

Dönem Sonu Projesi-Gürültü Giderme ve Ses Temizleme Sistemi Raporu

Hazırlayan : ÖMER FARUK DERİN - 22060388

GitHub Linki :

https://github.com/omerfderin/Sinyaller_Sistemler_Proje

Kullanılan Dil : Python

1- Giriş

1.1 Projenin Amacı

Projenin amacı günlük yaşamda ses kayıtlarında ya da gerçek zamanlı ses uygulamalarında yer alan gürültünün FFT kullanılarak analiz edilmesi, engellenmesi ve sesin temizlenmesidir.

1.2 Projenin Kapsamı

Belirlenen proje amacına göre aşağıdaki hedefler tarafımda belirlenmiştir.

- 1. Gürültü Tanımlama : Ortamdaki gürültünün ses profilini belirlemek.**

- 2. Ses Analizi :** Sesin frekans ve genlik özelliklerini incelemek.
- 3. Gürültü Giderme :** Belirlenen gürültüyü ses sinyalinden ayırmak.

2 - Yöntem

2.1 Kullanılan Teknikler ve Algoritmalar

2.1.1 Fast Fourier Transform (FFT)

Bu projede ses sinyalinin zaman domaininden frekans domainine dönüşümünü sağlamak amacıyla manuel FFT methodları kullanılmıştır. Bu methodlar böl ve yönet mantığında çalışmaktadır.

2.1.2 Inverse Fast Fourier Transform (IFFT)

Frekans domainindeki işlenmiş sinyali tekrar zaman domainine dönüştürmek için IFFT kullanılmıştır.

2.1.3 Hann Penceresi

Çerçeveler birleştirildiği sırada aralarda keskin geçişler olabilmektedir, bu pencere çerçeveler arası geçişte yumuşak geçişi sağlamak ve ses kayıplarını önlemek amacıyla kullanılmıştır.

Hann Penceresi = $x[n] = 0.5 * (1 - \cos(2\pi*n/(N-1)))$

2.1.4 Wiener Filtreleme

Bu filtre, sinyal-gürültü oranına göre her frekans bileşeni için en iyi kazancı hesaplar. Bu sayede gürültüyü minimuma indirirken orijinal sinyal mümkün olduğunca korunacaktır.

Kazanç Formülü = $H[k] = (|S[k]|^2 / (|S[k]|^2 + \alpha^2 \cdot |N[k]|^2))^\beta$

α = Gürültü azaltma faktörü β = Minimum kazanç değeri

Bu değerler üzerinde oynama yaparak gürültü engellemede fayda sağlanabilir.

2.1.5 Spektral Çıkarma Yöntemi

Gürültü profilini çıkarmak için spektral çıkarma tekniği kullanılmıştır. Bu yöntem, gürültülü sinyalden gürültü örneğinin spektrumunu çıkararak temiz sinyali elde etmeyi amaçlar.

3 - Uygulama

3.1. Ses Kayıtlarının Alınması

Ses kayıtları sessiz, kalabalık ortam ve otoyol kenarında olmak üzere 4 farklı ortamda 5 kayıt olacak şekilde telefon yardımıyla alınmıştır. Ses kayıtlarının orijinal halini aşağıdaki linklerde bulabilirsiniz.

Sessiz -> [Sessiz.wav](#)

Kalabalık Ortam -> [Kalabalık.wav](#)

Kalabalık Ortam 2 -> [Kalabalık2.wav](#)

Otoyol Kenarı -> [Otoyol.wav](#)

Yemekhane -> [Yemekhane.wav](#)

Gerçek zamanlı filtrelemede ise sounddevice kütüphanesi yardımıyla giriş sesi çıkış aygıtına verilmiştir bu sayede anlık olarak ses duyulmaktadır.

3.2 Gürültü Analizi ve Temizliği

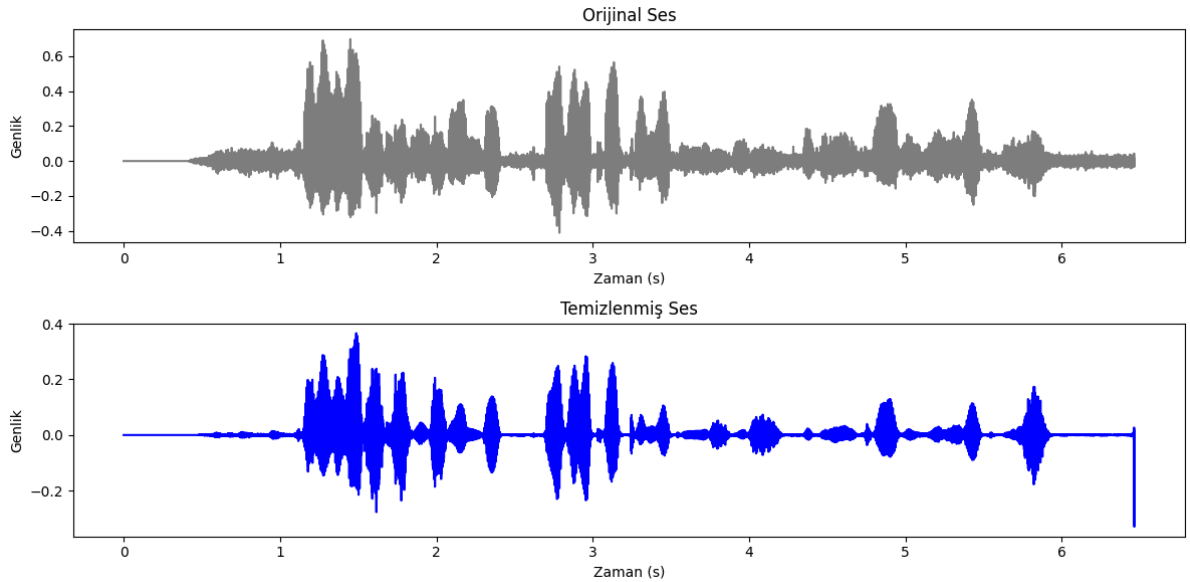
3.2.1 Ses Dosyasındaki Gürültünün Giderilmesi ve Sesin Temizlenmesi

Kod çalıştırıldığında öncelikle kullanıcıdan temizlenecek ses dosyasının ismi alınır ve sonrasında bu dosya **noise_filter()** methoduna verilir. Bu methodda öncelikle ses dosyasının hangi frame boyutlarında işleneceği 2'nin kuvvetleri olacak şekilde belirlenir (FFT bu şekilde daha iyi çalışıyor).

Sonrasında ise ses dosyası okunur, ses verisi ve sample rate alınır sonrasında ise bu ses verisine normalleştirme uygulanır. Uygulanan normalleştirme sonrasında en başta belirlediğimiz çerçeve boyutunda Hann penceresi oluşturulur. Daha sonra gürültüyü belirlemede kullanacağımız gürültü başlangıç ve bitiş sample değerlerini belirliyoruz ve boş bir gürültü profili dizisi oluşturuyoruz. Daha sonra ise belirlediğimiz çerçeve boyutunda atlama boyutunu da kullanarak bu çerçevenin gücünü bir diziye kaydediyoruz aynı zamanda işlenen çerçeve sayısını da kaydediyoruz. Hesaplanan her çerçevenin gücü için bu çerçevelerin gürültü profilini çıkarıyoruz. Wiener filtresinde gürültü engellemede büyük rol oynayan alpha, beta değerlerini belirliyoruz. Sonrasında her çerçeveye tekrardan Hann penceresi uyguluyoruz ve sonrasında FFT ile frekans uzayına geçiyoruz ardından ise sinyalin genlik, faz ve güç değerlerini

elde ediyoruz, faz deęerini korumak bizim iin nemli o yzden saklıyoruz. Daha sonra ise nceden belirledięimiz grlt profilindeki grlty temsil eden indeksleri buluyoruz (pozitif olanları) ve sonrasında bu sinyalin bu indekslerindeki grltnn gcn ve bu grltden elde edebileceęimiz kazanç deęerini hesaplıyoruz bu sayede yeni temizlenmiř sinyalde kullanacaęımız yeni genlikleri buluyoruz ve bu genlikleri her erevenin uygun indexlerine uyguluyoruz. Sonrasında ise IFFT kullanarak tekrardan zaman uzayına geiyoruz ve sinyale tekrardan Hann penceresi uyguluyoruz ve sesimiz grltden byk oranda arındırılmıř oluyor. Bu iřlemden sonra ise ıkıř dosyasındaki sesin normalleřtirilmesi saęlanıyor. Daha sonra ise temizlenmiř sesin kaydedileceęi dosyaya temiz ses profili ereve ereve kaydediliyor bylece iřlemimiz sonlanmıř oluyor.

rnek karřılařtırma



Gürültüsü Giderilmiş Ses Kayıtları

Sessiz Temiz -> [Sessiz_temiz.wav](#)

Kalabalık Temiz -> [Kalabalık_temiz.wav](#)

Kalabalık 2 Temiz -> [Kalabalık2_temiz.wav](#)

Otoyol Temiz.wav -> [Otoyol_temiz.wav](#)

Yemekhane Temiz.wav -> [Yemekhane_temiz.wav](#)

3.2.2 Gerçek Zamanlı Gürültü Engelleme ve Sesin Temizlenmesi

Kod çalıştırıldığında gürültü profili kaydı için **record_noise()** methodu belirlenen adaptasyon süresi boyunca ortam gürültüsünü analiz eder. Bu süreçte belirlenen çerçeve boyutunda ses verilerini okuyoruz ve her bir çerçeveye Hann penceresi uyguluyoruz. Sonrasında FFT algoritması kullanarak zaman uzayından frekans uzayına geçiş yapılıyor ve spektral genlik değerleri hesaplanıyor. Toplanan tüm gürültü çerçevelerinin medyan değeri alınarak ortamın gürültü profili elde ediliyor ve bu profil güvenlik faktörü ile çarpılarak güvenlik payı sağlanıyor. Gürültü profili çıkarıldıktan sonra sistem gerçek zamanlı işleme moduna geçiyor ve **process_audio()** methodu çalışmaya başlıyor. Ardından mikrofondan gelen ses sinyalinin sürekli olarak işlemeye başlıyoruz.

Gelen ses verilerini öncelikle bir giriş tamponunda biriktiriyoruz ve tampon dolduktan sonra belirlenen atlama boyutu kullanılarak örtüşen çerçeveler halinde işleme alıyoruz. Her ses çerçevesine tekrardan Hann penceresi uyguladıktan sonra FFT ile frekans uzayına geçiyoruz ve spektral genlik ile faz bilgilerini ayrı ayrı hesaplıyoruz, faz bilgisi ses kalitesi için

korunuyor. Daha sonra frekans spektrumunu düşük, orta ve yüksek frekans olmak üzere üç farklı banda ayırıyoruz ve her banda farklı gürültü azaltma faktörleri uyguluyoruz. Bu bantlarda spektral çıkarma yöntemi kullanarak gürültü profilindeki enerji miktarını orijinal sinyalden çıkarıyoruz ve spektral taban belirleyerek sinyalin tamamen kaybolmasını engelliyoruz. Aynı zamanda kazanç değerleri bir önceki çerçeve ile belirlenen yumuşatma oranında yumuşatılarak ani değişimler önleniyor. Ardından temizlenmiş spektral verileri IFFT ile tekrar zaman uzayına dönüştürüyoruz ve çıkış sinyaline tekrardan Hann penceresi uygulayarak çerçeveler arası geçişleri düzeltiyoruz. Sonrasında örtüşen çerçeveleri birleştiriyor ve ağırlıklı ortalamasını alıyoruz. Çıkış sinyali belirlenen çıkış kazancı ile yükseltildikten sonra (gürültü engelleme bir miktar da olsa konuşanın sesini azaltabiliyor) belirlenen eşik değerinin altındaki zayıf sinyaller büyük oranda azaltılarak artık gürültü minimize ediliyor. Ardından sinyal normalize edilerek çıkış cihazına gönderiliyor ve bu işlem döngüsü gerçek zamanlı olarak sürekli tekrarlanarak mikrofondan gelen ses anında temizlenerek çıkışa aktarılıyor. Son olarak ise gerçek zamanlı uygulamada elde edilen sonucun kaydedileceği dosyanın kullanıcı tarafından girilmesi isteniyor ve böylece işlemimiz son bulmuş oluyor.

Normal Ses Kaydı -> [gercek_zaman_normal.wav](#)

Gerçek Zamalı Gürültü Engelleme Uygulanmış Kayıt -> [gercek_zamanlı.wav](#)

4 - Sonular

Ses dosyası zerinden gerekleřtirilen grlt engelleme ve ses temizleme sistemi genel olarak grlty byk oranda engelleyebiliyor ayrıca ses dosyasını iřleme sresi de bana kalırsa kabul edilebilir bir dzeyde. Fakat bu sistemin kt yanlarından bahsedecek olursak yksek bass ieren grltleri engellemede biraz zayıf kaldıėını dřnyorum ayrıca uygulamanın alıřma biimi mikrofondan mikrofonta deėiřebilmekte bu nedenle grlt engellemede kullanılan bazı deėerler mikrofonta gre ayarlanmalı.

Gerek zamanlı grlt engelleme sistemine gelirsek ise grlt gerek zamanda byk oranda engellenebiliyor fakat bazen art arda farklı trden grltler geldiėi durumda arka planda hafif cızırtılar duyulabiliyor. nceki sistemde olduėu gibi bass ieren ve mzikal grltleri engellemede eksiklikler mevcut bunun iin ekstra iřlemler gerekebilir, yine aynı řekilde mikrofona deėiřiminde deėerler zerinde oynama yapılmalıdır. Ayrıca sistemin daha iyi alıřabilmesi iin grltl bir ortamda alıřtırılıp belirlenen adapte olma sresince ortama adapte olması gerekmekte bu řekilde daha iyi sonular alınabilmektedir.

5 - Tartıřma

İki sistemin de amacına byk oranda ulařtıėını dřnyorum fakat sonular kısmında bahsettiėim gibi bu sistemler farklı mikrofonde farklı sonular verebileceėinden dolayı ayarlar mikrofonta zel yapıldıėı takdirde sistem daha iyi alıřacaktır. Ayrıca FFT iin ktphane fonksiyonu yerine FFT'nin manuel bir řekilde koda dahil edilmesi nedeniyle normal FFT ile elde

edeceğimiz sonuçlar arasında ufak farklar olabilir çünkü manuel eklediğimiz FFT içerisinde optimizasyonlar bulunmuyor.

Sesin kalitesine gelecek olursak ise çerçeveler arası geçiş Hann penceresi tarafından iyi bir şekilde sağlanmış olduğundan seste takılma gibi sorunlar fazlaca görülmemekte.

Gürültü profili çıkarımında kullanılan yöntemler açısından değerlendirme yapacak olursak, ses dosyası sisteminde aritmetik ortalama, gerçek zamanlı sistemde medyan yöntemi kullanılmıştır. Aritmetik ortalama hesaplama açısından daha verimli olmakla birlikte, medyan yöntemi ani değişimlere karşı daha dayanıklı sonuçlar vermektedir. Bu nedenle kendi amaçları için uygun çözümlerdir.

Filtreleme konusuna değinecek olursak farklı frekanslar üzerinde farklı ve agresif bir filtreleme ses konusu bu da ses kalitesinin düşmesine sebep olabiliyor. Bu sorunu çözmek için istenildiği takdirde filtrenin agresifliği azaltılabilir. Agresif filtreleme aynı zamanda bazı noktalarda kendi sesimin azalmasına sebep oluyordu bunu çözmek için genlik değerlerini biraz arttırmam gerekti bu şekilde sesim normale yakın düzeye ulaşmış oldu. Gerçek zamanlı sistemin en büyük sıkıntısı ise müzik ve konuşmanın beraber olduğu durumlarda görülüyor, müzik içerisinde bir insan sesi olduğu zaman ise bu daha sıkıntılı bir durum haline geliyor bu yüzden müzik için daha kapsamlı bir yaklaşım izlenmesi gerektiğini düşünüyorum.