T.C.

İSTANBUL SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ

MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

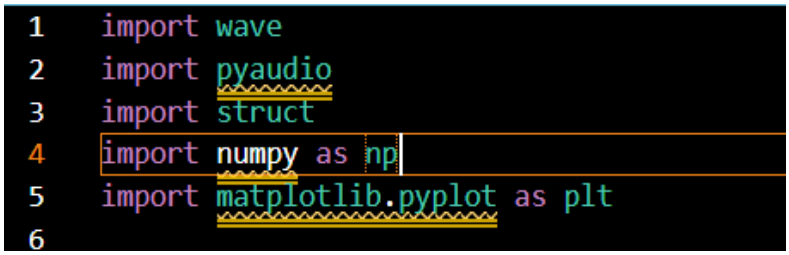
BIL206 – İŞARETLER VE SİSTEMLER DERSİ

ÖDEV 5- KONUŞMA SESİNDE SESSİZ BÖLGELERİN OTOMATİK OLARAK KIRPILMASI

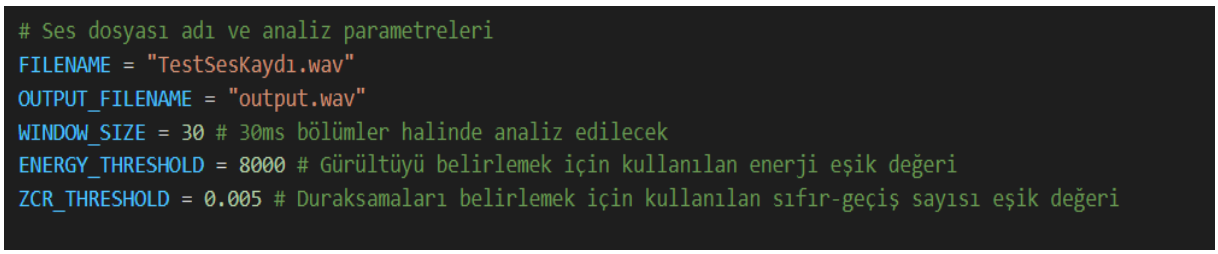
İbrahim Serhat Aktaş - 210601020
Kutay Can Batur - 210601009
Mert Tosun - 210601027

ÖĞRETİM GÖREVLİSİ: Prof, Dr. Halis Altun

Bu ödevde ses kayıtlarının yazdığımız kodlar tarafından analiz edilip belirli bir enerji altında kalan seslerin kırpılması ve kaydedilmesi aynı zamanda bu işlemlerin grafiklerle gösterilmesini amaçladık ve ödev sonunda amaçladığımız şeyleri yaparak ödevi tamamladık.



Ödev için gerekli kodları yazarken yukarıdaki kütüphanelerden yararlandık.



Sesin analizini yaparken 30ms'lik bölümlere ayırdık ve analiz kısmını bu şekilde gerçekleştirdik.

Kodda gözüktüğü gibi enerji eşik değerini 8000 olarak belirledik ve duraksamalar için eşik değeri 0.005 olarak koda yazdık.

```
# Bölümü belirleyin: konuşma veya gürültü/duraksama
if energy >= ENERGY_THRESHOLD and zcr >= ZCR_THRESHOLD:
    label = "konuşma"

    # konuşma bçlümünü yeni ses dosyasına yaz
    output_wf.writeframes(data)

    # temp değişkeni, ses bölümleri arasındaki dalgalanmayı azaltmak için kullandık.
    # konuşma olarak algılnan bir bölümün hemen sonrasındaki bölümü sessizlik olsa bile konuşma olarak # kaydediyoruz.
    # oynatma hızını normalleştirmek için bunu ekledik.
    temp = 1

else:
    if temp > 0:
        label = "konuşma"

        # konuşma bçlümünü yeni ses dosyasına yaz
        output_wf.writeframes(data)
        temp -= 1
    else:
        label = "gürültü/duraksama"
```

Burada da bir if döngüsü kullanarak dosyadaki sesin analizini yaptık eğer ses belirli bir enerji değerinin altındaysa o kısımların farklı bir listede tutulmasını sağladık.

```
    # Enerji grafiğini çiz
    ax[1].set_ylim([0, np.max(energies) * 1.1])
    time_points = np.linspace(0, num_frames / sample_rate, len(energies))
    ax[1].plot(time_points, energies, color='blue')

    # ZCR grafiğini çiz
    ax[2].set_ylim([0, np.max(zcrs) * 1.1])
    ax[2].plot(time_points, zcrs, color='green')
```

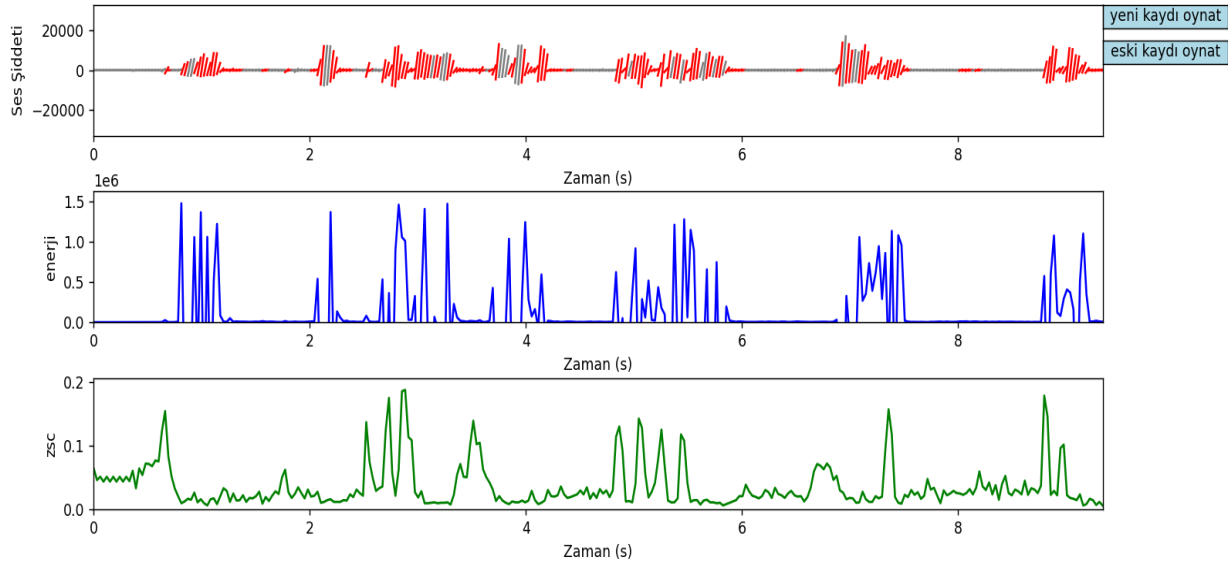
Daha sonra analizini yaptığımız bu sesin grafiklerini yukarıdaki kodu kullanarak oluşturduk.

```
# Enerji hesapla
energy = np.sum(signal ** 2) / len(signal)

# Sıfır-geçiş sayısını hesapla
zcr = np.sum(np.abs(np.diff(np.sign(signal)))) / (2 * len(signal))
```

Ses dosyasındaki sıfır geçiş sayılarını yani sessiz noktaları ve enerjiyi hesaplarken yukarıdaki iki kod bloğunu kullandık.

```python
# Çalma butonunu oluştur
play_button_ax = plt.axes([0.90, 0.91, 0.1, 0.04])
play_button = Button(play_button_ax, 'yeni kaydı oynat', color='lightblue', hovercolor='skyblue')

# Çalma butonunu oluştur
play_button2_ax = plt.axes([0.90, 0.85, 0.1, 0.04])
play_button2 = Button(play_button2_ax, 'eski kaydı oynat', color='lightblue', hovercolor='skyblue')
```

Yukarıdaki kodlarla iki adet buton oluşturarak ilk olarak ses dosyasının ilk halini oynatabilir diğer buton sayesinde revize edilmiş sesi dinleyebiliriz.



Yukarıda kodumuzun ekran çıktısında gördüğümüz grafikler gözüküyor. İlk grafik genel ses şiddeti grafiği ikinci grafik enerjinin nerelerde artıp azaldığını gösteriyor üçüncü grafikte ise sessizlik eşik değer grafiğini görüyoruz. Sağ taraftaki butonlarda önceki kısımda bahsetmiş olduğum ilk ses ve revize edilmiş sesin oynatılması için eklediğimiz butonları görüyoruz.