ELM368 - DÖNEM PROJESİ

SES SİNYALİ İLE HARF TANIMA

Aykut Şahin, Hasan Polat, Dilara Üzünlü
171024034, 171024057, 171024077
aykutsaahhin@gmail.com, hasanpolatpsn@gmail.com, dilarauzunlu1@gmail.com

ÖZET

Bu projede "F" ve "Z" harflerinin daha önceden kaydedilmiş seslerinden oluşan bir veritabanında kayıtları gerçekleştirilen seslerin giriş olarak kullanılarak bu seslerin kişilerden bağımsız olarak hangi harfe karşılık geldiğinin bulunması amaçlanmıştır. Problemin çözümünde MFCC, Dynamic Time Warping(DTW), FIR filtre kullanılmıştır. Girişten alınan ses sinyalinin gürültüsünü temizlemek için kullanılan FIR "bandpass" filtresinin çıkışları elde edilmiştir. Bu çıkış sinyallerinden MFCC ile elde edilen güç spektrum katsayıları DTW algoritması kullanılarak veritabanındaki sinyaller ile girişte verilen sinyalin birbirine olan uzaklıkları hesaplanıp, her sinyalin birbirine olan uzaklık değeri elde edilmiştir. Veritabanındaki sinyallerden hangisi girişte verilen sinyale daha yakın bir mesafeye sahipse sesin karşılığı olarak veritabanındaki sinyale denk gelen harf algoritma tarafından seçilmiştir.

ANAHTAR KELİMELER

Mel-Frequency Cepstral Coefficient(MFCC), Dynamic Time Warping(DTW), FIR Bandpass filtre, Fast Fourier Transform(FFT)

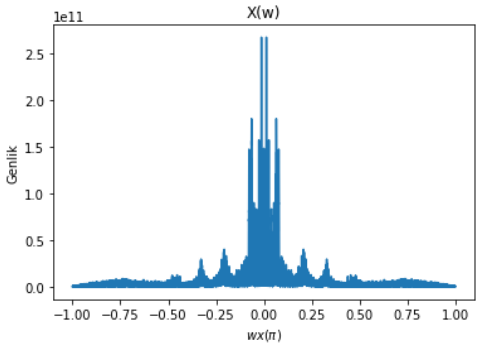
1.GİRİŞ

"F" ve "Z" harflerinin sesleri ekipte bulunan üç kişi tarafından her biri üçer tane olacak şekilde toplamda on sekiz(18) kere kaydedilerek oluşturulan veritabanı kullanılarak, oluşturulacak olan algoritmaya verilecek giriş ses sinyalinin kişiden bağımsız olarak hangi harfe tekabül ettiği bulunması gerekmektedir. "F" ve "Z" harflerinin okunuşları kayıt edilirken insanların ses tonları, cinsiyetleri, ortam gürültüsü gibi sinyali etkileyecek parametreler farklı olduğundan dolayı frekanslar farklı çıkabilmektedir fakat gürültüden maksimum şekilde arındırılmış sinyalde izlenecek örüntü neredeyse aynı olduğundan dolayı bu frekanslar çok büyük farklar göstermeyecek olup, bu frekansların izleyeceği örüntü problemin çözümü için önem arz edecektir. Dolayısıyla, seslerin okunuşunda elde edilecek olan sinyal güç verilerinin birbirine yakın olacağı düşünülmektedir. Sonuç olarak, ses sinyal güç verilerinin birbirine olan yakınlıklarının ölçülmesiyle elde edilecek yakınsama değerleri harflerin tespit edilmesi için kullanılabilecek bir karar mekanizmasına ihtiyaç duyulacağı aşikardır.

2.DENEYLER VE ANALİZ

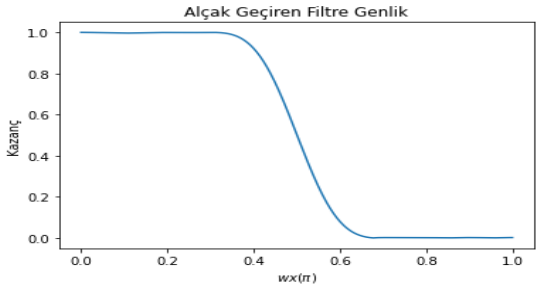
Projede kullanılacak olan "F" ve "Z" harflerinin ses sinyallerinin kayıtları için "PyAudio" ve "Wave" modülleri kullanılmaktadır. Oluşturulacak veritabanında yer alacak ses sinyallerinin hepsinin örnekleme frekanslarının ve formatlarının uygun bir karşılaştırılma ortamı oluşturulması için aynı olması gerektiğinden dolayı toplam on sekiz(18) tane olması gereken ses sinyallerinin kayıtları 16K Hz örnekleme frekansında ve dört(4) saniye olmak üzere "Jupyter Notebook" üzerinden oluşturulup, güncel dizin üzerindeki "database" klasörüne kaydedilmiştir. Bu ses dosyaları üç(3) farklı kişi tarafından üç(3) farklı ortamda oluşturulduğundan her ortamın gürültüsünün, kişi ses tonlarının ve diğer şartlarının aynı olmadığı göz önünde bulundurularak bir filtre tasarlanmasına karar verilmiştir.

2.1 Filtre Tasarımları

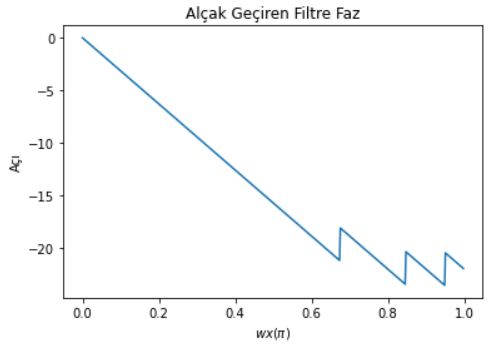


Şekil - 1

Şekil-1'de görüldüğü gibi kaydedilen seslerden bir örneğin FFT ile elde edilen genlik grafiği X(w) başlığında görüldüğü şekilde elde edilmiştir. Farklı alçak geçiren filtrelerle yapılan denemeler sonucunda gürültü seslerinin hem düşük hem yüksek frekanslarda olduğu fark edilmiştir.

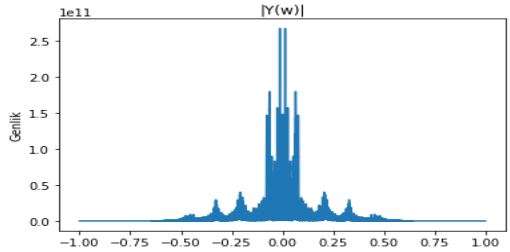


Şekil – 2



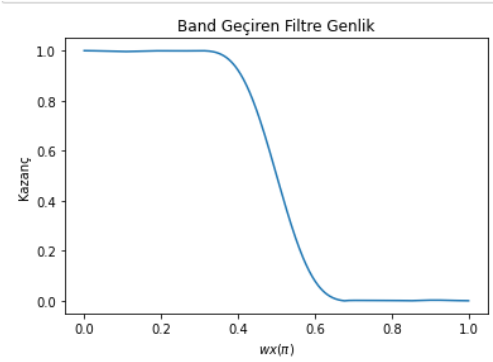
Şekil – 3

Şekil-2 ve Şekil-3'de tasarlanmış alçak geçiren filtrenin sırasıyla genlik ve faz grafikleri gözükmektedir. Bu filtrenin kesim frekansı 0.5π Hz, derecesi yirmi bir(21), kazancı da bir(1) olarak belirlenmiştir.



Şekil – 4

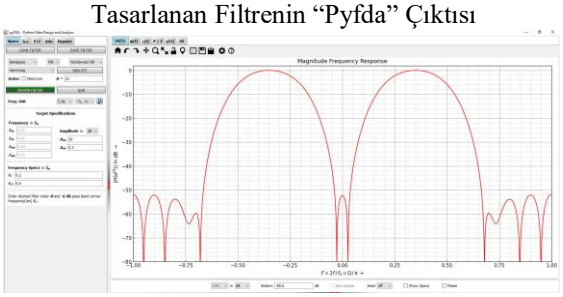
Şekil-4'de görüldüğü üzere giriş sinyaline Şekil - 2'de yer alan alçak geçiren filtrenin uygulanmasıyla yüksek frekanslı gürültü değerlerinden arındırılsa bile düşük frekanslı gürültü değerleri hala geçirmekteydi.



Şekil – 5

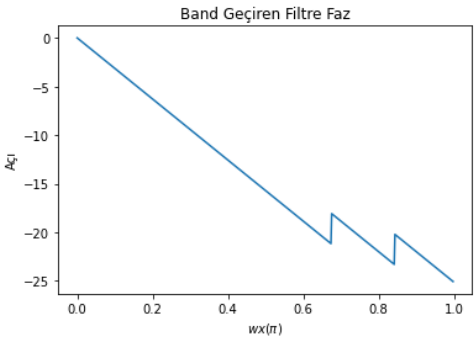
Dolayısıyla, Şekil-5 ile belirtilen bant geçiren filtre genliği grafiğinde görüldüğü üzere kazancı bir(1), kesim frekansı 4k Hz olan FIR filtre ile hem çok düşük hem yüksek frekanslı gürültü değerleri durdurulmaya çalışılmıştır. Filtrenin derecesi 21 ve türü "Hamming" olarak seçilmiştir. Filtre türü ise FIR olarak seçilmiştir çünkü FIR filtreler IIR filtrelere göre kararlılık anlamında daha avantajlıdır ve doğrusal faz düşünüldüğünde grup gecikmesi anlamında daha iyi performans almak mümkündür.

Tasarlanan Filtrenin "Pyfda" Çıktısı



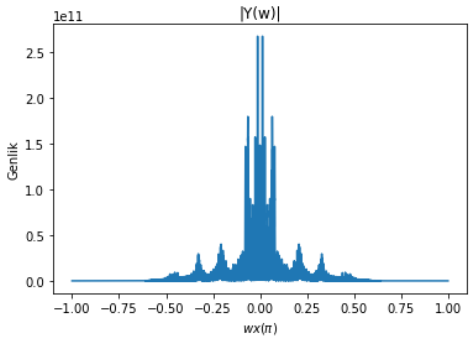
Şekil – 6

Şekil-6'de yer alan "Pyfda" üzerinden tasarlanmış bu bant geçiren filtrenin yukarıda belirtilen davranış karakteristiği daha açık bir şekilde görülmektedir. Kesim frekanslarını 0.5π Hz olması için "Fc2" parametresi 0.5 olacak şekilde belirlenmiştir. "Fc" parametresi alçak frekanslı değerlerin durdurulması için Şekil-6'de ortada yer alan küçük lobu oluşturulmak üzere 0.2 olarak belirlenmiştir.



Şekil - 7

Şekil-7'de görüldüğü üzere FIR türünde tasarlanan bant geçiren filtrenin faz grafiğinin doğrusal olduğu görülmektedir.



Şekil – 8

Şekil-8'de görüldüğü üzere Şekil- 4'ten çok farklı bir grafik çıktısı elde edilmemiştir. Yüksek frekans gürültü değerlerinin arındırıldığı görülmesine karşın düşük frekans değerlerinde olan değişiklik Şekil – 8'e yansımamıştır. Bunun nedeni bu farkı gösterecek hassasiyette bir grafik çiziminin mümkün olmamasıdır. Fakat her iki ses çıkışı

dosyasını dinlendiğinde düşük frekanslı gürültü değerlerinin bant geçiren filtrede filtrelendiği net bir şekilde duyulmuştur.

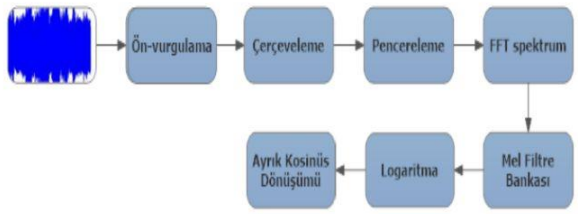
Filtre çıkışlarından elde edilen gürültüden arındırılmış seslerin birbirleriyle olan benzerliklerini tespit etmek amacıyla kullanılması gereken bir yöntem tespit edilmesi gerekmektedir. Yapılan araştırmalar sonucu keşfedilen MFCC yöntemi insan kulağının frekans seçiciliğini taklit ederek iyi bir şekilde sesleri birbirinden ayırt edici değerler elde edilmesini sağlamaktadır.

2.2 Mel-Frequency Cepstral Coefficients

[1] MFCC ses sinyalinin kısa zamanlı güç spektrumunun Mel ölçeği üzerindeki ifadesidir. Sesin bir tür kepstral (Kepstral, Fourier analizinde tahmin edilen sinyal spektrumunun logaritmasının ters Fourier dönüşümünün hesaplanmasının sonucudur) gösteriminden türemiştir. MFCC'de frekans bantları, insan ses sistemi tepkisini yaklaşık olarak Mel ölçeğinde eşit aralıklarla yerleştirilmiştir. Böylece ses sıkıştırmasında daha iyi bir ses temsili sağlanabilmektedir.

Mel frekans kepstral katsayıları (MFCC) MFCC, konuşma tanımada en çok kullanılan özniteliklerden biridir. MFCC, algı temelli sesi temsil eden özniteliklerdir. Mel frekans kepstral katsayıları (MFCC) öznitelik çıkarımının basamakları şekilde gösterilmiştir.

MFCC Basamakları



Şekil – 9

[1] Ses işaretinin istenen dB aralıklarına getirilmesi için bir ön-vurgulayıcı filtreden geçirilir. Bu işlem, çerçeveleme işleminde önce işareti spektral olarak düzleştirmek ve dış etkilere daha az duyarlı hale getirip performansı artırmak için kullanılmaktadır. Konuşma işareti, parametrelerin sabit kaldığı kabul edildiği çerçeve olarak adlandırılan küçük parçalara ayrılmalıdır. Pencereleme işleminin amacı; çerçeveleme işlemi sonucunda oluşabilecek spektral süreksizliğin önüne geçilmektedir. Elde edilen işaretin FFT ile spektrumu bulunarak frekans uzayında işlem yapmak üzere hesaplanmış olur. Bu spektrumu Mel ölçeğinde ifade ederek bant genişliklerinin daha dar, orta ve yüksek frekans bandının daha iyi modellenmesi sağlanır. Logaritma işlemi ile güç spektrumu elde edilir. Ardından ayrık kosinüs dönüşümü ile elde edilen spektrumun katsayıları MFC katsayılarına eşittir.

Mel ölçeği aşağıdaki denklemde (Denklem 1) belirtildiği gibi logaritmik bir işlem ile frekans değerlerinin sıkıştırılıp Hz biriminden mel ölçeğine geçişi sağlanmaktadır. [2]

$$mel(f) = 2595 \log_{10}(1 + (\frac{f}{700})) \text{ (Denklem 1)}$$

Mel güç spektrumu katsayıları, Yk (k=1,2, …, K); mel frekans kepstral katsayıları (cy) ile ifade edilmektedir. Mel katsayıları aşağıdaki denklemlerde (Denklem 2-3-4) verildiği şekilde hesaplanmaktadır. [2]

$$c_y(n) = u_n + \sum_{K=0}^{K-1} (logY_k) * \cos\left((2 * k + 1)n * \frac{\pi}{2 * K}\right) \ (Denklem\ 2)$$

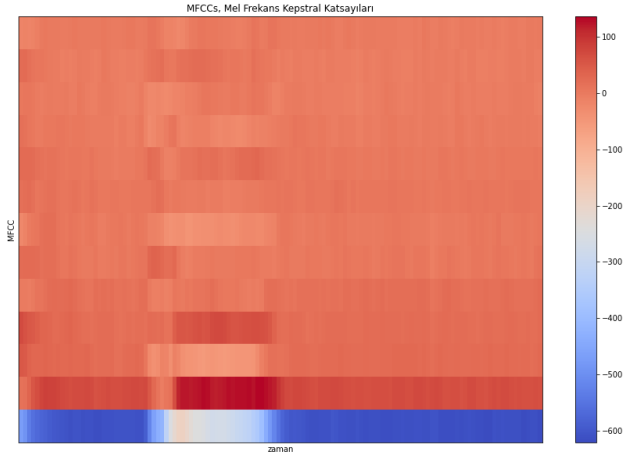
$$u_n = sqrt(K); n = 0 \ (Denklem\ 3)$$

$$u_n = sqrt(2 * K); n > 0 \ (Denklem\ 4)$$

Burada n, mel frekansı kepstral katsayı indeksini; k, Mel filtresi indeksini; K, toplam Mel filtresi sayısını göstermektedir.

[5] Buraya kadar anlatılan işlemler "python_speech_features" modülündeki "mfcc" fonksiyonu kullanılarak gerçekleştirilmiştir. Veritabanımız da bulunan bir ses kaydının Mel Frekans Kepstral Katsayılarının zaman ekseninde, katsayıların renk dağılımı Şekil-10'da görülmektedir.

'F' Harfinin MFCC Grafiği

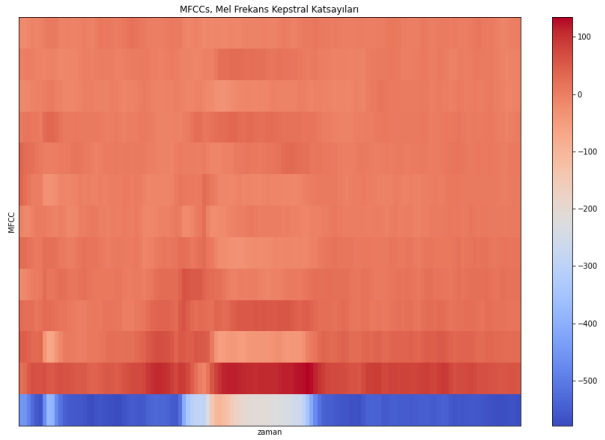


Şekil – 10

Yatay eksendeki belirli bir zaman aralığında renkte oluşan dalgalanmayı dikey olarak göstermektedir.

Bu dalgalanma, ses dalgasında oluşan frekans farkının göstergesidir.

‘Z’ Harfinin MFCC Grafiği



Şekil - 11

Şekil-10 ve Şekil – 11’de aynı kişinin sırasıyla ‘F’ ve ‘Z’ harflerini seslendirdiği ses dosyalarının MFCC grafikleri görülmektedir. Bu grafikler aynı kişinin aynı ortamda söylediği farklı iki harfin katsayılardaki etkisini göstermektedir. Bu etkiler her harfin farklı katsayı dağılımları oluşturduklarının göstergesidir.

Veritabanında bulunan tüm ses işaretlerinin MFC katsayıları ile yine veritabanından seçilen giriş işaretinin MFC katsayılarının birbirleriyle olan benzerlikleri karşılaştırılarak en benzer olan ses dosyası bulunması gerekmektedir. Bu sayede giriş olarak alınan ses işaretinin hangi harf olduğu tespit edilmektedir. Bu karşılaştırmayı yapmak üzere DTW (Dynamic Time Warping) algoritması kullanılmasına karar verilmiştir.

2.3 Dynamic Time Warping

[3] Dinamik zamanda bükme olarak ifade edebileceğimiz bu algoritma temel olarak zaman ekseninde değişen iki farklı dizinin benzerliğini ölçmektedir. İki farklı dizinin birbirlerine olan uzaklıkları bu algoritma sayesinde hesaplanmaktadır. [4]

DTW algoritması şu şekilde açıklanabilir; dizilerin kümülatif Öklid uzaklıkları hesaplanır. Tek boyutta iki değer arasındaki Öklid uzaklığı iki değerin farkının mutlak değeri alınarak bulunur. Kümülatif uzaklık; bir elemanın sadece kendi eşine değil, daha önce eşleşen elemanların tümünün eşlerine olan uzaklıklarının toplamıdır. Kümülatif uzaklık hesaplanırken; her bir eleman çifti arasında hesaplanan Öklid uzaklığına, bu elemanlardan bir birim gerideki eleman çiftleri için hesaplanan kümülatif uzaklıkların en küçüğü eklenir. Böylece eşleşen son elemanlara varıldığında, hesaplanan son kümülatif uzaklık değeri olabilecek en küçük değerinde olur.

DTW Deney Sonuçları

```
Hasan F1 mesafesi:54863.08725583002
Hasan F2 mesafesi:52679.89668171367
Hasan F3 mesafesi:54522.08895757814
Hasan Z1 mesafesi:61053.010076207385
Hasan Z2 mesafesi:54977.68295596177
Hasan Z3 mesafesi:55991.228445910194
Dilara F1 mesafesi:51898.282388378175
Dilara F2 mesafesi:57597.87976747291
Dilara F3 mesafesi:59278.114663965214
Dilara Z1 mesafesi:58566.978684993344
Dilara Z2 mesafesi:54150.98172231325
Dilara Z3 mesafesi:59603.27524146708
Aykut F1 mesafesi:0.0
Aykut F2 mesafesi:33395.782357918804
Aykut F3 mesafesi:33879.56308373447
Aykut Z1 mesafesi:42476.48215181282
Aykut Z2 mesafesi:38964.19955356863
Aykut Z3 mesafesi:40311.76059666079
Bulunan harf:f
```

Şekil - 12

Şekil-12’de yapılan deneyin sonuçları görünmektedir. Bu deneyde veritabanımızda olan ‘f’ seslerinden biri giriş olarak alınıp bu ses veritabanındaki tüm sesler ile karşılaştırıldı. Sonuçlarda görüldüğü üzere on sekiz ses dosyasından biri ile tamamen aynı bulundu ve farkı sayısal değer olarak sıfır bulundu. Eşleşen ses dosyasının hangi harf olduğuna göre bulunan sonuç ekrana bastırıldı.

Sözde Kod

```
Input:  x - length of .wav file
        y - data values of .wav file
Output: yn - output data values

Xw = fft(y)
Yw = Xw*Hw
//Hw,frequency response of the bandpass
filter.
yn = ifft(Yw)

yn_mfcc1 = mfcc(yn)
//yn_mfcc2...
//yn_mfcc3...
...

//test_mfcc is going to be compared with
mfcc of database files
yn_dtw[1] = dtw(test_mfcc,yn_mfcc1)
//yn_dtw[2] ...
//yn_dtw[3] ...
...
        for i {
                getMinValue(yn_dtw[i])
        }

//if minvalue indice is between 1-9, result
is f.
//else result is z.
        if(i<10)
                print Result is f
        else
                print Result is z
```

Şekil – 13

Oluşturmuş olduğumuz algoritmanın sözde kodu Şekil-13'te görünmektedir. Filtre uygulanmış giriş işaretlerinin MFC katsayılarının DTW algoritması ile uzaklıklarının hesaplanmasıyla elde edilen değerlerin en küçüğünün bulunmasına karşılık gelir. En küçük değer bir karar mekanizmasına tabi tutulup hangi harf olduğuna karar verilir.

Sonuçlara göre; veritabanımızdaki seslerin birbirleriyle karşılaştırıldığında kişiden bağımsız olarak harf doğru olarak beklendiği gibi gözlemlenmiştir.

Veritabanının dışında olan bir sesin giriş olarak alınıp veritabanındaki seslerle karşılaştırıldığında sonuçların kesin olarak doğru olmadığı gözlemlendi. Veritabanında olmayan kişilerden birinin söylediği harfleri tamamen doğru olarak tespit ederken diğer bir kişinin söylediği harfleri doğru olarak tespit edilemeyebildiği gözlemlenmiştir. Bunun sebebi ise dışardan gelen ses işaretlerinin kaydedildiği ortam, kişinin konuşma tavrı gibi farklı parametrelere bağlı olarak sonuçlar yanlış elde edilebilmektedir.

3.SONUÇ VE YORUM

Bu çalışmada; proje önerisinde kullanılacağı belirtilen yöntemlerin dışına çıkılmadan amaca ulaşıldı. Tasarlanan filtre ile gürültü azaltılmış olup sonuçların benzerliği karşılaştırıldığı bir çözüm planlanıp uygulandı. Eğer ki çok daha karmaşık gürültülerin olduğu bir ortamda ses kayıtları yapılmış olsaydı mevcut filtre ile yapılan filtreleme işlemi yakınsama için yetersiz olabilirdi. Bu durumda, mevcut filtreye ek olarak ses işaretlerindeki gürültüyü minimuma indirmek için farklı yöntemlere başvurulması gerekebilirdi. Örneğin gürültülü ortamda ses kaydı yapacak bir kişinin o ortamın gürültüsünü ek olarak kaydetmesiyle beraber bu gürültünün asıl ses işaretinden silinmesi bir yöntem sayesinde mümkündür. Gürültünün çok fazla ve karışık olduğu ses kayıtlarıyla işlem yapılması durumunda, bu yöntem sayesinde başarılı bir şekilde sonuçlar elde edilebilirdi.
Bu çalışma ile istenilen formatta ses kayıtları yapıp tasarlanan filtrelerden geçirilip önerdiğimiz algoritmalar kullanılarak ses üzerindeki etkilerini gözlemleyerek ses analizinin gerçekleştirmesi öğrenildi. Bu analizden yola çıkarak elde ettiğimiz sonuçları karşılaştırarak hangi harf olduğunu tespit edildi. Ses analizinde kullanılan yöntemleri ve bu yöntemlerin Python dili kullanarak nasıl uygulandığı öğrenildi.

4.KAYNAKÇA

[1]https://dergipark.org.tr/tr/download/article-file/202719

[2]https://web.itu.edu.tr/~kartalya/proje1/proje_sunum.pdf

[3]https://towardsdatascience.com/dynamic-time-warping-3933f25fcdd

[4]https://github.com/slaypni/fastdtw

[5]https://python-speech-features.readthedocs.io/en/latest/