

# ELM368 – DÖNEM PROJESİ

## ÇOCUK VE YETİŞKİN SESİ AYIRT ETME

Senanur Ağaç, Selen Erdoğan  
1901022038, 200102002043  
selenerdogan2019@gtu.edu.tr, s.agac2020@gtu.edu.tr

### ABSTRACT (ÖZET)

Proje konumuz çocuk ve yetişkin sesi ayırt etmedir. Bu konuyu ele almamızdaki amaçlar şunlardır: Bir çocuk kaçırıldığında veya kaybolduğunda, yetişkinlerin arama kurtarma ekipleri, çocuğun sesini diğer seslerden ayırt etmek için bu teknolojiye yararlanabilirler. Ayrıca, çocuk ve yetişkin seslerinin ayırt edilmesi, telefon görüşmeleri veya sesli asistanlar gibi uygulamalarda, kullanıcıların ses tanıma işlevlerini geliştirmek için de kullanılabilir.

### ANAHTAR KELİMELER

FIR Filtre, Spektrum Analizi, Frekans Analizi, Yüksek Geçiren Filtre, Bant Geçiren Filtre, Ses Ayırımı, Spectrogram, Fourier Dönüşümü, Genlik Spektrumu

## 1. Giriş

Çocukların sesleri yetişkinlere kıyasla daha yüksek frekansta ve tonlamaya sahiptir. Bu projede, bu özelliklerden faydalanarak, verilen örnek ses dosyalarının çocuklara mı yoksa yetişkinlere mi ait olduğu, filtreleme ve analiz işlemleri ile belirlenecektir. Seslerin frekans bileşenlerine ve genliklerine bakılarak, seslerin çocuk ya da yetişkin kaynaklı olduğu tespit edilecektir. Filtreler ve Fourier analizi, seslerin frekans spektrumlarından gelen bilgiyi kullanarak, seslerin çocuk mı yoksa yetişkin mi olduğunu belirlemek için kullanılacaktır.

## 2. Deneyler ve Analiz

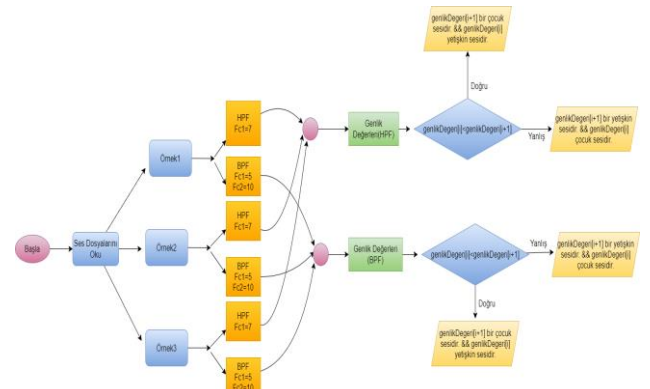
Örnek ses dosyalarının fourier dönüşümleri alınıp frekans analizleri yapılmıştır. Genlik spektrumları ve bu spektrumların filtreden geçirilmiş halleri incelenmiştir. Tüm örnekler filtrelerden geçirilip sonuçlar ortaya konulmuştur. Filtrelenmiş ve filtrelenmemiş veriler, grafikler ve ortalama değerler karşılaştırılmıştır. Analizlerden elde edilen veriler doğrultusunda, verilen bir ses dosyasının yetişkinlere mi çocuklara mı ait olduğunu tespit edebilen bir algoritma oluşturulmuştur.

Çözüm sürecinde ilk adım, çocuk ve yetişkin seslerinin ayırt edici özelliklerinin belirlenmesiydi. Bu farkların belirlenmesinin ardından, bu ayırt edici özelliklerin verilen ses örneklerinden nasıl çıkarılabileceği üzerine araştırmalar yapıldı. Ses analizleri için uygun örnekler

seçildi ve analizler yapıldı. Kuramsal düzeyde oluşturulan fikirler pratiğe dökülerek sonuçlar gözlemlendi. Bu gözlemler ve elde edilen veriler ışığında, çocuk ve yetişkin seslerini ayırt edebilen bir algoritma geliştirildi. Veriler el ile yeniden incelendi ve bu eldeki incelemeler, algoritmanın ürettiği sonuçlarla karşılaştırıldı. Bu şekilde, algoritmanın çocuk ve yetişkin seslerini başarıyla ayırt edip edemediği değerlendirildi. Farklı kütüphane yapıları projeye uygun olarak kullanıldı. 'librosa', 'numpy', 'scipy.io.wavfile', ve 'matplotlib.pyplot' Python kütüphaneleridir ve ses işleme, veri işleme ve veri görselleştirme gibi işlemleri gerçekleştirebilmek için kullanıldı. 'librosa': Bu kütüphane, ses ve müzik analizi için kullanıldı. 'numpy': Python'da bilimsel hesaplamaları gerçekleştirmek için kullanılan bir kütüphanedir. Çok boyutlu dizi işlemleri, lineer cebir işlemleri, Fourier dönüşümü vb. birçok işlevi kullanabilmek amacıyla kullanıldı. 'scipy.io.wavfile': Bu modül, WAV ses dosyalarını okuma ve yazma işlevlerini sağlamak için kullanıldı. 'matplotlib.pyplot': Bu kütüphane, Python'da çeşitli türlerde grafikler ve figürler oluşturmak için ve spektrogramlar, frekans spektrumları ve diğer ses analizi grafiklerini oluşturmak için kullanılır.

- Sonuçlar beklenildiği gibi elde edildi.
- Uygulanan metodlar doğru bir şekilde sonuç verdi.

## 2.2 Tespit Metodu

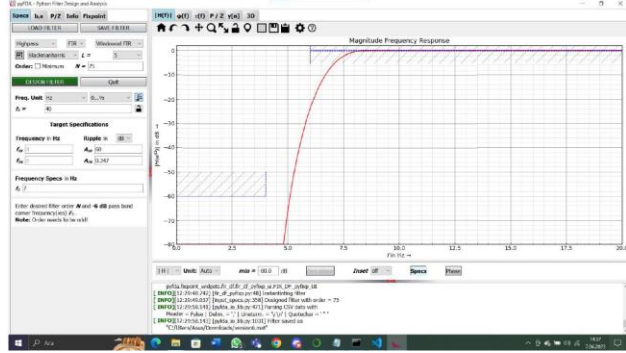


Şekil 1 Karar Algoritması Akış Diyagramı

Yapılan araştırmalara ve elde edilen örneklerin frekans spektrumlarının incelenmesi ile iki farklı filtrenin oluşturulmasına karar verilmiştir. Bu iki filtre sonucunda elde edilen değerler gerekli fonksiyon yapıları sayesinde koda dökülmüştür. Elde edilen magnitude değerleri kıyaslanarak karar algoritması başarılı bir şekilde oluşturulmuştur.

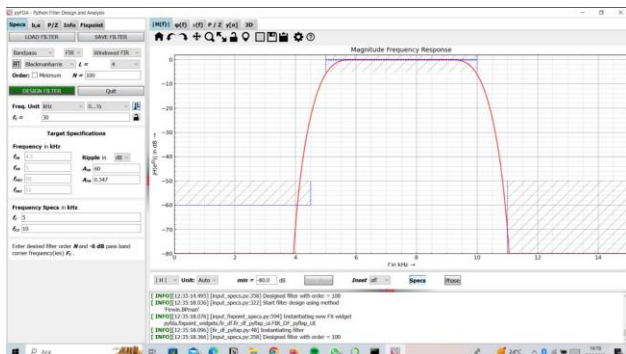
## 2.2 Uygun Filtre Oluşturma

Sistemde kullanılmak istenen Yüksek Geçiren ve Bant Geçiren Filtreler pyFDA programı ile tasarlanmıştır. Birinci filtremiz Yüksek Geçiren Filtredir. Şekil 2’de gösterilen FIR Yüksek Geçiren Filtre oluşturulurken Windowed FIR Blackmanharris Metodu kullanıldı. Kesim frekansı 7000 Hz olan bu filtre, metod seçimleri ripple oluşumlarına göre dikkatlice ayarlandı. Filtrenin gerilim değerine uygun olarak tasarım yapıldı. Durdurma bandı köşe frekansı Fsb değeri 4, geçirme bandı köşe frekans Fpb değeri 6 Hz olarak uygun görüldü. Filtrenin derecesi N=75 olmuştur.



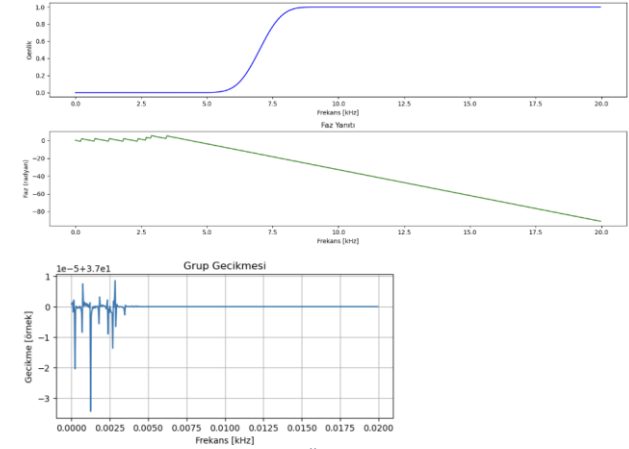
Şekil 2 Yüksek Geçiren Filtre PYFDA

FIR Bant Geçiren Filtre oluşturulurken Windowed FIR Blackmanharris Metodu kullanıldı. Şekil 3’de gösterilen alçak kesim frekansı 5000 Hz, yüksek kesim frekansı 10000 Hz olması istenilen bu filtreyi oluşturmak için bir önceki adımda da yapılan işlemlere dayanmıştır. Alçak durdurma bandı köşe frekansı FSB = 4.5 kHz, alçak geçirme bandı köşe frekansı FPB = 5 kHz, yüksek geçirme bandı köşe frekansı FPB2 = 10 kHz, yüksek durdurma bandı köşe frekansı FSB2 = 11 kHz olarak alınmıştır. Filtrenin derecesi N = 100 olmuştur.

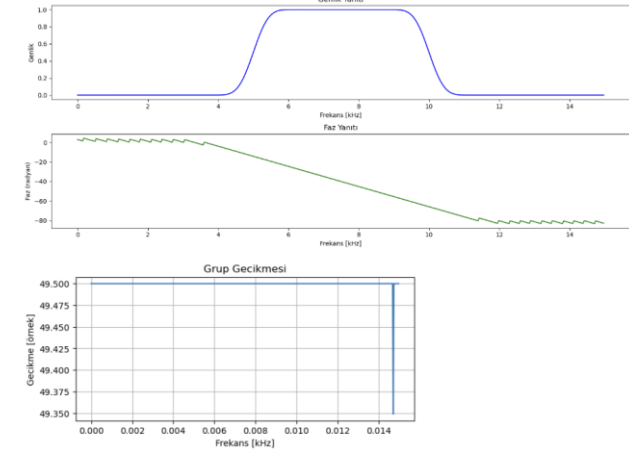


Şekil 3 Bant Geçiren Filtre PYFDA

Oluşturduğumuz filtrelerin özellikleri Şekil 4 ve Şekil 5’de gösterildiği gibidir. Veri setine uygun genlik yanıtı, faz yanıtı ve grup gecikmesi olan filtreler oluşturulmaya dikkat edilmiştir. Frekans değerleri spektrum grafiklerine göre seçilmiştir.



Şekil 4 Yüksek Geçiren Filtre Özellikleri

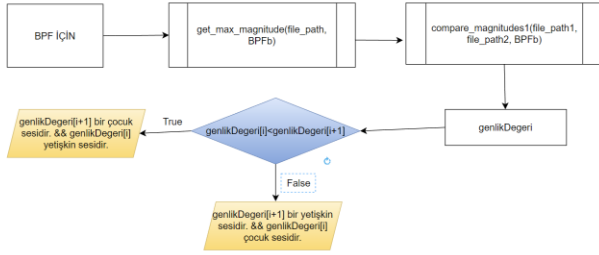


Şekil 5 Bant Geçiren Filtre Özellikleri

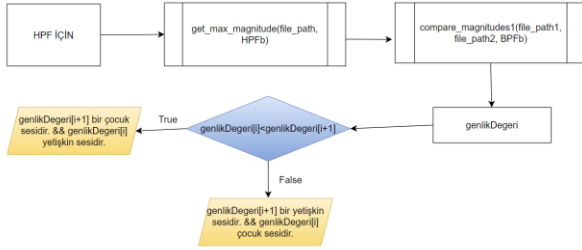
## 2.3 Karar Verme Fonksiyonları

Yukarıda da bahsedildiği gibi filtrelenen ses dalgalarının magnitude değerlerinde farklılık karar verme fonksiyonlarımızın temelini oluşturmuştur.

Yüksek geçiren filtre sonucunda elde edilen verileri anlamak için yazılan kod, verilen ses dosyalarının üst geçiren bir filtre uygulanmış hâllerindeki genlik değerlerinin maksimumlarını hesaplar ve sonuçları ekrana yazdırır. Bu sayede girilen veri setlerinin ses analizi yapıp, hangi sesin çocuk mu yetişkin mi olduğunu karar verilir. Yüksek magnitude değerine sahip ses dosyası çocuk sesine, alçak magnitude değerine sahip ses dosyası yetişkin sesine ait olduğunu gösterir.



Şekil 6 Bant Geçiren Filtreden Elde Edilen Değerlerin İşlenme Algoritması



Şekil 7 Yüksek Geçiren Filtreden Elde Edilen Değerlerin İşlenme Algoritması

## 2.4 Karar Verme Algoritmasının Uygulanması

Şekil 6 ve Şekil7’de gösterilen fonksiyonlar sayesinde verilen veri setindeki yetişkin ve çocuk seslerinin Max Magnitude değerleri bulunmuştur ve compare\_magnitudes fonksiyonları sayesinde verilen ses dosyalarının çocuğa mı yetişkine mi ait olduğu anlaşılmıştır. Her iki filtre için de, özelliklerine uygun yazılan bu fonksiyonlar ses dosyalarını parametre olarak almıştır. Veri olarak filtrelenmiş işaretlerin max amplitud değerlerini döndüren bu fonksiyonlar Python dilinde yazılmış olup, verilen veri setlerinin ses analizini başarılı bir şekilde yapmaktadır.

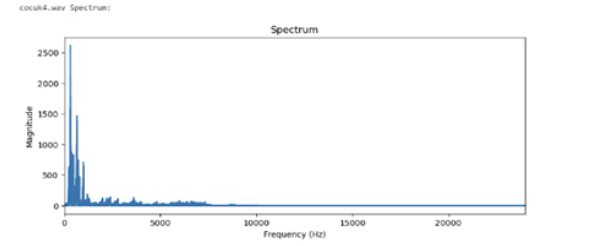
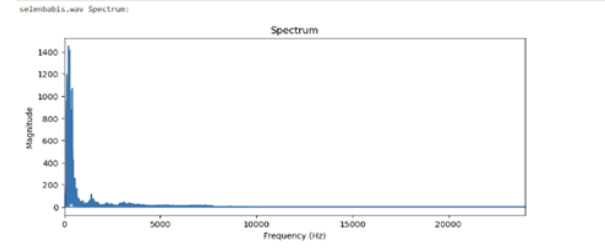
### 2.5 Uygulama

Python dilinde ve Jupyter Notebook ortamında, ses işleme, analiz ve uygulamaları gerçekleştirmek için kullanılan çeşitli kütüphaneler bulunur. Bu uygulama dosyasında, numpy, matplotlib, pyplot, librosa ve scipy gibi kütüphaneler kullanılarak ses işleme işlemleri gerçekleştirilmiştir. İşlemler, önceden tanımlanmış değişkenler ve pyFDA aracılığıyla oluşturulan filtrelerin katsayılarıyla yapılmıştır. Ses sinyalleri, zaman alanında grafiklendirilmiş ve Fourier dönüşümü alınarak frekans spektrumları ve spektrogram grafikleri oluşturulmuştur.

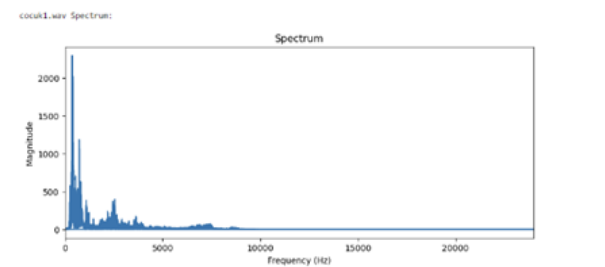
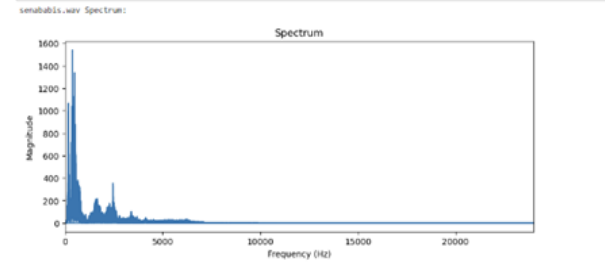
### 2.1 Grafik, Tablo ve Şekiller

Şekil 8-9-10’da gösterilen grafikler, ses sinyalinin frekans bileşenlerini gösteren bir spektrum grafiğini temsil ediyor. Spektrum grafiği, frekansın yatay ekseninde ve genişliğin dikey ekseninde gösterildiği bir grafikdir. Bu grafik, sesin frekans içeriği hakkında bilgi verir.

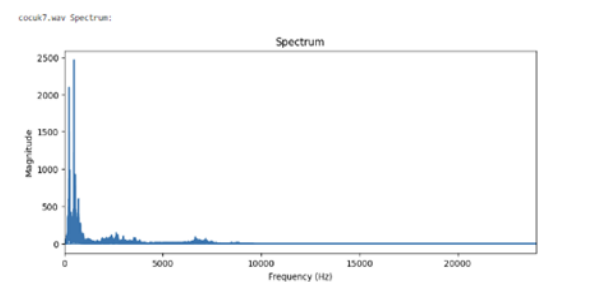
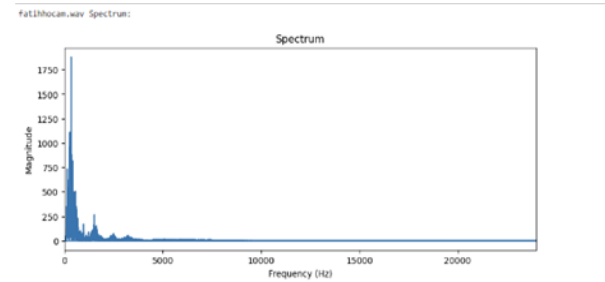
## Spektrum Grafikleri



Şekil 8 1.Veri Setinin Spektrum Grafiği

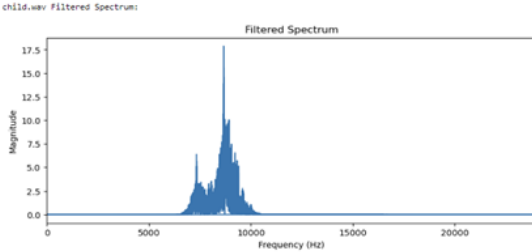
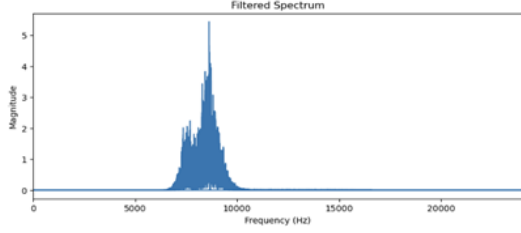


Şekil 9 2.Veri Setinin Spektrum Grafiği



Şekil 10 3.Veri Setinin Spektrum Grafiği

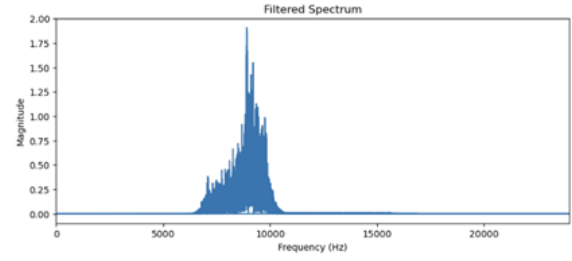
## 2.1.1 Filtrelenmiş Veri Setleri



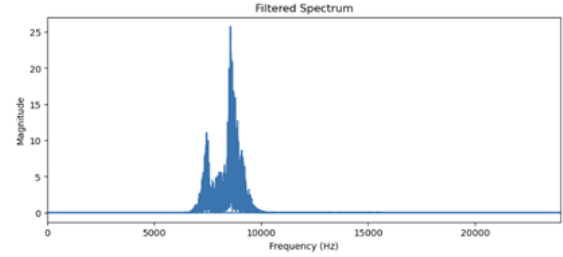
adult.wav Max Magnitude for BPF: 5.43465526955387  
child.wav Max Magnitude for BPF: 17.867929434301463

Şekil 11 1.Veri Setinin Bant Geçiren Filtre ile Filtrelenmesi

adult.wav Filtered Spectrum:



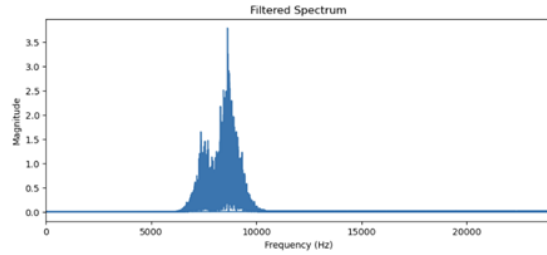
child.wav Filtered Spectrum:



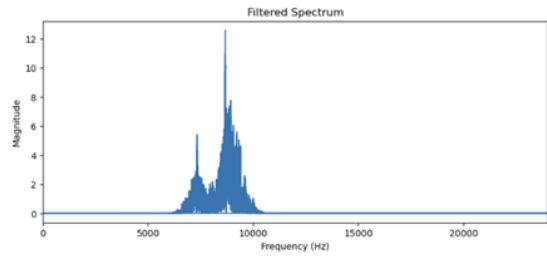
adult.wav Max Magnitude for BPF: 1.9079104514146747  
child.wav Max Magnitude for BPF: 25.77121050034731

Şekil 13 2.Veri Setinin Bant Geçiren Filtre ile Filtrelenmesi

adult.wav Filtered Spectrum:



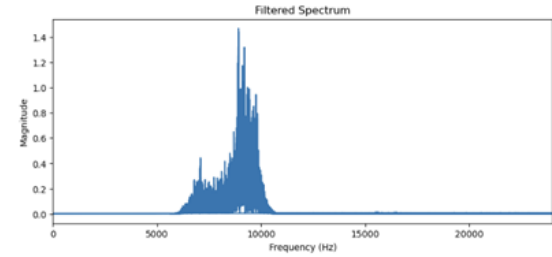
child.wav Filtered Spectrum:



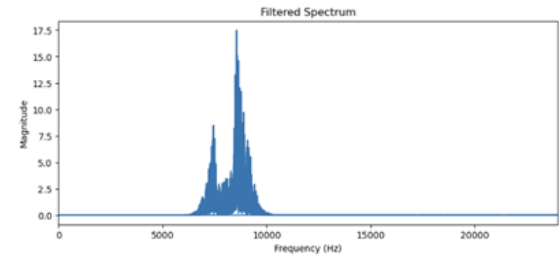
adult.wav Max Magnitude for HPF: 3.781746085587157  
child.wav Max Magnitude for HPF: 12.5885068193809

Şekil 12 1.Veri Setinin Yüksek Geçiren Filtre ile Filtrelenmesi

adult.wav Filtered Spectrum:

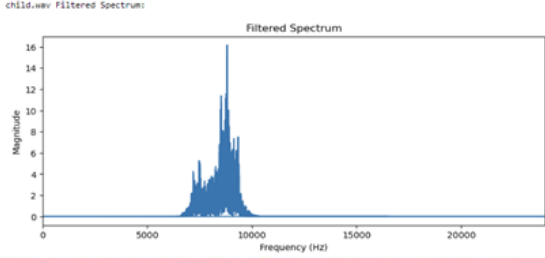
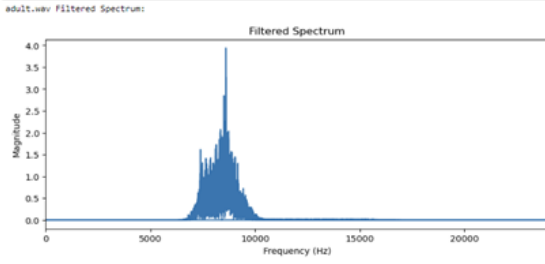


child.wav Filtered Spectrum:



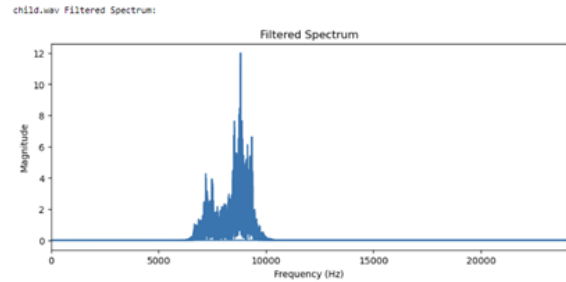
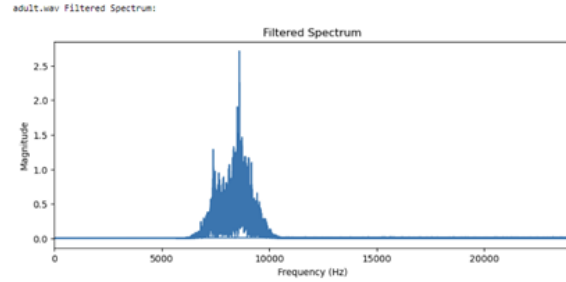
adult.wav Max Magnitude for HPF: 1.4688905010560374  
child.wav Max Magnitude for HPF: 17.48098771451666

Şekil 14 2.Veri Setinin Yüksek Geçiren Filtre ile Filtrelenmesi



adult.wav Max Magnitude for BPF: 3.939065095928071  
child.wav Max Magnitude for BPF: 16.164897195650784

Şekil 15 3. Veri Setinin Bant Geçiren Filtre ile Filtrelenmesi



adult.wav Max Magnitude for HPF: 2.713324918779853  
child.wav Max Magnitude for HPF: 12.00153898741709

Şekil 16 3. Veri Setinin Yüksek Geçiren Filtre ile Filtrelenmesi

Şekil 11-12-13-14-15-16' da görüleceği üzere filtrelenmiş seslerin genlik spektrumları istenilen sonuçları vermektedir. Çocuk seslerinin gereken frekans aralıklarında daha büyük genlik değerlerine sahip olduğu görülmekteyken, yetişkin seslerin gereken frekans aralıklarında daha küçük genlik değerlerine sahip olduğu görülür. Karar mekanizmaları da bu yapıya göre oluşturulmuştur. Elde edilen çıktılar Şekil 17-18-19-20'de gösterildiği gibidir.

## Karar Mekanizmaları

```
In [84]: compare_magnitudes1('senababis.wav', 'a1.wav', HPFb)
The higher magnitude is for a1.wav, this indicates a child voice.
The lower magnitude is for senababis.wav, this indicates an adult voice.
```

Şekil 17 2. Veri seti için HPF karar mekanizması

```
In [86]: compare_magnitudes('senababis.wav', 'a1.wav', BPFb)
The higher magnitude is for a1.wav, this indicates a child voice.
The lower magnitude is for senababis.wav, this indicates an adult voice.
```

Şekil 18 2. Veri seti için BPF karar mekanizması

```
In [88]: compare_magnitudes1('fatihhocam.wav', 'a7.wav', HPFb)
The higher magnitude is for a7.wav, this indicates a child voice.
The lower magnitude is for fatihhocam.wav, this indicates an adult voice.
```

Şekil 19 3. Veri seti için HPF karar mekanizması

```
In [90]: compare_magnitudes('fatihhocam.wav', 'a7.wav', BPFb)
The higher magnitude is for a7.wav, this indicates a child voice.
The lower magnitude is for fatihhocam.wav, this indicates an adult voice.
```

Şekil 20 3. Veri seti için BPF karar mekanizması

## 3. Sonuç ve Yorum

Bu projede amaç çocuk ve yetişkin seslerini birbirinden ayırabilmektir. Buradaki amaç bir kaybolma durumunda ekiplerin çocuk ve yetişkin seslerini ayırabilecek bir sisteme sahip olmalarıdır. Ayrıca bu proje telefon görüşmeleri veya sesli asistanlar gibi uygulamalarda, kullanıcıların ses tanıma işlevlerini geliştirmek için de kullanılabilir. Bu doğrultuda geliştirilen algoritma, çocuk ve yetişkin seslerini başarıyla ayırt edebilmektedir. Ses analizleri ve filtreleme işlemleri sonucunda elde edilen veriler, algoritmanın çocuk sesleri ile yetişkin seslerini doğru bir şekilde sınıflandırdığını göstermektedir. Frekans analizleri yapılarak ses dosyalarının genlik spektrumları incelenmiştir. Çocuk sesleri genellikle daha yüksek frekans bileşenlerine sahipken, yetişkin sesleri daha düşük frekanslarda yoğunlaştığı görülmüş ve bu farklılık kullanılarak bir algoritma tasarlanmıştır. Tasarlanan algoritma doğru bir şekilde çocuk ve yetişkin seslerini ayırt etmektedir. Filtreleme işlemleri ile filtrelenmiş seslerin frekans spektrumlarını daha belirgin hale getirmiş ve çocuk seslerinin karakteristik tonlamasını ortaya çıkarmıştır. Filtrelenmiş verilerin analizi göz önüne alındığında algoritmanın çocuk seslerini daha iyi tanımlayabildiğini göstermektedir. Ayrıca yapılan incelemeler ve karşılaştırmalar, algoritmanın doğruluk oranının yüksek olduğunu ve çocuk ve yetişkin seslerini başarıyla ayırt ettiğini göstermektedir. Ancak, bazı durumlarda yanlış sınıflandırmaların olabileceği de gözlemlenmiştir. Bu durumlar, daha karmaşık örnekler ve benzer frekans özelliklerine sahip seslerin varlığından kaynaklanabilir. Ek olarak daha uzun ses dosyasında algoritmanın başarılı olma durumu daha net gözükmemektedir. Sonuç olarak, geliştirilen algoritma çocuk ve yetişkin seslerini ayırt etme konusunda olumlu

sonuçlar vermektedir. Ancak, daha fazla veri seti ve farklı koşullar altında test edilerek algoritmanın performansının daha da iyileştirilmesi mümkündür. Bu proje sayesinde

sinyal işlemenin temelleri öğrenilmiştir. Ses analizi ve işleme konusunda deneyim kazanılmıştır. Ses özelliklerinin çıkarılması ve sınıflandırma yöntemleri üzerine çalışılmıştır. Çocuk ve yetişkin sesi arasındaki farklılıkları belirlemek için özelliklerin kullanılması denenmiştir.

Veri setinin oluşturulması, etiketlenmesi ve model eğitimi gibi aşamalar gerçekleştirilmiştir. Veri setinin kalitesi ve çeşitliliği, çözümün performansını etkileyebilir. Çözümü geliştirmek için daha fazla ve çeşitli veri toplanabilir ve kullanılabilir. Farklı ses özellikleri veya özellik kombinasyonları denenebilir.

Daha gelişmiş ve karmaşık modeller kullanılabilir.

Farklı filtreler, ön işleme yöntemleri veya öznetelik seçimi teknikleri denenebilir. Ses analizi ve işleme konusunda pratik deneyim kazanılmıştır.

Veri seti hazırlama ve etiketleme süreçleri üzerinde çalışılmıştır. Çocuk ve yetişkin sesinin analizinde karşılaşılan zorluklar ve stratejiler hakkında bilgi edinilmiştir.

## Kaynaklar

[1] <https://www.youtube.com/watch?v=sTFGBM1gvTU>

[2] <https://tomverbeure.github.io/2020/10/11/Designing-Generic-FIR-Filters-with-pyFDA-and-Numpy.html>

[3] <https://matplotlib.org/>

[4] <https://librosa.org/doc/latest/index.html>