

**PARALEL PROGRAMLAMA İLE SES TANIMA
İŞLEMİNİN GERÇEKLEŞTİRİLMESİ**

**2013
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ**

Emrah ÖZKAYNAK

**PARALEL PROGRAMLAMA İLE SES TANIMA İŞLEMİNİN
GERÇEKLEŞTİRİLMESİ**

Emrah ÖZKAYNAK

**Karabük Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalında
Yüksek Lisans Tezi
Olarak Hazırlanmıştır**

**KARABÜK
Haziran 2013**

Emrah ÖZKAYNAK tarafından hazırlanan "PARALEL PROGRAMLAMA İLE SES TANIMA İŞLEMİNİN GERÇEKLEŞTİRİLMESİ" başlıklı bu tezin Yüksek Lisans Tezi olarak uygun olduğunu onaylarım.

Yrd. Doç. Dr. Salih GÖRGÜNOĞLU
Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

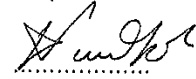


Bu çalışma, jürimiz tarafından oy birliği ile Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir. 26 / 06 / 2013

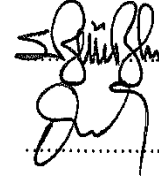
Ünvanı, Adı SOYADI (Kurumu)

İmzası

Başkan : Yrd. Doç. Dr. İlhami Muharrem ORAK (KBÜ)



Üye : Yrd. Doç. Dr. Salih GÖRGÜNOĞLU (KBÜ)

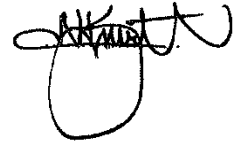


Üye : Yrd. Doç. Dr. Mustafa AKTAŞ (KBÜ)

24/07/2013

KBÜ Fen Bilimleri Enstitüsü Yönetim Kurulu, bu tez ile, Yüksek Lisans derecesini onamıştır.

Prof. Dr. Nizamettin KAHRAMAN
Fen Bilimleri Enstitüsü Müdürü



“Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim.”

Emrah ÖZKAYNAK

ÖZET

Yüksek Lisans Tezi

PARALEL PROGRAMLAMA İLE SES TANIMA İŞLEMİNİN GERÇEKLEŞTİRİLMESİ

Emrah ÖZKAYNAK

Karabük Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:

Yrd. Doç. Dr. Salih GÖRGÜNOĞLU

Haziran 2013, 52 sayfa

Ses tanıma, teknolojinin gelişmesi ile birlikte günümüzde birçok alanda kullanılmaktadır. Bu çalışmada, ses tanıma kullanılan yöntemler karşılaştırmalı olarak incelenmiştir. Matlab'ın Paralel Programlama aracı kullanılarak bu yöntemlerin ses tanıma uygulamasındaki zaman performansı değerlendirilmiştir. Bu çalışmada ses tanıma kullanılan yöntemler Temel Bileşenler Analizi (TBA), Yapay Sinir Ağları (YSA), Destek Vektör Makineleri (DVM) ve K-Ortalama (K-Means) yöntemleridir. Elde edilen sonuçlar, ses tanıma işlemlerinde en iyi başarı oranını Yapay Sinir Ağlarının, en düşük başarı oranını K-Ortalama Algoritmasının verdiğini, Temel Bileşenler Analizi ve Destek Vektör Makinelerinin ses tanımadaki başarı oranının bu ikisi arasında yer aldığını göstermektedir. Yapılan deneysel çalışmalarda, bu dört farklı yöntemin başarı oranının %90 üzerinde olduğu görülmektedir. Bu çalışmada ayrıca çok sayıda ses verisinin bulunduğu uygulamalarda, zaman performansını artırmak için paralel programlamadan

yararlanılmıştır. Algoritmaların paralel programlama ile kullanıldığında ne kadar bir hızlanma gerçekleştiği gösterilmiştir.

Anahtar Sözcükler : Ses tanıma, konuşma tanıma, konuşmacı tanıma, temel bileşenler analizi, yapay sinir ağları, destek vektör makineleri, k-ortalama algoritması, paralel programlama.

Bilim Kodu : 902.1.014

ABSTRACT

M.Sc. Thesis

VOICE RECOGNITION WITH PARALLEL PROGRAMMING

Emrah ÖZKAYNAK

Karabük University

Graduate School of Natural and Applied Sciences

Department of Computer Engineering

Thesis Advisor:

Assist. Prof. Dr. Salih GÖRGÜNOĞLU

Juy 2013, 52pages

Voice recognition, along with the development of technology is used in many areas today. In this study, the methods for identifying audio are evaluated comparatively. The performances of these methods in voice recognition are assessed by using Matlab's Parallel Programming tool. In this study, the methods used in identifying sound are Principal Component Analysis, Neural Networks, Support Vector Machines and K-Means methods. The results show that the best success rate of voice recognition processes the performances of is achieved by Artificial Neural Networks, and the lowers success rate is produced by K-Mean algorithm. The performance of Principal Component Analysis and Support vector machines are in between these two algorithms. In experimental studies, it is observed that the success rates of the four different methods are over 90%. In this study since the application deals with large amounts of audio data, parallel programming is used to in performance of improve the performance. Acceleration in performance of algorithms by using parallel programming is shown.

Key Word : Voice recognition, speech recognition, speaker recognition, principal component analysis, artificial neural networks, support vector machines, k-means algorithm and parallel programming.

Science Code : 902.1.014

TEŐEKKÜR

Bu tez alıřmasının planlanmasında, arařtırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteęini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle alıřmamı bilimsel temeller ışığında şekillendiren sayın hocam Yrd. Do. Dr. Salih GÖRGÜNOĐLU'na sonsuz teşekkürlerimi sunarım.

Sevgili eşime, kızıma ve aileme manevi hiçbir yardımı esirgemedен yanımda oldukları için tüm kalbimle teşekkür ederim.

İÇİNDEKİLER

	Sayfa
KABUL.....	ii
ÖZET	iv
ABSTRACT.....	vi
TEŞEKKÜR.....	viii
İÇİNDEKİLER	ix
ŞEKİLLER DİZİNİ.....	xii
ÇİZELGELER DİZİNİ	xv
SİMGELER VE KISALTMALAR DİZİNİ	xvi
BÖLÜM 1	1
GİRİŞ	1
BÖLÜM 2	3
SES TANIMA SİSTEMLERİ.....	3
2.1. SES TANIMA SİSTEMLERİNİN AVANTAJLARI	4
2.2. KONUŞMA TANIMA SİSTEMLERİ.....	5
2.3. KONUŞMACI TANIMA SİSTEMLERİ.....	5
BÖLÜM 3	7
LİTERATÜR TARAMASI.....	7
BÖLÜM 4	10
SES ÖZELLİKLERİNİN ÇIKARTILMASI	10
4.1.SESİN SAYISALLAŞTIRILMASI	11
4.1.1 Örnekleme	11
4.1.2. Sesli İfadenin Tespit Edilmesi.....	12
4.2. SES İŞLEME.....	14
4.2.1. Pencereleme.....	14

	Sayfa
4.2.2. Sesin Filtrelenmesi	16
4.2.2.1. FIR Filtreler	16
4.3. ÖZİNİTELİK VEKTÖRLERİNİN ELDE EDİLMESİ.....	19
4.3.1. Hızlı Fourier Dönüşümü (FFT)	19
4.3.2. Mel-Frekans Dönüşümü	20
4.3.3. Mel-Frekans Kepstrum Katsayısı	21
4.3.4. Normalizasyon.....	21
BÖLÜM 5	22
DENEYSEL YÖNTEMLER	22
5.1. DESTEK VEKTÖR MAKİNELERİ (DVM).....	22
5.2. TEMEL BİLEŞENLER ANALİZİ (TBA).....	24
5.3. K-ORTALAMA (K-MEANS)	26
5.4. YAPAY SİNİR AĞLARI (YSA).....	30
5.4.1. Temel YSA Bileşenleri ve Yapısı	31
5.5. PARALEL PROGRAMLAMA	33
BÖLÜM 6	37
UYGULAMA SONUÇLARI.....	37
6.1. YSA İLE SES TANIMA İŞLEMİ.....	38
6.2. DVM İLE SES TANIMA İŞLEMİ	41
6.3. TBA İLE SES TANIMA İŞLEMİ.....	42
6.4. K-MEANS İLE SES TANIMA.....	44
BÖLÜM 7	46
SONUÇ VE ÖNERİLER	46
KAYNAKLAR	48
ÖZGEÇMİŞ	53

ŞEKİLLER DİZİNİ

Sayfa

Şekil 2.1 Ses Tanıma sistemlerinin genel çalışma prensibi	4
Şekil 4.1 Sesin sayısallaştırılması	11
Şekil 4.2 “Ankara” kelimesinin ses sinyali	11
Şekil 4.3 ‘Ankara’ kelimesinin başlangıç ve bitiş tespiti sonrası ses sinyali	11
Şekil 4.4 Hamming penceresi	11
Şekil 4.5 ‘Ankara’ kelimesinin hamming penceresi uygulandıktan sonra durumu. ..	11
Şekil 4.6 ‘Ankara’ kelimesinin filtreleme işlemi öncesi ses sinyali	11
Şekil 4.7 ‘Ankara’ kelimesinin filtreleme işlemi sonrası ses sinyali	11
Şekil 5.1 Destek vektör makinesi.....	22
Şekil 5.2 En iyi altdüzlem	22
Şekil 5.3 K-Means uygulanmadan önceki karmaşık veri seti	28
Şekil 5.4 Sınıflandırılacak iki farklı veri türü için rastgele seçilmiş merkezler	29
Şekil 5.5 İki farklı renkte sınıflandırılan veriler.....	29
Şekil 5.6 Hesaplanan merkezlere göre sınıflandırılan veriler	30
Şekil 5.7 Temel yapay sinir ağı hücresi yapısı.....	31
Şekil 5.8 Seri programlamanın çalışma mantığı	33
Şekil 5.9 Paralel programlamanın çalışma mantığı.....	34
Şekil 6.1 Bir kelimenin farklı kişilere ait ses grafikleri	38
Şekil 6.2 Ses Tanımda kullanılan YSA modeli.....	39

ÇİZELGELER DİZİNİ

	<u>Sayfa</u>
Çizelge 5.1. Seri hesaplama örneği	35
Çizelge 5.2. Paralel hesaplama örneği	35
Çizelge 6.3. Ses tanımada kullanılan kelimeler	37
Çizelge 6.2. YSA modeline ait parametre verileri	39
Çizelge 6.3. Filtrelenmemiş seslerin YSA ile yapılan tanıma işlemi sonuçları	40
Çizelge 6.4. Filtrelenmiş seslerin YSA ile yapılan tanıma işlemi sonuçları.....	40
Çizelge 6.5. Filtrelenmemiş seslerin DVM ile yapılan tanıma işlemi sonuçları.....	42
Çizelge 6.6. Filtrelenmiş seslerin DVM ile yapılan tanıma işlemi sonuçları.....	42
Çizelge 6.7. Filtrelenmemiş seslerin TBA ile yapılan tanıma işlemi sonuçları	43
Çizelge 6.8. Filtrelenmiş seslerin TBA ile yapılan tanıma işlemi sonuçları.....	44
Çizelge 6.9. Filtrelenmemiş seslerin K-Means ile yapılan tanıma işlemi sonuçları .	45
Çizelge 6.10. Filtrelenmiş seslerin K-Means ile yapılan tanıma işlemi sonuçları	45

SİMGELER VE KISALTMALAR DİZİNİ

KISALTMALAR

ADC	: Analog Dijital Dönüştürücü
DFT	: Discrete Fourier Dönüşümü
DVM	: Destek Vektör Makineleri
FFT	: Fast Fourier Transform
FIR	: Finite Impulse Response
GB	: Giga Byte
GHz	: Giga Hertz
Hz	: Hertz
K-Means	: K-Ortalama
Khz	: Kilo Hertz
LPC	: Linear Predictive Coding
MA	: Moving Avarage
MFCC	: Mel-Frekans Kepstrum Katsayısı
RBF	: Radyal Tabanlı Sigma Genişlik Fonksiyonu
SMM	: Saklı Markov Model
TBA	: Temel Bileşenler Analizi
TG	: Toplam Genlik Hesabı
YSA	: Yapay Sinir Ağları

BÖLÜM 1

GİRİŞ

Ses insanları veya canlı türlerini birbirinden ayırmak için kullanılan biyometrik bir özelliktir. Var olan bir ses sinyali işlenerek, bu sesin insan veya başka bir nesneden geldiği tespit edilebilir. İnsanların sesleri birbirinden farklılık gösterdiğinden ses tanıma amacı ile kullanılabilir. Ses sinyalinden bir insanın konuştuğu kelimeler elde edilerek, çeşitli cihazları kontrol etmek veya yönlendirmek de mümkündür. Bu nedenlerle ses sinyalinin işlenmesi ve kullanımı oldukça önemlidir.

Ses tanıma ile bir takım işlemleri gerçekleştiren yazılımlar, ses ile kontrol edilebilen internet tarayıcıları, ev telefonu veya cep telefonu üzerinden sesli imza olarak da kullanılan konuşmaya dayalı otomatik tanıma sistemleri, herhangi bir şekilde engeli bulunan özellikle görme ve hareket engellilerin elektronik veya mekanik cihazları kontrol edebilmeleri için geliştirilmiş kontrol sistemleri ses tanımada kullanılan sistemlere örnek olarak verilebilir [1,2].

Yine yaygın olarak kullanılmaya başlayan bir diğer uygulama ise sesle kontrolü sağlanabilen internet tarayıcı sistemleridir. Bu sistemin çalışma yapısına bakıldığında kullanılan mobil telefonlar üzerinden web tarayıcıları sesle kontrol sağlanabilmektedir [3]. Buna benzer olarak, mobil veya sabit telefonlar üzerinden yapılabilen bankacılık ya da rezervasyon işlemlerinde de sesli imza teknolojisi kullanılmaya başlanmıştır. Daha önceden sisteme tanıtılan ses ile daha sonraki girişlerde kullanıcının sesi karşılaştırılarak ses tanıma işlemi gerçekleştirilir. Bu sayede telefon tuşlarıyla yapılan tanımlama işlemlerinde harcanan zaman sesli imza ile minimuma indirilerek zamandan tasarruf da sağlanmış olmaktadır [4]. Günlük hayatta ses tanımanın sağladığı faydalara bakılacak olursa; mobil telefonlarda yaygın olarak kullanılan sesli arama sistemleri, konuşma tanıma ile kontrol edilebilen cihazlar hızla günlük hayattaki yerlerini almayı başarmışlardır [5].

Genel olarak ses tanıma ile ilgili yapılan sistemlere bakıldığında ses tanıma işlemlerindeki performansın artırılması, herhangi bir koşula bağlı kalmadan ses tanıma işlemi yapabilen sistemlerin geliştirilmesi üzerine yapılan çalışmalar ağırlık kazanmaktadır [6].

Hazırlanan bu çalışma, yedi bölümden oluşmaktadır. Birinci bölümde, çalışmada esas alınan konuşma olayının önemi, iletişimdeki yeri ve konuşma tanınmanın kullanım alanları hakkında bilgi verilmektedir.

İkinci bölümde, ses tanıma sistemleri hakkında genel bilgi verilerek, konuşmacı tanıma sistemleri detaylandırılmıştır.

Üçüncü bölümde, bu alanda daha önce yapılmış çalışmalar incelenmiştir.

Dördüncü bölümde, ses tanıma işlemine geçilmeden önce ses üzerinde yapılan ön işlemler ve seslerin özneliliklerinin çıkartılması hakkında bilgi verilmiştir.

Çalışmanın beşinci bölümünde, bu çalışmada ses tanıma işlemleri için kullanılan yöntemler, yöntemlerin modellemesinin ve matematiksel hesaplamaların daha hızlı sonuçlanması için kullanılan paralel programlama hakkında bilgiler verilmiştir.

Altıncı bölümde, deneysel çalışmalar sonucu elde edilen bulgular hakkında bilgiler verilmiştir.

Yedinci ve son bölümde ise yapılan çalışmanın sonuçları ile ilgili yorumlara ve önerilere yer verilmiştir.

BÖLÜM 2

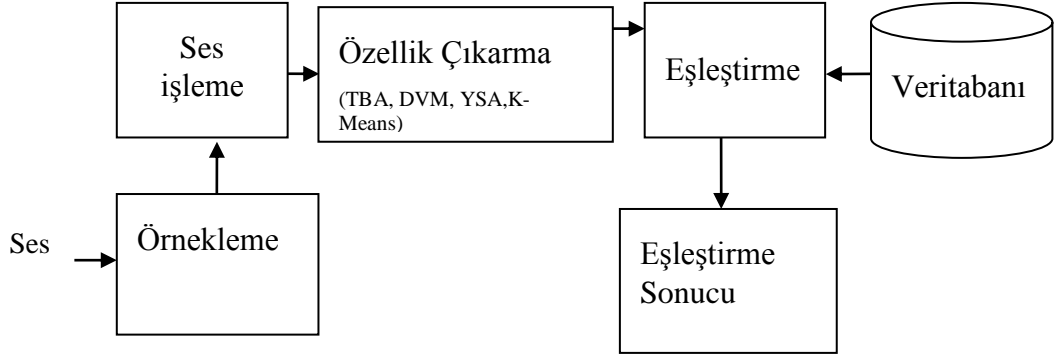
SES TANIMA SİSTEMLERİ

Teknolojide yaşanan gelişmelere paralel olarak insanlarla makineler arasındaki iletişim de çeşitli yöntemler kullanılmaya başlanmıştır. İnsan ile makine arasında iletişim kurmak için geliştirilen yöntemlerden biri de ses tanımadır. Günlük hayatta insanın en sık kullandığı iletişim yöntemi konuşmadır.

Sesle bir kişisel bilgisayara birtakım işler yaptırmak günümüzde üzerinde yoğun olarak çalışılan konulardan birisidir. Güvenilir bir şekilde kullanıcının konuştuğunu ekrana yazabilen veya kullanıcının konuşarak verdiği emirleri yerine getiren bilgisayarlar, birçok alanda yapılan işlemleri kolaylaştırmaktadır. Bunlara örnek olarak, dikte paketleri, sesli tarama, telefon üzerinden konuşma tanıma tabanlı otomatik sistemler, görme hareket vb. engelleri olan insanlara makinelerle iletişimi sağlamak için bir takım alternatifler sunan sistemler verilebilir [7]. Ses tanıma alanındaki yaygın uygulamalardan biride telefon üzerinden bankacılık rezervasyon gibi işlemlerdir. Daha önce telefon tuşları kullanılarak uzun zaman alan işlemler ses tanıma tabanlı sistemlerle daha hızlı, ucuz ve kolay hale getirilmiştir. Yakından tanıdığımız sesli arama yapan cep telefonları, sese duyarlı ev eşyaları daha önce saydığımız örnekler gibi hayatımızı kolaylaştırmaktadır [8].

Ses tanıma sistemleri genel olarak iki farklı alt başlıkta incelenebilir. Bunlardan birincisi konuşma tanıma, diğeri ise konuşmacı tanıma sistemleridir. Bu iki sistem hem kullanıldıkları yerler bakımından hem de ürettikleri çözümler bakımından birbirlerinden tamamen farklı sistemlerdir [9].

Ses tanıma sistemlerinin genel çalışma prensibi Şekil 2.1.'de verilmiştir.



Şekil 2.1 Ses Tanıma sistemlerinin genel çalışma prensibi.

2.1. SES TANIMA SİSTEMLERİNİN AVANTAJLARI

Ses tanıma sistemleri uygulamada pek çok yerde kullanıldıklarından dolayı sağladığı faydalar açısından da tercih edilme sebeplerinden biri olmuş durumdadır. Ses tanıma sistemlerinin sağladığı faydaları özetle sıralanacak olursa;

Kolay Veri Girişi: Ses tanıma sistemlerinin en önemli faydalarından bir tanesi veri girişinin kolaylıkla yapılabilmesidir. Girilen verinin ses olması, ayrıca veri girişi olarak bir mikrofon ya da telefon gibi cihazlardan rahatlıkla yapılabilmesi alışlagelmiş geleneksel veri giriş yöntemlerine göre avantajlıdır [10].

Verilerin Hızlı Hazırlanması: Ses tanıma sistemleri, veri girişinin konuşma dili gibi kolay yöntemlerle yapılabilmesi açısından zamandan tasarruf sağladığı gibi verilerin daha hızlı toplanmasını da kolaylaştırmaktadır. Örneğin konuşma ile veri girişi klavye gibi bir giriş ünitesinden daha hızlı olabilmektedir. Bu da üretim sektörlerinde veri toplanmasının daha hızlı yapılabilmesinde ve zamandan tasarruf sağlanmasında fayda sağlamaktadır [10].

Serbest Kullanım: Kişinin yaptığı iş bakımından zaman zaman kullandığı cihazlar ya da konumu açısından klavye, mouse vb. giriş üniteleri aracılığı ile veri girişi yapamayacak durumlarda olabiliyor. Ses tanıma sistemleri hareket serbestliği bakımından da kullanımda faydalar sağlamaktadır [10].

Mekândan Bağımsız Veri Giriş İmkânı: Ses tanıma sistemleri geleneksel veri giriş yöntemlerinden farklı olarak uzaktan veri girişine de olanak tanımaktadır. Örneğin

bir telefon aracılığı ile ses tanıma sistemlerinden faydalanarak uzaktaki bir sisteme veri girişi sağlanabilmektedir. Bu avantajı bakımından ses tanıma sistemleri, uzaktan servis hizmeti veren birçok hizmet birimi tarafından tercih edilmektedir [10].

2.2. KONUŞMA TANIMA SİSTEMLERİ

Konuşma tanıma sistemleri, ses tanımanın iki önemli alt başlıklarından bir tanesidir. Konuşma tanıma sistemlerinin başlıca görevi kişinin konuştuğu kelime ya da cümleyi tahmin edebilmek ve buna en yakın sonuçları veri setinden getirebilmektir [11]. Konuşma tanıma sistemleri kullanım yerleri ve amaçları bakımından sürekli konuşma tanıma ve hiyerarşik konuşma tanıma olarak iki kısma ayrılmaktadır [12].

Sürekli konuşma tanıma, konuşanın söylediği her kelime ya da cümlenin metne dönüştürülmesidir.

Hiyerarşik konuşma ise veri setindeki belli komutların söylenmesi üzerine çalışan bir yapıdır. Genellikle sesli komut sistemlerinde tercih edilir.

2.3. KONUŞMACI TANIMA SİSTEMLERİ

Konuşmacı tanıma, kısaca belli kelime ya da kelimelerden bir kişinin tanınması işlemidir. Aslında günümüzde sıkça karşılaştığımız bir durumdur. Bir kişiyi görmeden sadece sesini duyarak tanıdığımız çokça olmuştur. Telefondaki sesin kime ait olduğunu, ya da gözlerimiz kapalı iken daha önceden tanıdığımız bir kişiyi sesinden ayırt edebiliriz. Ya da kalabalık ve sesli bir ortamda tanıdığımız bir kişinin sesini ayırt ederek kişi tanımlaması yapabiliriz [13].

Konuşmacı tanıma sistemleri, insanların bu özelliklerinden faydalanarak, kimlik tanıma işlemlerini makinelere yaptırma amacıyla tasarlanan ve üzerinde yoğun olarak çalışılan bir konudur. Konuşmacı tanıma sistemleri de kendi içerisinde üç sınıfa ayrılmaktadır. Bunlardan birincisi konuşmacı doğrulama, ikincisi konuşmacı saptama ve üçüncüsü ise metinden bağımsız konuşmacı tanımadır [14].

Konuřmacı doęrulama sistemi, daha nce sisteme ses rnekleri tanıtılmıř bir konuřmacının, sisteme giriř esnasında beyan ettięi kimlięe ait olup olmadıęının doęrulanmasıdır. Eęer giriř yapmak isteyen kullanıcı bilgisi ile sesi rtřyorsa sistem kabul cevabı verirken, aksi bir durumda ret cevabı vermektedir [15].

Konuřmacı saptama sistemi, daha nce ses rnekleri alınan kullanıcılar ile sisteme giriř yapmak isteyen kullanıcının ses rneklerinin karřılařtırılmasıdır. Eęer giriř esnasında kullanıcıdan alınan ses rnekleri veri tabanındaki herhangi bir ses rneęi ile uyuyor ise o ses rneęine ait kiřinin bilgileri cevap olarak dnerken, herhangi bir ses rneęi ile uyuřmaması durumunda sistem ret cevabı vermektedir [15].

Metinden baęımsız konuřmacı tanıma sistemi ise konuřmacılardan alınan ses rnekleri dıřında bařka bir ifade ile sisteme giriř yapıldıęında tanıma iřlemi gerekleřtirebilme olanaęı vermektedir. Metin baęımlı konuřmacı tanıma sistemlerinden farklı olarak belirlenmiř parola ya da kelimeler kullanılmadan tanıma iřlemi bařarı ile gerekleřtirilebilmektedir [15].

BÖLÜM 3

LİTERATÜR TARAMASI

Ses alanında yapılan çalışmalar incelendiğinde, en eski çalışmanın AT&T BellLabs tarafından üretilen ve “Voder” olarak adlandırılan ilk elektronik konuşma sentezleyicisi olduğu bilinmektedir. Konuşma sentezleyicisi, gürültülü ortamlarda elde edilen kayıtlardaki gürültüleri tespit ederek bunların temizlenmesini sağlamaktadır [16].

Texas Instruments tarafından “Speakand Spell” isminde bir oyuncak, geliştirilmiştir. Bu oyuncakın elektronik devresi “Speakand Spell” için bir konuşma çipi tasarlanmıştır. Tasarlanan çipin en önemli avantajı insanın konuşma şekline yakın konuşma sentezleme işlemi yapabilmesidir [17].

Speech Works isminde bir şirket, telefon hattı üzerinden otomatik konuşma tanıma sistemlerini geliştirmiş ve bunun üretimini sağlamıştır. Otomatik konuşma tanıma sisteminde tanıma işlemi için Saklı Markov Model (SMM) yöntemi kullanılmıştır ve başarı sağlanmıştır. SMM için kullanılan özellik vektörleri ise Mel-kepstrum algoritması ile hesaplanarak oluşturulmuştur [18].

Dragon Systems, tarafından üretilen kelime tabanlı dikte yazılımı YSA ile tasarlanmış ve belli kelimelere bağlı komutları tanıma işlemleri gerçekleştirilerek, o komutların aracılığı ile belli başlı işlemlerin yapılması sağlanmıştır [19].

Benzeghiba vd., yaptıkları çalışmada konuşma tanımada aksanın, cinsiyetin, yaşın, konuşma tarzının etkilerinin iklim ve bölgelere göre değişiklik gösterdiğini tespit etmişlerdir. Bu tespiti yaparken geçmişte çeşitli yöntemlerle birçok farklı alanda yapılan ses tanıma çalışmalarını hangi yöntemlerin ne tür farklılıklardan etkilendiklerini de araştırmışlardır [20].

Valente, konuşma tanıma alanında yaptığı çalışmasında hiyerarşik konuşma tanımanın sürekli konuşma tanımaya göre daha verimli olduğunu tespit etmiştir. Valente çalışmasında, sürekli ve hiyerarşik konuşmalardan topladığı ses örneklerini çok katmanlı YSA modeli kullanarak eğitmiş ve testlere tabi tutmuştur [21].

Kim ve Stern, yaptıkları çalışmada ses tanımada ses gürültülerini azaltmak için Multi-Band yöntemini araştırmışlardır. Çok katmanlı filtre özelliği olan çalışmada gürültüleri sınıflandırıcı olarak Bayes sınıflandırma yöntemini kullanmışlardır. Ses örneklerinin özellik vektörleri için mel-kepstrum yönteminden yararlanmışlardır [22].

Kingsbury vd., modülasyon spektrogram kullanarak gürültülerden arındırılmış güçlü bir konuşma tanıma sistemi üzerinde çalışma yapmışlardır. Modülasyon spektrogram, seste bulunan belirli seviyedeki gürültüleri alçak geçiren filtre ile temizlerken, seste bulunan yankı, ses tekrarı gibi güçlü gürültüleri ise band geçiren filtre ile temizlemekteydi. Çalışmanın konuşma tanıma kısmında ise YSA ve SMM kullanarak modülasyon spektrogram ile temizledikleri ses örneklerini sınıflandırmışlardır [23].

Campbell, geliştirdiği konuşmacı tanıma sisteminde kimlik doğrulama işlemi yapmış ve %98'lik bir başarı elde etmiştir. Geliştirilen sistemde ses örnekleri Hamming, Hanning gibi çeşitli pencereleme yöntemlerinden geçirilerek alçak geçiren filtrelerde temizlendikten sonra, kepsral analiz yöntemi ile sadeleştirilmiş ve YSA ile eğitilmiştir [24].

Reynolds vd., kimlik doğrulama yapan bir konuşmacı tanıma sistemi geliştirmişlerdir. Sistemde tanıma işlemi için Gaussian yöntemi kullanılmıştır [25].

Çakır ve Okutan, ses kontrollü web tarayıcı yazılımı geliştirmişlerdir. Yazılımda ses örneklerinin işlenmesi ve tanınması için YSA ve SMM' den oluşan hibrit bir model kullanmışlardır. Yapılan testlerin sonucunda sistemin başarıyla çalıştığını tespit edilmiştir [26].

Eray, DVM ile ses tanıma işlemini başarı ile uygulamıştır. Çalışmada 400 adet ses örneğinden 200 adet ses örneğini eğitim, 200 adet ses örneğini ise test için kullanmıştır. Yapılan testler sonucunda DVM'nin %91 oranında başarılı olduğunu tespit etmiştir [27].

Güvensan ve Tayşi, ev aletlerinin tanınması için çevresel seslerin sınıflandırılması üzerine bir çalışma yapmışlardır. Evlerde kullanılan elektrikli makinelerin çıkardıkları seslerin sınıflandırılması üzerine geliştirilen çalışmada yöntem olarak K En Yakın Komşuluk algoritması DVM karşılaştırılmak amaçlı kullanılmıştır. Yapılan testler sonucunda DVM'nin %98' lik bir başarı sağladığını tespit etmişlerdir [28].

Caner ve Üstün, YSA ile çalışan bir konuşmacı tanıma sistemi geliştirerek kabul edilebilir bir hata payı ile başarılı sonuçlar elde etmişlerdir. Yapılan çalışmada 7 farklı kişiden alınan ses örnekleri işlenerek gerekli öznelikleri çıkartılmış YSA ile eğitildikten sonra yapılan testlerde 7 kişi içerisinde 5 kişiyi %90 üzerinde başarı oranı tanınmasını sağlamışlardır [29].

Küçüker vd., güvenlik sistemleri için Mel Frekans Kepstrum Katsayıları ve YSA modelini kullanarak konuşmacı tanıma sistemi geliştirmişlerdir. Yapılan çalışmada 3 kullanıcıdan alınan Kapi kelimesine ait ses örnekleri Mel Frekans Kepstrum Katsayıları ile işlenerek YSA'da eğitilmiş ve sınıflandırma işlemi yapılmıştır. Yapılan testler sonucunda %100 başarı sağlanmıştır [30].

Edizkan vd., sesle kontrol edilebilen gezgin araç çalışması yapmışlardır. Yapılan çalışmada ses tanıma işlemini Ortak Vektör Yaklaşımı ve SMM kullanarak gerçekleştirmişlerdir. 'ileri', 'geri', 'sağa dön', 'sola dön' ve 'dur' kelimelerinden oluşan sesli komut sistemini başarı ile uygulamışlardır [31].

BÖLÜM 4

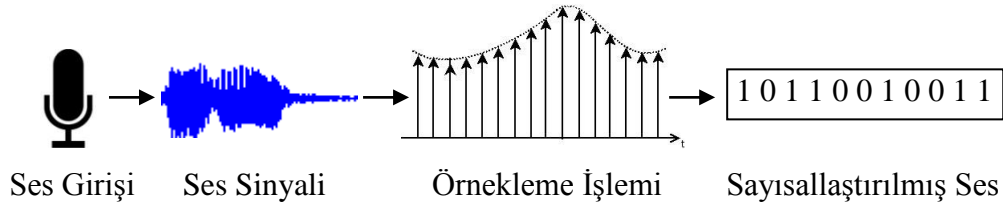
SES ÖZELLİKLERİNİN ÇIKARTILMASI

Bilgisayarda ses tanıma işleminin gerçekleştirilebilmesi için sesin algılanmasından itibaren tanıma sürecine kadar pek çok ön işlemin uygulanması gerekmektedir. Ses tanıma işlemi de yapılacak tanıma işleminin durumuna göre uygun ve sağlıklı ön işlemlerin önceden belirlenmesi, başarılı bir ses tanıma işlemi için çok önemli bir aşamadır. Sesler üzerinde yapılan her ön işlem ses tanımının başarılı olmasında önemli bir etken iken yanlış seçilen ön işleme tekniği de ses tanımının başarısızlığına bir etkindir [32]. Ses tanımının alt konuları olan konuşma tanıma ve konuşmacı tanıma işlemleri için kullanılacak ses örnekleri her zaman aynı ön işlemlerden geçirilmek durumunda değildir. Aynı ön işlemlerden geçirilerek de konuşma ve konuşmacı tanıma işlemleri başarı ile yapılabilirken, bazen de farklı ön işleme teknikleri de kullanmak gerekebilir. Hatta bu farklılık konuşmacı tanıma işlemleri için de kendini gösterebilir. Örneğin konuşmacı kimliği tespiti için kullanılacak ses örneklerinin ön işlemleri için kullanılacak teknikler, konuşmacı tanıma ya da metinden bağımsız konuşmacı tanıma işlemlerinde kullanılacak ses örneklerinin ön işlemleri için yeterli olmayabilir. Bazen de ses tanıma aşamasında kullanılacak yöntemler için farklı ön işleme teknikleri kullanmak gerekebilir. YSA için kullanılacak ses örneklerinin öznitelik vektörlerinde ön işlemler için Hamming Pencereleme ve Finite Impulse Response (FIR) yeterli olur iken bu durum SMM yönteminde yeterli ya da geçerli olmaz. Çünkü SMM için kullanılan ses örneklerinin öznitelik vektörlerinin çıkarımı sadece Fast Fourier Transform (FFT) dönüşümü ile yeterli olabildiğinden FFT için kullanılacak ses örneklerinin Hamming Pencereleme ve Doğrusal Öngörülü Kodlama (LPC) katsayı hesaplamasından geçmesi yeterli olabilmektedir [33].

Sesin tanıma sürecine girişi bir mikrofon ya da benzer bir görevi yerine getiren kayıt aracı ile yapılır.

4.1.SESİN SAYISALLAŞTIRILMASI

Bir analog sinyalin sayısallaştırılması için örnekleme işleminin yapılması gerekmektedir. Şekil 4.1.'de örnek bir sayısallaştırma işlemi gösterilmektedir. Ses kayıt cihazlarında gürültü gidermek için filtre bulunmasına rağmen ses sinyalinin gürültüden arındırılmasına ve duru bir ses kaydı yapılmasına yeterli değildir. Diğer taraftan profesyonel ses kayıt stüdyolarında kullanılan geliştirilmiş ses kayıt cihazlarındaki filtreler duru bir sesi elde etmede oldukça başarılıdır. Sesin sayısallaştırılması için analog olarak alınan ses sinyalinin Analog Dijital Dönüştürücü (ADC) cihazlarla işlenmesi gerekmektedir. Bilgisayarda ses kayıt işlemlerinde ADC olarak ses kartı kullanılmaktadır. Bu çalışmada sesler 8 bit ve mono olarak kayıt altına alınmıştır. Sesin sayısallaştırılması aşamasında Şekil 4.1.'de gösterildiği gibi ve örnekleme işlemi yapılmaktadır [34].



Şekil 4.1. Sesin sayısallaştırılması.

4.1.1 Örnekleme

Analog bir sinyali sayısallaştırmak için sinyalin belli bir düzen de sayılar şeklinde gösterimi yapılması gerekmektedir. Bu işlem örnekleme ile yapılmaktadır. Ses sinyalinin içerdiği özelliklerin korunarak yapılan örnekleme, sayısallaştırılacak ses sinyalinin sahip olduğu en yüksek frekans değeri sıklığının en az iki katı sıklıkta gerçekleşmesi ile mümkündür [35].

Rabiner ve Schafer tarafından ifade edilen örnekleme; bir $x_u(t)$ sinyali, $\Omega \geq 2\pi F_N$ olmak üzere $X_u(j\Omega) = 0$ şeklinde bant sınırına sahip fourier $X_u(j\Omega)$ dönüşümüne

sahip ise; $xu(t), \frac{1}{T} > 2F_N$ şeklinde ve $-\infty < n < \infty$ olmak üzere $Xu(nT)$ şeklinde eşit örnekler yerleştirilerek yeniden oluşturularak ifade edilebilir.

F_N ile gösterilen Nyquist Frekansı, örnekleme frekansının yarsını ifade eder. Başka bir ifade ile örnekleme frekansı 100 Hz İse Nyquist Frekansı $F_N = 50 \text{ Hz}$ olacaktır.

Yüksek frekanslı sinyallerin çok düşük oranlarda örnekleme yapılması sonucu elde edilen sinyalin içerdiği özellikler örnekleme yapılmadan önceki sinyale göre çok farklılık gösterebilmektedir. Bu nedenle örnekleme yapılacak sinyalin frekans değerine uygun örnekleme oranı belirlemek ve uygulamak önemlidir [35].

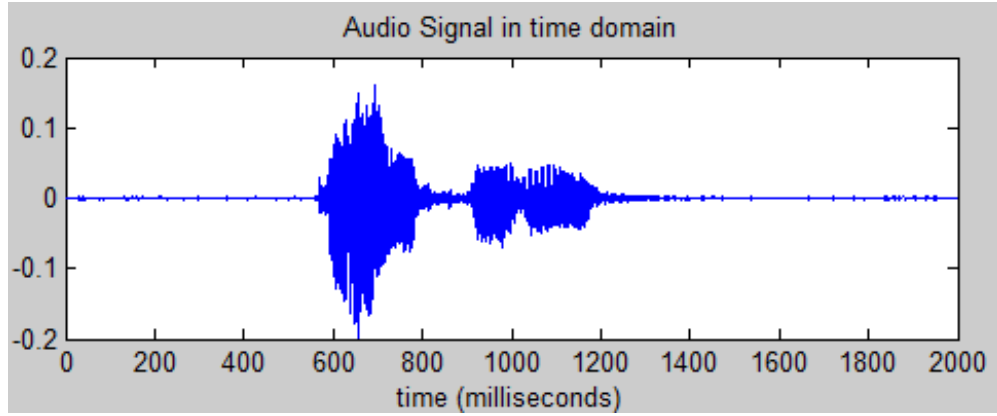
4.1.2. Sesli İfadenin Tespit Edilmesi

Bir ses örneği içerisinde birbirinden farklı ama ses için oldukça önemli iki farklı özellik bulunmaktadır. Bunlar genlik ve frekans özellikleridir. Genlik özelliği sesin şiddeti ve sesin enerjisi hakkında bilgi içerirken, frekans özelliği ise sesin tizlik ve peslik gibi özelliklerini içermektedir. Ses tanıma işleminde, ses sinyalinin analizi ve özelliklerinin birbirinden ayrıştırılması ile gerçekleşmektedir. Aynı kelimeye ait iki farklı ses sinyalinin genlik ve frekans değerleri birbirinden farklılık gösterebilir. Bu farklılığa fonem adı verilmektedir. Ses tanıma sistemleri, bu farklılıklardan faydalanarak kelimeye ait fonemlerin saptanması ve sınıflandırılması üzerine çalışır [35].

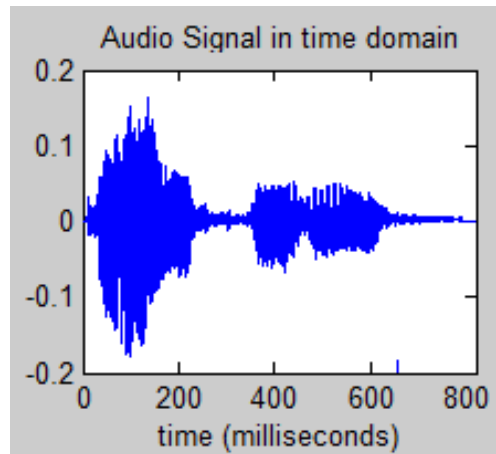
Kayıt edilen bir ses örneğinde ifadenin tespit edilmesi için kullanılan en genel yöntem, konuşmanın geçtiği kısımları sesin genliğine bakarak belirlemektir. Sesin belli bir çerçeve içerisindeki toplam genlik hesabını bularak ses içerisinde konuşmanın geçtiği yerleri tespit etmek mümkün olabilmektedir [35]. Toplam Genlik Hesabı (TG) adı verilen bu yöntem Eşitlik 4.1’de verilmiştir.

$$TG = \sum_{t=1}^n x(t) \quad (4.1)$$

Bu şekilde yapılan hesaplama sonucunda elde edilen toplam genlik değeri sesin enerji toplamı anlamına da gelmektedir, belirli bir eşik değerin üstündeysse ses örneği içerisindeki konuşmanın başladığı anlamına gelmektedir. Aynı şekilde elde edilen toplam enerji miktarı belirlenen eşik değerin altına düşüyse bu durumda konuşmanın sonlandığını anlamına gelmektedir. Gerçekleştirilen bu yöntemle ses örneği içerisindeki konuşmanın başlangıç ve bitişini tespit ederek konuşma dışındaki alanları temizlemek mümkün olmaktadır. Şekil 4.1 ve Şekil 4.2’de ‘Ankara’ kelimesinin ses kaydı içerisindeki konuşmanın tespit edilmeden ve tespit edildikten sonra sessiz kısımları atılmış durumdaki ses örnekleri gösterilmiştir.



Şekil 4.2. ‘Ankara’ kelimesinin ses sinyali.



Şekil 4.3. ‘Ankara’ kelimesinin başlangıç ve bitiş tespiti sonrası ses sinyali.

Bu tez çalışmasında kullanılan kelimelere ait ses örnekleri 8000 Hz Örnekleme frekansı ile sayısallaştırılmıştır.

4.2. SES İŞLEME

Ses işleme başlığı altında incelenen bütün yöntemler ses tanıma işleminden önceki son aşamaları oluşturmaktadır. Ses örnekleri, ses işleme aşamasındaki yöntemlerden geçtikten sonra ses tanıma işlemleri için kullanıma hazır hale gelmektedir. Ses işleme içerisinde, pencereleme, sayısal filtreleme, normalizasyon, ses kodlaması ve son olarak da sesin öznelilik vektörlerinin çıkarılması aşamaları bulunmaktadır.

4.2.1. Pencereleme

Ses sinyalleri, ses işleme aşamalarından önce belli sayılarda örnek içeren parçalara ayrılarak işleme aşamasına hazır hale getirilirler. İşte bu örnekler içeren parçalara pencere adı verilmektedir. Aslında pencereleme işlemi belli bir çerçevedeki ses sinyalinin başlangıç ve bitiş kısımlarını söndürme orta kısmını ise kuvvetlendirme işlemi olarak da tanımlanabilir. Bu işlem ses sinyalini çarpan fonksiyonlar ile yapılmaktadır. Ses işleme aşamasında ses sinyalinin pencereleme işleminden geçirilmesi çok sıkça yapılmaktadır. Bunun avantajı ses sinyalindeki yoğun örneklerin kuvvetlendirilerek belirgin hale getirilmesidir. Pencereleme işleminden geçen bir ses sinyalinin özneliliklerinin bulunması daha kolay hale gelir [36].

Bu tez çalışmasında ses sinyallerini pencereleme işlemleri için Hamming Pencereleme yöntemi kullanılmıştır. Hamming penceresi için kullanılan fonksiyon işlemi Eşitlik 4.2.'de gösterilmiştir.

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) \quad (4.2)$$

Burada N pencereleme yönteminin uygulanacağı çerçevenin uzunluğunu belirtir. Çerçeve dışında fonksiyonun değeri sıfırdır. n ise çerçevenin sırasıyla tüm

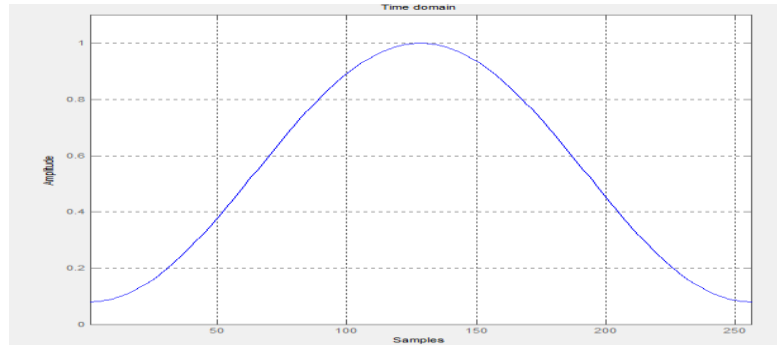
değerlerinin indisini belirtir. Çerçevenin tüm değerleri bu fonksiyonun ürettiği değerlerle çarpıldıktan sonra orta kesiminde bulunan değerler baskınlaştırılmış olur.

Hamming penceresinden elde edilen $y(n)$ çıkışı, Eşitlik 4.3 'de gösterilmiştir.

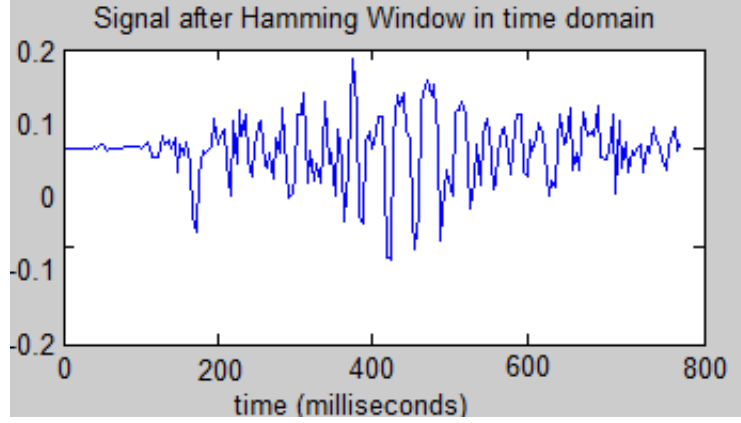
$$y(n) = w(n).x(n) \quad (4.3)$$

Sonuç olarak bakıldığında Hamming penceresinin de sinyalin orta kısmını belirginleştirmesi açısından filtre etkisi yaptığı görülmektedir. Şekil 4.4'de ses sinyaline uygulanan Hamming Pencereleme yöntemi, Şekil 4.5'de 'Ankara' kelime - sine ait ses örneğinin Hamming penceresinden geçirildikten sonraki durumu gösterilmiştir.

Hamming penceresinden sonra elde edilen çıkış sinyalinde 'Ankara' kelimesinin başlangıç ve bitiş noktalarının söndüğü orta kısımın ise daha çok belirginleştiği Şekil 4.5'de görülmektedir.



Şekil 4.4. Hamming penceresi.



Şekil 4.5. ‘Ankara’ kelimesinin hamming penceresi uygulandıktan sonra durumu.

4.2.2. Sesi Filtrelenmesi

Sesi filtrelemenin genellikle iki amacı vardır. Bunlardan birincisi ses içerisindeki konuşmanın daha belirgin hale getirilmesi diğeri ise ses kaydında bulunan ve istenmeyen gürültülerin ortadan kaldırılmasını sağlamaktır. Sesi yakalayan, kayıt eden mikrofon, telefon gibi cihazların özelliğinden ya da kayıt yapılan ortamın gürültüsünden dolayı oluşan gürültü, genellikle rastgele, düzgün olmayan bozucu etkiler olarak görünür ve ses sinyali içerisinde sesin karakteristik özellikleri önemli ölçüde etkiler. Seslerdeki gürültüler doğrusal filtreleme teknikleri kullanılarak temizlenebilmektedir. Ses işleme aşamasında filtreleme işlemleri için pek çok yöntem bulunmaktadır. Bu yöntemler içerisinde en sık kullanılan filtreleme yöntemi olan FIR, hem tasarlanması hem de uygulanması kolay olmakla birlikte elde edilen sonuçların başarılı olması sebebiyle tercih edilmektedirler [37].

4.2.2.1. FIR Filtreler

Sonlu impulse cevabı olan Finite Impulse Response kısaca FIR olarak adlandırılır. FIR filtrelerine, noniteratif filtreler, konvolüsyon filtreleri veya Moving Average (MA) filtreleri de denir. FIR filtrenin çıkışı bir sonlu konvolüsyon olarak ifade edilebilir. Girişi $x(n)$ olan bir sinyalin, o andaki girişlerin ve önceki girişlerin ağırlıklı toplamı $y(n)$ ile ifade edilir [38,39]. FIR filtrenin matematiksel ifadesi Eşitlik 4.4’de gösterilmiştir.

$$y_n = b_0x_n + b_1x_{n-1} + b_2x_{n-2} + \dots + b_qx_{n-q} \quad (4.4)$$

Burada , x_n , filtre edilecek giriş dizisini ; y_n , filtrelenmiş çıkış dizisini ve b de , FIR süzgeç katsayılarını gösterir.

FIR filtrelerinin en önemli özellikleri, katsayı simetrisine sahip olmalarından dolayı lineer faz meydana getirmeleri, her zaman kararlı olabilmeleri ve konvolüsyon kullanılarak filtreleme fonksiyonu yerine getirilebilmeleridir.

FIR filtreler, bir ayrık zaman sisteminin belirtilmiş, istenen frekans cevabının (yaklaşık) tahmini ile tasarlanır. En genel teknikler, bir lineer-faz cevabını sürdürürken istenen büyüklük cevabını yaklaştırır.

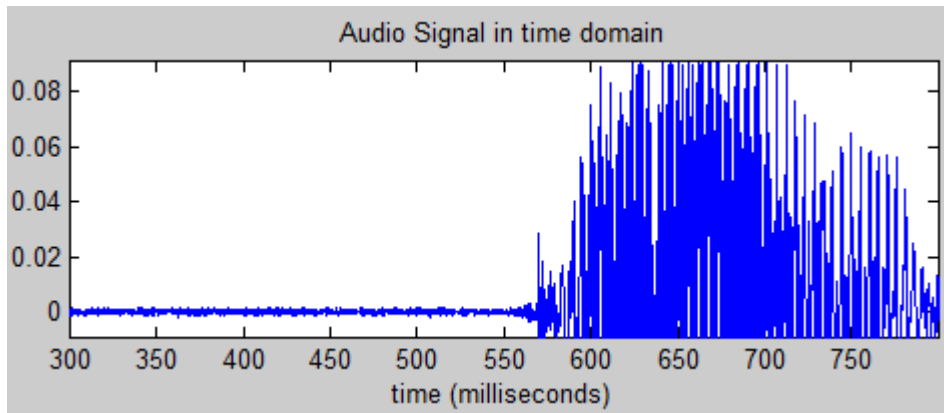
FIR filtrelerinin dizaynında sıkça kullanılan metotların başında, pencere dizayn metodu gelmektedir. Bir FIR filtresini pencerelemeyle dizayn etmek için ideal bir frekans cevabıyla başlamak gerekmektedir. İdeal frekans cevabı ile başladıktan sonra dürtü cevabı hesaplanır ve bir sonlu sayıda katsayı ortaya çıkarmak için dürtü cevabı kesilir. İdeal dürtü cevabının kesilmesi FIR filtre frekans cevabında ani geçişlerdeki salınım davranışına neden olur. Pencereleme ile FIR filtrelerinin dizayn edilmesi basittir ve hesap bakımından kolaydır. Ayrıca FIR filtrelerinin dizaynında en hızlı yoldur [38].

FIR filtrelerin avantajlarına bakacak olursak;

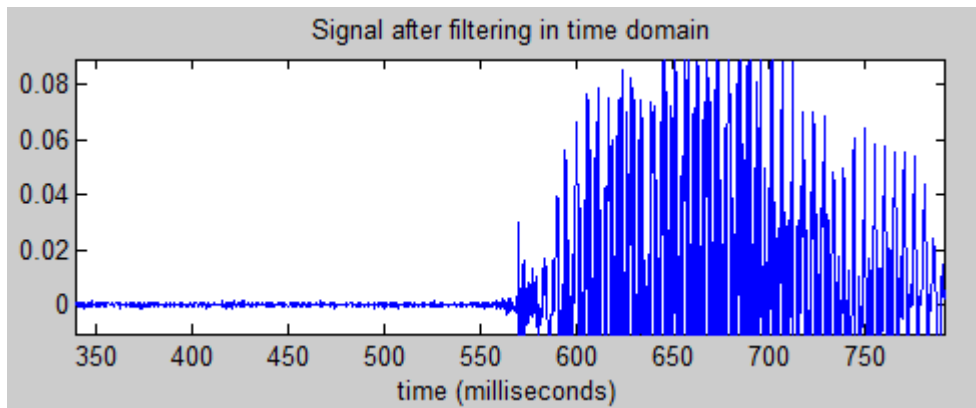
- 1) FIR filtrelerler önceden belirlenmiş genlik frekans karakteristiğini sağlayacak şekilde, tam olarak lineer fazlı olarak tasarlanabilmektedirler. Bununla birlikte herhangi bir frekans karakteristiğini yaklaşık olarak sağlayabilmektedirler.
- 2) FIR filtreler doğrudan konvülasyon veya FFT kullanılarak hızlı konvolüsyon (katlama) yöntemleri ile gerçekleştirilebilmektedir. FIR filtrenin bu şekilde gerçekleştirilmesi yinelemesiz işlem olarak da adlandırılır.
- 3) FIR filtrenin yinelemesiz işlem olarak gerçekleştirilmesi filtrenin daima kararlı olması anlamına gelmektedir

- 4) Diğer filtreleme yöntemlerinin gerçekleştirilmesinde yuvarlatma hataları ortaya çıkmaktadır. Ancak, FIR filtrelerin nonrecursive gerçekleştirilmesinde bu hatalar düşük seviyelerde olduğu için önemsiz olmaktadır.
- 5) Keskin kesim frekanslı FIR filtre tasarımında filtre için katsayı hataları diğer filtre tasarımlarına göre daha azdır.

Şekil 4.6 ve Şekil 4.7’de bir ses sinyalinin FIR filtresinden geçirilmeden önceki durumu ile FIR filtresinden geçirildikten sonraki durumu gösterilmiştir.



Şekil 4.6. ‘Ankara’ kelimesinin filtreleme işlemi öncesi ses sinyali.



Şekil 4.7. ‘Ankara’ kelimesinin filtreleme işlemi sonrası ses sinyali.

Bu çalışmada kullanılan ses örneklerine ait sinyaller 5.dereceden 6 Khz söndürme bandına sahip FIR alçak geçiren filtre ile filtrelenmiştir.

4.3. ÖZİNİTELİK VEKTÖRLERİNİN ELDE EDİLMESİ

Ses tanıma sistemlerinde genellikle eldeki ses verileri ile sistemin eğitilmesi işlemleri gerçekleştirilir. Daha sonra ise eğitilmiş sistemin test edilmesi işlemleri gerçekleştirilir. Burada en önemli husus eğitilecek ya da test edilecek ses verilerini temsil edecek öznelik vektörlerinin uygun yöntemlerle elde edilmesidir. Elde edilen bu öznelikler ile eğitim kütüphanesi oluşturulur ve nihayet tanınmak istenen ses verisine ait öznelik verisi eğitim kütüphanesinde karşılaştırılarak tanıma işlemi gerçekleştirilmiş olur. Ses örneklerine ait sinyalleri analiz ederken sesi temsil edecek özneliklerin çıkarımı için genellikle spektral analiz işlemi yapılır.

Bu tez çalışmasında ses verilerinin öznelik vektörlerinin çıkarımı için uygulanan spektral analiz işlemleri, sırasıyla Hamming Pencereleme, FFT, Mel-Frekans Dönüşümü ve Mel-Frekans Kepstrum Katsayılarından oluşmaktadır. Hamming Pencereleme işlemi hakkında bilgiler Bölüm 4.2.'de verilmiştir.

4.3.1. Hızlı Fourier Dönüşümü (FFT)

Ses işleme aşamalarında çerçeveleme, pencereleme, filtreleme aşamalarından geçen ses örneğine ait ses sinyalin öznelik vektörlerinin elde edilmesi için genlik spektrumunun incelenmesi gerekmektedir. Pencerelemiş bir ses sinyalinin genlik spektrumunun hesaplanması FFT işlemini uygulamak ile mümkün olmaktadır. FFT bir ses sinyalini, kendisini meydana getiren frekanslara ayırmaktadır. Bir rengin oluşması için farklı renklerin bir araya gelmesi gerektiği gibi, ses sinyali de farklı frekanslardaki seslerin birleşmesiyle oluşur. Aynı kelimenin birkaç defa seslendirilmesinde zaman ve genlik bakımından farklılıklar ortaya çıkmaktadır. Bu nedenle bir ses sinyalinin ham halini algılamak ve tanıma işlemlerinde kullanmak zordur. Ses sinyali bu nedenlerden dolayı çeşitli yöntemlerle işlenir ve tanıma işlemlerinde kullanılır. FFT yöntemi de sesin kendine has özelliklerinin ortaya çıkarılmasında kullanılan tekniklerden bir tanesidir. Aynı sözcüğü ifade eden iki ses sinyalinin ham görüntüsüne bakıldığında benzerlikleri bulmak oldukça zor iken FFT işlemi uygulanmış aynı kelimelere ait ses sinyallerinin benzerliklerini tespit etmek daha kolay olmaktadır. FFT, Ayırık Fourier Dönüşümü (DFT) ' nü hızlı bir şekilde hesaplayan bir yöntemdir. DFT, sinyal işlemede

kullanılan temel işlemlerden biridir. Fakat DFT hesaplaması uzun ve karmaşık olmasından dolayı sinyal işlemede zaman kaybına ve yapılan işlem sayısının artmasına neden olmaktadır. Bu soruna çözüm olarak geliştirilen FFT algoritmaları DFT hesaplamalarında etkili çözüm üretmektedir. Bu yöntemin etkili olmasının sebebi, DFT hesaplaması gereken birçok problemin çözümünde kullanılabilmesi ve DFT ile benzer sonuçları vermesidir. FFT işlemi DFT işleminden zaman performansı ve işlem sayısı bakımından çok daha hızlı ve kolay işlem yapmaktadır [40].

N adet örnekle ifade edilen bir ses sinyalinin FFT hesaplaması Eşitlik 4.6'da gösterilmiştir.

$$x_n = \sum_{k=0}^{N-1} x_k e^{-2\pi kn/N} \quad (4.6)$$

Eşitlik 4.6'da x_n , FFT işlemi uygulanacak ses sinyallerini, N ise bu ses sinyallerinin mevcut örnek sayılarını ifade etmektedir [40,41].

4.3.2. Mel-Frekans Dönüşümü

Mel olarak adlandırılan birim, insan kulağını yapay olarak taklit edebilecek şekilde tasarlanmış bir birimdir. Yani mel birimi, bir ses sinyalinin frekans ekseninde ifade edilmesine göre değil, insan kulağının algıladığı şekle göre ifade edilmesine göre geliştirilmiştir. Bu birimle elde edilen seriye mel ölçeği adı verilir. Bir ses sinyalinin frekans ölçeğinden mel ölçeğine dönüşümü için kullanılan matematiksel yöntem Eşitlik 4.7'de gösterilmiştir [40,41].

$$f_{mel} = 2595 \log_{10} \left(1 + \frac{f_{doğrusal}}{700} \right) \quad (4.7)$$

Burada $f_{doğrusal}$, sinyalin frekans domainindeki karşılığı, f_{mel} ise sinyalin mel birimindeki karşılığını ifade etmektedir. Eşitlik 4.7'ye göre mel ölçeği ses sinyalinin 1000 Hz'den düşük frekans değerleri için doğrusal değerlerde bir dağılım gösterirken, 1000 Hz'den yüksek frekans değerleri için ise logaritmik değerlerde dağılım göstermektedir [40,41].

4.3.3. Mel-Frekans Kepstrum Katsayısı

Ses tanımada kullanılacak olan ses örneklerinin özniteliklerini hesaplamada son aşama olan Mel-Frekans Kepstrum Katsayısı (MFCC), sinyali her çerçevesinde ters fourier transformasyon işlemi ile frekans domaininden zaman domainine geçişini sağlar. Bu işlemin sonucunda MFCC katsayıları elde edilerek öznitelik çıkarma işlemi sonlandırılmış olur. MFCC işlemine ait matematiksel ifade Eşitlik 4.8'de gösterilmiştir [40].

$$MFCC_i = \sum_{k=1}^n x_k \cos\left[i\left(\frac{k-1}{2}\right) \frac{\pi}{n}\right] \quad (4.8)$$

Burada $i = 1,2,3,4 \dots M$ 'ye kadar devam etmektedir. M ise, kepstrum katsayılarının sayısını ifade etmektedir. x_k, k ise ($k = 1,2,3 \dots n$), k . filtrenin log-enerji çıktısını ifade etmektedir. n ifadesi MFCC katsayısını göstermektedir. Her çerçevede n boyutlu vektörler ses sinyalinin öznitelik vektörlerinin boyutunu ifade etmektedir [40,41].

4.3.4. Normalizasyon

MFCC işleminden elde edilen ses sinyallerine ait öznitelik vektörlerinin boyutları çok fazla olmasından dolayı verileri işlemek hem zaman açısından hem de işlem açısından zor olmaktadır. Bu nedenle öznitelik vektörlerinin boyutları daha küçük boyutlara yani minimum seviyeye çekilmesi gerekmektedir. Yapılan bu normalizasyon işlemine vektör kuantumlama adı verilmektedir.

Normalizasyon işlemi, büyük sayılardaki öznitelik vektörlerini belli sayıdaki kümelere indirgeyerek, her küme için merkezi vektör hesaplar. Hesaplanan bu merkezi vektör o kümedeki tüm vektörleri temsil eder. Bu şekilde m adet öznitelik vektörü sabit sayıda bir vektör dizisi ile ifade edilmiş olur. min-max normalizasyon işlemi Eşitlik 4.9.'da gösterilmiştir [42].

$$X(n) = \frac{(X_n - X_{min})}{(X_{max} - X_{min})} \quad (4.9)$$

BÖLÜM 5

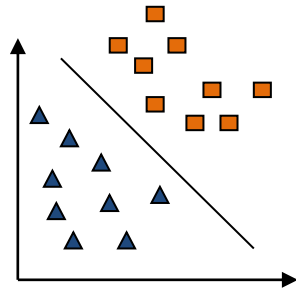
DENEYSEL YÖNTEMLER

Bu bölümde ses tanıma işlemlerinde kullanılacak DVM, TBA, K-Means, YSA ile sistem performansını artırmak için test edilecek Paralel Programlama yöntemleri hakkında bilgiler verilmiştir.

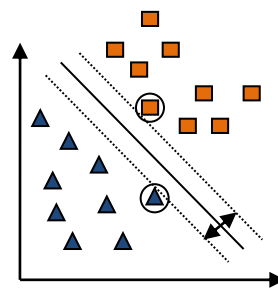
5.1. DESTEK VEKTÖR MAKİNELERİ (DVM)

DVM istatistiksel öğrenme yöntemine dayalı bir sınıflandırma algoritmasıdır. DVM, doğrusal ve doğrusal olmayan verilerin sınıflandırılmasında başarılı olmasından dolayı tercih edilmektedir. DVM, Görüntü Tanıma, Ses Tanıma, İmza Tanıma vb. örüntü tanıma çalışmalarında yaygın olarak kullanılan yöntemlerden biridir.

DVM iki sınıfı birbirinden ayıran en uygun hiper-düzlemin (doğrunun) bulunması mantığı ile çalışmaktadır. DVM iki sınıfa ait verileri sınıflandırmak için iki farklı düzlem (doğru) bulmaya çalışır. Şekil 5.1'de ve Şekil 5.2'de farklı sınıfa ait verilerin birbirinden doğrusal olarak ayrılması ve en ideal alt düzlemin bulunması gösterilmektedir [43].



Şekil 5.1 Destek vektör vakinesi.



Şekil 5.2 En iyi altdüzlem.

Sınıflandırmadaki en temel amaç iki sınıfı birbirinden ayıran en iyi sınırı bulmaktır. En iyi sınırı belirlerken dikkat edilmesi gereken en önemli şey oluşturulacak düzlem her iki sınıf arasındaki en uzak düzlem olmasıdır. İki sınıfın tam ortasından geçen bir sınır olması gerekmektedir. Sınıra en yakın eğitim vektörleri destek vektörleri olarak adlandırılırlar [43].

DVM' de doğrusal olarak ayrılabilen verilerin her biri $\{-1,+1\}$ şeklinde iki sınıf olarak ifade edilmektedir. R^n in elemanı olan veriler x_i ile sınıfları da $y_i=\{-1,+1\}$ ifade edilir. Buradan $i=1,\dots,N$ kümesi S ile ifade edilebilir. Buradaki amaç R^n elemanı olan veri kümesini y_i kümesindeki etiketlere ait olan bir alt düzleme ayırarak aynı sınıfa ait verileri bir alt düzlemde bir arada tutmaktır. Burada x_i veri kümesi için, $i=1,\dots,N$ alınırsa

$$y_i(w \cdot x_i + b) \geq 1 \quad (5.1)$$

koşulunu sağlayan bir w var ise bu veriler doğrusal olarak denilebilir. Buradaki (w,b) Eşitlik 5.1'deki denkleme sahip bir alt düzlemi ifade etmektedir [43].

$$w \cdot x + b = 0 \quad (5.2)$$

Eşitlik 5.2'deki işlem veri noktası ile etiketinin alt düzlemin aynı tarafında olmasını sağlamaktadır. Alt düzlemden rastgele bir x_i veri noktasına olan uzaklığı (d_i) hesaplamak için Eşitlik 5.3 kullanılmaktadır.

$$d_i = \frac{w \cdot x_i + b}{\|w\|} \quad (5.3)$$

Eşitlik 5.1 ve Eşitlik 5.3 birleştirildiğinde Eşitlik 5.4' deki denklem elde edilir.

$$y_i d_i \geq \frac{1}{\|w\|} \quad (5.4)$$

Buradaki işlemlerle birlikte kullanılacak karar fonksiyonunu Eşitlik 5.5 deki gibi elde edilir.

$$f(x) = \text{sign}(w \cdot x + b) \quad (5.5)$$

DVM’de en önemli noktalardan biri de en uygun temsilin seçimidir. Eğer sınıfı temsil edecek vektör doğru seçilirse elde edilecek sonuçlar ve yapılacak sınıflandırmalar da en doğru şekilde gerçekleşir [43].

5.2. TEMEL BİLEŞENLER ANALİZİ (TBA)

Bu yöntem büyük boyutlara sahip veri kümelerinden anlamlı veriler çıkararak veri kümelerini temsil edebilecek yeni veriler oluşturan bir analiz yöntemidir. TBA’nın en önemli özelliği, büyük boyutlardaki veri kümelerinin benzerlik ya da farklılıklarını ortaya koymasındır. Büyük boyutlardaki veri kümelerinin analiz işlemleri zor olmakla birlikte hata oranları da yüksek çıkabilmektedir. TBA, yüksek boyutlardaki veri kümelerini analiz edebilme ve düşük hata oranları ile sınıflandırma yapılmasına olanak sağlaması açısından birçok sistemlerde tercih edilen bir yöntemdir [44].

TBA yöntemi, örüntü tanıma sistemlerinde yaygın olarak kullanılmakta olan istatistiksel bir yöntemdir. Görüntü işleme, görüntü tanıma sistemlerindeki yüksek performansı sağlaması TBA’nın tercih edilmesi sebeplerinden birisi olmuştur.

TBA’nın çalışma mantığına bakıldığında veri kümelerinde, verilerin dağılımını en iyi şekilde veren ve temsil edebilen vektörü bulmaktır. Bu vektörler, veri kümelerine karşılık gelen kovaryans matrislerinin öz vektörleri olarak adlandırılırlar. Bu vektörlere özyüzler de denilmektedir [45].

TBA’nın uygulamasına bakıldığında öncelikle, veritabanındaki veri kümelerinin $\Gamma_1, \Gamma_2, \Gamma_3, \dots, \Gamma_m$ gibi vektörlere dönüşümü sağlanır. Burada uygulanacak veri kümesinin türüne göre x veri kümesi elde edilir. Örneğin uygulanacak veri kümesi $N \times N$ boyutunda bir resim ise x veri kümesi $M \times N^2$ ’lik vektörlerden oluşur. Uygulanacak veri kümesi M adet ses verisinden oluşuyor ise bu durumda x veri kümesi $M \times N$ ’lik vektörlerden oluşur. Oluşturulan x veri kümesindeki vektörlerden Eşitlik 5.6’daki ortalama vektör hesaplanır [46].

$$y = \frac{1}{M} \sum_{i=1}^M \Gamma_i \quad (5.6)$$

Bu şekilde hesaplanan ortalama vektörler daha önce oluşturulan x veri kümesindeki vektörlerden çıkarılarak Eşitlik 5.7.'deki fark vektörleri hesaplanır.

$$f_i = r_i - y \quad (5.7)$$

Hesaplanan fark vektörlerinden $A = [f_1, f_2, f_3, \dots, f_m]$ matrisi elde edilir. $N \times M$ 'lik A matrisi transpozesi ile çarpılarak Eşitlik 5.8.'deki kovaryans matrisi hesaplanmış olur.

$$C = A \cdot A^T \quad (5.8)$$

Veri kümesinin boyutu büyüdükçe hesaplanan kovaryans matrisin boyutu da büyük olacağından dolayı özdeğer ve özvektörlerin hesaplama işlemleri zorlaşacaktır. Bu nedenle oluşturulacak kovaryans matrisin boyutunu küçük tutmak gerekmektedir. Bunun için de Eşitlik 5.9.'da gösterilen L kovaryans matrisi hesaplanır.

$$L = A^T \cdot A \quad (5.9)$$

TBA için hesaplanan kovaryans matrisin özdeğer ve özvektör ayrışması için Eşitlik 5.10'daki eşitleme işlemi yapılır.

$$Cv = \lambda v \quad (5.10)$$

Burada C kovaryans matris, λ skaler bir değer, v sıfırdan farklı bir sütun vektör olmak üzere, Eşitlik 5.10'u sağlayan λ sayısı C 'nin özdeğeri, v ise λ ile ilişkili özvektördür.

Özdeğerlerin büyükten küçüğe doğru sırlanması sonucunda, sıralı bir şekilde oluşturulan P adet özvektör, bir matrisin sütunlarını oluşturacak şekilde dizilecek olursa $W = [w_1, w_2, w_3, \dots, w_p]$ özvektörleri elde edilmiş olur. Son olarak da veri kümelerinin özniteliklerini hesaplama işlemi Eşitlik 5.11'de gösterildiği gibi hesaplandıktan sonra öznitelik çıkarma işlemi tamamlanmış olmaktadır.

$$y^i = W^T \cdot x^i, \quad i = 1, 2, 3, \dots, M \quad (5.11)$$

Öznitelikler belirlenerek eğitim aşaması tamamlanmıştır. Bir sonraki aşama, sınıflandırma aşamasıdır. Sınıflandırma aşaması, eğitimde kullanılmayan bir test görüntüsü verildiğinde, eğitim aşamasındaki öznitelikler kullanılarak test vektörünün, eğitim aşamasındaki sınıflardan birine atanmasıdır. TBA temeline uyumlu ve yüksek performans veren en yaygın sınıflandırıcı, en yakın komşu sınıflandırıcısıdır. Bu sınıflandırıcıda öncelikle hangi sınıfa ait olduğu bilinmeyen test vektörünün özniteliği hesaplanır ve eğitim aşamasında belirlenen özniteliklerle karşılaştırılır. Test vektörünün özniteliğine en yakın uzaklığa sahip olan eğitim örneğinin sınıfı, test vektörünün sınıfı olarak belirlenir.

Öznitelik belirleme işlemi tamamlandıktan sonra sistemin test aşamasında sınıflandırma işlemine geçilir. Bu aşamada bir verinin eğitilen verilerin öznitelikleri kullanılarak herhangi bir sınıfa ait olma ya da ait olmama işlemleri yapılır. TBA' da en yüksek performans veren sınıflandırma Öklid uzaklığı hesaplanarak yapılan sınıflandırmadır.

$$U_{\text{öklid}} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (5.12)$$

Burada eğitim için hesaplanan p öznitelik vektörleri ile test için kullanılacak verinin hesaplanan q öznitelik vektörleri arasındaki uzaklıklar Eşitlik 5.12'de gösterilen Öklid hesabına göre bulunarak minimum uzaklığa sahip eğitim verisi ve test verisinin sınıfı tespit edilmiş olur [44 - 46].

5.3. K-ORTALAMA (K-MEANS)

Verileri sınıflandırmak için kullanılan en eski ve iyi bilinen kümeleme algoritmalarından biri K-Means algoritmasıdır. Genel mantığı girilen n adet veri nesnesinin nitelik ve özelliklerine göre k adet sınıfa ayrılmasıdır. Bu işlem girilen verilerin en yakın veya benzer oldukları kümenin merkezi etrafına yerleştirilmesi esasına dayanır [47].

Algoritmaya K-Means denilmesinin sebebi, algoritmanın başlangıçta sabit bir k sayısına ihtiyaç duymasından kaynaklanmaktadır. K sabiti küme sayısını ifade eder ve nesnelerin birbirlerine yakınlıklarına göre kaç sınıfa ayrılacaklarını belirler. Bu durumda k kümeleme işlemi bitinceye kadar değişmeyen bir sabittir. Veri setine en uygun küme sayısı deneme-yanılma yoluyla bulunmak zorundadır. Algoritma uygulanmadan önce veriler gürültü ve istisna durumlardan arındırılmalıdır. Yüksek orandaki bu gürültü ve istisnalar algoritma ile hesaplanan ortalamayı değiştireceği için yanlış sonuçlar üretilir. K-Means algoritması gürültü ve istisnalara çok duyarlıdır. Sınıflandırmaya giren her nesne yalnızca bir kümeye dâhildir. K-Means algoritması sadece sayısal veri setleri üzerinde işlem yapmamıza olanak sağlar [48].

K-Means algoritmasının işlem basamakları şöyledir:

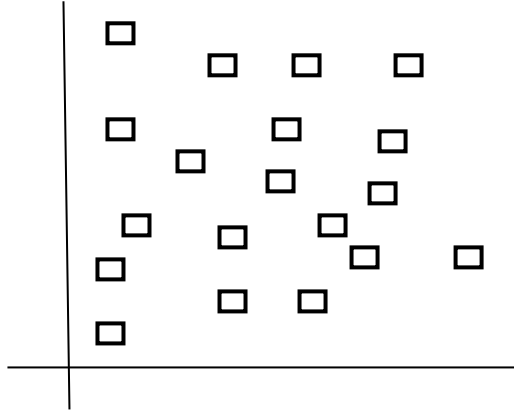
1.Adım: Algoritmanın ilk adımında kümelemek istediğimiz veri setimizin kaç sınıfa ayrılacağını gösteren k sabiti belirlenir. Fakat sınıflandırdığımız küme elemanlarının değerleri birbirine çok yakın olursa bu seçim rastgele de yapılabilir ve birbirine uzak elemanlar küme merkezi olarak seçilebilir.

2.Adım: Her nesnenin seçilen merkez noktalara olan uzaklığı hesaplanır. Elde edilen sonuçlara göre tüm nesneler k adet kümeden kendilerine en yakın olan kümeye yerleştirilir. Kullanılan veri setindeki elemanların kümelerin merkezlerine olan uzaklıklarının hesaplanması için birçok geometrik yöntem kullanılmaktadır. Bunlardan biri, kümelerin arasındaki sınırların belirlenerek elemanların hangi kümeye yakın olduğunun bulunmasıdır. Bu yöntemde kümelerin merkez noktaları bir doğru ile birleştirilerek daha sonra bu doğruyu dik kesen başka bir doğru ile kümelerin kapsama alanları belirlenir. Kullanılan bilgiler iki boyuttan daha fazla boyutlara ulaştığında doğru yerine düzlemler de kullanılır. Noktalar arası uzaklık hesaplanmasında kullanılan diğer bir yöntem Eşitlik 5.13'de gösterilen Öklit bağıntısıdır. Öklit bağıntısı çok yaygın ve tercih edilen bir yöntemdir. Bu bağıntıda noktalar arası uzaklıklar hesaplanarak noktaların merkeze olan yakınlığı dikkate alınır.

$$d(i,j) = \sqrt{\sum_{k=1}^p (X_{ij} - X_{jk})^2} \quad (5.13)$$

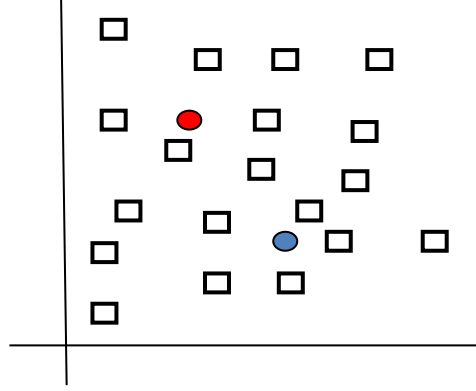
3.Adım: Her bir kümeye eklenen yeni elemanlar ile küme elemanlarının ağırlıkları hesaplanarak yeni bir küme merkezi bulunur. Yeni merkezlerin bulunmasından sonra tekrar elemanların merkezlere olan uzaklıkları hesaplanarak elemanlar kendilerine en yakın merkezin bulunduğu kümelere dâhil edilir. Bu şekilde optimum değerler elde edilene kadar her bir döngüde elemanlar başka kümelere dahil edilebilir.

4.Adım: Merkez noktalar değişmeyene kadar 2. ve 3. adımlar tekrar edilir. K-Means algoritması uygulanan bir veriseti Şekil 5.3’de gösterilmiştir.



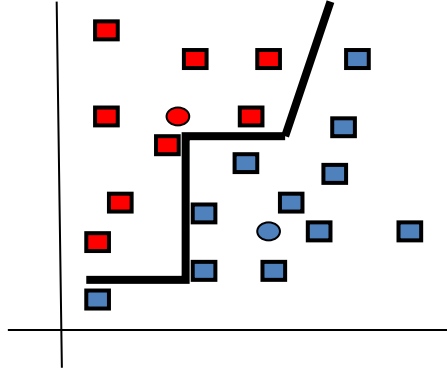
Şekil 5.1 K-Means uygulanmadan önceki karmaşık veri seti.

Şekil 5.4’de verilen ve uzayda koordinatları kodlanmış olan örnekler için iki adet hedef küme tanımlanmıştır. Yani iki ayrı sınıf ve bu sınıflara ait olan karakteristikler belirlenmiştir.(k=2)



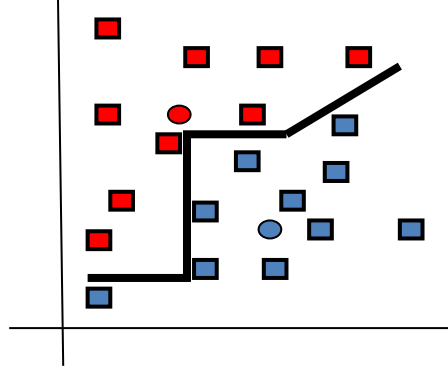
Şekil 5.2 Sınıflandırılacak iki farklı veri türü için rastgele seçilmiş merkezler.

Bu sınıf tanımlarının uzaklıklarına göre bütün örnekler Şekil 5.5.'de gösterildiği gibi hangi sınıfa daha yakınsa o sınıfa eklenir.



Şekil 5.3 İki farklı renkte sınıflandırılan veriler.

Daha önceden sınıflandırılan örneklerin merkezleri tekrar hesaplanır ve yeni merkezlere göre Şekil 5.6'de gösterildiği tekrar sınıflandırma işlemi yapılır.



Şekil 5.4 Hesaplanan merkezlere göre sınıflandırılan veriler.

Merkezleri hareket ettirdikten sonra örneklerden bazıları yeni merkezlere daha yakın olabilir. Buna göre örnek kümeleri sınıflandırılması yeniden güncellenir. Bu son iki işlem (merkez bulma ve tekrar sınıflandırma işlemleri) yeni bulunan merkezler bir önce bulunan merkez değerleriyle aynı olana kadar tekrarlanarak sınıflandırma işlemi aynı işlemlerle tekrar devam eder [48].

5.4. YAPAY SİNİR AĞLARI (YSA)

Bu bölümde, yapay sinir ağları hakkında kısa genel bilgi verildikten sonra, ses tanıma uygulamalarında kullanılmış başarılı algoritmalar incelenmiştir.

YSA, insan beyninin yeni bilgiler öğrenme ve öğrenilmiş bilgiler ile yeni durumlara karşı cevap oluşturabilme, keşfedebilme gibi yetenekleri otomatik olarak gerçekleştirmek amacı ile geliştirilen hesaplama sistemleridir. YSA'ların her çalışmada öne çıkarılan özelliklerine göre tanımları yeniden yapılmaktadır [49].

YSA, basit işlem birimlerinden oluşan, deneysel bilgi biriktirmeye yönelik doğal bir eğilimi olan ve bunların kullanımına imkan veren, yoğunlukla paralel dağıtılmış bir işlemcidir.

Bu işlemci iki şekilde beyne benzer.

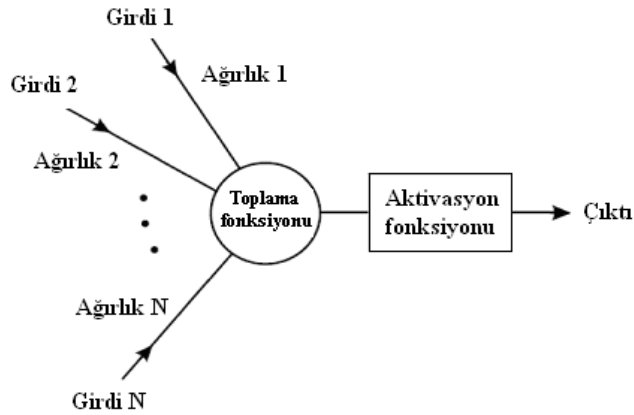
- 1) Bilgi, ağ tarafından bir öğrenme süreciyle çevreden kazanılır.

2) Bilgi biriktirmek için sinaptik ağırlıklar olarak da bilinen nöronlar arası bağlantı güçleri kullanılır [50].

Özetle YSA, insan beyninin sinir hücreleri vasıtasıyla öğrenme ve tepki verme mekanizmasını model alan doğrusal olmayan ve yüksek karmaşıklığa sahip bir bilgi işleme sistemi olarak düşünülebilir.

5.4.1. Temel YSA Bileşenleri ve Yapısı

Şekil 5.7.'de yapısı görülen bir biyolojik nöron, hücre gövdesi (soma), bu gövdeyi çevreleyen dendritler ve gövdeye bağlı aksondan oluşmaktadır. Biyolojik nöronlar, dendritleri vasıtasıyla diğer biyolojik nöronlardan gelen uyarıları alırlar. Bu uyarılar, hücre içindeki elektrokimyasal bağlantıları sağlayan sinapslarla ölçeklendirildikleri şekilde gövdeye iletilirler. Hücre boyunca iletilen sinyaller aksona ulaştığında belirli bir esik değerinin üzerindeyse nöron aktif, altında YSA nöron pasif olur [51].



Şekil 5.5 Temel yapay sinir ağı hücresi yapısı [51].

Girdiler: Dış çevreden ya da sistemdeki diğer hücrelerden gelen bilgilerdir.

Ağırlıklar: Bir YSA hücresine etki eden girdilerin matematiksel ağırlık katsayılarıdır ve bağlantılı oldukları girdilerin hücreleri ne ağırlıkta etkileyeceğini belirler. Her bağlantı noktası için ayrı ayrı ağırlık değerleri vardır. Böylece ağırlıklar, girdilerin hücrenin sergileyeceği davranışları belirlemede rol oynamaktadır.

Toplama fonksiyonu: Toplama fonksiyonu genellikle YSA hücresine gelen girdiler ile bu girdilere ait ağırlıkların çarpımlarının toplamını ifade eder. Bu fonksiyon, çarpımlar toplamının yanı sıra maksimum, minimum, normalizasyon benzeri hesaplamaları içerebileceği gibi ağ tasarımcısının kendi tanımladığı her hangi bir fonksiyon da olabilmektedir.

Aktivasyon fonksiyonu: Bu fonksiyon, toplama fonksiyonu ile elde edilen sonucu seçilen bir aktivasyon işleminden geçirerek YSA hücresinin çıktısını belirler.

Aktivasyon fonksiyonları yapı itibariyle doğrusal olabildiği gibi, günlük hayatta karşılaşılan karmaşık ve doğrusal olmayan problemlere çözüm getirilebilmesi için sıklıkla doğrusal olmayan yapıda da olabilirler [51,52].

Tasarlanan yapıya en uygun aktivasyon fonksiyonu deneme-yanılma yoluyla ya da tecrübe ile en uygun sonucu verecek şekilde seçilmelidir. Çünkü bu fonksiyonlar, toplam fonksiyonu sonuçlarını seçilen aktivasyon fonksiyonuna uyarlayarak katmanların çıkışlarını belirli değerlere sınırlandırır ve atandıkları katmandaki tüm YSA hücrelerine aynı şekilde etkiler. Ayrıca ağır yapısına göre de aktivasyon fonksiyonu önem arz etmektedir.

Çıktılar: Aktivasyon fonksiyonu sonucu elde edilen değer, YSA hücresinin çıktısıdır. Çıktılar, diğer YSA hücrelerine girdi olabileceği gibi dış ortama çıktı olarak da gönderilebilirler. Ancak burada dikkat edilmesi gereken nokta, bir YSA hücresinin birden fazla girdisi olabilmesine karşın sadece bir çıktısının olmasıdır.

Tek bir YSA hücresi basit problemlere çözüm sunabilmiştir ancak özel veya (EXOR) gibi daha karmaşık işlemler için yetersiz kalmıştır. Bu sebeple çok katmanlı ve çok hücreli ağ yapıları geliştirilmiştir. Bu yapıda üç katman bulunmaktadır. Giriş katmanı, gizli katmanlar ve çıkış katmanı.

Giriş katmanı, dış dünyadan gelen bilgilerin alındığı ve ağa sevk edildiği katmandır. Bu katmanda bilgi işleme yapılmamaktadır.

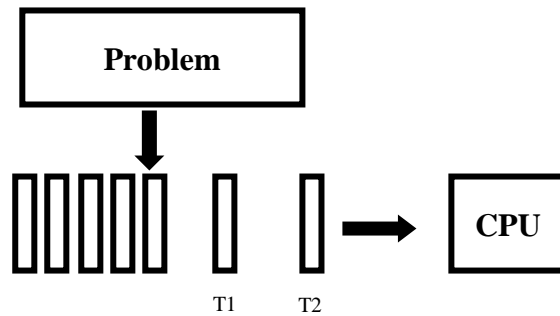
Gizli katman(lar), girdi katmanı ile çıktı katmanı arasında yer alır. Giriş katmanından gelen bilgiler, gizli katmanlarda çeşitli algoritmalarla işlenerek çıkış katmanına gönderilir. Bir YSA' da, gerçekleştirilmek istenen uygulamaya göre bir veya daha fazla gizli katman bulunabilir.

Çıkış katmanı, gizli katman(lar)dan gelen bilgileri işleyerek YSA' nın çıkışını oluşturan katmandır. Bir YSA' da tek çıkış katmanı bulunur. Ancak çıkış katmanında birden fazla nöron bulunabilir.

Çok katmanlı ağı eğitiminde bilinen girdi ve hedeflenen çıktılar ile eğitim yapılır. Eğitimde beklenen çıktıların elde edilebilmesi için ağırlıklar ve toplama fonksiyonundaki eşik değer güncellenir. Güncelleme beklenen çıktı değerine karşılık ağı verilen girdi için ürettiği çıktı arasındaki hata değerine göre hesaplanır [51,52].

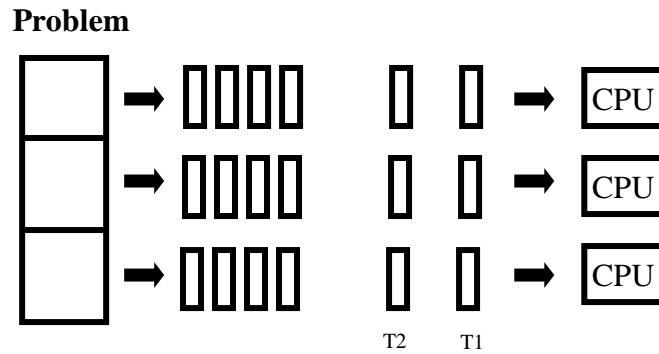
5.5. PARALEL PROGRAMLAMA

Paralel programlama işleminin mantığını daha iyi anlatabilmek için seri programlama hakkında biraz bilgi verilecek olursa; seri programlama bir işlemci ile tek bir bilgisayar kullanılarak işlemlerin gerçekleştirilmesi olayına denir [53]. Bu sebeple bu türden çalışan bir sistemde aynı anda sadece bir işlem gerçekleştirilebilmektedir. Bu yüzden birden fazla işlemin aynı anda çalıştırılması düşünülen durumlara uygun değildir. Seri programlamanın mantığı Şekil 5.8'de gösterilmiştir.



Şekil 5.8 Seri programlamanın çalışma mantığı [53].

Seri programlamada aynı anda sadece bir işlemin yapılmasından dolayı bir işlem bitmeden diğer işlemin yapılması mümkün olmamaktadır. Kısaca işlemler birbiri ardından sıra gerçekleştirilmekte ve her bir işlem için farklı zamanlara ihtiyaç duyulmaktadır. Sistemde yalnızca bir bilgisayarın bulunması sebebiyle bilgisayarın performansı düşmekte ve arzu edilen düzeyde verimli çalışmamaktadır. Performansın ve verimliliğin yükseltilebilmesi ve hedeflere daha rahat ulaşılabilmesi için paralel programlamanın kullanılması daha uygun düşmektedir. Paralel programlama, aynı anda birden fazla bilgisayarın aynı işlemin yapılabilmesi için işe koşulması olayıdır. Seri programlamada tek bir bilgisayar ve tek bir işlemcinin kullanılmasından kaynaklanan ve verim düşüklüğüne sebep olan durum paralel programlamada ortadan kaldırılmıştır. Çünkü işlemler, paralel programlamada her bir bilgisayardaki her bir işlemci veya bir bilgisayardaki çoklu çekirdek teknolojisine sahip bir işlemci üzerinden yürüdüğünden bir problem parçalanarak gerçekleştirilir. İşlemin her parçası aynı anda gerçekleştirilmeye çalışılır. Bu sebeple bilgisayarın performansı seri programlamaya göre daha yüksektir. Paralel programlama mantığı Şekil 5.9'de gösterilmiştir [53].



Şekil 5.9 Paralel programlamanın çalışma mantığı [53].

Paralel programlamada gerçekleştirilmesi gereken işlem önce belirli parçalara ayrılır ve her bir parça aynı anda ya aynı ağ üzerinde kurulu bulunan bilgisayarlara veya aynı bilgisayarın değişik işlemcilerine iletilir ve işlem paralel olarak gerçekleştirilir. Bu durum yapılan işlemin özelliğine göre hız, verimlilik ve performans olarak sonuca yansır.

Çizelge 5.1'de ve Çizelge 5.2'de verilen seri ve paralel hesaplama örneğinde 6 adet sayının toplanmasına bakılacak olursa ilk bakışta göze çarpan adım sayısındaki

azalmadır. Aynı zamanda seri hesaplama göre paralel hesaplamaların 1,75 kat daha hızlı olarak çalıştığı görülmektedir. Örnekte iki işlemcili bir bilgisayar üzerinde belirtilen işlem gerçekleştirilmiştir [54].

Çizelge 5.1. Seri hesaplama örneği [53].

$1+2+3+4+5+6$
$3+3+4+5+6$
$6+4+5+6$
$10+5+6$
$15+6$
21

Çizelge5.2. Paralel hesaplama örneği [53].

$1+2+3+4+5+6$
$3+7+5+6$
$10+11$
21

Bilgisayarların dizaynında seri programlama sistemleri yerine daha fazla paralel programlama sistemlerinin kullanılmasının en önemle gerekçesi işlemlerde yoğun hesaplamaların bulunduğu uygulamalarda uygulama zamanının azaltılmasından kaynaklanmaktadır. Bir paralel programlama sistemindeki uygulama zamanı değişik faktörlere bağlı olarak değişmektedir. Bilgisayarlarda aranan en önemli özellikler arasında yer alan hızlanma ve verimlilik ölçümleri paralel programlamada avantaj olarak karşımıza çıkmaktadır. Bu özellikler seri programlamanın uygulama zamanlarıyla karşılaştırıldıklarında avantajlar sağlamaktadır [53,54].

Performans: Seri programlama ile paralel programların karşılaştırılmasında yapılan çalışmalarda uygulama sürelerine dikkat edilir. Sürenin kısalması performansın iyi olarak kabul edilmesini sağlar.

Hızlanma(Hız):. Seri programlama ve paralel programlama arasındaki farklardan birisi de hızlanmadır. Bir işlemin gerçekleştirilebilmesindeki zaman kısalığı

bilgisayarın hızı ile ilgili olup hızlanma ile ilgili Eşitlik 5.14’de gösterilen formülden de anlaşıldığına göre seri programlama ile paralel programlamanın hızlanmaları karşılaştırıldığında oran ne kadar küçük ise bilgisayarın hızlanmasının o kadar iyi olduğu sonucuna ulaşılır [53].

$$Hız = \frac{SeriÇalışmaSüresi}{ParalelÇalışmaSüresi} \quad (5.14)$$

BÖLÜM 6

UYGULAMA SONUÇLARI

Yapılan uygulamada eğitim ve test verileri olarak 25 kişiden 16'ar adet ses verisi alınmıştır. Toplamda 200 adet ses verisi eğitim amaçlı, 200 adet ses verisi de test amaçlı kullanılmıştır. Uygulama için geliştirilen yazılım için MATLAB 7.9.0 kullanılmıştır. Uygulamaların çalıştırılması için 2 çekirdekli, 2,53 Ghz işlemcili, 4 GB RAM belleğe sahip bir bilgisayar kullanılmıştır.

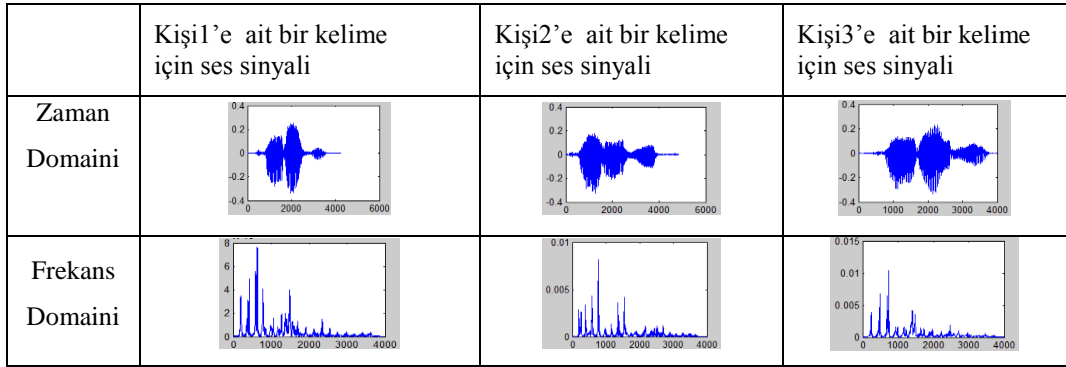
Eğitim ve test amaçlı kullanılan sesler için Türkçe kelimeler ve şehir isimlerinden yararlanılmıştır. Bu kelimeler seçilirken de kelimelerin ağızın hareketlerine göre okunuş biçimlerine dikkat edilmiştir. Ses kayıtları için kullanılan kelimeler Çizelge 6.1.'de verilmiştir.

Çizelge 6.1. Ses tanımada kullanılan kelimeler.

Eğitim verisi	Ankara	İstanbul	İzmir	Adana	Konya	Malatya	Şanlıurfa	Elazığ
Test Verisi	Bursa	Niğde	İzmit	Aydın	Van	Rize	Karaman	Sinop

Seslerin kaydedilmesi esnasında ses kayıtlarında kullanılan filtreleme özelliğine sahip profesyonel bir kayıt cihazı kullanılmıştır. Ses kayıt programı olarak, geliştirilen yazılımdaki ses ekleme modülü kullanılmıştır. Sesleri örnekleme frekansı 8000 Hz olarak belirlenmiş ve kayıtlar mono olarak yapılmıştır. Seslerin bilgisayar ortamına kayıt formatı olarak .wav dosyalama biçimi tercih edilmiştir. Ses kayıtlarının uzunluğu iki saniye ve tek kelimedenden oluşmaktadır. Bu nedenle elde edilen her bir kayıt 16000 örnekten oluşmaktadır. Yani her bir ses kaydı 16000X1 adet sütun vektörden veriler halinde oluşmaktadır.

Uygulamada kullanılan ses kayıtları 16000 gibi yüksek veri içerdiği için bu verilerle işlem yapılması hem zaman performansı açısından hem de sonuç performansları açısından zor olduğu için bu ses verilerini temsil edecek özellik vektörlerinin çıkarılması gerekmektedir. Elde edilen ses kayıtları ön işlemlerden geçirildikten sonra MFCC hesaplanarak her bir ses verisi için özellik vektörleri çıkarılmıştır. Ses tanımada kullanılan ses kayıtlarında örnekler Şekil 6.1.'de verilmiştir.

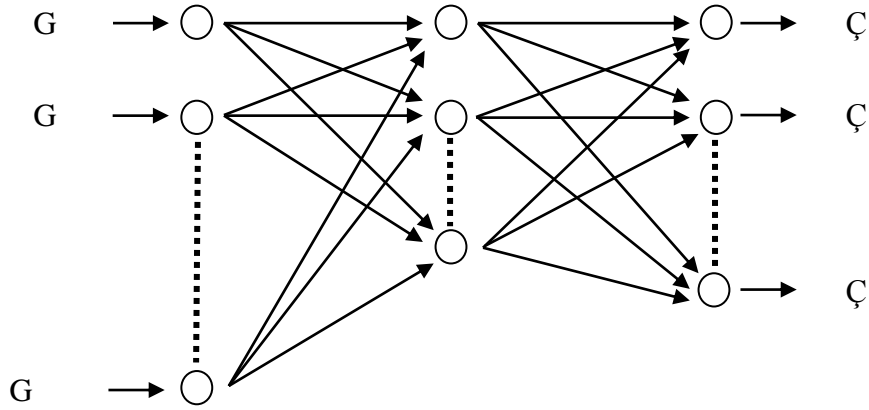


Şekil 6.1 Bir kelimenin farklı kişilere ait ses grafikleri.

6.1. YSA İLE SES TANIMA İŞLEMİ

Yapılan uygulamada YSA yapısı olarak çok katmanlı algılayıcılar kullanılmıştır. Çok katmanlı ağlar, giriş katmanı, saklı katmanlar ve çıkış katmanından oluşmaktadır. Bu yapıda giriş katmanındaki nöron sayısı toplam ses verisi sayısına seçilir. Çıkış katmanı ise tanıma işlemi yapılacak kişi sayısına bakılarak seçilir. Saklı katmandaki kullanılan nöron sayısı ise test aşamasındaki başarı sonuçlarına göre değiştirilerek başarıya uygun nöron sayısı seçilmiştir. Bu çalışmada kullanılan YSA modelinde giriş katmanında 200 adet nöron, saklı katmanda 10 adet nöron ve çıkış katmanında ise 25 adet nöron kullanılmıştır. Kullanılan YSA'nın modeli Şekil 6.2.'de gösterilmiştir.

YSA, giriş katmanındaki hesaplanmış giriş vektörlerini alarak çıkıştaki sonuç vektörleri ile aralarındaki hata oranını bulur ve bu hata oranını minimize etmek için saklı katmanlarda bulunan nöronlar arasındaki ağırlık vektörlerinin sayısal değerlerini değiştirir.



Şekil 6.2. Ses Tanımda kullanılan YSA modeli.

Ses tanıma işleminde amaç çıkış katmanında yer alan nöronlar içerisinde test edilen kişiye ait nöronun çıkış değerinin 1 olmasını diğer kişilere ait nöron çıkış değerlerinin ise 0 olmasını sağlamaktır.

Ses tanıma işlemlerinde performansı etkileyen önemli unsurlardan biri de eğitim ve test süreleridir. Bu nedenle YSA modelinin başarısında eğitim ve test sürelerinin ölçülmesi işlemi de yapılmıştır. Kullanılan YSA modeline ait parametre verileri Çizelge 6.2.'de verilmiştir.

Çizelge 6.2. YSA modeline ait parametre verileri.

Katman Sayısı	3		
Katmandaki Nöron Sayısı	Giriş	Gizli	Çıkış
	200	10	25
Tasvir Fonksiyonu	Lineer	Sigmoid	Sigmoid
Başlangıç Ağırlıkları	-1 ile 1 arasında belirlenmiştir.		
Öğrenme Oranı	0.300000		
Momentum Faktörü	0.500000		
Maks. İterasyon Sayısı	400		
Minimum Eğitim Hatası	0.002000		

YSA için deneysel çalışmada giriş, saklı ve saklı katmandaki nöron sayısı 10 ve katmanlardaki transfer fonksiyonlar hiperbolik tanjant sigmoid olarak seçilmiştir. Uygulama 400 iterasyon ile çalıştırılmıştır. Uygulama sonuçlarına ait bilgiler Çizelge 6.3. ve Çizelge 6.4.'de verilmiştir.

Çizelge 6.3. Filtrelenmemiş seslerin YSA ile yapılan tanıma işlemi sonuçları.

Kişi Sayısı	Eğitim Süresi (sn)	Paralel Programlama Eğitim Süresi (sn)	Test Süresi (sn)	Paralel Programlama Test Süresi (sn)	Hız Oranı	Hata Değeri	Başarı Oranı (%)
10	359	212	2,6	1,4	1,69	0,18	96
14	461	284	2,9	1,6	1,62	0,22	96
18	519	341	3,1	1,7	1,52	0,26	95
20	571	380	3,2	1,8	1,50	0,39	95
23	658	450	3,5	1,9	1,46	0,48	94
25	739	543	3,6	2,1	1,36	0,55	94

Çizelge 6.4. Filtrelenmiş seslerin YSA ile yapılan tanıma işlemi sonuçları.

Kişi Sayısı	Eğitim Süresi (sn)	Paralel Programlama Eğitim Süresi (sn)	Test Süresi (sn)	Paralel Programlama Test Süresi (sn)	Hız Oranı	Hata Değeri	Başarı Oranı (%)
10	337	206	2,4	1,2	1,63	0,06	99
14	418	259	2,9	1,3	1,61	0,08	99
18	476	323	3,0	1,6	1,47	0,09	99
20	524	371	3,1	1,7	1,41	0,11	98
23	597	473	3,3	1,8	1,26	0,14	98
25	649	540	3,5	1,9	1,20	0,16	98

YSA ile yapılan ses tanıma işlemlerinin sonuçlarında, filtrelenmemiş seslerin tanıma oranının filtrelenmiş seslere göre daha düşük olduğu görülmektedir. Kişi sayısının artması tanıma işleminde filtrelenmiş sesler için tanımaya çok fazla olumsuz etki göstermemişken, filtrelenmemiş sesler için olumsuz etki göstermektedir. Filtrelenmemiş seslerin YSA işlemleri için hesaplanan süreleri, filtrelenmiş seslerin YSA işlemleri için hesaplanan sürelerine göre daha fazla olduğu tespit edilmiştir.

Paralel programlamanın işlem sürelerini kısaltmada belirgin bir etkisi olduğu ve işlem sayısı arttıkça performansının ortaya çıktığı görülmektedir.

6.2. DVM İLE SES TANIMA İŞLEMİ

Yapılan uygulamada, ses tanıma işlemi için MFCC hesaplanarak özellik vektörleri çıkarılan ses verileri DVM sınıflandırıcısına eğitim aşamasında giriş veri kümesi olarak verilmiştir. DVM sınıflandırıcı, sistemin modellenmesi için gerekli katsayıyı hesaplaması yaptıktan sonra sistemin modellenmesi tamamlanmış ve eğitim hataları bulunmuştur. Uygulamada DVM sınıflandırıcısının çekirdek fonksiyonu olarak Radyal Tabanlı Sigma Genişlik Fonksiyonu (RBF) kullanılmıştır. RBF parametrelerinin aldığı değerlere göre sistemin eğitim ve test hataları değişken gösterdiğinden dolayı RBF parametreleri, en iyi sonucu verecek şekilde deneme yoluyla bulunmuştur.

Uygulamanın tasarımında her bir ses verisine ait özellik vektörü için bir adet destek vektör makinesi kullanılmıştır. Bu şekilde uygulama 25 adet ses verisi için 25 adet destek vektör makinesinden oluşmaktadır. Eğitim aşaması tamamlandıktan sonra test aşaması için herhangi bir kişiye ait ses verisi test verisi olarak sisteme verildiğinde o kişiye ait destek vektör makinesi +1 çıkışı üretmiş, diğer kişilere ait destek vektör makineleri ise -1 çıkışı üretmişlerdir. Bu durum dışında oluşan bütün çıkışlar tanıma işleminin başarısız olduğu sonucunu vermektedir. Uygulama sonuçlarına ait bilgiler Çizelge 6.5 ve Çizelge 6.6'de verilmiştir.

Çizelge 6.5. Filtrelenmemiş seslerin DVM ile yapılan tanıma işlemi sonuçları.

Kişi Sayısı	Eğitim Süresi (sn)	Paralel Programlama Eğitim Süresi (sn)	Test Süresi (sn)	Paralel Programlama Test Süresi (sn)	Hızlandırma Oranı	Hata Değeri	Başarı Oranı (%)
10	99	54	0,6	0,1	1,83	0,06	97
14	113	62	0,8	0,3	1,82	0,07	96
18	118	65	0,9	0,4	1,81	0,09	94
20	129	74	1,1	0,6	1,74	0,13	92
23	136	78	1,3	0,7	1,73	0,15	91
25	145	88	1,4	0,9	1,65	0,19	90

Çizelge 6.6. Filtrelenmiş seslerin DVM ile yapılan tanıma işlemi sonuçları.

Kişi Sayısı	Eğitim Süresi (sn)	Paralel Programlama Eğitim Süresi (sn)	Test Süresi (sn)	Paralel Programlama Test Süresi (sn)	Hızlandırma Oranı	Hata Değeri	Başarı Oranı (%)
10	97	51	0,5	0,1	1,89	0,03	99
14	106	57	0,6	0,2	1,85	0,03	99
18	108	60	0,8	0,3	1,79	0,04	98
20	111	63	1,0	0,4	1,76	0,04	97
23	116	66	1,1	0,6	1,75	0,04	96
25	121	70	1,3	0,8	1,71	0,05	95

DVM ile yapılan ses tanıma işlemlerinde, filtrelenmemiş seslerin tanıma oranının filtrelenmiş seslere göre daha düşük olduğu görülmektedir. Filtrelenmemiş seslerin DVM işlemleri için hesaplanan süreleri, filtrelenmiş seslerin DVM işlemleri için hesaplanan sürelerine göre biraz daha fazla olduğu anlaşılmaktadır. Paralel programlamanın işlem sürelerini kısaltmada belirgin bir etkisi olduğu ve işlem sayısı arttıkça performansının ortaya çıktığı görülmektedir.

6.3. TBA İLE SES TANIMA İŞLEMİ

Yapılan uygulamada ön işlemlerden geçirilmiş ses verilerinin TBA algoritması ile özdeğerleri ve özvektörleri hesaplanmıştır. Hesaplanan bu değerler daha sonra test

aşamasında tanıma işlemlerinde kullanılan ses verilerinin özdeğerleri ve bu özdeğerler kullanılarak hesaplanan ağırlık vektörleri karşılaştırılarak minimum uzaklığa sahip ses verisinin bulunması sağlanmıştır. Ağırlıkların karşılaştırılmasında Eşitlik 5.2.' de verilen Öklid uzaklığı formülü kullanılmıştır.

Karşılaştırma işlemlerinde hangi ses verilerinin tanındığı, hangi ses verilerinin tanınmadığının tespiti, hesaplanan uzaklıkların belirlenen eşik değerin altında veya üstünde kalmasına bakılarak yapılmıştır. En uygun eşik değeri en uygun başarı oranının yakalanmasına kadar deneme yolu ile bulunmuştur. Uygulama sonuçlarına ait bilgiler Çizelge 6.7 ve Çizelge 6.8' de verilmiştir.

Çizelge 6.7. Filtrelenmemiş seslerin TBA ile yapılan tanıma işlemi sonuçları.

Kişi Sayısı	Eğitim Süresi (sn)	Paralel Programlama Eğitim Süresi (sn)	Test Süresi (sn)	Paralel Programlama Test Süresi (sn)	Hız Oranı	Ortalama Uzaklık Değeri	Başarı Oranı (%)
10	7	5	0,52	0,36	1,27	246,8	92
14	9	7	0,56	0,45	1,26	287,6	90
18	11	9	0,61	0,56	1,24	321,5	86
20	14	11	0,65	0,59	1,23	367,2	86
23	18	14	0,78	0,65	1,22	398,4	85
25	20	16	0,98	0,83	1,20	421,6	83

Çizelge 6.8. Filtrelenmiş seslerin TBA ile yapılan tanıma işlemi sonuçları.

Kişi Sayısı	Eğitim Süresi (sn)	Paralel Programlama Eğitim Süresi (sn)	Test Süresi (sn)	Paralel Programlama Test Süresi (sn)	Hız Oranı	Ortalama Uzaklık Değeri	Başarı Oranı (%)
10	6	4	0,48	0,39	1,60	135,9	95
14	8	5	0,55	0,42	1,53	149,6	95
18	10	7	0,59	0,46	1,42	196,3	95
20	12	9	0,63	0,49	1,33	199,2	94
23	14	11	0,70	0,55	1,27	225,8	94
25	16	13	0,82	0,63	1,23	234,4	93

TBA ile yapılan ses tanıma işlemlerinde, filtrelenmemiş seslerin tanıma oranının filtrelenmiş seslere göre daha düşük olduğu görülmektedir. Filtrelenmemiş seslerin TBA işlemleri için hesaplanan süreleri, filtrelenmiş seslerin TBA işlemleri için hesaplanan sürelerine göre biraz daha fazla olduğu anlaşılmaktadır. Paralel programlamanın işlem sürelerini kısaltmada belirgin bir etkisi olduğu ve işlem sayısı arttıkça performansının ortaya çıktığı görülmektedir.

6.4. K-MEANS İLE SES TANIMA

Yapılan uygulamada ön işlemlerden geçirilmiş Mel-kepstrum katsayıları hesaplanmış ve normalizasyon işlemi uygulanmış ses verilerinin K-Means algoritması ile sınıflandırma işlemi yapılmıştır. Sınıflandırılacak küme sayısı kişi sayısı kadar belirlenmiştir. Başka bir deyişle 25 farklı kişiye ait ses tanıma işlemi gerçekleştirildiği için k sabiti 25 adet olarak alınmıştır. Sınıflandırma işlemi 6 iterasyonda istenilen sonuçları verecek yapıya gelmiştir. Daha sonra test için kullanılan ses verileri tek tek K-Means algoritmasına verilerek seslerin hangi sınıfa ait olduğuna dair tahminleme yöntemi ile ses tanıma işlemi tanıma gerçekleştirilmiştir.

Karşılaştırma işlemlerinde hangi ses verilerinin tanındığı, hangi ses verilerinin tanınmadığının tespiti, hesaplanan uzaklıkların hangi kişiye daha yakın olduğuna bakılarak yapılmıştır. Minimum uzaklık hangi kişiye ait ise tanıma işlemi yapılacak

sesin sınıfı da o kişinin bulunduğu sınıf olarak tespit edilmiştir. Uygulama sonuçlarına ait bilgiler Çizelge 6.9 ve Çizelge 6.10’ da verilmiştir.

Çizelge 6.9. Filtrelenmemiş seslerin K-Means ile yapılan tanıma işlemi sonuçları.

Kişi Sayısı	Eğitim Süresi (sn)	Paralel Programlama Eğitim Süresi (sn)	Test Süresi (sn)	Paralel Programlama Test Süresi (sn)	Hız Oranı	Ortalama Uzaklık Değeri	Başarı Oranı (%)
10	14	9	0,89	0,63	1,55	223,3	91
14	18	13	0,95	0,75	1,38	236,9	88
18	20	15	1,17	0,99	1,33	275,4	86
20	25	20	1,36	1,22	1,25	302,6	83
23	27	23	1,49	1,35	1,17	326,5	81
25	30	26	1,68	1,48	1,15	336,7	80

Çizelge 6.10. Filtrelenmiş seslerin K-Means ile yapılan tanıma işlemi sonuçları.

Kişi Sayısı	Eğitim Süresi (sn)	Paralel Programlama Eğitim Süresi (sn)	Test Süresi (sn)	Paralel Programlama Test Süresi (sn)	Hız Oranı	Ortalama Uzaklık Değeri	Başarı Oranı (%)
10	13	10	0,69	0,62	1,30	169,6	93
14	14	11	0,75	0,72	1,27	183,3	93
18	19	16	0,97	0,99	1,18	192,8	92
20	21	18	1,24	1,03	1,16	199,7	91
23	23	21	1,26	1,14	1,09	214,3	91
25	27	23	1,53	1,23	1,05	224,6	90

K-Means ile yapılan ses tanıma işlemleri sonucunda, filtrelenmemiş seslerin tanıma oranının filtrelenmiş seslere göre daha düşük olduğu görülmektedir. Filtrelenmemiş seslerin K-Means işlemleri için hesaplanan sürelerin, filtrelenmiş seslerin K-Means işlemleri için hesaplanan süreye göre biraz daha fazla olduğu anlaşılmaktadır. Paralel programlamanın işlem sürelerini kısaltmada belirgin bir etkisi olduğu ve işlem sayısı arttıkça performansının ortaya çıktığı görülmektedir.

BÖLÜM 7

SONUÇ VE ÖNERİLER

Kolay elde edilmesi ve işlenmesinin kolay olması nedeni güncel yaşamda birçok alanda ses sinyali kullanılmaktadır. Ses sinyalinin yaygın olarak kullanılması da ses tanıma, konuşma tanıma, konuşmacı tanıma alanlarında çok çeşitli çalışmalar yapılmasını mümkün kılmaktadır. Ses tanıma çalışmalarında ortaya çıkan ne önemli sorunlardan bir tanesi gürültülü ortamlarda ses tanıma işlemlerinin başarısız sonuçlar vermesidir. Bununla birlikte gürültüleri azaltmak amacıyla kullanılan filtreleme yöntemlerinin yetersizliği de ses tanıma işlemlerini etkileyen olumsuz faktörlerin başında gelmektedir. Ses tanıma işlemlerinde kullanılan yaygın yöntemlerin dışında farklı alanlarda kullanılan ve başarılı olan yöntemlerin, uygun koşullar sağlandığında ses tanıma işlemlerinde kullanılması ses tanıma işlemlerinin yaygınlığını artırmada etkili olacaktır. Bununla beraber gelişen teknoloji ile birlikte işlemlerin daha kısa zamanda yapılabilmesine olanak sağlayan Paralel Programlama yönteminin de ses tanıma alanında uygulanması, bu alanda yapılacak çalışmalara örnek teşkil edecektir.

Bu çalışmada çoğunlukla yüz tanıma sistemlerinde kullanılmış olan TBA algoritması ile ses tanıma işlemlerinde başarılı oldukları bilinen YSA, DVM ve K-Means yöntemlerinin ses tanıma sistemlerinde kullanımı ve başarısı incelenmiştir. Öncelikle farklı kişilerden alınan ses örnekleri ile bir ses veritabanı oluşturulmuştur. Bu ses veri tabanı kullanılarak TBA, YSA, SVM ve K-Means algoritmaları ile ses tanıma işlemi gerçekleştirilmiş ve tanıma performansları karşılaştırılmıştır. Ses tanıma işlemleri yapılırken seslerin hem filtrelenmemiş hem de filtrelenmiş şekilde tanıma işlemlerinde kullanılması filtrelemenin de tanıma ses tanıma işlemlerine etkisini değerlendirme imkanı vermiştir. Elde edilen sonuçlar TBA algoritmasının ses tanımada da yüksek başarı oranına sahip olduğunu göstermektedir. Filtrelenmemiş seslerin ses tanıma işlemlerinde tanımayı zorlaştırdığı ve ses tanıma oranını düşürdüğü yapılan deneysel çalışmanın sonuçlarında tespit edilmiştir. Ses tanıma

işlemlerinde başarılı bir tanıma işlemi için seslerin mutlaka iyi bir filtreleme algoritmasından ya da filtreleme cihazından geçirilmesi gerekmektedir.

Karmaşık ve uzun süren işlemleri daha kısa zamanda gerçekleştirebilmesi yönünden kullanılan paralel programlama yöntemi bu çalışmada başarı ile uygulanmıştır. İşlem sürelerini kısaltma yönünden başarılı bir performans gösteren bu yöntem algoritmaların başarı performanslarında herhangi bir olumsuzluk etki oluşturmamıştır. Kullanılan verilerin büyüklüğü ve yapılan işlemlerin yoğunluğunun artması paralel programlamanın performansını görmede etkili olmaktadır. Ses tanıma işlemlerinde seri programlamada uzun zaman gerektiren bir işlemin süresini paralel programlama ile çok daha kısa süreye indirmek yapılan bu çalışmada görülmüştür. Paralel programlama yönteminin başarısı, uygulamaların çalıştırıldığı bilgisayarlarda başka işlemlerin yapılmasına bağlı olarak değişkenlik göstermektedir.

KAYNAKLAR

1. Öcal, K., “Otomatik konuşma tanıma algoritmalarının uygulamaları”, Yüksek Lisans Tezi, *Ankara Üniversitesi Fen Bilimleri Enstitüsü*, Ankara, 4-10 (2005).
2. Alleva, P., Huang, X. D., and Hwang, M. Y., “Improvements on the pronunciation prefix tree search organization”, *In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Atlanta, 133–136 (1996).
3. Axelrod, S., Goel, V., Gopinath, R., Olsen, P. and Visweswariah, K., “Discriminative estimation of subspace constrained Gaussian mixture models for speech recognition”, *IEEE Transactions on Speech and Audio Processing*, 15 (1), 135-163 (2007).
4. Chow, Y., “Maximum mutual information estimation of hmm parameters for continuous speech recognition using the n-best algorithm”, *In IEEE BBN Systems and Technologies Corp*, Cambridge, 701–704 (1990).
5. Bahl, L. R., Jelinek, F. and Mercer, R. L., “A maximum likelihood approach to continuous speech recognition”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5: 179–190 (1983).
6. Ney, H., Haeb-Umbach, R., Tran, B. H. And Oerder, M., “Improvements in beam search for 10000- word continuous-speech recognition”, *IEEE Transactions on Speech and Audio Processing*, 2 (2): 353–356 (1994).
7. He, X., Deng, L. and Chou, W., “A novel learning method for hidden markov models in speech and audio processing”, *In IEEE Workshop on Multimedia Signal Processing (MMSP)*, Victoria, 80–85 (2006).
8. Lee, S. M., Fang, S. H., Hung, J. W. and Lee, L. S., “Improved MFCC Feature Extraction by PCA-Optimized Filter Bank for Speech Recognition”, *Automatic Speech Recognition and Understanding*, Asru, 49-52 (2001).
9. Lima, A., Zen, H., Nankaku, Y., Miyajima, C., Tokuda, K. and Kitamura, T., “On the use of kernel pca for feature extraction in speech recognition”, *IEICE Trans. Inf. & Syst.*, E87-D (12): 2802-2811 (2004).
10. Baygün, M. K., “Türkçe komutları tanıyan ses tanıma sisteminin geliştirilmesi”, Yüksek Lisans Tezi, *Pamukkale Üniversitesi Fen Bilimleri Enstitüsü*, Denizli, 3-20 (2006).
11. Aygün, O., “Türkçe konuşmacı tanıma sistemlerinde dalgacık dönüşümü”, Yüksek Lisans Tezi, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*, Konya, 7-11 (2006).

12. Roch M. and Hurtig R. R., "The integral decode: a smoothing technique for robust HMM-based speaker recognition.", *IEEE Trans. On Speech and Audio Processing*, 10: 315-324 (2002).
13. Rodriguez-Porcheron, K. D. and Faundez-Zanuy M., "Speaker recognition with a MLP classifier and LPCC codebook.", *IEEE Acoustics, Speech and Signal Processing (ICASSP)*, Arizona: 1005-1008 (1999).
14. Mengüšoğlu, E., "Bir Türkçe sesli ifade tanıma sisteminin kural tabanlı tasarımı ve gerçekleştirimi", Yüksek Lisans Tezi, *Hacettepe Üniversitesi Fen Bilimleri Enstitüsü*, Ankara, 3-24 (1999).
15. Caner M and Üstün S. V., "Yapay sinir ağları ile konuşmacı kimliğini tanıma uygulaması", *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 12 (2): 279-284 (2005).
16. Dempster, A. P., Laird, N.M. and Rubin, D.B., "Maximum likelihood from incomplete data via the EM algorithm.", *J. Roy. Stat. Soc.*, 39 (1): 1-38 (1977).
17. Juang, B. H., Levinson, S. E. and Sondhi, M. M., "Maximum likelihood estimation for multivariate observations of Markov Chains.", *IEEE Transactions on Information Theory*, 32 (2): 307-309 (1986).
18. Levinson, S. E., Rabiner, L. R. and Sondhi, M. M., "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition-A unified view.", *Bell Syst. Tech. J.*, 62 (4): 1035-1074 (1983).
19. Rabiner, R., "A tutorial on hidden markov models and selected applications in speech recognition.", *Proceedings of the IEEE*, 77 (2): 257-286 (1989).
20. Benzeghiba M., Mori, R. D., Deroo, O., Dupont, S., Erbes, T., Jouviet, D., Fissore, L., Laface, P., Mertins, A., Ris, C., Rose, R., Tyagi, V. And Wellekens C., "Automatic speech recognition and speech variability: A review", *Speech Communication*, Boston, 763-786 (2007).
21. Valente, F., "Hierarchical and parallel processing of auditory and modulation frequencies for automatic speech recognition", *In Proceedings of Speech Communication*, Makuhari, 790-800 (2010).
22. Kim W. and Stern, R. M., "Mask classification for missing-feature reconstruction for robust speech recognition in unknown background noise", *Speech Communication*, Indiana, 1-11 (2011).
23. Kingsbury, B., Morgan, N. and Greenberg, S., "Robust speech recognition using the modulation spectrogram", *In Proceedings of Speech Communication*, Montreal 117-132 (1998).

24. Campbell J. P., "Speaker recognition: a tutorial", *IEEE Senior Member*, Texas, 1437-1462 (1998).
25. Reynolds, D. A., Quatieri, T. F. and Dunn, R. B., "Speaker Verification Using Adapted Gaussian Mixture Models", *Digital Signal Processing*, 10: 19-41 (2000).
26. Çakır, H. and Okutan, B., "Ses kontrollü web tarayıcı", *Bilişim Teknolojileri Dergisi*, 4 (1):, 11-18 (2011).
27. Eray, O., "Destek vektör makineleri ile ses tanıma uygulaması", Yüksek Lisans Tezi, *Pamukkale Üniversitesi Fen Bilimleri Enstitüsü*, Denizli, 74-75 (2008).
28. Güvensan, M. A., "Ev aletlerinin tanınması için çevresel faktörlerin sınıflandırılması", *2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, Zurich, 1-4 (2010).
29. Caner, M. and Üstün, S. V., "Yapay sinir ağları ile konuşmacı tanıma kimliğini tanıma", *Mühendislik Bilimleri Dergisi*, 12 (2):, 279-284 (2006).
30. Küçüker, A. and Yüzkollar, C., Şanslı, A. and Şen, F., "Güvenlik sistemleri için mel frekans kepsrum katsayıları ve yapay sinir ağları kullanılarak konuşmacı tanıma", *Speaker Recognition Using MFCC and Artificial Neural Network For Security Systems*, 2 (2):, 8-15 (2006).
31. Edizkan, R., Tiryaki, B., Büyükcan, T. and Uzun, İ., "Ses komut tanıma ile gezgin araç kontrolü", *IX. Akademik Bilişim Konferansı*, Kütahya, 583-587 (2007).
32. Bülbül, H. İ. and Karacı, A., "Bilgisayar ortamında sesli komutları tanıma: Örüntü tanıma yöntemi", *Kastamonu Eğitim Dergisi*, 15 (1):, 45-62 (2007).
33. Sigmund, M., "Speaker recognition - identifying people by their voices", PhD Thesis, *Brno University of Technology Faculty of Electrical Engineering and Computer Science Institute of Radio Electronics*, Brno, 6-12 (2000).
34. Furui, S., Kikuchi, T., Shinnaka, Y. and Hori, C., "Speech-to-text and speech-to-speech summarization of spontaneous speech", *Speech and Audio Processing, IEEE Trans.*, 12 (4): 401-408 (2004).
35. Coleman, J., "Introducing Speech and Language Producing", *Cambridge University Press*, 301-310 (2005).
36. Huang, X., Acero, A., Hon and H. W., "Spoken Language Processing: A Guide to Theory, Algorithm and System Development 1st Ed.", *Prentice Hall PTR*, New Jersey, 1-980 (2001).
37. Smith, S. W., "The Scientist's and Engineer's Guide to Digital Signal Processing 2nd Ed.", *California Technical Publishing*, San Diego, 1-650 (1999).

38. Kaya T, “Genetik algoritma ile sayısal filtre tasarımı”, Yüksek Lisans Tezi *Fırat Üniversitesi Fen Bilimleri Enstitüsü*, Elazığ, 31-40 (2006).
39. Baygün, M. K., “Türkçe komutları tanıyan ses tanıma sistemi geliştirilmesi”, Yüksek Lisans Tezi *Pamukkale Üniversitesi Fen Bilimleri Enstitüsü*, Denizli, 20-21 (2006).
40. Ting, H., Yingchun, Y. and Zhaohui, W.,”Combinning MFCC and Pitch to Enhance the Performance of the Gender Recognition”, *8th International Conference on Signal Processing*, Beijing, 16-20 (2006).
41. Nabiyev, V. V. and Yücesoy, E., “Konuşmacı cinsiyetinin temel frekansa göre belirlenmesi”, *1.Mühendislik ve Teknoloji Sempozyumu*, Ankara, 33-41 (2008).
42. Yakut, M. and Reisoğlu, N.,”Fast video vector quantization”, *23rd International Symposium on Computer and Information Sciences*, İstanbul, 1-4 (2008).
43. Wu, J. D. and Liu, C. T., “Finger-vein pattern identification using SVM and neural network technique”, *Expert System Appl*, 38:14284–14289 (2011).
44. Turk, M. and Pentland, A., Eigenfaces for recognition”, *Journal of Cognitive Neurosci*, 3: 71-86 (1991).
45. Gorgunoglu, S., Oz, K. and Bayır, S., “Performance analysis of eigenfaces method in face recognition system”, *2nd international Symposium on Computing in Science & Engineering*, Aydın, 136-143 (2011).
46. Yazar, I., Yavuz, H. S. and Çay, M. A., “Temel bileşen analizi yönteminin ve bazı klasik robust uyarlamalarının yüz tanıma uygulamaları”, *Eskişehir Osmangazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 22 (1): 49-63 (2009).
47. Qiu, D., “A comparative study of the K-means algorithm and the normal mixture model for clustering: bivariate homoscedastic case”, *Journal Statistic Planning Inference*, 140: 1701–1711 (2010).
48. Sarma, T. H., Viswanath, P., Reddy, B. E., “Speeding-up the kernel k-means clustering method: A prototype based, hybrid approach”, *Pattern Recognition Letters*, 34: 564-573 (2013).
49. Oral, L., “Uyarlamalı sinir-bulanık sınıflayıcı ile parmak izi tanıma ve dalgacık tabanlı momentlerle özellik çıkarma”, Yüksek Lisans Tezi *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü*, Isparta, 50-55 (2010).
50. Haykin, S., “Neural Networks”, *Macmillan College Publishing Company*, Prentice Hall, 312-364 (1994).

51. Öztemel, E., “Yapay Sinir Ağları”, *Papatya Yayıncılık*, 169-200, İstanbul (2003).
52. Başaran, S., “Yapay sinir ağları kullanarak konuşmacı tanıma”, Yüksek Lisans Tezi *Uludağ Demirel Üniversitesi Fen Bilimleri Enstitüsü*, Bursa, 22-40 (2007).
53. Altıntaş, V. and Yeğenoğlu, V. D., “Görüntü işlemede seri ve paralel programlamanın performansı”, *6th International Advanced Technologies Symposium*, Elazığ, 131-134 (2011).
54. Akıncı, Ö., “MPI Programlamaya Giriş Ve Motivasyon”, http://www.uybhm.itu.edu.tr/documents/basarim09sunum/01_Giris_ve_Motivasyon_akinci_v3.pdf (2013).

ÖZGEÇMİŞ

Emrah ÖZKAYNAK 1979 yılında Konya’ da doğdu. 2005 yılında Zonguldak Karaelmas Üniversitesi Bilgisayar Öğretmenliği bölümünden mezun oldu. Ses tanıma ve ses işleme sistemleri üzerinde çalışmalar yapmaktadır. Hala Karabük Üniversitesi Bilgi İşlem Daire Başkanlığı’nda Öğretim Görevlisi olarak görev yapmaktadır.

ADRES BİLGİLERİ

Adres : Karabük Üniversitesi
Fen Bilimleri Enstitüsü
Balıklarkayası Mevkii / KARABÜK

Tel : (507) 796 37 21
E-posta : emrahozkaynak@gmail.com

Emrah ÖZKAYNAK tarafından hazırlanan “PARALEL PROGRAMLAMA İLE SES TANIMA İŞLEMİNİN GERÇEKLEŞTİRİLMESİ” başlıklı butezin Yüksek Lisans Tezi olarak uygun olduğunu onaylarız.

Yrd. Doç. Dr. Salih GÖRGÜNOĞLU
Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından oy birliği ile Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir. .../.../2013

Ünvanı, Adı SOYADI (Kurumu) İmzası _____

Başkan : Yrd. Doç. Dr. Salih GÖRGÜNOĞLU (KBÜ).....

Üye :

Üye:

Üye:

Üye:

...../...../2013

KBÜ Fen Bilimleri Enstitüsü Yönetim Kurulu, bu tez ile, Yüksek Lisans derecesini onamıştır.

Prof. Dr. Nizamettin KAHRAMAN
Fen Bilimleri Enstitüsü Müdürü