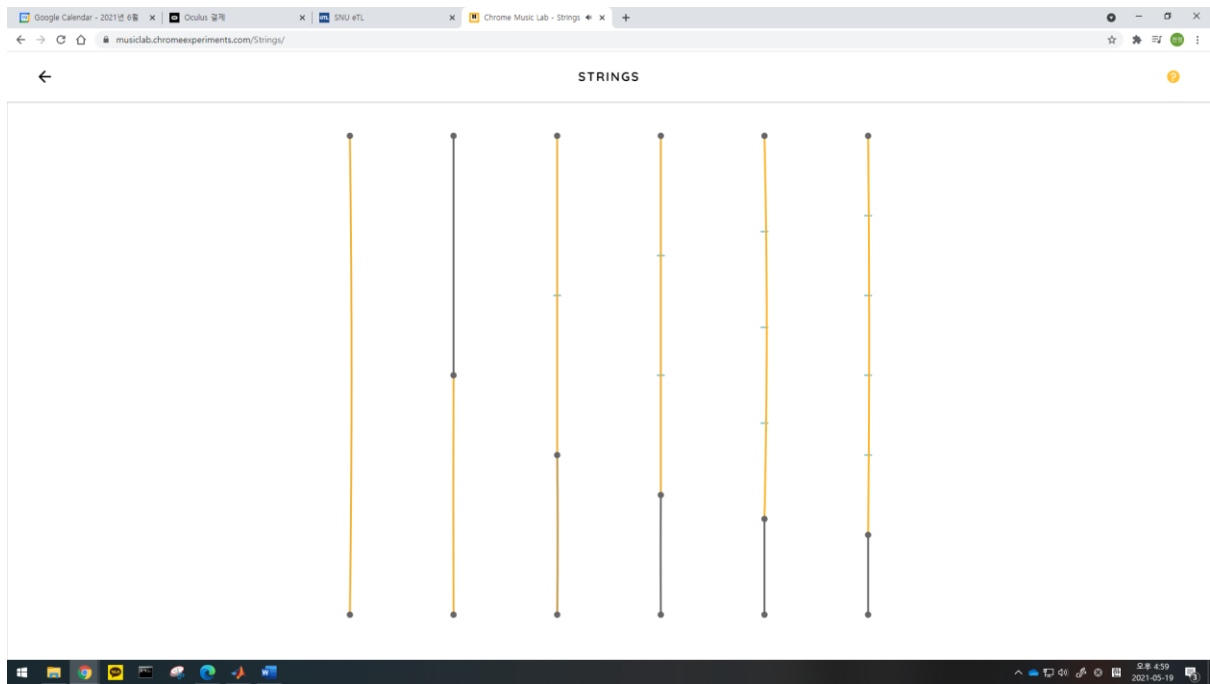


먼저 버스커버스커의 정류장을 불러보았다. 특히 캡처한 모습은 곡의 앞 부분인데, 이 부분의 음역대는 꽤 낮다. Spectrogram으로 확인해보았을 때도 진폭이 큰 부분(붉게 표시되는 부분)이 화면의 아래쪽에 위치하는 것을 확인할 수 있다.



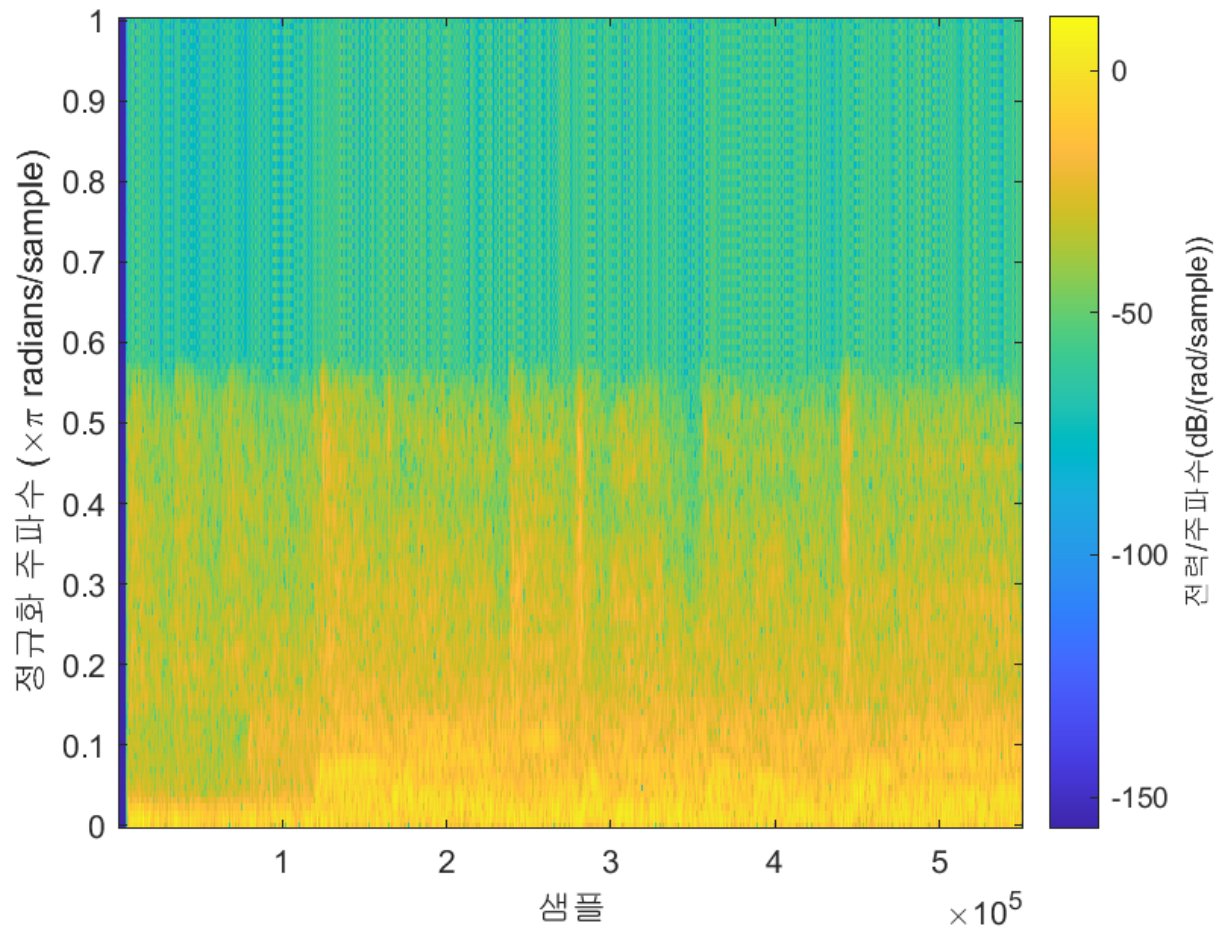
다음으로 MC THE MAX의 One love를 불러보았다. 특히 캡처한 부분은 음이 높은 부분인데, 진폭이 큰 부분이 이전보다 위쪽에 위치하는 것을 확인할 수 있다. 그리고 붉게 표시되는 부분의 폭도 훨씬 좁아진 것을 확인할 수 있다.

(c)



많은 실험 중 Strings를 선택해보았다. 줄의 길이에 따라 음의 높낮이가 달랐다. 줄이 길수록 낮은 음, 줄이 짧을수록 높은 음이 재생되었다.

(d)



야구팀 한화 이글스의 김성근 감독 응원가 일부를 Spectrogram으로 나타내었다. 음원 파일의 샘플 레이트는 44100Hz이었고 spectrogram에서 Window는 길이 128의 Hamming Window, Overlap Length는 120으로 설정하였다.

(e)

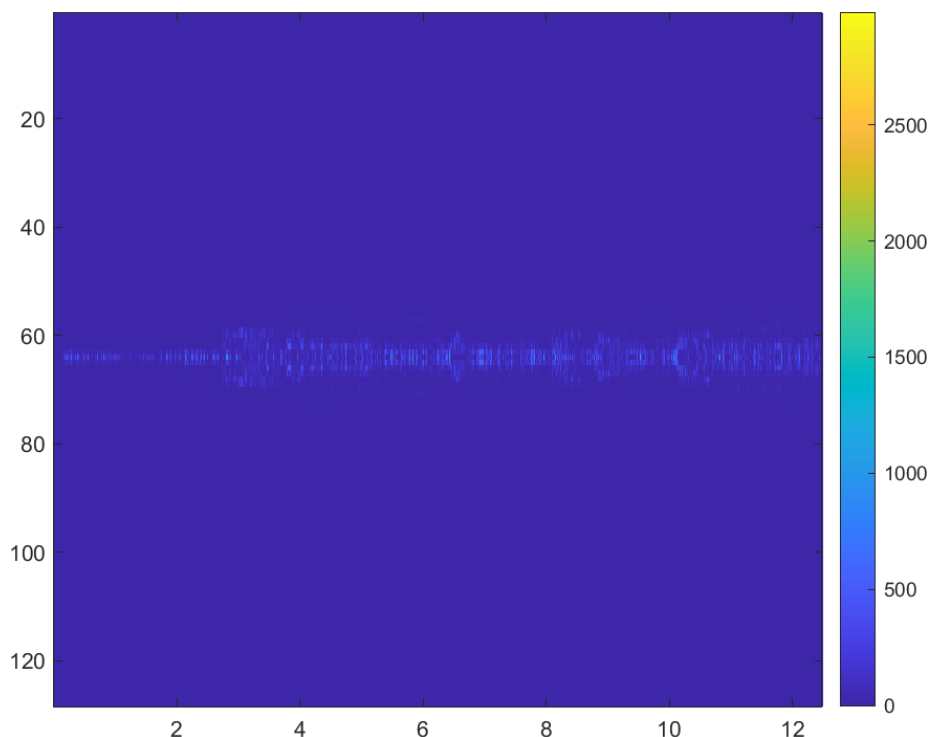
Continuous Time STFT는 임의의 Window Function을 설정하여 원래 함수에 곱한 뒤 Fourier Transform을 하는 식으로 이루어진다. 이 때 Window Function은 과제 1번에서 확인하였듯이 신호의 특정 부분을 남기고 나머지 부분을 없애는 용도로 사용된다. Window Function을 특정 시간만큼 평행이동하고 원래 함수에 곱하여 Fourier Transform을 함으로써 그 시간 주변의 짧은 기간동안의 주파수 스펙트럼을 확인할 수 있다.

Discrete Time STFT도 CT STFT와 크게 다르지 않다. Window Function을 설정하고 특정 값만큼 평행이동하여 원래 함수에 곱한 뒤 Fourier Transform을 함으로써 그 값 주변의 작은 범위에서의 주파수 스펙트럼을 확인할 수 있다.

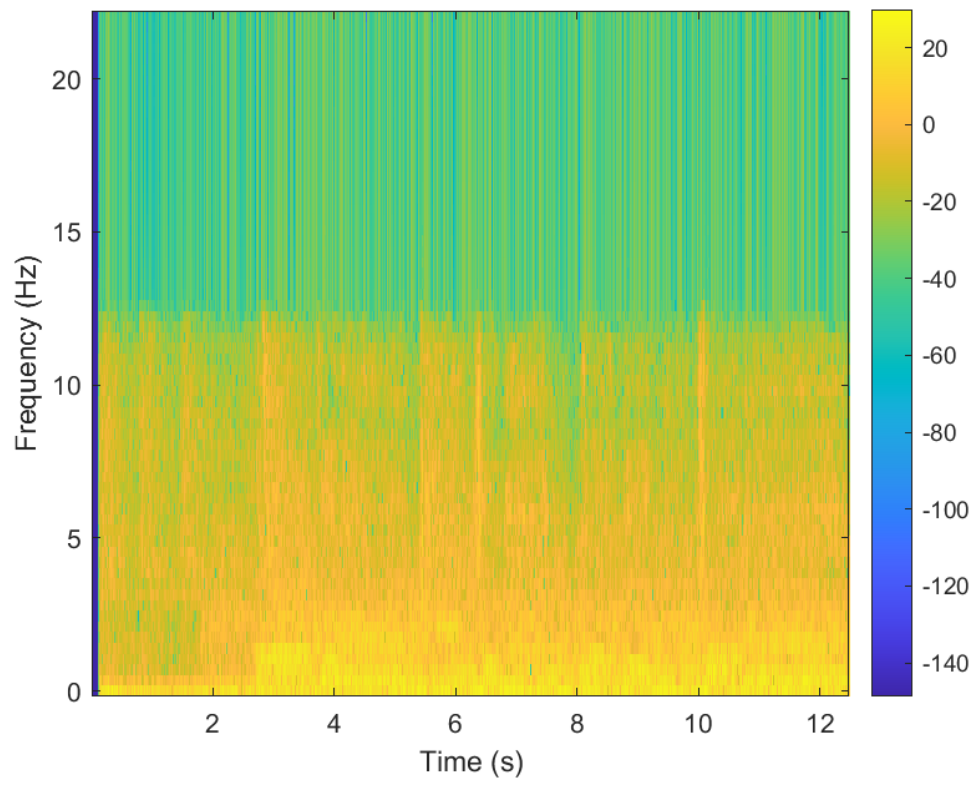
이후 STFT 결과를 제공하여 각 주파수에서의 에너지를 구할 수 있고, 밑이 10인 로그를 취하고 20을 곱하여 데시벨(dB) 단위로도 나타낼 수 있다.

(f)

길이 128의 Hamming Window를 적용하고 Overlap Length를 120으로 적용하여 2.(d)과 동일한 음원을 STFT하였다. 이를 그래프로 나타낸 모습은 아래와 같다.



(d)에서 사용한 matlab 내장 spectrogram과 비슷한 형태를 유도하기 위해 STFT 결과에 대해 밑이 10인 로그를 취하고 20을 곱하여 다시 그래프를 그려 보았다.



비슷한 형태를 얻었다.

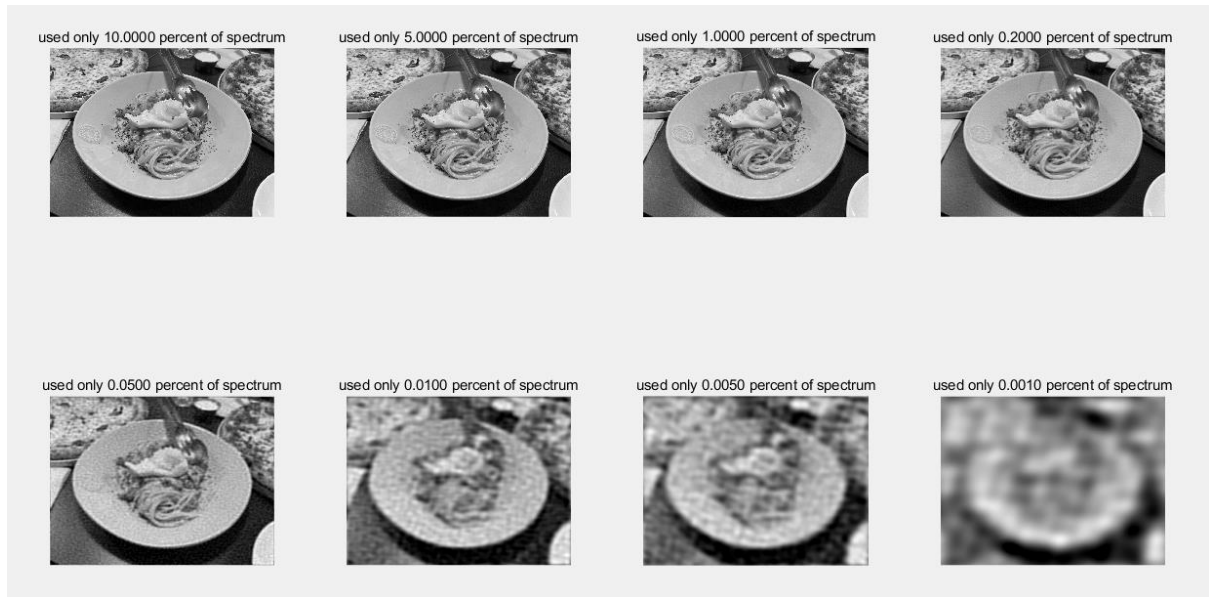
3.

원본 이미지

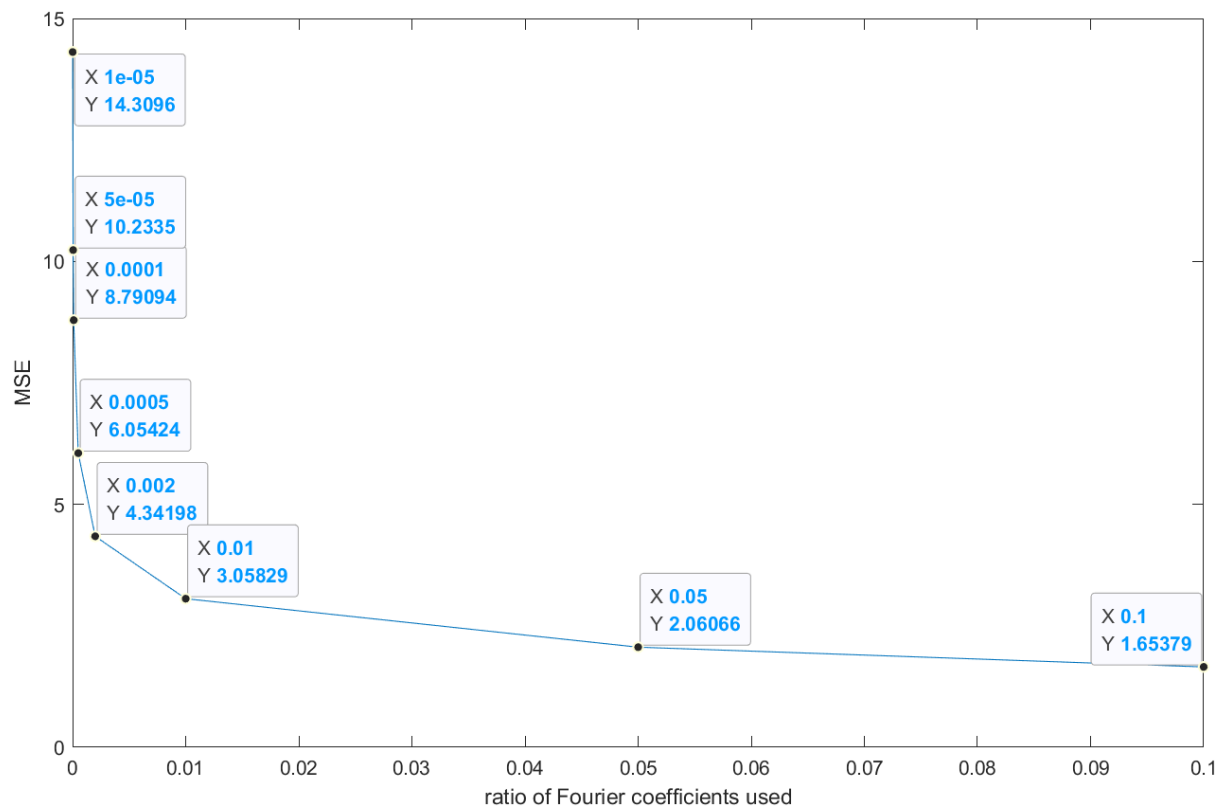
original image



처리 이미지



MSE



Keep 값이 커질수록 Mean Square Error가 커졌는데, 대략 0.01 이하부터 MSE가 급격하게 커지기 시작하였다.

이미지를 육안으로 판별하였을 때는 keep 값 0.1에서 0.01까지는 27인치 모니터에서 (2, 2) subplot을 전체화면으로 보았을 때 유의미한 차이를 확인할 수 없었다. 즉 Keep 값 0.01까지는 일반적인 환경에서 이미지를 보는 이로 하여금 위화감을 주지 않는 적절한 수준이라고 할 수 있다. 하지만 keep 값이 이보다 커짐에 따라 이미지에 노이즈가 느껴지기 시작해서 세밀한 표현이나 경계 표현이 매우 뭉개지기 시작했다. Keep값 0.0001부터는 구체적인 형태 표현의 수준도 낮아지기 시작했으며 keep 값 0.00001부터는 이미지의 내용물을 인식할 수 없게 되었다.

즉 사용하는 스펙트럼의 범위를 좁힘에 따라 이미지의 세부 표현의 정도가 낮아지기 시작하며, 어느 정도까지의 범위 내에서는 큰 변화를 보여주지 않지만 일정 수준 이하로 좁히게 되면 이미지 자체의 표현력을 잃어버리기에 이른다.