Konuşmacıdan Bağımsız Kelime Tanıma
Speaker Independent Word Recognition

Nurefşan Sertbaş
Elektronik ve Haberleşme Mühendisliği
İstanbul Teknik Üniversitesi
İstanbul, Türkiye
sertbasn@itu.edu.tr

Shamyrat Zyrriyev
Elektronik ve Haberleşme Mühendisliği
İstanbul Teknik Üniversitesi
İstanbul, Türkiye
zyriyev@itu.edu.tr

Meryem Meray Yağmur
Elektronik ve Haberleşme Mühendisliği
İstanbul Teknik Üniversitesi
İstanbul, Türkiye
yagmurm@itu.edu.tr

***Özetçe* —Bu çalışmada, sınırlı sayıdaki kelimlerden oluşan bir kümedeki seslerin bilgisayar tarafından tanınması üzerinde durulmuştur. Tasarlanan kelime tanıma sistemi kişiye bağımlı değildir. Farklı kişilerden ses örnekleri alınıp kodkitabı oluşturulmuştur. Kodkitabı baz alınarak giriş olarak alınan sesin tanınması ve eşleştirilmesi yapılmıştır. Bu gerçekleme sırasında kullanılan çeşitli ses işleme algoritmaları anlatılmıştır. Yapılan testlerde %100'lere varan doğru kelime tanıma yüzdesine ulaşıldığı gözlemlenmiştir.**

Anahtar Kelimeler—algoritma, ses işleme, lpc,vektör kuantalama, öklid, otokorelasyon,levinson-durbin, özilişki

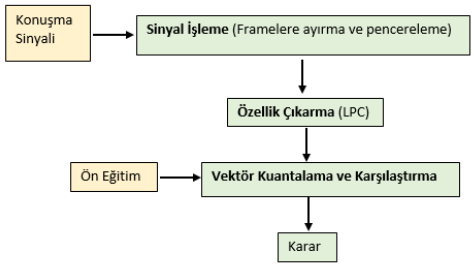
***Abstract*—Main purpose of this paper is recognizing words with the help of computer. Designed word recognition project is independent from the speaker. Also, it realized on bounded set of words. Sample words are taken from different speakers in order to build codebook. The codebook is used as a reference when comparison process of test word. In this paper, the steps of speech recognition is given. According to simulation results accuracy reaches up to %100.**

Keywords—algorithm, speech processing, linear predictive coding,vector quantization, ecluidien distance ,autocorrelation,levinson-durbin,

I. Giris

Günümüz teknolojisinin hızlı gelişimi, mobil cihazların hayatımızın her alanına girişiyle birlikte insan-insan iletişimi yerini ciddi ölçüde insan-makine etkileşimine bırakmştır. Tüm bu gelişmeleri destekleyici olarak bilgisayarın konuşma sinyalini algılaması ve anlamlandırması önerilebilir. Kısıtlı sayıdaki kelimeden oluşan bir öneğitimle yüksek doğruluk seviyelerine ulaşmak mümkün olabilmektedir. Kelime tanıma sistemlerinin çalışma ilkesi, giriş verisinin daha önce kaydedilmiş şablonlarla karşılaştırılmasına dayanır [2]. Karşılaştırma sonucunda mevcut girişe en yakın olan seçilir. Ses sinyalleri zamanla çok hızlı değişen işaretler olduğundan iki ses sinyalini karşılaştırmak oldukça maliyetli bir işlemdir. Bu sebeple ses sinyalleri üzerinde birtakım işlemler ve dönüşümler yapılarak mümkün olduğunca herhangi bir veri kaybına izin vermeden, işlenecek veri miktarının azaltılması esas alınır [14]. Bu aşamada bilgi kaybının olmaması veya minimum düzeyde tutulması için parametre seçimi oldukça önemlidir. Kelime tanıma uygulamalarında uygun forma getirilen sesi karakterize eden birtakım veriler elde edilir. Bu amaçla en çok kullanılan yöntem Doğrusal Öngörü Kodlama (LPC) ve Mel Frekans Kepstral Katsayıları (MFCC) olarak verilebilir [8]. Karşılaştırma aşamasında da bu verilerin birbirlerine benzerlik oranları kullanılarak tahmin yapılır.

Projede yapılacak olan temel adımlar Şekil 1'de gösterilmiştir.



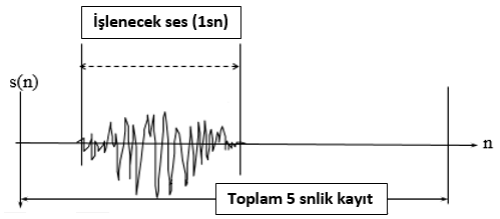
Şekil 1: Sistem blok diyagramı

Bu çalışmada, bir bilgisayarın söylenen sınırlı sayıdaki kelimeleri algılayıp kullanıcının seçeceği sınırlı sayıdaki dillerden birine çevrilmesi hedeflenmiştir. Tasarlanan sesli sözlük sisteminde, sesin alınması ve işlenmeye hazır hale getirilmesiyle öznitelik çıkarma için kullanılan yöntemler II de, mevcut kelimelerle karşılaştırılması ve karar verilmesi aşamasında yapılan işlemler III de anlatılmıştır.

II. Ön İşleme

A. Kayıttan Ses Olmayan Kısımların Çıkartılması

Öznitelik vektörü çıkarma aşamasından önce alınan kaydın ses olan ve olmayan kısımlarının ayrıştırılması gerekmektedir. Bu nedenle konuşmacının kelimeye başladığı ve kelimeyi bitirdiği noktaları tespit etmek amaçlı ses dosyaları uygun forma getirilir [17]. Bu işlem Şekil 2'de gösterilmiştir.



Şekil 2: Ses kaydının kesilmesi

B. Öznitelik Çıkarma

Öznitelik vektörü, analiz edilecek sinyalin taşıdığı bilgiyi mümkün olan en iyi şekilde karakterize eder [7]. Öznitelik çıkarmanın gerekliliği işlenecek veri miktarını azaltmasından gelir. Bu sayede yapılacak karşılaştırma işlemlerinin karmaşıklığı azalır.

Projede öznitelik vektörlerinin çıkartılması sırasında en çok kullanılan algoritmalardan LPC (Doğrusal Öngörü Kodlama) kullanılmıştır [6]. Öznitelik vektörünün çıkartılması sırasında izlenilmesi gereken adımlar şu şekilde verilebilir [4] .

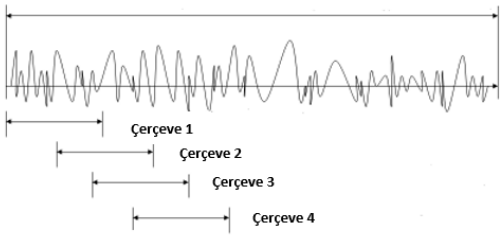
1) Çerçeve Bloklama
2) Pencereleme
3) LPC Katsayılarının Çıkarılması

1) Çerçeve Bloklama : Ses sinyalleri zamanla çok hızlı değişen sinyaller olduklarından ancak 20-100 ms gibi çok küçük zaman aralıklarında kararlı(durağan) oldukları kabul edilmiştir. Yapılan literatür araştırmalarında bu aralıkta en iyi performansın 20-30 ms de alındığı görülmüştür.

Projede her çerçeve uzunluğu 500 birim olarak alınmış ve örnekleme frekansı 48000 Hz ile 1 sn lik ses verileri üzerinde çalışılmıştır. Dolayısıyla

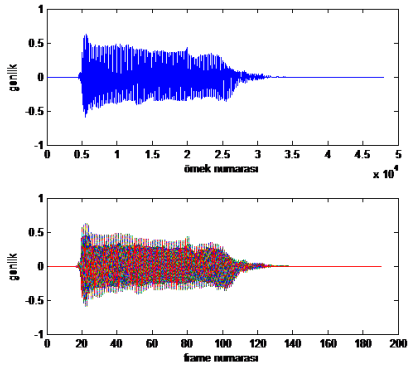
$$cerceve\ suresi = \frac{[cerceve\ boyutu]}{1/f_s} \tag{1}$$

Denklem 1 gereğince çerçeve süresi 500/fs yani 10.41 ms olarak hesaplanmıştır.



Şekil 3: Ses sinyali için çerçeveleme

Şekil 4'de gösterildiği üzere her çerçeve kendisinden bir önceki çerçevenin belli bir kısmını örter. Örtme yönteminin amacı bir çerçeveden diğerine geçişim yumuşak olmasını sağlamaktır [5]. Aşağıda örnek olarak 'elma' kelimesi için sinyalin kendisinin ve çerçevelenmiş halinin zamanla değişimi gösterilmiştir.



Şekil 4: Sinyalin çerçevelere ayrılması [11]

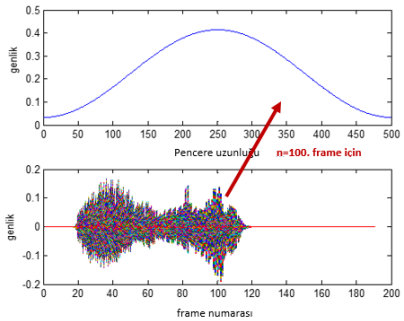
2) Pencereleme: Elde edilen çerçeveler, çerçeve başında ve sonundaki süreksizlikleri ortadan kaldırmak amacıyla pencerelenirler. Benzetimin bu aşamasında en çok kullanılan pencerelerden biri olan Hamming penceresi kullanılmıştır.

$$w(n) = 0.54 + 0.46cos(\frac{2\pi n}{N-1}) \tag{2}$$

Elde edilen pencereleme fonksiyonu her çerçeveye ayrı ayrı uygulanır.

$$s(n) = s(n)w(n) \tag{3}$$

Pencereleme sırasında pencerenin varsayılan genliği 1 olduğundan, her çerçeve için genliği normalize edilmiştir. Benzetimdeki pencere kullanımı çıktıları aşağıda gösterilmiştir.



Şekil 5: Hamming penceresi kullanımı

3) LPC Katsayılarının Çıkarılması: Sayısal işaret işlemede seslerin tanınması için her biri farklı algoritmalar içeren birden çok metod kullanılır. Bunlardan en önemlisi Doğrusal Öngörü Kodlamadır [1]. Ses sinyalinin bir sonraki değerinin bir önceki değerin parametrelerine bakılarak öngörülmesi prensibi ile gerçeklenir. LPC tercih edilmesinin başlıca sebepleri:

1.Sesin düşük bit hızında iletilmesi
2.Saklama alanı maliyetlerinin düşük olması
3.Gerçeğe yakın sonuçların elde edilmesi

Normalde R ses sinyalinin özilişki matrisini göstermek üzere $R.a = r$ formunda bir denkleminin çözülmesi ile LPC katsayıları yani denklemdeki a vektörü bulunabilir. R matrisi n x n lik özilişki katsayılarını, a ise n x 1 lik LPC katsayı matrisidir. Bu denklem Wiener-Hopf denklemi olarak bilinir. Ancak burada R^{-1}in yani özilişki matrisinin tersinin hesaplanması gerekir. Bu da büyük miktarda ses verileri kullanıldığı durumda maliyetli bir işlem haline gelmektedir [13].

Çözüm Toeplitz matrisi ile oldukça basit hale indirgenebilir. Levinson-Durbin algoritması Toeplitz matrisinin kullanımıyla hesaplamalarda ciddi bir kolaylık sağlayarak karmaşıklık seviyesini düşürür. Bu yöntemde LPC katsayıları özilişki formülü (ard-arda gelen değerler arasındaki benzerlik) hesaplanarak yapılır [16], [12].

Levinson-Durbin işlemi:

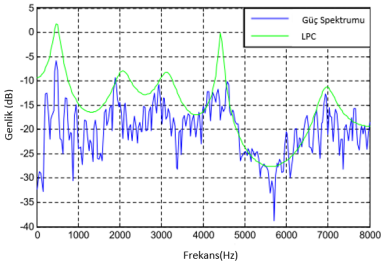
$$\begin{bmatrix} r(1) & r(2)^* & ... & r(n)^* \\ r(2) & r(1) & ... & r(n-1)^* \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ r(n) & ... & r(2) & r(n) \end{bmatrix} \begin{bmatrix} a(2) \\ a(3) \\ . \\ . \\ . \\ a(n+1) \end{bmatrix} = \begin{bmatrix} -r(2) \\ -r(3) \\ . \\ . \\ . \\ -r(n+1) \end{bmatrix}$$

Buradaki r(n) özilişki vektörü, $r(n)^*$ özilişki vektörü konjugesi, a(n) LPC katsayıları, n ise süzgecin derecesidir.

$$r(n) = \sum_{i=1}^{p} a_i(n) r(n-1) \tag{4}$$

Denklem 4'den ve yöntemin adından da anlaşılacağı üzere bir önceki duruma bağlılık söz konusudur.

Benzetim sırasında öngörülen filtre derecesi 17 dir. Yukarıda anlatılan Levinson-Durbin algoritması baz alınarak 17 adet a_k katsayısı hesaplanmıştır



Şekil 6: LPC katsayılarının orjinal genlik spektrumu ile karşılaştırılması [13]

III. KARŞILAŞTIRMA VE EŞLEŞME

A. Vektör Kuantalama

İşlemler sonucunda elde edilen öznitelik vektörlerinin boyutunun büyük oluşu,yüksek hesaplama maliyetleri gibi nedenlerden ötürü bu veriyi işlemek çok zaman alır. Bu nedenle çalışmada vektörlerin daha küçük boyutlara indirgenmesi için en etkili yöntemlerden biri olan Vektör Kuantalama yöntemi kullanılmıştır [3], [10]. Daha az yer kaplaması, daha hızlı hesaplama yapabilmesi ve düşük karmaşıklık seviyesi vektör kuantalamanın avantajları arasında sayılabilir [15]. Bu yöntemin en önemli dezavantajı ise kuantalama sırasında değerlerin kaybedilmesidir. Özellikle kuantalama seviyesinin az olduğu durumlarda sistemin performansını ciddi ölçüde düşürür. Vektör kuantalamada amaç bilgiyi performans kaybı olmadan minimum miktarda veri ile ifade etmektir.

B. Öklid Uzaklığı Yöntemi

İki nokta arası mesafe ne kadar azsa noktaların, projedeki seslerin, birbirlerine benzerlikleri o kadar artmaktadır [9]. Öklid uzaklığının hesaplanması aşağıda verilmiştir.

$$\sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{5}$$

p Eğitim verisi noktaları
q Test verisi noktaları

IV. GERÇEKLEŞTIRILEN ÇALIŞMA

Kullanıcı ara yüzü programını oluşturmak için MATLAB kullanılmıştır.Kullanıcı 'Ses Kaydet' tuşu ile ses kaydını yapıp 'Kaydedilen Sesi Dinle' tuşu ile boşluklardan arındırılmış 1 sn lik ses kaydını dinleyebilir. Verilen dillerden dilediğini seçer. 'Çevir' tuşuna bastığında kaydedilen ses dosyası üzerinde yukarıda bahsedilen ilgili işlemler yapılır ve önceden oluşturulmuş sistemdeki kayıtlı ses verileri ile karşılaştırma yaparak benzetim sonucunu ve istenen dildeki karşılığını ilgili kutucuklara yazar. Şekil 8'de ise örnek olarak 'elma' sözcüğü için geliştirilen kullanıcı arayüzü programı çıktısı verilmiştir.

Şekil 7: Kullanıcı arayüzü

V. Sonuç ve Öneriler

Proje tamamlandıktan sonra yapılan denemelerde parametre seçiminin kompleksiteye etkisi ve hesaplama süresinin ne şekilde değiştiği gözlemlenmiştir. Ayrıca, benzetimin daha başarılı çalışabilmesi için eğitim setinin daha da büyütülmesi, daha fazla kişiden ses örneklerinin alınması gerekliliği görülmüştür.
Kelime bazında doğruluk oranları Tablo 1 de gösterilmiştir.

Kelime	*Başarım oranları*
Araba	60
Armut	50
Çanta	80
Deniz	30
Elma	10
Kablo	50
Kaşık	100
Köpek	70
Mavi	50
Üzüm	70
Tamam	30

Tablo I: Benzetim sonucu doğruluk oranları

Benzetim sırasında da aynı sözcüğü aynı kişinin farklı seslendirmelerinde dahi farklı vektörler çıkmakta bibirleriyle eşleşmedikleri durumlar gözlemlenmiştir. Bu duruma çözüm olarak daha fazla ses örneğinin toplanması öngörülmektedir. Ayrıca kelime seçiminin doğruluk oranına etkisi de yine Tablo 1'den görülebilmektedir. En iyi başarım 'Kaşık', 'Çanta' ve 'Köpek' kelimelerinden alınmıştır. Bu da kelime seçiminin etkisini ortaya çıkarmaktadır. İleriki aşamalarda verimliliğin arttırılması için zaman eşleştirme(dynamic time wrapping) işlemi yapılabilir. Bu metod sayesinde kelimelerin söylenme süreleri sıkıştırılıp genişletilme ile eldeki referanslarla karşılaştırılmaları esas alınır.

VI. Kaynakça

[1] J. Bradbury. Linear predictive coding. 2000.
[2] B. B. İbrahim Gelegin. Ayrık kelime tabanlı bir konuşma tanıma sistemiyle bilgisayar kontrolü. 2011.
[3] A. Buzo, J. Gray, A., R. Gray, and J. Markel. Speech coding based upon vector quantization. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 28(5):562–574, Oct 1980.
[4] M. N. Do. An automatic speaker recognition system, 2015.
[5] S. Çenesiz and M. Öztürk. Ses tanıma, 2010.
[6] J. F. Frigon and V. Teplitsky. Implementation of linear predictive coding (lpc) of speech, Spring 2000.
[7] S. Furui. Speaker-independent isolated word recognition using dynamic features of speech spectrum. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 34(1):52–59, Feb 1986.
[8] S. Furui. *Digital speech processing, synthesis, and recognition*. Marcel Dekker, New York, 1989.
[9] J. Gray, A. and J. Markel. Distance measures for speech processing. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 24(5):380–391, Oct 1976.
[10] R. M. Gray. Vector quantization. *ASSP Magazine, IEEE*, 1(2):4–29, 1984.
[11] G. M. M. K. Linga Murthy. Isolated word recognition using lpc and vector quantization. *IJRET: International Journal of Research in Engineering and Technology*, 1:479–482, November 2012.
[12] T. MathWorks. lpc, 2015.
[13] N. A. Meseguer. Speech analysis for automatic speech recognition. 2009.
[14] L. R. Rabiner and B.-H. Juang. *Fundamentals of speech recognition*, volume 14. PTR Prentice Hall Englewood Cliffs, 1993.
[15] S. M. S. Venugopal, B. Murugan. Sopc based word recognition system. Nios II Embedded Processor Design Contest—Outstanding Designs 2005, 2005.
[16] C. G. Si (Laura) Cai, Pritish Gandhi. The music really speaks to me. Technical report, Carnegie Mellon University, 2009.
[17] K. H. Wong. An example of a speech recognition system.