

T.C.
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MEL FREKANSI KESPTAL KATSAYILARININ BULUNMASI, DOĞRUSAL
ÖNGÖRÜLÜ KODLAMA YÖNTEMLERİ VE YAPAY SİNİR AĞLARI
KULLANARAK TÜRKÇE KONUŞMA TANIMA VE YÖNTEMLERİN
KARŞILAŞTIRILMASI

YÜKSEK LİSANS TEZİ

Mehmet ÖZTÜRK

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Tez Danışmanı: Prof. Dr. Ömerülfaruk ÖZGÜVEN

NİSAN 2025

T.C
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**MEL FREKANSI KESPTRAL KATSAYILARININ BULUNMASI, DOĞRUSAL
ÖNGÖRÜLÜ KODLAMA YÖNTEMLERİ VE YAPAY SİNİR AĞLARI
KULLANARAK TÜRKÇE KONUŞMA TANIMA VE YÖNTEMLERİN
KARŞILAŞTIRILMASI**

YÜKSEK LİSANS TEZİ

Mehmet ÖZTÜRK
(36223615903)

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Tez Danışmanı: Prof. Dr. Ömerülfaruk ÖZGÜVEN

NİSAN 2025

İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜ

**Mel Frekansı Kesptral Katsayılarının Bulunması, Doğrusal Öngörölü Kodlama
Yöntemleri Ve Yapay Sinir Ağları Kullanarak Türkçe Konuşma Tanıma Ve
Yöntemlerin Karşılaştırılması**

Tezi Hazırlayan: Mehmet ÖZTÜRK
Yüksek Lisans Tezi

TEŞEKKÜR

Bu tez çalışmasının her aşamasında yardım, öneri, bilgi, tecrübe ve desteklerini esirgmeden beni her konuda yönlendiren danışman hocam Sayın Prof. Dr. Ömerülfaruk ÖZGÜVEN'e teşekkür ederim.



ONUR SÖZÜ

Yüksek lisans tezi olarak sunduğum “Mel Frekansı Kesptral Katsayılarının Bulunması, Doğrusal Öngörülü Kodlama Yöntemleri Ve Yapay Sinir Ağları Kullanarak Türkçe Konuşma Tanıma Ve Yöntemlerin Karşılaştırılması” başlıklı bu çalışmanın bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın tarafımdan yazıldığına ve yararlandığım bütün kaynakların hem metin içinde hem de kaynakçada yöntemine uygun biçimde gösterilenlerden oluştuğunu belirtir, bunu onurumla doğrularım.

Mehmet ÖZTÜRK



İÇİNDEKİLER

TEŞEKKÜR.....	i
ONUR SÖZÜ	ii
ÇİZELGELER DİZİNİ.....	v
ŞEKİLLER DİZİNİ.....	vi
SEMBOLLER VE KISALTMALAR	vii
ÖZET.....	viii
ABSTRACT.	ix
1. GİRİŞ.....	1
1.1 Tez Çalışmasının Amacı ve Kapsamı.....	1
1.2 Literatür Taraması ve Değerlendirilmesi.....	4
2. KONUŞMA TANIMA SİSTEMLERİ.....	7
2.1 Konuşma Tanıma Sistemlerinin Sınıflandırılması	7
2.1.1 Konuşmanın türüne göre konuşma tanıma sistemi türleri.....	8
2.1.2 Konuşmacı modeline dayalı konuşma tanıma türleri	9
2.1.3 Kelime hazinesine dayalı konuşma tanıma türleri.....	10
2.2 Konuşma Tanıma Sistemlerinin Yapısı.....	10
2.2.1 Öznitelik çıkarma	11
2.2.2 Sınıflandırma	11
2.2.3 Akustik model	12
2.2.4 Dil modeli	13
2.3 Konuşma Tanıma Sistemlerinin Uygulamaları	13
2.4 Konuşma Tanıma Sistemlerinde Performans	15
2.4.1 Kelime hata oranı (KHO)	15
2.4.2 Hız	16
2.5 Konuşma Veritabanları.....	16
2.6 Konuşma Araçitleri.....	18
3. ÖZNİTELİK ÇIKARMA YÖNTEMLERİ	20
3.1 Mel Frekansı Kepstral Katsayıları.....	20
3.1.1 Önvurgulama	21
3.1.2 Çerçeveleme	21
3.1.3 Pencereleme.....	22
3.1.4 Hızla fourier dönüşümü.....	22
3.1.5 Mel frekansına çevirme	23
3.1.6 Logaritma	25
3.1.7 Ayrık kosinüs dönüşümü.....	25
3.2 Doğrusal Öngörülü Kodlama.....	25
3.2.1 Önvurgulama	27
3.2.2 Çerçeveleme	27
3.2.3 Pencereleme.....	27
3.2.4 Otokorelasyon analizi	27
3.2.5 DÖK analizi.....	28
4. SINIFLANDIRMA YÖNTEMLERİ	29
4.1 Dinamik Zaman Bükmesi.....	29
4.1.1 DZB algoritmasının çalışması	31
4.2 Yapay Sinir Ağları.....	33
4.2.1 Biyolojik nöron.....	34
4.2.2 Yapay nöron	34
4.2.3 YSA'nın tarihçesi	37

4.2.4 Yapay sinir ağı yapısı	38
4.2.5 Yapay sinir ağlarından en çok kullanılan modeller	38
4.2.6 YSA’da öğrenme stratejileri	39
4.2.7 YSA’da öğrenme kuralları	39
5. DİNAMİK ZAMAN BÜKMESİ UYGULAMASI.....	41
5.1 MFKK –DZB Teknikleriyle Konuşma Tanıma.....	42
5.1.1 MFKK –DZB teknikleriyle konuşma tanımanın performansı.....	44
5.2 DÖK –DZB Teknikleriyle Konuşma Tanıma	45
5.2.1 DÖK –DZB teknikleriyle konuşma tanımanın performansı.....	47
6. YAPAY SİNİR AĞI İLE TÜRKÇE RAKAM TANIMA UYGULAMASI	49
6.1 MFKK –Yapay Sinir Ağı ile Türkçe Rakam Tanıma Uygulaması	50
6.1.1 MFKK –Yapay sinir ağı ile Türkçe rakam tanıma uygulamasının performansı	52
6.2 DÖK –Yapay Sinir Ağı ile Türkçe Rakam Tanıma Uygulaması	53
6.2.1 DÖK –Yapay sinir ağı ile Türkçe rakam tanıma uygulamasının performansı ..	55
7. MİKRODENETLEYİCİ UYGULAMASI.....	57
8. SONUÇ... ..	60
KAYNAKLAR.....	62
EKLER.....	65
ÖZGEÇMİŞ.....	89

ÇİZELGELER DİZİNİ

Çizelge 2.1: Konuşma tanıma teknolojisinin kullanıldığı alanlar ve uygulamalar.....	14
Çizelge 2.2: Konuşma veritabanları karakteristikleri.....	17
Çizelge 2.3: Konuşma tanıma için mevcut araç setleri.....	18
Çizelge 3.1: Mel ölçeğin merkez frekansları ve bant genişlikleri	24
Çizelge 4.1: Biyolojik nöronla yapay nöronun karşılaştırılması.....	35
Çizelge 5.1: Matlab’da geliştirilen konuşma tanıma sisteminin tanıdığı kelimeler.....	41
Çizelge 5.2: MFKK-DZB OKT sisteminin kelimeleri doğru ve yanlış tanıma miktarları..	44
Çizelge 5.3: DÖK-DZB OKT sisteminin kelimeleri doğru ve yanlış tanıma miktarları ...	47
Çizelge 6.1: Konuşma tanıma sisteminin tanıdığı kelimeler.....	49
Çizelge 6.2: MFKK-YSA OKT sisteminin kelimeleri doğru ve yanlış tanıma miktarları..	52
Çizelge 6.3: DÖK-YSA OKT sisteminin kelimeleri doğru ve yanlış tanıma miktarları....	55



ŞEKİLLER DİZİNİ

Şekil 2.1: Ses işleme ve konuşma tanımanın sınıflandırılması.....	8
Şekil 2.2 : Otomatik konuşma tanıma sistemlerinin yapısı.....	10
Şekil 3.1 : MFKK katsayılarını elde etme basamakları.....	21
Şekil 3.2: Çerçeveleme İşlemi.....	22
Şekil 3.3: Bir konuşma parçası işareti	23
Şekil 3.4 : Şekil 3.3'deki işaretin hızlı fourier dönüşümü alınmış hali.....	23
Şekil 3.5: Mel ölçeğe dizilmiş üçgen süzgeç dizileri.....	24
Şekil 3.6 : Konuşma işaretinin süzgeç dizisinden geçirildikten sonraki durumu	25
Şekil 3.7: DÖK öznitelik çıkarımı işlem adımları.....	26
Şekil 4.1: İki işaretin öklid mesafesi eşleştirmesi	30
Şekil 4.2: İki işaretin DZB eşleştirmesi	30
Şekil 4.3: İki zaman serisinin grafiği.....	31
Şekil 4.4: Boş bir maliyet matrisi.....	32
Şekil 4.5: Doldurulmuş maliyet matrisi.....	32
Şekil 4.6: Son bükülmüş yol.....	33
Şekil 4.7: Yapay Nöron.....	35
Şekil 4.8: Sigmoid fonksiyonu.....	36
Şekil 4.9: Bir yapay sinir ağının yapısı.....	38
Şekil 5.1 : Geliştirilen konuşma tanıma sisteminin blok diyagramı.....	42
Şekil 5.2 : MFKK ve DZB teknikleriyle konuşma tanıma programı kullanıcı arayüzü.....	43
Şekil 5.3 : DÖK ve DZB teknikleriyle konuşma tanıma programı kullanıcı arayüzü.....	46
Şekil 6.1: Konuşma tanıma sisteminin yapısı (Yapay Sinir Ağının eğitimi aşaması).....	49
Şekil 6.2: Konuşma tanıma sisteminin yapısı (Yapay Sinir Ağının test aşaması).....	49
Şekil 6.3: MFKK –Yapay Sinir Ağı Uygulaması.....	50
Şekil 6.4: MFKK –Yapay Sinir Ağı Uygulamasının Eğitim Aşaması.....	51
Şekil 6.5: MFKK –Yapay Sinir Ağı Uygulamasının MATLAB Kullanıcı Arayüzü.....	52
Şekil 6.6: DÖK –Yapay Sinir Ağı Uygulaması.....	53
Şekil 6.7: DÖK –Yapay Sinir Ağı Uygulamasının Eğitim Aşaması.....	54
Şekil 6.8: DÖK –Yapay Sinir Ağı Uygulamasının MATLAB Kullanıcı Arayüzü.....	55
Şekil 7.1: Mikrodenetleyici Uygulamasının Donanım Mimarisi.....	57
Şekil 7.2: Mikrodenetleyici Uygulamasının Yazılım Mimarisi.....	58

SEMBOLLER VE KISALTMALAR

OKT	: Otomatik Konuşma Tanıma – Automatic Speech Recognition (ASR)
MFKK	: Mel Frekans Kepstral Katsayıları – Mel Frequency Cepstral Coefficients(MFCC)
DÖK	: Doğrusal Öngörülü Kodlama – Linear Predictive Coding (LPC)
DZB	: Dinamik Zaman Bükmesi – Dynamic Time Warping (DTW)
YSA	: Yapay Sinir Ağı – Artificial Neural Network (ANN)
SMM	: Saklı Markov Modeli – Hidden Markov Model (HMM)
DVM	: Destek Vektör Makinesi – Support Vector Machine (SVM)
KHO	: Kelime Hata Oranı – Word Error Rate (WER)
GZF	: Gerçek Zaman Faktörü – Real Time Factor (RTF)
KTO	: Kelime Tanıma Oranı – Word Recognition Rate (WRR)
DSA	: Derin Sinir Ağları – Deep Neural Network (DNN)
ADÖ	: Algısal Doğrusal Öngörü - Perceptual Linear Prediction (PLP)
SGO	: Sıfır Geçiş Oranı – Zero Crossing Rate (ZCR)
KVE	: Kısa Vadeli Enerji – Short Term Energy (STE)
SA	: Sinir Ağı - Neural Network (NN)

ÖZET

Yüksek Lisans Tezi

MEL FREKANSI KESPTRAL KATSAYILARININ BULUNMASI, DOĞRUSAL ÖNGÖRÜLÜ KODLAMA YÖNTEMLERİ VE YAPAY SİNİR AĞLARI KULLANARAK TÜRKÇE KONUŞMA TANIMA VE YÖNTEMLERİN KARŞILAŞTIRILMASI

Mehmet ÖZTÜRK

İnönü Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik Elektronik Mühendisliği Anabilim Dalı

89+IX sayfa

2025

Danışman: Prof. Dr.Ömerülfaruk ÖZGÜVEN

Bu tezde MATLAB programı kullanılarak konuşmacı bağımlı, yalıtılmış kelimeler otomatik konuşma tanıma sistemleri geliştirilmiştir. Konuşma tanıma sistemleri öznitelik çıkarma ve sınıflandırma ana bölümlerinden oluşur. Bu çalışmada öznitelik çıkarma yöntemlerinden Mel Frekans Kepstral Katsayıları ile Doğrusal Öngörülü Kodlama teknikleri ve sınıflandırma tekniklerinden ise Dinamik Zaman Bükmesi ile Yapay Sinir Ağları kullanılmıştır. Dinamik Zaman Bükmesi ve Yapay Sinir Ağı kullanan sınıflandırma ve konuşma tanıma yöntemlerinde öznitelik katsayılarının bulunması için Mel Frekans Kepstral Katsayıları ve Doğrusal Öngörülü Kodlama metot ve algoritmaları kullanılmıştır. Dolayısıyla 4 farklı konuşma tanıma sistemi geliştirilmiştir. Bu 4 yöntemin çıkış cevapları karşılaştırılmıştır. Öznitelik çıkarma yöntemlerinden Mel Frekans Kepstral Katsayıları yönteminin Doğrusal Öngörülü Kodlama yöntemine göre daha iyi sonuçlar verdiği gözlemlenmiştir. Ayrıca Dinamik Zaman Bükmesi yönteminin de Yapay Sinir Ağlarına oranla hem Kelime Tanıma Oranı açısından hem de kelime hazinesi açısından daha iyi sonuçlar ürettiği gözlemlenmiştir.

Anahtar Kelimeler: Otomatik Konuşma Tanıma, Mel Frekans Kepstral Katsayıları, Doğrusal Öngörülü Kodlama, Dinamik Zaman Bükmesi, Yapay Sinir Ağları, Öznitelik Çıkarma, Sınıflandırma

ABSTRACT

Master Thesis

FINDING MEL FREQUENCY CEPSTRAL COEFFICIENTS, LINEAR PREDICTIVE CODING METHODS AND TURKISH SPEECH RECOGNITION USING ARTIFICIAL NEURAL NETWORKS AND COMPARISON OF THE METHODS

Mehmet ÖZTÜRK

Inonu University
Graduate School of Nature and Applied Sciences
Department of Electrical-Electronics Engineering

89+IX pages

2025

Supervisor: Professor Ömerülfaruk ÖZGÜVEN

In this thesis, speaker dependent, isolated words automatic speech recognition systems are developed using MATLAB program. Speech recognition systems consist of feature extraction and classification main parts. In this study, Mel Frequency Cepstral Coefficients and Linear Predictive Coding techniques are used as feature extraction methods and Dynamic Time Warping and Artificial Neural Networks are used as classification techniques. Mel Frequency Cepstral Coefficients and Linear Predictive Coding methods and algorithms are used to find feature coefficients in classification and speech recognition methods using Dynamic Time Warping and Artificial Neural Network. Therefore, 4 different speech recognition systems are developed. The output responses of these 4 methods are compared. It is observed that Mel Frequency Cepstral Coefficients method gives better results than Linear Predictive Coding method among feature extraction methods. It is also observed that Dynamic Time Warping method produces better results both in terms of Word Recognition Rate and vocabulary compared to Artificial Neural Networks.

Keywords: Automatic Speech Recognition, Mel Frequency Cepstral Coefficients, Linear Predictive Coding, Dynamic Time Warping, Artificial Neural Network, Feature Extraction, Classification

1. GİRİŞ

İnsanların birbiri ile iletişimde zaman içinde telgraf, telefon, telsiz, bilgisayar ve cep telefonu gibi birçok teknolojiler kullanılmıştır. Bu teknolojiler insan-makine etkileşiminin ortaya çıkmasına neden olmuştur. İlk zamanlarda basit tuşlu sistemler ile gerçekleştirilen insan-makine etkileşimi günümüzde yerini dokunmatik sistemlere bırakmıştır. Bu durumda insanların makine veya elektronik cihazlarla iletişimi sınırlı sağlanmaktadır. İnsanların makinelerle ana dillerini konuşarak iletişim kurması, makinelerin komut ve kontrol mekanizmasının daha hızlı bir şekilde çalıştırabilmesine olanak vermektedir. Bu nedenle Otomatik Konuşma Tanıma(OKT) sistemleri geliştirilmekte ve kullanım alanı her geçen gün artmaktadır. Otomatik Konuşma Tanıma ses işaretlerinin işlenerek metne dönüştürülmesi işlemidir (Oyucu ve diğ.,2019).

OKT sistemleri iletişim, eğitim, elektronik eşya, savunma sanayi, adalet ve istihbarat gibi alanlarda sıklıkla kullanılmaktadır. OKT sistemleri ile metne aktarılan konuşmalar üzerinde yapay zekâ teknolojileri kullanılarak konuşma anlama ve duygu analizi gibi işlemler de gerçekleştirilmektedir. Konuşma anlama ile komuta kontrol işlemleri yapılabildiği gibi aynı zamanda kişisel asistanlar da geliştirilebilmektedir. Kişisel asistanlara Apple'ın Siri'sini ve Google'ın Gemini'sini örnek verebiliriz. Duygu analizi ise bir konuşmanın içeriğine göre konuşmacının içerisinde bulunduğu duygu durumunun belirlenmesidir. Konuşma içerisindeki duyguların analizi ile özellikle çağrı merkezlerinde hizmet kalitesinin artırılması amaçlanmaktadır. Ayrıca konuşma metinleri analiz edilerek müşteri ilişkileri yönetilmekte ve müşterilere özel fırsatlar sunulmaktadır (Oyucu ve diğ.,2019).

1.1 Tez Çalışmasının Amacı ve Kapsamı

Bu tezde MATLAB programı kullanılarak konuşmacı bağımlı, yalıtılmış kelimeler otomatik konuşma tanıma sistemleri geliştirilmiştir. Bir OKT sistemi temel olarak iki bölümden oluşur. Öznitelik çıkarma ve sınıflandırma. Bu tezde öznitelik çıkarma yöntemlerinden ikisi olan Mel Frekanslı Kepstral Katsayıları ile Doğrusal Öngörülü kodlama yöntemleri kullanılmıştır. Sınıflandırma yöntemleri olarak da Dinamik Zaman Bükmesi ve Yapay Sinir

Ağları kullanılmıştır. Dolayısıyla dört farklı Otomatik Konuşma Tanıma Sistemi geliştirilmiştir. Bu OKT sistemleri birbirleriyle kıyaslanmıştır.

Mel Frekansı Kepstral Katsayıları (MFKK) belki de en iyi bilinen ve en popüler olan öznelik çıkarma yöntemidir. MFKK'ler, insan kulağının kritik bant genişliklerinin frekansla bilinen değişimine dayanmaktadır. MFKK tekniği, doğrusal aralıklı filtreler ve logaritmik aralıklı filtreler olmak üzere iki tür filtre kullanır. Konuşmanın fonetik açıdan önemli özelliklerini yakalamak için sinyal, Mel frekans ölçeğinde ifade edilir. Bu ölçek, 1000 Hz'nin altında doğrusal bir frekans aralığına ve 1000 Hz'nin üzerinde logaritmik bir aralığa sahiptir. Normal konuşma dalga formu, konuşmacıların ses tellerinin fiziksel durumuna bağlı olarak zaman zaman değişebilir. Konuşma dalga formlarının kendisinden ziyade, MFKK'ler söz konusu değişimlere daha az duyarlıdır. Mel Frekansı Kepstral Katsayıları sesi temsil eden Mel ölçeğine dayalı katsayılardır. Kepstral kelimesi, spektrumun logaritmik bir ölçeği olan Kepstrum kelimesinden gelir (ve spektrum kelimesindeki ilk dört harfi tersine çevirir). İlk olarak, konuşma verileri 20-30 ms'lik çerçevelere bölünür. Her 10-20 ms'de yeni bir çerçeve başlatılır ve bu, örnekleme periyodu haline getirilir ve pencerelerin birbiriyle örtüşmesine neden olur. Çerçevelere Hamming penceresi fonksiyonu uygulanır. Sonra, her konuşma verisi karesinde Hızlı Fourier Dönüşümü gerçekleştirilir ve büyüklük bulunur. Bir sonraki adım, sinyali Mel ölçekli filtre bankaları adı verilen bir frekans çarpıtılmış log filtre bankası kümesiyle filtrelemeyi içerir. Filtre bankasının üçgen bir bant geçiş frekansı tepkisi vardır ve aralık ve bant genişliği sabit bir Mel frekans aralığıyla belirlenir. Log filtre bankaları, bir tonun algılanan perdesinin veya frekansının bir ölçüsü olan logaritmik bir ölçek olan Mel ölçeğine göre frekans eksenini boyunca düzenlenir ve böylece insan işitme ölçeği simüle edilir. Son olarak, cepstral katsayıların hesaplanması için Ters Ayırık Fourier Transformu kullanılır (Vimala ve Radha, 2012).

En güçlü sinyal analiz tekniklerinden biri Doğrusal Öngörülü Kodlama (DÖK) yöntemidir. Konuşmanın DÖK'si, konuşmanın temel parametrelerini tahmin etmek için baskın teknik haline gelmiştir. Hem konuşma parametrelerinin doğru bir tahminini sağlar hem de konuşmanın etkili bir hesaplama modelidir. DÖK'nin arkasındaki temel fikir, bir konuşma örneğinin geçmiş konuşma örneklerinin doğrusal bir kombinasyonu olarak yaklaşık olarak hesaplanabilmesidir. Gerçek konuşma örnekleri ile tahmin edilen değerler arasındaki kare farkların toplamını (sonlu bir aralıkta) en aza indirerek, benzersiz bir parametre veya tahmin katsayıları kümesi belirlenebilir. Bu katsayılar, konuşmanın DÖK'sinin temelini oluşturur. Analiz, konuşmanın doğrusal tahmin modelini zaman içinde hesaplama yeteneği sağlar. Bu

nedenle tahmin katsayıları, cepstral katsayılar olarak bilinen daha sağlam bir parametre kümesine dönüştürülür (Vimala ve Radha, 2012).

Dinamik Zaman Bükmesi, zaman veya hız açısından değişebilen iki dizi arasındaki benzerliği ölçmek için kullanılan bir algoritmadır. Örneğin, yürüme desenlerindeki benzerlikler, bir videoda kişi yavaş yürüyor olsa ve bir diğerinde daha hızlı yürüyor olsa veya bir gözlem sırasında hızlanmalar ve yavaşlamalar olsa bile tespit edilebilir. Dinamik Zaman Bükmesi video, ses ve grafiklere uygulanmıştır, aslında doğrusal bir gösterime dönüştürülebilen herhangi bir veri Dinamik Zaman Bükmesi ile analiz edilebilir. İyi bilinen bir uygulama, farklı konuşma hızlarıyla başa çıkmak için otomatik konuşma tanıma olmuştur. Genel olarak, Dinamik Zaman Bükmesi bir bilgisayarın belirli kısıtlamalarla verilen iki dizi (örneğin zaman serisi) arasında optimum bir eşleşme bulmasını sağlayan bir yöntemdir. Diziler, zaman boyutunda belirli doğrusal olmayan değişikliklerden bağımsız olarak benzerliklerinin bir ölçüsünü belirlemek için zaman boyutunda doğrusal olmayan bir şekilde bükülür. Süreklilik, Dinamik Zaman Bükmesinde diğer desen eşleştirme algoritmalarına göre daha az önemlidir; Dinamik Zaman Bükmesi, eşleşmenin gerçekleşmesi için yeterince uzun segmentler olması koşuluyla, eksik bilgi içeren dizileri eşleştirmek için özellikle uygun bir algoritmadır. Optimizasyon süreci, dinamik programlama kullanılarak gerçekleştirilir, dolayısıyla adı da buradan gelir (Anusuya ve Katti, 2009).

Yapay Sinir Ağı, beynin bilgiyi işlemesi gibi biyolojik sinir sistemlerinden esinlenen bir bilgi işleme paradigmasıdır. Belirli bir sorunu çözmek için birlikte çalışan çok sayıda birbirine bağlı işlem elemanından (nöron) oluşur. Temel olarak, sinir ağları bazen nöronlar veya hücreler olarak adlandırılan basit birimlerden oluşturulur. Yapay Sinir Ağı, insan beynine veya biyolojik sinir ağına benzer hesaplamalar yapma yeteneğine sahip bilgi işleme cihazlarıdır. Çoğu Yapay Sinir Ağında esas olarak üç farklı katman türü sunulur. İlk katmana giriş katmanı denir. Ana görevi dış dünyadan girdi almaktır. Çıkış katmanı dış dünyaya çıkışları aktarır. Üçüncü katman ise ara katmandır. Yapay Sinir Ağlarında bilgi, hücrelerin ağırlıklarında tutulur.

Tezin ikinci bölümde OKT sistemine genel bir bakış yapılmıştır. OKT sistemini oluşturan bölümler incelenmiştir. Üçüncü bölümde öznitelik çıkarma tekniklerinden Mel Frekanslı Cepstral Katsayıları ve Doğrusal Öngörülü Kodlama Katsayıları çıkarma teknikleri incelenmiştir. Dördüncü bölümde sınıflandırma yöntemlerinden Dinamik Zaman Bükmesi ve Yapay Sinir Ağları yöntemleri incelenmiştir. Beşinci bölümde MATLAB'la geliştirilen

Dinamik Zaman Bükmesi uygulamaları ve performansları anlatılmıştır. Altıncı bölümde MATLAB’la geliştirilen Yapay Sinir Ağı kullanarak Türkçe Rakam Tanıma uygulamaları ve performansları anlatılmıştır. Yedinci bölümde OKT sisteminin mikrodenetleyici uygulaması gerçekleştirilmeye çalışılmıştır. Sekizinci bölüm olan sonuç bölümünde ise OKT sistemlerinin performansları değerlendirilmiştir.

1.2 Literatür Taraması ve Değerlendirilmesi

Sözcük ve Hece Tabanlı Konuşma Tanıma Sistemlerinin Karşılaştırılması, adlı tez çalışmasında Yakar Ö., öznitelik çıkarma yöntemi olarak mel frekansı kepsstral katsayılarını kullanmıştır. Sınıflandırma yöntemi olarak da dinamik zaman bükmesi, çok katmanlı algılayıcı, saklı markov modeli ve destek vektör makinesini kullanmıştır. Sözcük tabanlı sistemler için kelime tanıma oranları dinamik zaman bükmesinde %96, çok katmanlı algılayıcıda %82,6 saklı markov modelinde %89,4 ve destek vektör makinesinde %90,7 olarak bulunmuştur (Yakar, 2016).

MFKK, DÖK ve Sinir Ağları Kullanarak Ayırık Kelime Tanıma, adlı çalışmada Gamit M.R. ve Dhameliya K. geliştirdikleri otomatik konuşma tanıma sisteminde öznitelik çıkarma yöntemleri olarak MFKK ve DÖK kullanmışlardır. Sınıflandırma için ise sinir ağlarını kullanmışlardır. MFKK için kelime tanıma oranı %82,33 olarak bulunmuştur. Ayrıca MFKK ve DÖK birleştirilerek oluşturulan öznitelik modülünü kullanan sistemin kelime tanıma oranı ise %86,66 olarak ölçülmüştür (Gamit ve Dhameliya, 2015).

Konuşma Tanıma Tekniği : Bir İnceleme, adlı çalışmada Das S. öznitelik çıkarma için MFKK, DÖK ve ADÖ kullanmıştır. Sınıflandırma yöntemleri olarak da DZB ve SMM kullanmıştır. MFKK ve DZB tekniklerinde kelime tanıma oranı %90,4 olarak ölçülmüştür. MFKK ve SMM tekniklerinde kelime tanıma oranı %86,2 olarak ölçülmüştür. DÖK ve DZB tekniklerinde kelime tanıma oranı %76,4 olarak ölçülmüştür. DÖK ve SMM tekniklerinde kelime tanıma oranı %80,5 olarak ölçülmüştür. ADÖ ve DZB tekniklerinde kelime tanıma oranı %85,6 olarak ölçülmüştür. ADÖ ve SMM tekniklerinde kelime tanıma oranı %77,6 olarak ölçülmüştür (Das, 2012).

DÖK, MFKK, SGO ve KVE Tabanlı Öznitelikler Kullanılarak Yapay Sinir Ağı Sınıflandırıcıları ile Ayırık Kelimelerin Tanınması, adlı çalışmada Das, B. P. ve Parekh, R. öznitelik çıkarma için DÖK, MFKK, SGO ve KVE özniteliklerini kullanmıştır. Sınıflandırma yöntemi olarak da yapay sinir ağlarını kullanmıştır. Geliştirilen sistem konuşmacı bağımsız bir sistemdir. DÖK kullanan sistemde doğruluk % 37,5 olmuştur.

MFKK kullanan sistemde kelime tanıma oranı %51,25 olmuştur. DÖK, MFKK, SGO ve KVE özniteliklerinin birleştirilmesi ile oluşturulan sistemde ise kelime tanıma oranı %85 olmuştur (Das ve Parekh, 2012).

SA ve MFKK'nin Konuşma Tanıma için Birlikte Kullanımının Analizi, adlı çalışmada Tanweer S., Mobin A. ve Alam A., öznitelik çıkarma tekniği olarak MFKK kullanmışlardır. Sınıflandırma yöntemi olarak da yapay sinir ağlarını kullanmışlardır. Geliştirdikleri sistem konuşmacı bağımsızdır ve elde ettikleri konuşma tanıma oranı ise %59,2'dir (Tanweer ve diğ., 2014).

Yapay Sinir Ağları ile Konuşma Tanıma, adlı tez çalışmasında Dede G., öznitelik çıkarma yöntemi olarak MFKK kullanmıştır. Sınıflandırma yöntemi olarak da üç çeşit yapay sinir ağı kullanmıştır. Kullanılan yapay sinir ağları çok katmanlı algılayıcı, elman ağı ve olasılıksal ağıdır. Çok katmanlı algılayıcının kelime tanıma oranı 98,75'dir. Elman ağının kelime tanıma oranı %99,375'dir. Olasılıksal ağın kelime tanıma oranı ise %100'dür (Dede, 2008).

Yapay Zeka Tabanlı Konuşma Tanıma Sistemi, adlı tez çalışmasında Şaşmaz M. B., öznitelik çıkarma yöntemi olarak MFKK ve dalgacık dönüşümü kullanmıştır. Sınıflandırma yöntemi olarak da yapay sinir ağları ile uyarlamalı sinirsel bulanık ağ yöntemlerini kullanmıştır. MFKK ve yapay sinir ağları kullanan sistemde kelime tanıma oranı %98,95 olmuştur. MFKK ve uyarlamalı sinirsel bulanık ağ kullanan sistemde kelime tanıma oranı %97,4 olmuştur. Dalgacık dönüşümü ve uyarlamalı sinirsel bulanık ağ yöntemlerini kullanan sistemde kelime tanıma oranı %29,7 olmuştur (Şaşmaz, 2011).

Farsça Konuşma Tanıma İçin Yeni Bir Yaklaşım, adlı çalışmasında Pour, M. M. ve Farokhi, F., öznitelik çıkarma yöntemi olarak MFKK ve ayrık dalgacık dönüşümü kullanmıştır. Sınıflandırma yöntemi olarak da yapay sinir ağları (çok katmanlı algılayıcı) kullanmıştır. Geliştirdiği sistemde kelime tanıma oranı %98 olmuştur (Pour ve Farokhi, 2009).

Çok Katmanlı Algılayıcı Kullanarak Konuşma Tanıma, adlı çalışmasında Ahad A., Fayyaz A., ve Mehmood T., öznitelik çıkarma yöntemi olarak MFKK kullanmışlardır. Sınıflandırma yöntemi olarak ise yapay sinir ağı (çok katmanlı algılayıcı) kullanmışlardır. Geliştirdikleri sistemde kelime tanıma oranı %94 olmuştur (Ahad ve diğ., 2002).

Konuşmacıya Bağımlı Ayrık Sözcük Tanıma, adlı tez çalışmasında Özişbakan T., öznitelik çıkarma yöntemi olarak MFKK ve DÖK kullanmıştır. Sınıflandırma yöntemi olarak da DZB kullanmıştır. Sistemi Türkçe ve İngilizce rakamlar için geliştirmiştir ve İngilizce rakamlarda daha iyi performans gözlemlemiştir. MFKK ve DZB teknikleri kullanarak geliştirdiği

sistemde kelime tanıma oranı %85-90 olmuştur. DÖK ve DZB teknikleri kullanarak geliştirdiği sistemde ise kelime tanıma oranı %75-85 olmuştur (Özişbakan, 2011).

MATLAB Kullanarak Tekrarlayan Sinir Ağı Tabanlı Konuşma Tanıma, adlı çalışmada James P. E., Kit M. H., Vaithilingam C. A. ve Chiat A. T. W., öznitelik çıkarma yöntemi olarak MFKK kullanmıştır. Sınıflandırma yöntemi olarak da tekrarlayan sinir ağıları kullanmıştır. Geliştirilen OKT sisteminin kelime tanıma oranı %89 olarak bulunmuştur (James ve diğ., 2020).

Tekrarlayan Sinir Ağlarını Kullanarak Konuşma Tanıma, adlı tez çalışmasında Elberri M. A., öznitelik çıkarma yöntemi olarak MFKK kullanmıştır. Sınıflandırma yöntemi olarak da evrişimli sinir ağıları kullanmıştır (Elberri, 2020).

Platformdan Bağımsız Bir Otomatik Konuşma Tanıma Sisteminin Tasarlanması ve Uygulanması, adlı tez çalışmasında Urgan D., öznitelik çıkarma yöntemi olarak MFKK kullanmıştır. Sınıflandırma yöntemi olarak da yapay sinir ağılarını kullanmıştır. Geliştirilen uygulamada kelime tanıma oranı %85 olarak elde edilmiştir (Urgan, 2012).

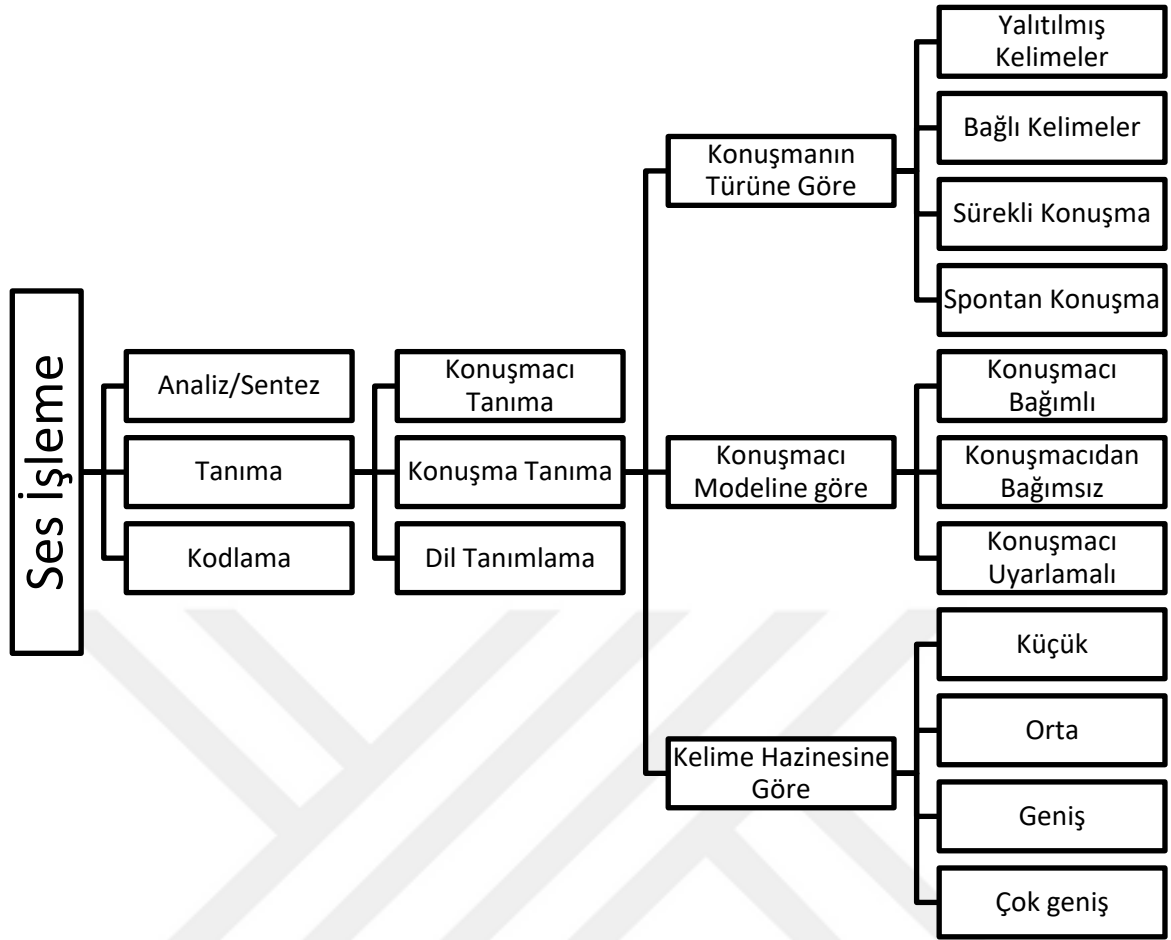
2. KONUŞMA TANIMA SİSTEMLERİ

Otomatik konuşma tanıma, insan konuşmasını (ses işareti) genellikle bir komut olarak kullanılan yazılı metne dönüştürme işlemidir. Otomatik konuşma tanıma sistemleri bu dönüşüm için makine öğrenimi veya yapay zeka teknolojileri kullanır. OKT transkripsiyon teknolojilerinin gelişmiş sürümleri artık doğal dil işlemeyi içeriyor. Bunlar, insanlar arasındaki gerçek konuşmaları yakalar ve bunları işlemek için makine zekasını kullanır. Yine de, OKT transkripsiyonu söz konusu olduğunda sonuçlar değişecektir. OKT tarafından sağlanan doğruluğu etkileyen birçok faktör vardır; bunlar arasında konuşmacı sesi, arka plan gürültüsü, ilgili kayıt ekipmanının kalitesi ve daha fazlası bulunur.

Bu bölümde, OKT sistemlerine genel bir bakış sunulmuştur. Önce OKT sistemleri sınıflandırılmıştır. Sonra geleneksel OKT sistemlerinin yapısı incelenmiştir. Öznitelik çıkarma, sınıflandırma, akustik model, dil modeli gibi OKT sistemlerinin bölümleri anlatılmıştır. Daha sonra OKT sistemlerinin uygulamalarından bahsedilmiştir. Ardından OKT sistemlerinin performansını belirleyen kelime hata oranı ve hız anlatılmıştır. Daha sonraki bölümde akustik modeller oluşturmak için kullanılabilecek konuşma veri tabanları anlatılmıştır. Son olarak da OKT sistemleri oluşturmaya yarayan konuşma araçları anlatılmıştır.

2.1 Konuşma Tanıma Sistemlerinin Sınıflandırılması

Ses işleme, ses analizi/sentezi, ses tanıma, ses kodlama olarak üç alana ayrılır. Ses tanıma ise konuşmacı tanıma, dil tanımlama ve konuşma tanıma olarak üç kısma ayrılır. Konuşma tanıma ise konuşmanın türüne göre, konuşmacı modeline göre ve kelime hazinesine göre alt alanlara ayrılır. Şekil 2.1’de ses işleme ve konuşma tanımayı tasnif eden bir şekil verilmiştir. Bu şekildeki sınıflandırma diğer alt bölümlerde açıklanmıştır.



Şekil 2.1: Ses işleme ve konuşma tanımanın sınıflandırılması (Anusuya ve Katti, 2009)

2.1.1 Konuşmanın türüne göre konuşma tanıma sistemi türleri

Konuşmanın türüne göre konuşma tanıma sistemlerini yalıtılmış kelimeler, bağlı kelimeler, sürekli konuşma ve spontan konuşma olarak dörde ayırıyoruz.

Yalıtılmış kelimeler, tekli ifadeleri, yani tek kelimeyi tanıyan yalıtılmış kelime tanıma sistemi. Yalıtılmış kelime tanıma, kullanıcının yalnızca tek kelimelik yanıt veya komut vermesinin gerektiği durumlar için uygundur, ancak birden fazla kelime girişi için bu çok doğal değildir. Uygulaması basit ve en kolay olanıdır çünkü sözcük sınırları açıktır ve sözcükler açıkça telaffuz edilmeye eğilimlidir; bu da bu türün en büyük avantajıdır. Bu tipin dezavantajı farklı sınırların seçilmesinin sonuçları etkilemesidir (Saksamudre ve diğ., 2015).

Bağlı kelimeler sistemi, yalıtılmış kelimelere benzer, ancak ayrı ifadelerin aralarında minimum duraklamayla "birlikte yürütülmesine" olanak tanır. Söylem, bilgisayar için tek bir anlamı temsil eden bir kelimenin veya kelimelerin seslendirilmesidir (Saksamudre ve diğ., 2015).

Sürekli konuşma tanıma sistemi, içeriği bilgisayar belirlerken kullanıcıların neredeyse doğal bir şekilde konuşmasına olanak tanır. Temel olarak, bilgisayar diktesidir. Bunda en yakın kelimeler, duraklama olmadan veya kelimeler arasında herhangi bir ayırım yapılmadan birlikte çalışır. Sürekli konuşma tanıma sisteminin geliştirilmesi zordur (Saksamudre ve diğ., 2015).

Spontan konuşma tanıma sistemi doğal konuşmayı tanır. Aniden ağızdan çıkan spontan konuşma doğaldır. Spontan konuşmaya sahip bir OKT sistemi, sözcüklerin birlikte çalıştırılması gibi çeşitli doğal konuşma özelliklerini işleyebilir. Kendiliğinden konuşma yanlış telaffuzu, yanlış başlangıçları ve sözcüksüzlüğü içerebilir (Saksamudre ve diğ., 2015).

2.1.2 Konuşmacı modeline dayalı konuşma tanıma türleri

Her konuşmacının benzersiz fiziksel bedeni ve kişiliği nedeniyle özel bir sesi vardır. Konuşma tanıma sistemi üç ana kategoriye ayrılır: Konuşmacı bağımlı modeller, konuşmacıdan bağımsız modeller ve konuşmacı uyarlamalı modeller.

Konuşmacı bağımlı sistemler belirli bir konuşmacı tipi için geliştirilmiştir. Bunlar genellikle belirli bir konuşmacı için daha doğrudur ancak diğer türdeki konuşmacı için daha az doğru olabilir. Bu sistemler genellikle daha ucuz, geliştirilmesi daha kolay ve daha hassastır. Ancak bu sistemler konuşmacıdan bağımsız sistemler kadar esnek değildir (Saksamudre ve diğ., 2015).

Konuşmacıdan bağımsız sistem, önceden herhangi bir eğitim gerektirmeden çeşitli konuşmacıları tanıyabilir. Herhangi bir konuşmacı tipi için çalışacak şekilde konuşmacıdan bağımsız bir sistem geliştirilmiştir. Çok sayıda farklı kullanıcıdan gelen girişi kabul etmesi gereken İnteraktif Sesli Yanıt Sisteminde (IVRS) kullanılır. Ancak dezavantajı, kelime dağarcığındaki kelime sayısını sınırlamasıdır. Konuşmacıdan bağımsız sistem uygulanması en zor olanıdır. Ayrıca pahalıdır ve doğruluğu konuşmacıya bağlı sistemlere göre daha düşüktür (Saksamudre ve diğ., 2015).

Konuşmacı uyarlamalı konuşma tanıma sistemi, konuşmacıya bağlı verileri kullanarak konuşmayı tanımak için en uygun konuşmacıya uyum sağlar ve uyarlama yoluyla hata oranını azaltır. Çalışmayı konuşmacının özelliklerine göre uyarlarlar (Saksamudre ve diğ., 2015).

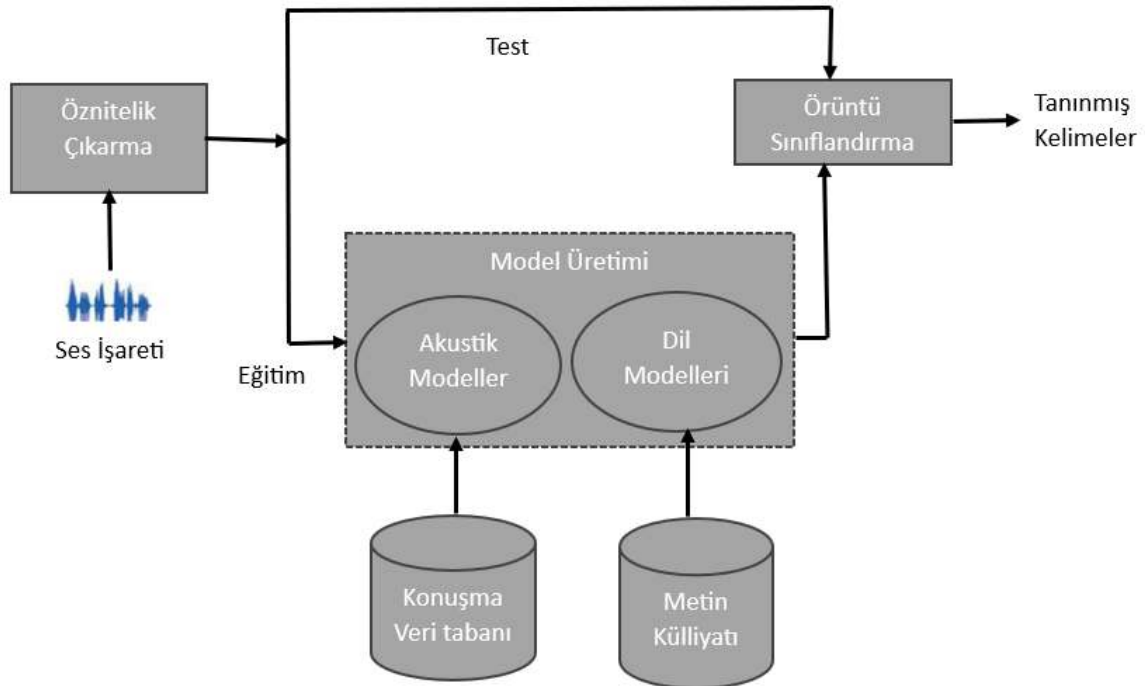
2.1.3 Kelime hazinesine dayalı konuşma tanıma türleri

Kelime hazinesine göre konuşma tanıma türleri dört tanedir. Küçük kelime hazinesine sahip konuşma tanıma sistemleri, orta kelime hazinesine sahip konuşma tanıma sistemleri, geniş kelime hazinesine sahip konuşma tanıma sistemleri ve çok geniş kelime hazinesine sahip konuşma tanıma sistemleri.

Küçük kelime hazinesine sahip konuşma tanıma sistemleri 1-100 arası kelime içerir. Orta kelime hazinesine sahip konuşma tanıma sistemleri 101-1000 arası kelime içerir. Geniş kelime hazinesine sahip konuşma tanıma sistemleri 1001-10000 arası kelime içerir. Çok geniş kelime hazinesine sahip konuşma tanıma sistemleri 10000'den fazla kelime içerir.

2.2 Konuşma Tanıma Sistemlerinin Yapısı

Şekil 2.2'de geleneksel konuşma tanıma sistemlerinin yapısı gösterilmiştir. Konuşma tanıma sistemleri öznitelik çıkarma, sınıflandırma, akustik modeller ve dil modelleri gibi kısımlardan oluşur. Akustik modeller konuşma veritabanları kullanılarak eğitilir ve oluşturulur. Dil modelleri de metin külliyatları kullanılarak oluşturulur ve eğitilir. Alt bölümlerde şekilde gösterilen konuşma tanıma sistemlerinin bölümleri açıklanmıştır.



Şekil 2.2 : Otomatik konuşma tanıma sistemlerinin yapısı (Aggarwal ve Dave, 2012)

2.2.1 Öznitelik çıkarma

Konuşma tanımanın ilk aşamasında kullanılan tekniklerin amacı sınıflandırma için öznitelik vektörleri çıkarmaktır. Öznitelik çıkarmada amaç çok fazla olan konuşma verilerinin, konuşmayı tanımlayabilecek vektörlere indirgenmesi ve bir sonraki aşama olan sınıflandırma için kullanışlı veriler üretmektir (Eskidere, 2007).

Öznitelik çıkarımı, yüksek boyutlu verilerden sınırlı miktarda yararlı bilginin çıkarılması yöntemidir. Konuşma sinyalinin boyutsallığını tipik 80:1 düzeyinde azaltırken, öznitelik çıkarıcılar orijinal konuşmanın ilgili özniteliklerinin çoğunu korur ve konu dışı bilgileri ortadan kaldırır. Böyle bir süreçteki hedeflerden biri, bir kelimenin belirli bir alt birimini başka bir alt birimden ayıran bilgiyi tamamen çıkarmaktır (Aggarwal ve Dave, 2012).

Öznitelik çıkarma, bir konuşmayı diğerinden ayırmada önemli bir rol oynadığı için konuşma tanımanın en önemli parçasıdır. Çünkü her konuşmanın, söylemlerde saklı farklı bireysel özellikleri vardır. Bu özellikler, önerilen ve konuşma tanıma görevi için başarıyla kullanılan çok çeşitli öznitelik çıkarma tekniklerinden çıkarılabilir. Ancak çıkarılan özellik, konuşma sinyaliyle ilgilenirken aşağıdaki gibi bazı kriterleri karşılamalıdır:

- Çıkarılan konuşma özelliklerini ölçmek kolaydır.
- Taklitçiliğe açık olmamalıdır.
- Bir konuşma ortamından diğerine çok az dalgalanma göstermeli.
- Zamanla istikrarlı olmalı.
- Konuşmada sıklıkla ve doğal olarak ortaya çıkmalıdır (Vimala ve Radha, 2012).

Çok kullanılan öznitelik çıkarma tekniklerinde bazılarının isimleri aşağıdadır:

- Mel Frekanslı Kepstral Katsayıları (MFKK)
- Doğrusal Öngörülü Kodlama (DÖK)
- Algısal Doğrusal Öngörü (ADÖ)
- Göreceli Spektra (RASTA-PLP)
- Ayırık Dalgacık Dönüşümü (ADD)

2.2.2 Sınıflandırma

İnsanlar günlük hayatta yüzleri tanıma, sesleri tanıma gibi sınıflandırma işlemlerini gerçekleştirmektedir. Sınıflandırma, kategori üyeliği olarak bilinen gözlemleri veya

örnekleri içeren bir eğitim seti temelinde yeni bir gözlemin ait olduğu bir grup kategorinin tanımlanmasıdır (Çakır, 2017).

Sınıflandırma birimi öznitelik çıkarma biriminden aldığı verileri kullanarak konuşulan kelimeleri tespit eder. OKT sistemlerinde kullanılan bazı sınıflandırma teknikleri aşağıdadır:

- Dinamik zaman bükmesi
- Saklı markov modelleri
- Yapay sinir ağları
- Destek vektör makineleri

2.2.3 Akustik model

Akustik model, fonetik birimlerin tanınması için akustik özellikleri temsil eden otomatik konuşma tanıma sistemi için en önemli bilgi kaynaklarından biridir. Akustik model oluştururken temel ve önemli konulardan biri temel modelleme birimlerinin seçimidir. Genel olarak konuşursak, konuşmanın hedef dili belirlendiğinde akustik modelleme için kullanılabilecek çeşitli alt kelime birimleri vardır. Farklı akustik modelleme birimleri, konuşma tanıma sisteminin performansı üzerinde dramatik bir fark yaratabilir (Karpagavalli ve Chandra, 2016).

Konuşmanın akustik modellemesi tipik olarak konuşma dalga formundan hesaplanan özellik vektör dizileri için istatistiksel temsillerin oluşturulması sürecini ifade eder. Saklı Markov Modeli (SMM), akustik modeller oluşturmak için en yaygın kullanılan istatistiksel modellerden biridir. Diğer akustik modeller, bölümsel modelleri, süper bölümsel modelleri (gizli dinamik modeller dahil), sinir ağlarını, maksimum entropi modellerini ve (gizli) koşullu rastgele alanları içerir. Akustik model, bir kelimeyi oluşturan farklı seslerin her birinin istatistiksel temsillerini içeren bir dosyadır. Bu istatistiksel temsillerin her birine fonem adı verilen bir etiket atanır. Akustik model, konuşma külliyatı adı verilen geniş bir konuşma veri tabanı alınarak ve bir dildeki her fonem için istatistiksel temsiller oluşturmak üzere özel eğitim algoritmaları kullanılarak oluşturulur. Her fonemin kendi SMM'si vardır. Konuşma sınıflandırıcı, kullanıcı tarafından konuşulan farklı sesleri dinler ve ardından akustik modelde eşleşen bir SMM arar. Konuşulan her kelime w , baz sesler adı verilen bir dizi temel sese ayrıştırılır. Akustik model, baz sese verilen belirli bir gözlemin olasılığını açıklar (Karpagavalli ve Chandra, 2016).

2.2.4 Dil modeli

Dil modeli, belirli bir dilde kabul edilebilir kelimelerin sırasına ilişkin kısıtlamaların bir koleksiyonudur. Bu kısıtlamalar, örneğin üretken bir dilbilgisi kurallarıyla veya basitçe bir eğitim derleminde tahmin edilen her kelime çiftine ilişkin istatistiklerle temsil edilebilir. Her ne kadar benzer sese sahip kelimeler olsa da insanlar genellikle bu kelimeyi tanımakta zorluk çekmezler. Bunun temel nedeni bağlamı bilmeleri ve aynı zamanda bağlamda hangi sözcük veya ifadelerin geçebileceği konusunda da oldukça iyi bir fikre sahip olmalarıdır. Bu bağlamı bir konuşma tanıma sistemine sağlamak, dil modelinin amacıdır. Dil modeli, dildeki geçerli kelimelerin neler olduğunu ve bunların hangi sırayla oluşabileceğini belirtir (Karpagavalli ve Chandra, 2016).

Dil modelleri genellikle eğitilir; yani n-gram olasılıkları, tipik olarak milyonlarca kelime belirteci içeren metin bütünündeki kelime dizilerinin gözlemlenmesiyle ve eğitim verilerindeki karışıklığın azaltılmasıyla tahmin edilir. Bununla birlikte, kafa karışıklığının azalmasının her zaman daha iyi konuşma tanıma sonuçlarına yol açmadığı gözlemlenmiştir. Bu nedenle, konuşma tanıma üzerindeki etkilerine dayalı olarak dil modellerini geliştiren algoritmalar, söylenen kelimelerin geçmişi göz önüne alındığında, konuşmacının daha sonra söyleyebileceği kelimelerin olasılık dağılımını belirleyen bir dil modeli özellikle çekicidir. Yaygın dil modelleri bigram ve trigram modelleridir. Bu modeller sırasıyla iki veya üç belirli kelimelerin bir dizide gruplandırılmasının hesaplanmış olasılıklarını içerir (Karpagavalli ve Chandra, 2016).

2.3 Konuşma Tanıma Sistemlerinin Uygulamaları

Son zamanlarda, büyük verinin ve bilgi işlem gücünün katlanarak büyümesiyle birlikte OKT teknolojisi, daha zorlu uygulamaların gerçeğe dönüştüğü aşamaya ilerledi. Örnekler arasında sesli arama ve mobil cihazlarla etkileşimler (örneğin, iPhone'da Siri, winPhone'da Bing sesli arama ve Andriod'da Google Now), ev eğlence sistemlerinde ses kontrolü (örneğin, xBox'ta Kinect) ve çeşitli konuşma merkezli bilgi işleme yer almaktadır. Bu tipik uygulamalardan bazıları dikte sistemleri, sesli kullanıcı arayüzleri, sesli arama, çağrı yönlendirme, ev aletleri kontrolü, komuta ve kontrol, sesle etkinleştirilmiş arama, basit veri girişi, eller ve gözler serbest uygulamalar ve engelli kişiler için öğrenme sistemidir (Karpagavalli ve Chandra, 2016).

OKT sistemleri iletişim, eğitim, elektronik eşya, savunma sanayi, adli ve istihbarat gibi alanlarda sıklıkla kullanılmaktadır. OKT sistemleri ile metin haline dönüştürülen konuşmalar üzerinde yapay zekâ teknolojileri kullanılarak konuşma anlama ve duygu analizi gibi işlemler de gerçekleştirilmektedir. Konuşma anlama ile komuta kontrol işlemleri yapılabildiği gibi aynı zamanda kişisel asistanlar da geliştirilebilmektedir . Duygu analizi ise bir konuşmanın içeriğine göre konuşmacının içerisinde bulunduğu duygu durumunun belirlenmesidir. Konuşma içerisindeki duyguların analizi ile özellikle çağrı merkezlerinde hizmet kalitesinin artırılması amaçlanmaktadır. Ayrıca konuşma metinleri analiz edilerek müşteri ilişkileri yönetilmekte ve müşterilere özel fırsatlar sunulmaktadır (Oyucu ve diğ., 2019).

OKT sistemleri sadece sosyal, kültürel ve ekonomik alanda değil aynı zamanda adli ve istihbari alanlarda da yaygın olarak kullanılmaktadır. İstihbari alanda şüpheli şahısların konuşmalarının OKT sistemleri ile metne aktarılması ve bu metinler üzerinde suç analizi yapılması çalışılmaktadır. Yine istihbarat amacıyla terör gruplarına ait karasal, uydu, kablo ve internet üzerinden yapılan radyo ve televizyon yayınlarında geçen konuşmalar OKT sistemleri ile metin haline getirilmektedir. Bu metinler üzerinde kelime yakalama veya büyük veri analizi yapılabilmektedir. Konuşulan ifadenin doğru bir şekilde metne aktarılması ise önem arz etmektedir. Bu nedenle OKT sistemlerinin başarımını arttırmak amacıyla birçok çalışma yapılmıştır. Fakat OKT sistemlerinin başarısı henüz istenilen seviyeye ulaşamamıştır(Oyucu ve diğ., 2019).

Çizelge 2.1: Konuşma tanıma teknolojisinin kullanıldığı alanlar ve uygulamalar (Oyucu ve diğ., 2019).

Kullanım Alanı	Uygulama	Girdi	Çıktı
İletişim	Telefon rehberinden sesli sorgulama	Konuşma İşareti	Konuşulan Kelimeler
Eğitim	Yabancı dil eğitiminde kelimelerin doğru okunuşunu öğretmek	Konuşma İşareti	Konuşulan Kelimeler
Elektronik	Fırın, çamaşır makinesi ve buzdolabı gibi ev eşyalarında konuşma tanıma	Konuşma İşareti	Konuşulan Kelimeler

Savunma	Savunma sanayide kullanılan araçların kontrol edilmesi	Konuşma İşareti	Konuşulan Kelimeler
Yapay Zeka	Robotların kontrolü	Konuşma İşareti	Konuşulan Kelimeler
Sağlık	Doktorların raporlarını yazmak	Konuşma İşareti	Konuşulan Kelimeler
Çağrı Merkezleri	Telefon görüşmelerinin otomatik yazıya dökülmesi ve kalite kontrol	Konuşma İşareti	Konuşulan Kelimeler

2.4 Konuşma Tanıma Sistemlerinde Performans

Konuşma tanıma sistemlerinin performansı sıklıkla doğruluk ve hız açısından tanımlanır. Doğruluk, Kelime Hata Oranı (KHO) ile ölçülürken, hız ise gerçek zaman faktörü (GZF) ile ölçülür (Das, 2012). Kelime hata oranı yerine bazen de kelime tanıma oranı(KTO) kullanılır.

2.4.1 Kelime hata oranı (KHO)

Kelime hata oranı, konuşma tanıma veya makine çeviri sisteminin performansının ortak bir ölçüsüdür. Performansı ölçmenin genel zorluğu, tanınan kelime dizisinin referans kelime dizisinden (doğru olduğu varsayılan) farklı bir uzunluğa sahip olabilmesi gerçeğinde yatmaktadır. KHO, fonem düzeyi yerine sözcük düzeyinde çalışan Levenshtein mesafesinden türetilmiştir. Bu sorun ilk önce tanınan sözcük dizisinin dinamik dize hizalaması kullanılarak referans (konuşulan) sözcük dizisiyle hizalanmasıyla çözülür. Kelime hata oranı daha sonra denklem 2.1'deki gibi hesaplanabilir:

$$KHO = \frac{S+D+I}{N} \quad (2.1)$$

Burada:

S , ikame sayısıdır.

D , silmelerin sayısıdır.

I , eklemelerin sayısıdır.

N , referanstaki kelime sayısıdır.

Bir konuşma tanıma sisteminin performansını bildirirken bazen bunun yerine Kelime Tanıma Oranı (KTO) kullanılır. Kelime tanıma oranı denklem 2.2'deki gibidir:

$$KTO = 1 - KHO = \frac{N-S-D-I}{N} = \frac{H-I}{N} \quad (2.2)$$

Burada:

$$H = N-(S+D) \quad (2.3)$$

doğru tanınan kelimelerin sayısıdır (Das, 2012).

2.4.2 Hız

Bir konuşma tanıma sisteminin hızı genellikle Gerçek Zaman Faktörü (GZF) cinsinden ölçülür. I süresindeki bir girişi işlemek P zaman alır, denklem 2.4'deki gibi tanımlanır:

$$GZF = \frac{P}{I} \quad (2.4)$$

$GZF \leq 1$ gerçek zamanlı işleme anlamına gelir (Das, 2012).

2.5 Konuşma Veritabanları

Amerika ve Avrupa dillerinde otomatik konuşma tanıma konusunda araştırma yapmak için pek çok konuşma veri tabanı mevcuttur. Yaygın olarak kullanılan veritabanlarından bazıları TIMIT, GlobalPhone, Aurora, Wall Street Journal, AN4, TI Digits, TI46, NTIMIT, RM1, RM2, Switch Board'dur. TIMIT okuma konuşması külliyatı, akustik-fonetik çalışmalar için konuşma verileri sağlamak ve otomatik konuşma tanıma sistemlerinin geliştirilmesi ve değerlendirilmesi için tasarlanmıştır. DARPA TIMIT akustik-fonetik sürekli konuşma külliyatı (TIMIT - Texas Instruments ve Massachusetts Teknoloji Enstitüsü), fonetik olarak dengeli, uyarılmış İngilizce konuşmanın kayıtlarını içerir. Sennheiser yakın konuşma mikrofonu kullanılarak 16 kHz hızında, 16 bit örnek çözünürlükte kaydedildi. TIMIT, Amerika Birleşik Devletleri'nin 8 ana lehçe bölgesinden 630 konuşmacının her biri tarafından konuşulan 10 cümleden oluşan toplam 6300 cümle içermektedir. Tüm cümleler ses düzeyinde manuel olarak bölümlere ayrıldı (Karpagavalli & Chandra, 2016).

Çizelge 2.2: Konuşma veritabanları karakteristikleri (Anusuya ve Katti, 2009).

Konuşma Veritabanı İsmi	Konuşmacı Sayısı	Birim Sayısı	Konuşma Biçimi	Kayıt Ortamı	Örnekleme Oranı kHz	Transkripsiyon Temeli
TI Digits	326	>2500 sayı	Okuma	Sessiz oda	20	Kelime
TIMIT	630	6300 cümle	Okuma	Sessiz oda	16	Ses
NTIMIT	630	6300 cümle	Okuma	Telefon	8	Ses
RM1	144	15024 cümle	Okuma	Sessiz oda	20	Cümle
RM2	4	10608 cümle	Okuma	Sessiz oda	20	Cümle
ATIS0	36	10722 ifade	Okuma Spontan	Ofis	16	Cümle
Switch Board (Kredi kartı)	69	35 diyalog	Spontan	Telefon	8	Kelime
Switch Board (Tamamı)	550	2500 diyalog	Spontan	Telefon	8	Kelime
ATC	100	30000 diyalog	Spontan	Radyo frekansı	8	Cümle

ATIS2	351	12000 ifade	Spontan	Ofis	16	Cümle
-------	-----	----------------	---------	------	----	-------

2.6 Konuşma Araçkitleri

Konuşma tanıma görevini mükemmelleştirmek için pek çok çalışma yapıldı. Yıllar boyunca gerçekleştirilen işlerin bir kısmı Cortana ve Siri gibi kişisel asistan araçları olarak elimizde bulunmaktadır. Bu alanda pek çok araştırma yapılmış olmasına rağmen çalışmaların çoğu kamuya açıklanmamıştır.

Çizelge 2.3: Konuşma tanıma için mevcut araç setleri (Malik ve diğ., 2020).

Araç İsmi	Programlama Dili	Açık Kaynak	Desteklenen Diller
Kaldi	C++	Evet	İngilizce
CMU Sphinx	Java	Evet	İngilizce, Fransızca, Mandarin
Julius	C	Evet	İngilizce, Japonca
HTK	C	Hayır	İngilizce
RWTH ASR	C++	Hayır	İngilizce

Kaldi, açık kaynaklı bir konuşma tanıma aracıdır. Bu araç C++ ile geliştirilmiştir ve birden fazla işletim sistemine kolayca kurulabilir. Şu anda bu araç seti yalnızca İngilizce dilini desteklemektedir. Kaldi, özellikleri çıkarmak için kullanılabilir; sınıflandırma görevlerini de gerçekleştirebilir. Özellikler, en sık kullanılan MFCC'yi, cepstral ortalama ve varyans normalizasyonunu (CMVN) ve i-vektörleri içeren birden fazla yöntem kullanılarak çıkarılabilir. Sınıflandırma görevini gerçekleştirmek için derin sinir ağları (DSA) kullanılır.

CMU Sphinx, başka bir açık kaynaklı konuşma tanıma aracıdır. Bu araç Java'da geliştirilmiştir ve İngilizce, Fransızca, Mandarin Çincesi, Rusça ve Almanca dahil olmak üzere çeşitli diller için önceden eğitilmiş modeller sağlayabilir. CMU Sphinx, özellikleri çıkarmak için MFCC'yi kullanır ve sınıflandırma görevini gerçekleştirmek için SMM tabanlı

bir model kullanır. Ayrıca Sphinx için dil modelleri oluşturmak için kullanılabilecek çevrimiçi bir araç da sağlar (Malik ve diğ., 2020).

Julius, başlangıçta Japon dilini tanımak için tasarlanmış açık kaynaklı bir konuşma tanıma aracıdır. Yıllar geçtikçe farklı araştırmaların da yardımıyla İngilizce dili için de kullanılabilir bir model geliştirildi. Julius'un kendisi, o dil için bir akustik model ve dil modeli mevcut olduğu sürece herhangi bir dil için bir tanıyıcı oluşturmak için kullanılabilen, dilden bağımsız bir kod çözme programıdır.

Saklı Markov Model Araç Takımı (HTK), Saklı Markov Modellerini işlemek ve oluşturmak için kullanılabilecek taşınabilir bir araç setidir. Bu araç setinin oluşturulmasının ardındaki asıl amaç, konuşma tanıma görevini gerçekleştirmektir ancak diğer örüntü tanıma problemleri için de kullanılabilir.

RWTH ASR konuşma tanıma için geliştirilen başka bir araç setidir. Bu araç seti hem konuşma tanıma hem de konuşmacı uyarlaması için kullanılabilir. Özellikleri çıkarmak için MFCC ve PLP'yi kullanır. Akustik modelleme GMM kullanılarak gerçekleştirilir. Bu araç setinin önemli sınırlamalarından biri yalnızca Linux ve macOS'ta mevcut olmasıdır (Malik ve diğ., 2020).

3. ÖZNİTELİK ÇIKARMA YÖNTEMLERİ

Öznitelik çıkarma, makine öğrenimi algoritmalarının performansını iyileştirmek için verilerin boyutunu veya karmaşıklığını azaltan bir tekniktir. Öznitelik çıkarma veri setini yalnızca önemli değişkenlerini veya niteliklerini içerecek şekilde basitleştirir ve veri analizini iyileştirir. Özellikle görüntü işleme, doğal dil işleme ve sinyal işleme gibi alanlarda büyük veri kümeleriyle çalışırken, çoğu işe yaramaz veya gereksiz olabilen birden fazla özellik içeren verilerle karşılaşmak olağandır. Öznitelik çıkarma verileri basitleştirir, bu özellikler orijinal verilerin temel özelliklerini yakalar ve daha verimli işleme ve analize olanak tanır.

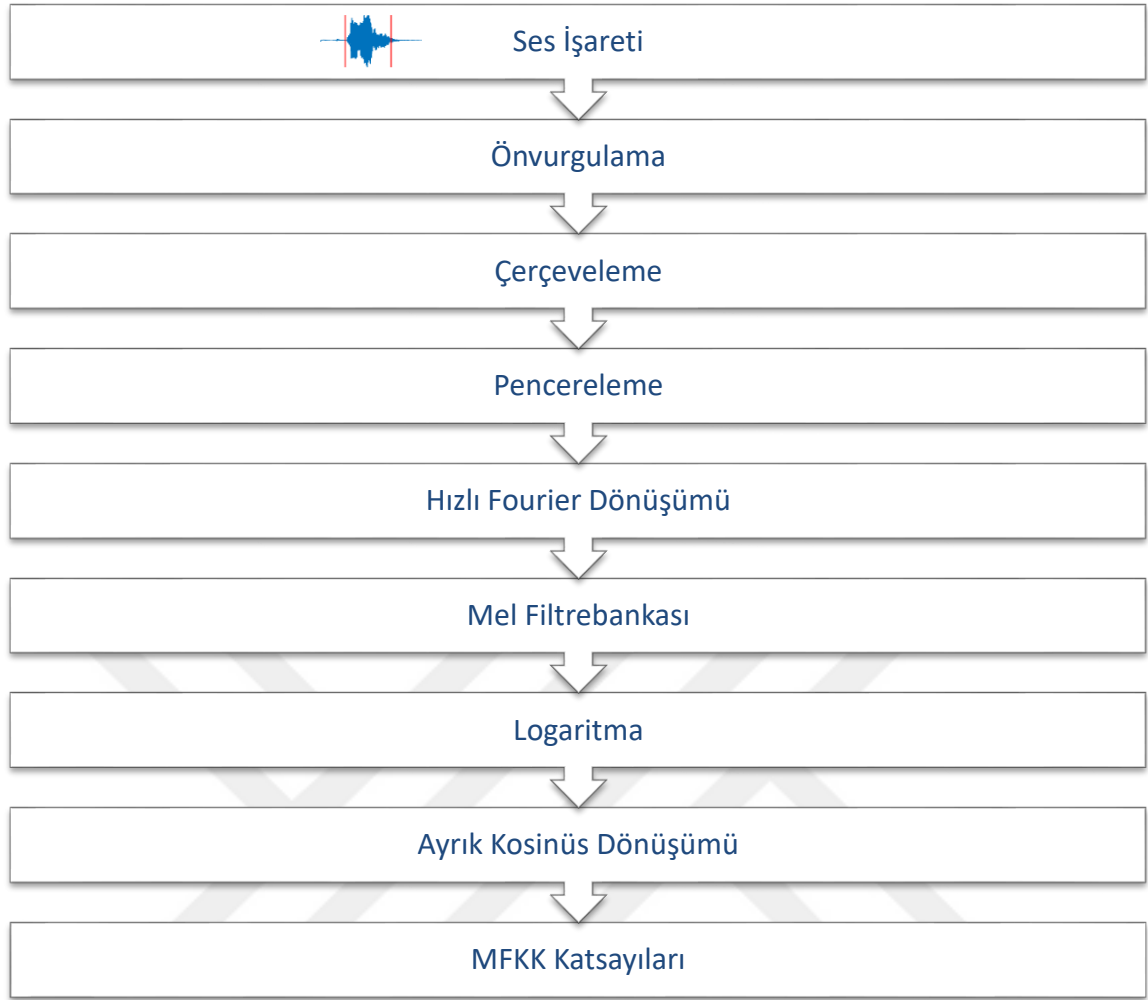
Bu bölümde öznitelik çıkarma yöntemleri anlatılmıştır. Alt bölümlerde öznitelik çıkarma yöntemlerinden Mel Frekanslı Kepstral Katsayıları ve Doğrusal Öngörülü Kodlama yöntemlerinin nasıl bulunduğu anlatılmıştır.

3.1 Mel Frekanslı Kepstral Katsayıları

Mel Frekanslı Kepstral Katsayıları (MFKK) algoritması ilk kez Davis ve Mermelstein tarafından 1980 yılında duyurulmuştur. Duyurulduktan sonra en çok kullanılan algoritmalarından biri olmuştur. MFKK kullanılarak yapılan konuşma tanıma uygulamalarında tanıma oranının oldukça yüksek çıktığı tespit edilmiştir (Başaran, 2007).

MFKK, konuşma örneklerinin benzersiz özelliklerini çıkarmak için kullanılır. İnsan konuşmasının kısa süreli güç spektrumunu temsil eder. MFKK tekniği, doğrusal aralıklı filtreler ve logaritmik aralıklı filtreler olmak üzere iki tür filtreden yararlanır. Konuşmanın fonetik olarak önemli özelliklerini yakalamak için sinyal Mel frekans ölçeğinde ifade edilir. Mel ölçeği esas olarak insan tarafından algılanan perde veya frekansın gözlemlenmesi çalışmasına dayanmaktadır. Ölçek mel birimlerine bölünmüştür. Mel ölçeği normalde 1000 Hz'nin altında doğrusal bir haritalamadır ve 1000 Hz'nin üzerinde Logaritmik olarak aralıklıdır (Chavan & Sable, 2013).

Şekilde 3.1'de MFKK elde edilmesinin adımları gösterilmektedir:



Şekil 3.1 : MFKK katsayılarını elde etme basamakları

3.1.1 Önvurgulama

Sinyal, yüksek frekansları vurgulayan bir filtreden geçirilir. Bu işlem yüksek frekanstaki sinyalin enerjisini artırır. Yüksek frekans aynı zamanda bilgi de içerir. Ön vurguyu belirtmek için kullanılan denklem, denklem 3.1’de gösterilmiştir:

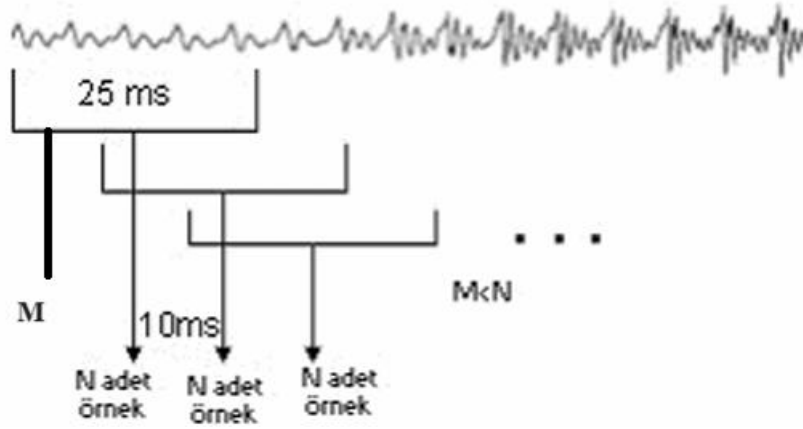
$$S(n) = X(n) - a * X(n - 1) \quad (3.1)$$

Burada $S(n)$ çıktı örneğini, $X(n)$ mevcut örneği, $X(n-1)$ geçmiş örneği ve a 'nın değeri 0,95 ila 1 arasındadır.

3.1.2 Çerçeveleme

Bu aşamada ses sinyali üzerinde kısa zamanlı spektral analiz yapabilmek için sürekli ses sinyali, N adet örnekten oluşan çerçevelere bölünür. İlk çerçeve N örnekten oluşurken ikinci çerçeve ilk çerçeveden M örnek sonra başlar ve ilk çerçevenin $N-M$ örnek kadar üzerine

biner. Kullanılan tipik değerler $N=256$ ve $M=100$ 'dür. Tipik çerçeveler 20 ms - 40 ms arasındadır. Şekil 3.3'te çerçeveleme işlemi görülmektedir (Eray, 2008).



Şekil 3.2: Çerçeveleme İşlemi(Eray, 2008).

3.1.3 Pencereleme

Bu aşamada ses sinyalinin başındaki ve sonundaki süresiz bölümleri en aza indirmek için her çerçeve pencerelenir. Burada amaç, pencere kullanılarak çerçevelerin başındaki ve sonundaki bilgi içermeyen bölümlerin kırılması böylece spektral bozulmayı engellemektir. Yaygın olarak kullanılan Hamming, Hanning, Blackman, Gauss, dikdörtgen ve üçgen pencereleme fonksiyonlarıdır. Hamming pencere fonksiyonunun matematiksel ifadesi denklem 3.2'deki gibidir (Öcal, 2005);

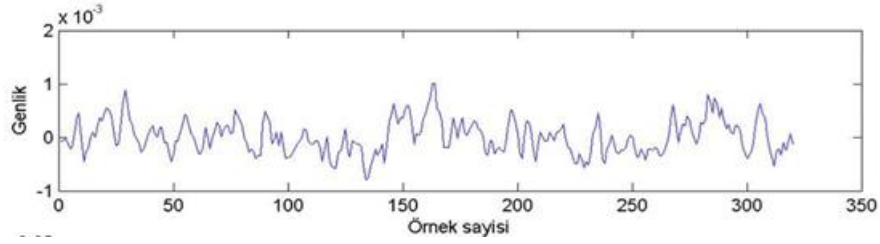
$$w[k + 1] = 0,54 + 0,46 \cos \left(2\pi \frac{k}{n-1} \right) \quad k = 0, \dots, n-1 \quad (3.2)$$

3.1.4 Hızla fourier dönüşümü

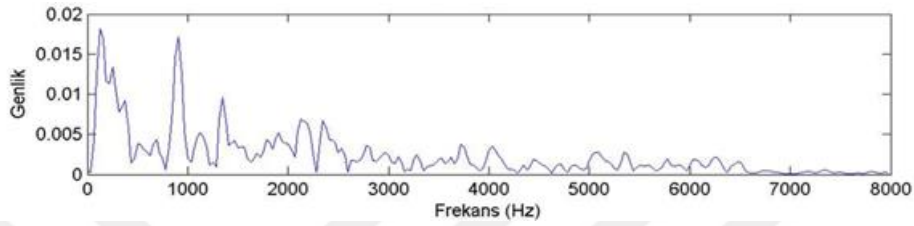
Bu aşamada pencerelenen işaretin hızlı fourier dönüşümü alınır. Hızlı fourier dönüşümü ile N örnekten oluşan zaman alanındaki her bir çerçeve, frekans alanına çevrilir. Hızlı fourier dönüşümü, ayrık fourier dönüşümünden üretilmiştir. Tipik olarak N değeri, 256 veya 512'dir. N örnek $\{x_n\}$ olarak denklem 3.3'deki gibi tanımlanır (Eskidere, 2007).

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-2\pi jkn/N} \quad n = 0, 1, 2, \dots, N-1 \quad (3.3)$$

Şekil 3.3’de bir konuşma işareti parçası görülmektedir. Şekil 3.4’de ise konuşma işaretinin hızlı fourier dönüşümü alınmış halinin grafiği verilmiştir.



Şekil 3.3 : Bir konuşma parçası işareti (Eskidere, 2007)



Şekil 3.4 : Şekil 3.3’deki işaretin hızlı fourier dönüşümü alınmış hali (Eskidere, 2007)

3.1.5 Mel frekansına çevirme

Normal bir insan kulağının frekansları algılaması doğrusal olmayan bir biçimdedir. Araştırmalara göre ölçüler, 1 kHz’e kadar doğrusal olarak, daha yüksek değerlerde ise logaritmiktir olarak artmaktadır. İnsan kulağının frekans cevabını karakterize eden bu ölçüye melodi ölçüsü denir. Bir frekans mel-frekansına çevirmek için denklem 3.4 kullanılır:

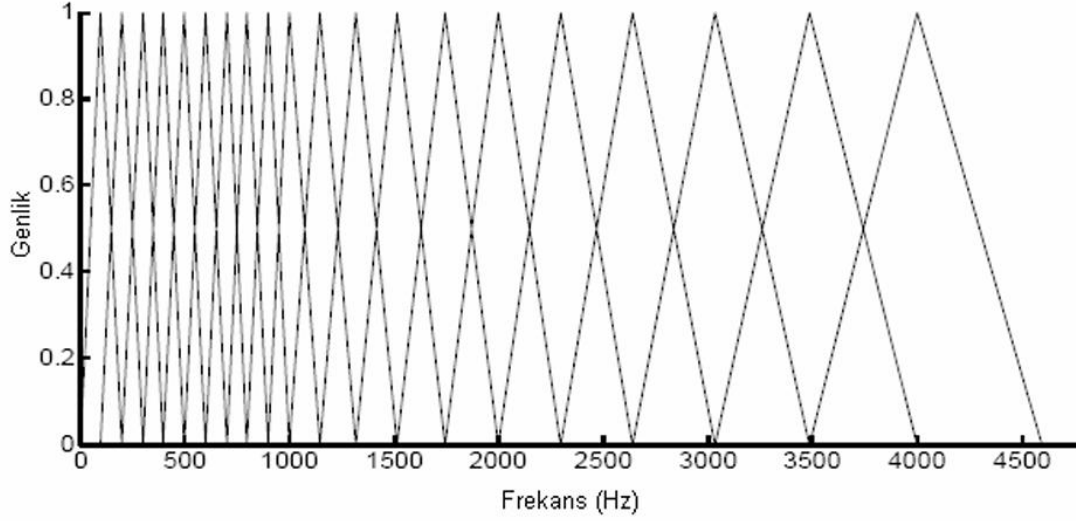
$$m(f) = 2595 * \log\left(1 + \frac{f}{700}\right) \quad (3.4)$$

Bu aşamada bant geçiren filtreler kullanılır. Kullanılan filtre bankı üçgen bant geçiren özelliktedir. Bu filtre bankı ses işaretinin hızlı fourier dönüşümü ile frekans alanına çevrilmiş haline uygulanır(Şaşmaz, 2011).

Şekil 3.5’te Davis ve Mermelstein tarafından tanımlanan üçgen süzgeç dizileri görülmektedir. Burada ilk 10 süzgecin merkez frekansları doğrusal olarak, sonraki 10 süzgeç ise logaritmik olarak yerleştirilmiştir. Tüm süzgeçler eşit genliğe sahiptir.

Çizelge 3.1’de Mel ölçeğin merkez frekansları ve bant genişlikleri verilmiştir.

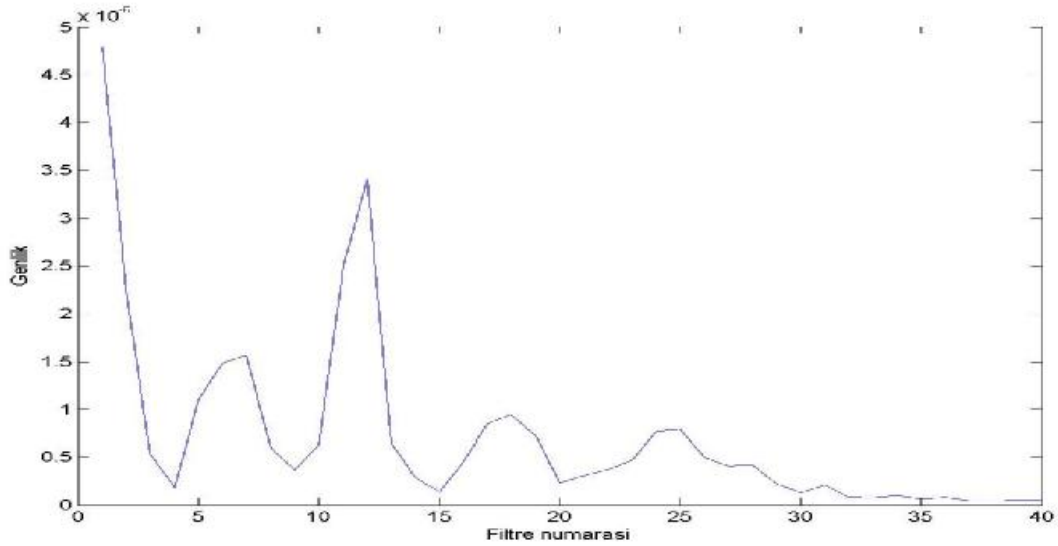
Şekil 3.6’da Konuşma işaretinin süzgeç dizisinden geçirildikten sonraki durumunun grafiği gösterilmiştir.



Şekil 3.5 : Mel ölçeğe dizilmiş üçgen süzgeç dizileri (Tunalı, 2005).

Çizelge 3.1: Mel ölçeğin merkez frekansları ve bant genişlikleri (Eskidere, 2007)

Süzgeç indeksi	Mel Ölçek Davis ve Mermelstein (1980)	
	Merkez fr. (Hz)	BW (HZ)
1	100	100
2	200	100
3	300	100
4	400	100
5	500	100
6	600	100
7	700	100
8	800	100
9	900	100
10	1000	124
11	1149	160
12	1320	184
13	1516	211
14	1741	242
15	2000	278
16	2297	320
17	2639	367
18	3031	422
19	3482	484
20	4000	556
21	4595	639
22	5278	734
23	6063	843
24	6964	969



Şekil 3.6 : Konuşma işaretinin süzgeç dizisinden geçirildikten sonraki durumu (Eskidere, 2007)

3.1.6 Logaritma

Konuşma işareti süzgeç dizisinden geçirildikten sonra logaritması alınır.

3.1.7 Ayrık kosinüs dönüşümü

Bu aşamada son olarak logaritması alınan işaretin ayrık kosinüs dönüşümü alınır. Böylece işaret frekans alanından tekrar zaman alanına geçer. Bu işlemin sonucunda Mel Frekansı Kepstral Katsayıları(MFKK) elde edilmiş olur. Her çerçeve için elde edilen vektörlere de öznelik vektörleri denir. Ayrıca her çerçeve için elde edilen vektörlere akustik vektörler de denir. Böylece konuşma işareti öznelik vektörleri veya akustik vektörlere dönüştürülür. Ayrık kosinüs dönüşümü denklem 3.5 ile hesaplanır:

$$mfkk_i = \sum_{k=1}^K (X_k) \cos \left[i * \left(k - \frac{1}{2} \right) * \frac{\pi}{2} \right] \quad i = 1, 2, \dots, K \quad (3.5)$$

K , kepstum katsayılarının sayısını, X_k , $i = 1, 2, \dots, K$, ise k . filtrenin log-enerji çıktısını göstermektedir(Şaşmaz, 2011).

3.2 Doğrusal Öngörülü Kodlama

Kodlamaya ve sıkıştırmaya imkan veren ses analiz teknikleri arasında yer alanlarından birisi, doğrusal öngörülü kodlama (DÖK) analizdir. Bu temel ses parametrelerinin tahmin edilmesi

ve düşük bit oranlarında ses ile haberleşme ve sesin depolanması için sistemin ifade edilmesinde vazgeçilmez bir tekniktir. Bu yöntemin önemi, ses parametrelerinin büyük oranda doğrulukla tahmin edilmesi ve matematiksel işlemlerdeki hız üstünlüğüdür. Temel olarak DÖK analizi, ses örneğinin, geçmiş ses örneklerinin doğrusal kombinasyonu olarak ifade edilmesi prensibine dayanmaktadır. Sonlu bir zaman aralığı için gerçek ses örnekleri ile doğrusal olarak tahmin edilen örneklerinin kareleri farkının toplamının minimize sonucu öngörü katsayıları elde edilir. Bu öngörü katsayıları, doğrusal kombinasyonların ağırlık katsayıları olacaktır (Özişbakan, 2011).

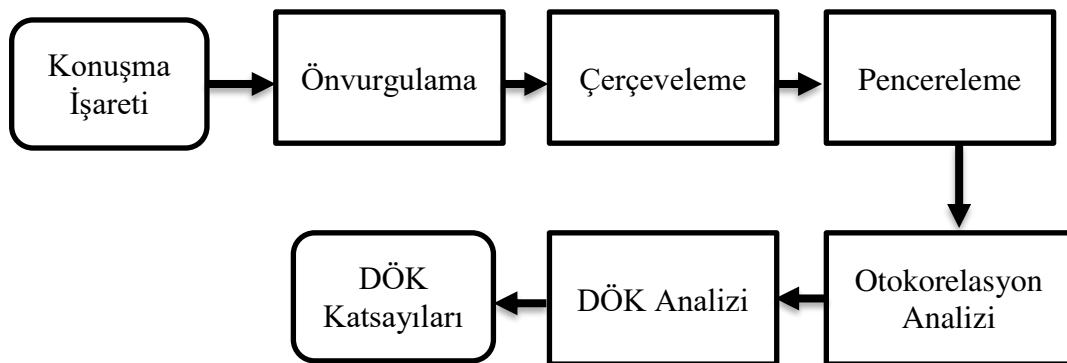
DÖK, doğrusal bir filtre olarak denklem 3.6 ve denklem 3.7 ile ifade edilir:

$$s(n) = a_1s(n-1) + a_2s(n-2) + \dots + a_ps(n-p) \quad (3.6)$$

$$s(n) = \sum_{i=1}^p a_i s(n-i) \quad (3.7)$$

DÖK, sıradaki örneğin, önceki bir dizi örnekten elde edilebileceği prensibi ile çalışır. Tahmin sonucu elde edilen örneğin asıl örnekle olan farkının yani hataların kareleri toplamının en aza indirilmesi için bir dizi parametre hesaplanır. Doğrusal öngörülül kodlama yöntemi ile p sayıda parametre hesaplanır. Burada p DÖK kodlayıcının derecesi; $a_1, a_2, \dots, a_p$ ise DÖK katsayıları olarak ifade edilir. DÖK parametreleri DÖK kodlayıcının çıkışıdır, bu çıkış DÖK kodlayıcının girişine verilen bir çerçeve örneğe karşılık gelen çıktıdır (Eray,2008).

DÖK öznitelik çıkarımı işlem adımları Şekil 3.7’de gösterilmiştir.



Şekil 3.7: DÖK öznitelik çıkarımı işlem adımları (Eray,2008).

3.2.1 Önvurgulama

Sinyal, yüksek frekansları vurgulayan bir filtreden geçirilir. Bu işlem yüksek frekanstaki sinyalin enerjisini artırır. Yüksek frekans aynı zamanda bilgi de içerir. Ön vurguyu belirtmek için kullanılan denklem, denklem 3.8'de gösterilmiştir:

$$S(n) = X(n) - a * X(n - 1) \quad (3.8)$$

Burada $S(n)$ çıktı örneğini, $X(n)$ mevcut örneği, $X(n-1)$ geçmiş örneği ve a 'nın değeri 0,95 ile 1 arasındadır.

3.2.2 Çerçeveleme

Bu aşamada ses sinyali N adet örnekten oluşan çerçevelere bölünür. İlk çerçeve N örnekten oluşurken ikinci çerçeve ilk çerçeveden M örnek sonra başlar ve ilk çerçevenin $N-M$ örnek kadar üzerine biner. Tipik çerçeveler 20 ms - 40 ms arasındadır. Kaydırma miktarı ise 10 ms -20 ms arasındadır. 16000 Hz'lik örnekleme frekansına sahip bir ses işareti için N ve M değerlerinin bulalım. Çerçevemiz 30 ms olsun. 30 ms saniye için N değeri $(16000/1000)*30$ ile hesaplanır ve 480 bulunur. Yani çerçeve boyumuz 480 örnektir. Çerçeveyi kaydırma miktarımız yani M değerini hesaplayalım. Kaydırma miktarımız 20 ms olsun. 20 ms için M değeri $(16000/1000)*20$ ile hesaplanır ve 320 bulunur. Yani çerçeveyi kaydırma miktarımız 320 örnektir.

3.2.3 Pencereleme

Bu aşamada ses sinyalinin başındaki ve sonundaki süreksiz bölümleri en aza indirmek için her çerçeve pencerelenir. Burada amaç, pencere kullanılarak çerçevelerin başındaki ve sonundaki bilgi içermeyen bölümlerin kırılması böylece spektral bozulmayı engellemektir. Yaygın olarak kullanılan Hamming, Hanning, Blackman, Gauss, dikdörtgen ve üçgen pencereleme fonksiyonlarıdır. Hamming pencere fonksiyonunun matematiksel ifadesi denklem 3.9'daki gibidir (Öcal, 2005);

$$w[k + 1] = 0,54 + 0,46 \cos \left(2\pi \frac{k}{n-1} \right) \quad k = 0, \dots, n - 1 \quad (3.9)$$

3.2.4 Otokorelasyon analizi

Pencerelenmiş sinyale otokorelasyon analizi uygulanır. p DÖK analizin derecesidir. p genellikle 8 ile 16 arası bir değer seçilir. Yapılan uygulamalarda p 14 seçildi. Otokorelasyon analizinin bir faydalı tarafı, sıfırcı otokorelasyonun ilgili çerçevenin enerjisini ifade ediyor

olmasıdır. Bir çerçevenin enerjisi konuşma tanıma sistemleri için önemli bir parametredir (Eray,2008).

3.2.5 DÖK analizi

Bu aşamada her bir çerçeveye ait $p+1$ otokorelasyondan DÖK parametre kümesi hesaplanır. Genellikle DÖK analizi sonucunda elde edilen DÖK katsayıları DÖK parametre kümesi olarak sıkça kullanılmaktadır. Otokorelasyon analizinden DÖK analizine geçişte Durbin metodu gibi bir yöntem kullanılabilir. Neticede elde edilen DÖK parametreleri, a_m DÖK katsayılarından oluşmaktadır.



4. SINIFLANDIRMA YÖNTEMLERİ

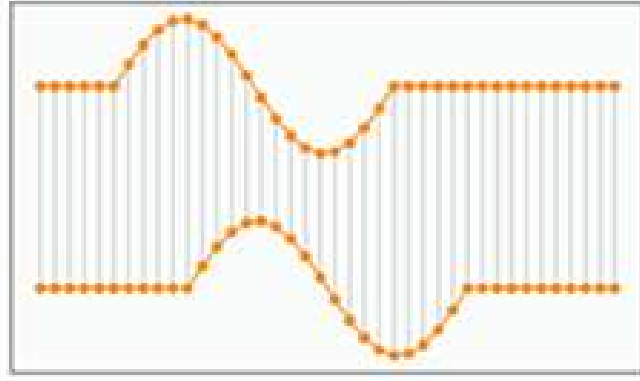
Sınıflandırma, bir makineye nesneleri kategorilere ayırmayı öğretir. Etiketli örneklerle bakarak öğrenir. Öğrendikten sonra, yeni öğelerin hangi kategoriye ait olduğuna karar verebilir, örneğin yeni bir e-postanın spam olup olmadığını belirleyebilir. Örneğin, bir sınıflandırma modeli, fare veya kedi olarak etiketlenen görüntülerin veri kümesi üzerinde eğitilebilir ve renk, doku ve şekil gibi özelliklerine göre yeni ve görülmemiş görüntülerin fare veya kedi olarak sınıfını tahmin etmek için kullanılabilir. Sınıflandırma, makine öğreniminde yeni veri noktalarının kategorisini girdi özelliklerine göre tahmin eden bir öğretmenli öğrenme tekniğidir. Sınıflandırma algoritmaları, doğru kategorinin bulunduğu etiketli verileri kullanarak özellikleri belirli kategorilere nasıl eşleyeceğini öğrenir. Bu işleme kategorileştirme veya kategorik sınıflandırma da denir. Sınıflandırmayı gerçekleştirmek için algoritmalar iki temel aşamada çalışır. Eğitim aşamasında, algoritma girdi verileri ile bunlara karşılık gelen etiketler veya kategoriler arasındaki ilişkiyi öğrenir. Eğitimden sonra model, öğrenilen kalıpları kullanarak gerçek dünya uygulamalarında yeni, görülmemiş verileri sınıflandırdığı çıkarım aşamasına girer. Sınıflandırmanın etkinliği büyük ölçüde bu aşamaların nasıl işlendiğine ve eğitim sırasında mevcut olan önceden işlenmiş verilerin kalitesine bağlıdır.

Bu bölümde sınıflandırma yöntemlerinden ikisi olan dinamik zaman bükmesi ve yapay sinir ağları incelenmiştir. Alt bölümlerde iki yöntem incelenmiştir.

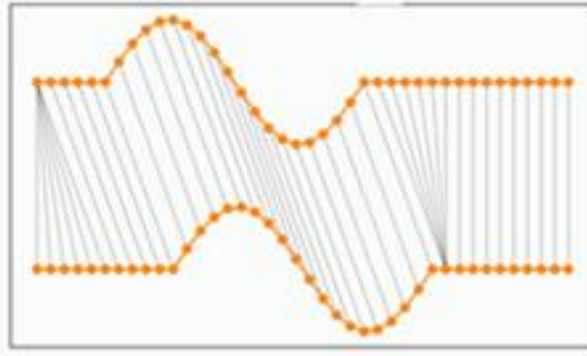
4.1 Dinamik Zaman Bükmesi

Dinamik Zaman Bükmesi (DZB) algoritması iki zaman serisinin birbirine benzer olup olmadığını tespit etmek için kullanılır. DZB şablon temelli bir sınıflandırma yöntemidir. Aynı kelime aynı kişi tarafından iki defa söylene bile konuşma sinyalinin uzunluğu farklılık arzedebilir. DZB'deki ana fikir, hem referans sinyalinin hem de giriş sinyalinin başlangıç ve bitiş zamanlarını, zaman eksenini boyunca uzatarak veya daraltarak doğrusal olmayan bir şekilde hizalamaktır. DZB yöntemi konuşma tanıma uygulamalarında yaygın olarak kullanılmaktadır (Urgun, 2012).

Dinamik Zaman Bükme, mümkün olan en iyi hizalamayı oluşturmak için çoktan bire karşılaştırmalara izin verir ve aralarındaki zamansal çarpıklıkları kullanır; Öklid ise bire bir nokta karşılaştırmasına izin verir. Dinamik Zaman Bükmenin iki değişken uzunluklu dizi veya zaman dizisi için kullanılmasının nedeni budur. Şekil 4.1’de iki işaretin öklid mesafesi eşleştirmesi ve Şekil 4.2’de iki işaretin DZB eşleştirmesi görülmektedir (Dynamic Time Warping(DTW) Algorithm in Time Series).



Şekil 4.1: İki işaretin öklid mesafesi eşleştirmesi



Şekil 4.2: İki işaretin DZB eşleştirmesi

DZB'de M ve N adlı iki zaman serisi arasındaki “D” ile temsil edilen mesafe, denklem 4.1’de formüle edilen dinamik programlama yaklaşımı kullanılarak hesaplanabilir. $M = (X_1, X_2, X_3... X_M)$ ve $N = (Y_1, Y_2, Y_3... Y_N)$, “X” ve “Y”nin serinin elemanları olduğu iki zaman serisi olarak kabul edilir.

$$D(i, j) = \text{minimum} \begin{cases} D(i, j - 1) \\ D(i - 1, j) \\ D(i - 1, j - 1) \end{cases} + d(X_i, Y_j) \quad (4.1)$$

Konuşma şablonlarının benzerliklerini belirlemek için şablonlar arasındaki mesafenin hesaplanması gerekir. Bu şablonların sınıflandırılması bu benzerliklere göre yapılmaktadır. Mesafe ölçümü, denklem 4.2’de formüle edilen Öklid mesafe ölçümü ile verimli bir şekilde yapılır;

$$d(x, y) = \sqrt{\sum_i (X_i - Y_i)^2} \quad (4.2)$$

Referans şablonları ile giriş şablonu arasındaki en az mesafeyi sağlayan şablon, en iyi eşleşen şablona karşılık gelir. Hızlı çalışan algoritması ve kolay uygulanması Dinamik Zaman Bükme yönteminin en avantajlı özellikleridir. Ancak eğitim seti arttıkça DZB'deki performans düşüyor(Urgun, 2012).

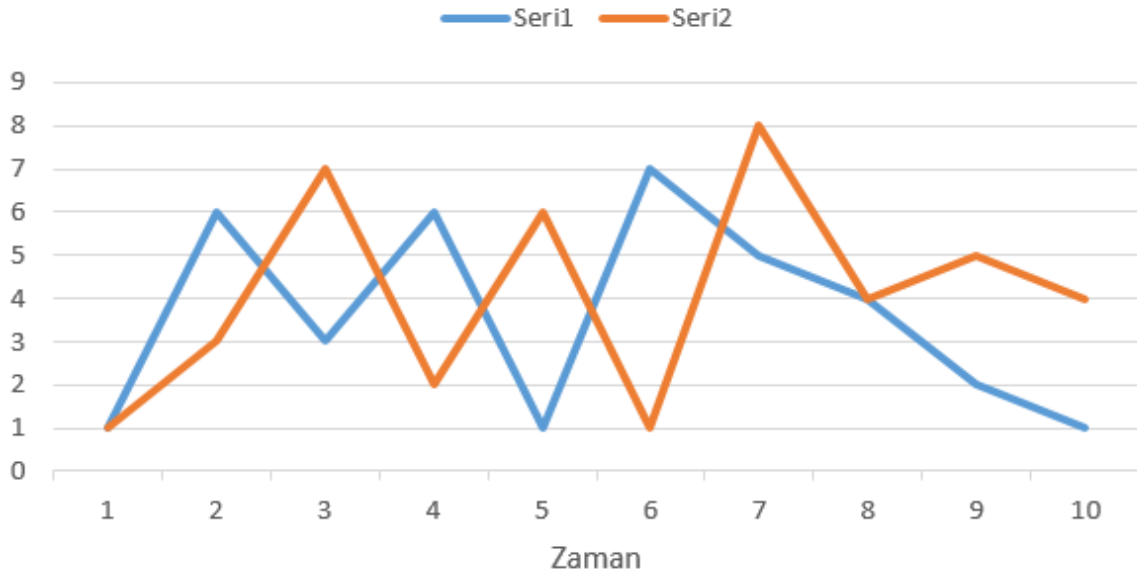
4.1.1 DZB algoritmasının çalışması

Algoritmanın çalışmasını bir örnekle açıklayalım. İki tane zaman serimiz olsun.

Seri1 = [1 6 3 6 1 7 5 4 2 1]

Seri2 = [1 3 7 2 6 1 8 4 5 4]

Şekil 4.2’de bu iki işaretin grafiği yer almaktadır.



Şekil 4.3: İki zaman serisinin grafiği

Önce Şekil 4.4’deki gibi boş bir maliyet matrisi oluşturalım.

1										
2										
4										
5										
7										
1										
6										
3										
6										
1										
	1	3	7	2	6	1	8	4	5	4

Şekil 4.4: Boş bir maliyet matrisi

Maliyet matrisimiz denklem 4.1'e göre doldurulur. Matrisin doldurulmuş hali aşağıdadır.

1	26	18	22	13	17	10	17	11	12	11
2	26	16	18	12	13	10	16	8	9	8
4	25	15	13	12	9	10	12	6	7	6
5	22	14	10	11	7	10	8	6	6	7
7	18	12	8	10	6	9	5	8	10	13
1	12	8	10	5	9	4	11	14	17	17
6	12	6	4	8	4	9	11	13	14	16
3	7	3	7	4	7	9	14	15	17	18
6	5	3	3	7	7	12	14	16	21	23
1	0	2	8	9	14	14	21	24	28	31
	1	3	7	2	6	1	8	4	5	4

Şekil 4.5: Doldurulmuş maliyet matrisi

Maliyet matrisinde sağ üst köşeden başlayıp, sol alt köşede sonlanacak yol tespit edilir. Bunun için sol üst köşedeki 4 hücrede en küçük olan sayı seçilir. Daha sonra seçilen hücrenin solundan ve altından seçilen dört hücreden en küçük değeri taşıyan hücre seçilir. Bu yolla

sol alt köşeye kadar devam edilir. Seçilen her hücre sağ üst köşeden sol alt köşeye kadar bükülmüş yolu tanımlar. Şekil 4.6’da bükülmüş yol gösterilmiştir.

Seri1	1	26	18	22	13	17	10	17	11	12	11
	2	26	16	18	12	13	10	16	8	9	8
	4	25	15	13	12	9	10	12	6	7	6
	5	22	14	10	11	7	10	8	6	6	7
	7	18	12	8	10	6	9	5	8	10	13
	1	12	8	10	5	9	4	11	14	17	17
	6	12	6	4	8	4	9	11	13	14	16
	3	7	3	7	4	7	9	14	15	17	18
	6	5	3	3	7	7	12	14	16	21	23
	1	0	2	8	9	14	14	21	24	28	31
		1	3	7	2	6	1	8	4	5	4
		Seri2									

Şekil 4.6: Son bükülmüş yol

Son aşamada da sonuç mesafe hesaplaması yapılır. Burada bükülmüş yol serisi aşağıdadır:

$$d = [11 \ 8 \ 6 \ 6 \ 6 \ 5 \ 4 \ 4 \ 4 \ 3 \ 2 \ 0]$$

mesafe = bükülmüş yol serisinin elemanlarının toplamı / bükülmüş yol serisinin uzunluğu

$$\text{mesafe} = (11 + 8 + 6 + 6 + 6 + 5 + 4 + 4 + 4 + 3 + 2 + 0) / 12$$

$$\text{mesafe} = 4,9167$$

4.2 Yapay Sinir Ağları

Yapay sinir ağları yapay zekâ alanının bir alt dalıdır. YSA biyolojik sinir ağından etkilenerek üretilmiştir. Zekâ veya örüntü tanımayı içeren birçok görevin otomatikleştirilmesi son derece zordur, ancak insanlar tarafından çok kolay bir şekilde gerçekleştirildiği görülmektedir. Örneğin, insanlar çeşitli nesneleri tanır ve çevrelerindeki büyük miktardaki görsel bilgilerden anlam çıkarırlar; görünüşe göre çok az çaba gerektirirler. Benzer görevleri yerine getiren bilgi işlem sistemlerinin, insanların bu görevleri nasıl yerine getirdiğini anlamaktan ve bu süreçleri fiziksel sınırlamaların izin verdiği ölçüde simüle etmekten büyük fayda sağlayacağı mantıklıdır. Bu, Sinir Ağlarının incelenmesini ve simülasyonunu gerektirir. Bir insanın sinir ağı, çok sayıda birbirine bağlı nöron (sinir hücreleri) içeren sinir

sisteminin bir parçasıdır. “Sinir” nöron için bir sıfattır ve “Ağ” grafik benzeri bir yapıyı ifade eder. Yapay Sinir Ağı, ana teması biyolojik sinir ağları benzetmesinden alınan bilgi işlem sistemlerini ifade eder. Yapay Sinir Ağları “Sinir Ağları”, yapay sinir sistemleri “paralel dağıtılmış işlem sistemleri” ve “bağlantıcı sistemler” olarak da anılmaktadır. Bir bilişim sisteminin bu güzel isimlerle anılması için sistemin, düğümlerin bazı basit hesaplamaları gerçekleştirdiği etiketli yönlendirilmiş grafik yapısına sahip olması gerekir. Temel grafik teorisinden, bir "Yönlendirilmiş Grafiğin" bir dizi "Düğüm" (köşeler) ve düğüm çiftlerini birbirine bağlayan bir dizi "Bağlantı" (kenarlar/bağlantılar/yaylar) içerdiğini biliyoruz. Bir sinir ağında, her düğüm bazı basit hesaplamalar gerçekleştirir ve her bağlantı, bir sinyalin ne kadar güçlendirildiğini veya azaltıldığını gösteren "Bağlantı Gücü" veya "Ağırlık" adı verilen bir sayıyla etiketlenen bir sinyali bir düğümden diğerine iletir. Bu sistem insan uzmanlığının ve bilgisinin alternatifidir. Yapay Sinir Ağları beyni yakından takip ederek modellenmiştir ve bu nedenle birçok terminoloji sinir biliminden alınmıştır (Dongare ve diğ., 2012).

4.2.1 Biyolojik nöron

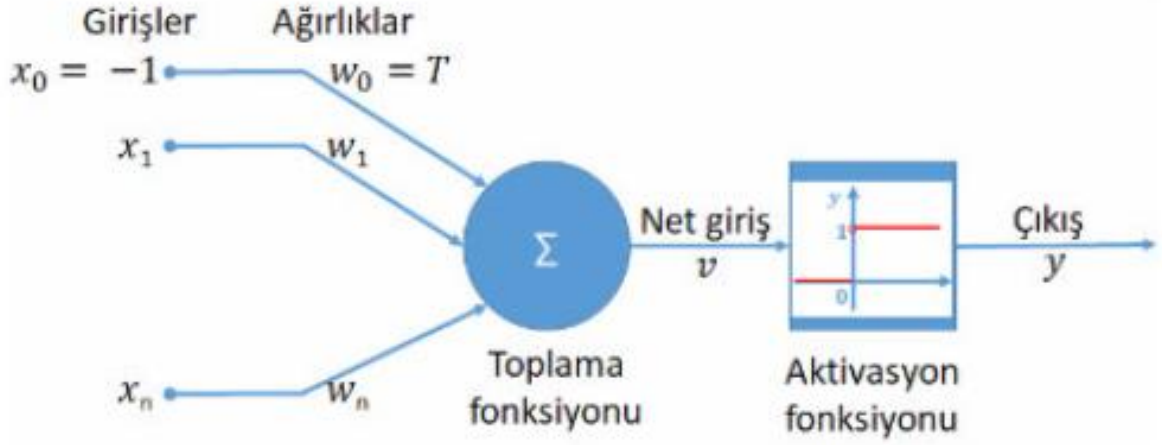
İnsan beyni, düşünme, hatırlama ve problem çözme yeteneklerine sahip karmaşık bir sistemdir. Beyin fonksiyonlarının bilgisayarla taklit edilmesine yönelik girişimlerin başarısı henüz kısmi olmaktan öteye gidememiştir.

Beynin temel birimi nörondur. Nöron, dentritler aracılığıyla sinyalleri alır ve birleştirir. Bileşke sinyalin yeterince güçlü olduğu durumda nöron ateşleme yapar ve sinyal, terminalleri aracılığıyla diğer nöronların dentritleriyle bağlantılı olan akson boyunca yol alır. Akson boyunca yol alarak nörona ulaşan sinyaller, elektriksel yükü değişken hızlarda ileten sıvı ile dolu çok küçük boşluklardan geçerler. Bu boşluklar, sinaptik bağlantılar olarak anılır. Sinaptik bağlantının empedans veya kondüktans değerinin ayarlanması, bellek oluşumu ve öğrenmeyi sağladığından kritik önemdedir.

4.2.2 Yapay nöron

Sinir ağları birbirine paralel olarak çalışan basit elemanlardan oluşur. Gerçek bir sinir hücresinin birimlerine eşdeğer bileşenlerle modellenen yapay sinir hücresi Şekil 4.7’de gösterilmiştir. Gövdenin giriş birimi olan bağlantıların her birinin kendine ait bir ağırlık çarpanı vardır. Ağırlık değeri pozitif veya negatif olabilir. Uygulanan sinyallerin ağırlık değeriyle çarpımları, iki kısımdan oluşan gövdenin ilk kısmında toplanır. Bu toplam, ikinci

kısmı tanımlayan aktivasyon fonksiyonunun argümanı olur. Çizelge 4.1’de biyolojik nöronla yapay nöronun karşılaştırılması yapılmıştır.



Şekil 4.7: Yapay Nöron

Çizelge 4.1: Biyolojik nöronla yapay nöronun karşılaştırılması

Biyolojik Nöron	Yapay Nöron
Dentrit	Girişler
Hücre çekirdeği	Düğümmler
Sinapsis	Ağırlıklar
Akson	Çıkış

Yapay nörona, yapay sinir hücresi, proses elemanı, hücre gibi farklı isimler de verilmektedir. Yapay nöronun 5 temel elemanı vardır (Öztemel, 2006).

Girişler:

Bir yapay sinir hücresine dış dünyadan gelen bilgilerdir. Yapay sinir hücresine dış dünyadan olduğu gibi başka hücrelerden veya kendi kendinden de bilgiler gelebilir.

Ağırlıklar:

Ağırlıklar bir yapay nörona gelen bilginin önemini ve hücre üzerindeki etkisini gösterir. Şekildeki w_i , Giriş 1 olan x_i ’in hücre üzerindeki etkisini gösterir. Ağırlıkların büyük ya da küçük olması önemli veya önemsiz olduğu anlamına gelmez. Bir ağırlığın değerinin sıfır olması o ağ için en önemli olay olabilir. Eksi değerler önemsiz demek değildir.

Toplama Fonksiyonu:

Bu fonksiyon, bir hücreye gelen net girişi hesaplar. Bunun için değişik fonksiyonlar kullanılır. En yaygın olanı ağırlıklı toplamı bulmaktır. Bu yöntemde her gelen giriş değeri kendi ağırlığı ile çarpılarak toplanır. Böylece ağırlıklı net girdi hesaplanmış olur. Bu işlem denklem 4.3 şeklinde formüle edilir:

$$NET = \sum_i^n x_i w_i \quad (4.3)$$

Burada X girişleri, W ise ağırlıkları, n ise bir hücreye gelen toplam girdi sayısını gösterir.

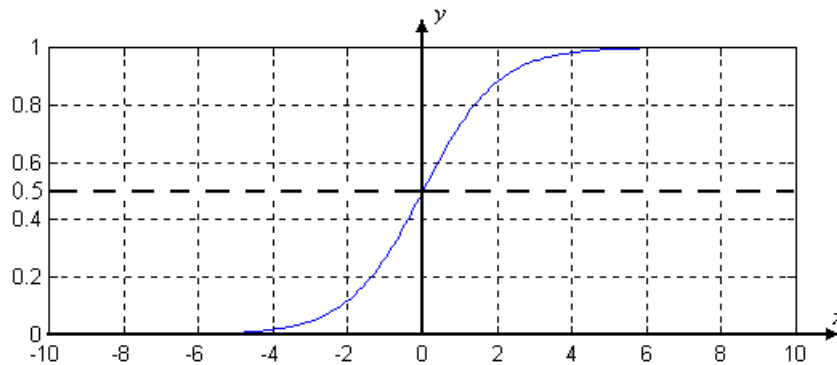
Aktivasyon Fonksiyonu:

Hücreye gelen net girdiyi işleyip hücrenin bu girdiye karşılık üreteceği çıkışı belirler. Toplama fonksiyonunda olduğu gibi aktivasyon fonksiyonu olarak da çıkışı hesaplamak için değişik formüller kullanılır. Bazı modeller bu fonksiyonun türevinin alınabilir bir fonksiyon olmasını şart koşar. Toplama fonksiyonunda olduğu gibi aktivasyon fonksiyonunda da ağırlıklı hücrelerinin hepsinin aynı fonksiyonu kullanması gerekmez. Bazı elemanlar aynı fonksiyonu diğerleri farklı fonksiyonları kullanabilirler.

Günümüzde yaygın olarak kullanılan çok katmanlı algılayıcı modelinde genel olarak aktivasyon fonksiyonu olarak sigmoid fonksiyonu kullanılır. Bu fonksiyon denklem 4.4 ile gösterilir:

$$F(NET) = \frac{1}{1+e^{-NET}} \quad (4.4)$$

Burada NET hücreye gelen NET girdi değerini gösterir. Bu değer toplama fonksiyonu kullanılarak belirlenir. Sigmoid fonksiyonunun grafiği Şekil 4.8’dir (Öztemel, 2006).



Şekil 4.8: Sigmoid fonksiyonu

Çıkışlar:

Aktivasyon fonksiyonu tarafından belirlenen çıkış değeridir. Üretilen çıkış dış dünyaya veya başka bir hücreye gönderilir. Hücre kendi çıktısını kendine giriş olarak da gönderebilir. Hücreden çıkan tek bir çıktı vardır. Tek çıktı birden fazla hücreye giriş olarak girmektedir.

4.2.3 YSA'nın tarihçesi

1940'lı yıllardan bu yana sinir ağları alanında bir dizi kayda değer ilerleme kaydedildi:

1940'lar-1950'ler: İlk Kavramlar

Sinir ağları, McCulloch ve Pitts tarafından yapay nöronların ilk matematiksel modelinin tanıtılmasıyla başladı. Ancak hesaplama kısıtlamaları ilerlemeyi zorlaştırdı.

1960'lar-1970'ler: Algılayıcılar

Bu dönem Rosenblatt'ın algılayıcılar üzerine yaptığı çalışmayla tanımlanmaktadır. Algılayıcılar, uygulanabilirliği ayrı ayrı doğrusal olarak çözülebilecek sorunlarla sınırlı olan tek katmanlı ağlardır.

1980'ler: Geriye Yayılım ve Bağlantıcılık

Çok katmanlı ağ eğitimi Rumelhart, Hinton ve Williams'ın geri yayılım yöntemini icat etmesiyle mümkün oldu. Birbirine bağlı düğümler aracılığıyla öğrenmeye vurgu yapan bağlantıcılık, ilgi kazandı.

1990'lar: Yükseliş ve Duraklama

Görüntü tanımlama, finans ve diğer alanlardaki uygulamalarla sinir ağları bir patlama yaşadı. Ancak sinir ağı araştırması, aşırı hesaplama maliyetleri ve şişirilmiş beklentiler nedeniyle bir duraklama yaşadı.

2000'ler: Diriliş ve Derin Öğrenme

Daha büyük veri kümeleri, yenilikçi yapılar ve gelişmiş işleme kapasitesi geri dönüşü teşvik etti. Derin öğrenme, çok sayıda katmanı kullanarak birçok disiplinde inanılmaz bir etkinlik göstermiştir.

2010'lar-Günümüz: Derin Öğrenmenin Hakimiyeti

Evrişimli sinir ağları (CNN'ler) ve tekrarlayan sinir ağları (RNN'ler), iki derin öğrenme mimarisi, makine öğrenimine hâkim oldu. Güçleri oyun oynama, resim tanıma ve doğal dil işlemedeki yeniliklerle ortaya çıktı.

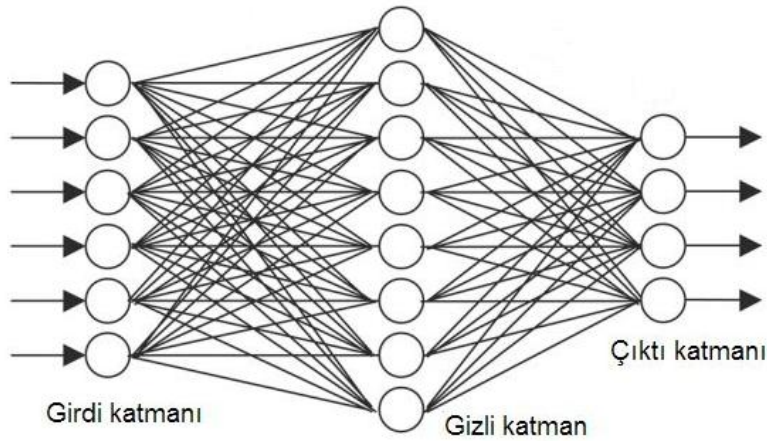
4.2.4 Yapay sinir ağının yapısı

Yapay sinir hücreleri bir araya gelerek yapay sinir ağını oluştururlar. Hücrelerin bir araya gelmesi rasgele olmaz. Genel olarak hücreler 3 katman halinde ve her katman içinde paralel olarak bir araya gelip ağı oluştururlar. Bu katmanlar:

Girdi katmanı: Bu katmandaki hücreler dış dünyadan bilgileri alarak ara katmanlara aktarmaktan sorumludurlar. Bazı ağlarda giriş katmanında herhangi bir bilgi işleme olmaz.

Ara katmanlar (gizli katmanlar): Giriş katmanından gelen bilgileri işleyerek çıkış katmanına iletirler. Bilgilerin işlenmesi ara katmanlarda gerçekleşir. Bir ağ için birden fazla ara katman olabilir.

Çıktı katmanı: Bu katmandaki hücreler ara katmanlardan gelen bilgileri işleyerek ağın giriş katmanından sunulan girdi örnek kümesi için üretilmesi gerekli çıktıyı üretirler. Üretilen çıktı dış dünyaya iletir.



Şekil 4.9: Bir yapay sinir ağının yapısı

4.2.5 Yapay sinir ağlarından en çok kullanılan modeller

Bir yapay sinir ağında hücrelerin bağlanması sonucu oluşan topoloji, hücrelerin sahip oldukları toplama ve aktivasyon fonksiyonları, öğrenme stratejisi ve kullanılan öğrenme kuralı ağın modelinin belirlemektedir. Günümüzde çok sayıda model geliştirilmiştir. Bunların en yaygın olarak kullanılanlarından bazıları aşağıdadır(Öztemel, 2006).

- Algılayıcılar
- Çok katmanlı algılayıcılar
- Vektör kuantizasyon modelleri (LVQ)
- Kendi kendini organize eden model (SOM)

- Adaptif rezonans teorisi modeller (ART)
- Radyal temelli ağılar (RBN)
- Elman ağı
- Probabilistik ağılar (PNN)

4.2.6 YSA'da öğrenme stratejileri

YSA gibi örneklerden öğrenen sistemlerde farklı farklı öğrenme stratejileri kullanılmaktadır. Öğrenmeyi gerçekleştirecek olan sistem ve kullanılan öğrenme algoritması bu stratejilere bağlı olarak değişmektedir. Genel olarak 3 öğrenme stratejisi kullanıldığı görülmektedir.

Öğretmenli (Supervised) öğrenme:

Bu stratejide sistemin olayı öğrenebilmesine bir öğretmen yardımcı olmaktadır. Öğretmen öğrenilmesi istenen olayla ilgili örnekleri Giriş/Çıkış kümesi olarak sisteme verir. Yani her örnek için her girişler hem de o girişler karşılığında oluşturulması gereken çıkışlar sisteme gösterilir. Sistemin yapacağı iş girişleri öğretmenin belirlediği çıkışlara haritalamaktır. Bu sayede olayın girişleri ile çıkışları arasındaki ilişkiler öğrenilir. Çok katmanlı algılayıcı ağı bu stratejiyi kullanan ağlara örnek olarak verilebilir(Öztemel, 2006).

Öğretmensiz (Unsupervised) öğrenme:

Bu stratejide sistemin öğrenmesine yardımcı olan herhangi bir öğretmen yoktur. Sisteme sadece giriş değerleri gösterilir. Örneklerdeki parametreler arasındaki ilişkileri sistemin kendi kendine öğrenmesi beklenir. Bu, daha çok sınıflandırma problemleri için kullanılan bir stratejidir. Yalnız sistemin öğrenmesi bittikten sonra çıktıların ne anlama geldiğinin gösteren etiketlendirmenin kullanıcı tarafından yapılması gerekmektedir(Öztemel, 2006).

Destekleyici (Reinforcement) öğrenme:

Bu stratejide öğrenen sisteme bir öğretmen yardımcı olur. Fakat öğretmen her giriş kümesi için olması gereken çıkış kümesini sisteme göstermek yerine sistemin kendisine gösterilen girişlere karşılık çıktısını üretmesini bekler ve üretilen çıkışın doğru veya yanlış olduğunu gösteren bir işaret üretir. Sistem öğretmenden gelen bu işareti dikkate alarak öğrenme sürecini devam ettirir(Öztemel, 2006).

4.2.7 YSA'da öğrenme kuralları

YSA gibi öğrenen sistemlerde öğrenme yukarıda anlatılan stratejilerden hangisi uygulanırsa uygulansın bazı kurallara göre gerçekleştirilir. Öğrenme sistemlerinde kullanılan değişik

öğrenme kuralları vardır. Yapay sinir ağlarında bu öğrenme kurallarının çoğu Hebb kuralına dayanmaktadır.

Hebb Kuralı:

Bilinen en eski öğrenme kuralıdır. Diğer öğrenme kurallarının temelini oluşturur. 1949 yılında geliştirilen bu kurala göre bir hücre diğer bir hücreden bilgi alırsa ve her iki hücrede aktifse (matematik olarak aynı işareti taşıyorsa) her iki hücrenin arasındaki bağlantı kuvvetlendirilmelidir. Bir hücre kendisi aktif ise bağlı olduğu hücreyi aktif yapmaya pasif ise pasif yapmaya çalışır. Diğer öğrenme kurallarının çoğu bu kuralı temel alarak geliştirilmiştir(Öztemel, 2006).

Hopfield kuralı:

Hebb kuralına benzemektedir. Yapay sinir ağı elemanlarının bağlantılarının ne kadar kuvvetlendirilmesi veya zayıflatılması gerektiği belirlenir. Eğer beklenen çıkış ve girişler ikisi de aktif/pasif ise öğrenme katsayısı kadar ağırlık değerleri kuvvetlendir/zayıflatılmaktadır. Ağırlıkların kuvvetlendirilmesi veya zayıflatılması öğrenme katsayısı yardımı ile gerçekleşir. Öğrenme katsayısı genel olarak 0-1 arasında kullanıcı tarafından atanan sabit ve pozitif bir değerdir(Öztemel, 2006).

Delta kuralı:

Hebb kuralının biraz daha geliştirilmiş şeklidir. Bu kurala göre beklenen çıkış ile gerçekleşen çıkış arasındaki farkı azaltmak için yapay sinir ağı elemanlarının bağlantılarının ağırlık değerlerinin sürekli değiştirilmesi ilkesine dayanarak geliştirilmiştir. Ağın ürettiği çıkış ile beklenen çıkış arasındaki hatanın karelerinin ortalamasını en az yapmak hedeflenir(Öztemel, 2006).

Kohonen kuralı:

Bu kurala göre ağın hücreleri ağırlıklarını değiştirmek için birbirleriyle yarışır. En büyük çıktıyı üreten hücre kazanan hücre olmakta ve bağlantı ağırlıkları değiştirilmektedir. Bu, o hücrenin yanındaki diğer hücrelere karşı daha kuvvetli hale gelmesi demektir. Hem kazanan hücrelerin hem de komşuları sayılan hücrelerin ağırlıklarını değiştirmesine izin verilmektedir (Öztemel, 2006).

5. DİNAMİK ZAMAN BÜKMESİ UYGULAMASI

Matlab’la geliştirdiğimiz konuşma tanıma sisteminin türü konuşmacı bağımlı, yalıtılmış kelimeler, küçük kelime hazinesine sahip bir konuşma tanıma sistemidir. Sistemin tanıdığı kelimeler aşağıdaki tabloda verilmiştir.

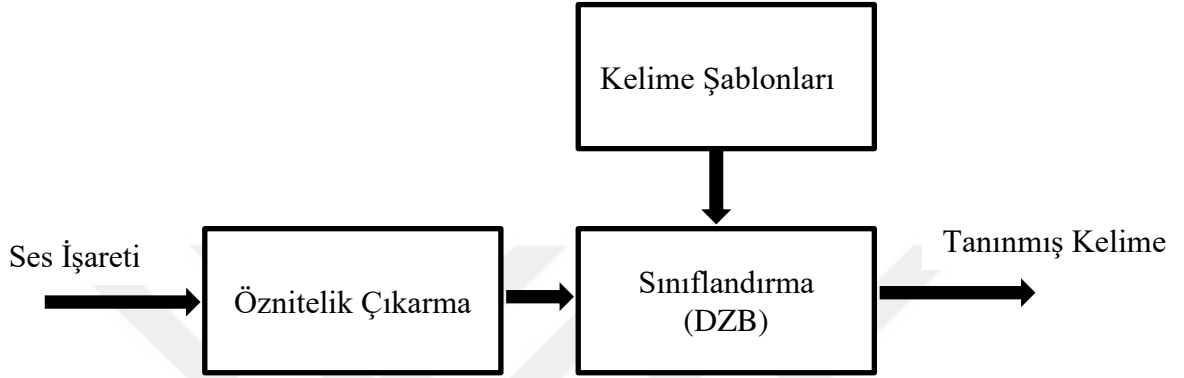
Çizelge 5.1 : Matlab’da geliştirilen konuşma tanıma sisteminin tanıdığı kelimeler

Sıfır	On	Bin	Sol
Bir	Yirmi	Milyon	Aç
İki	Otuz	Milyar	Kapa
Üç	Kırk	Evet	
Dört	Elli	Hayır	
Beş	Altmış	Başla	
Altı	Yetmiş	Dur	
Yedi	Seksen	İleri	
Sekiz	Doksan	Geri	
Dokuz	Yüz	Sağ	

Konuşma tanıma sisteminin blok diyagramı Şekil 5.1’de gösterilmiştir. Sistem iki temel bölümden oluşuyor. Öznitelik çıkarma bölümü ve sınıflandırma bölümü. Öznitelik çıkarma bölümünde ya MFKK katsayıları yada DÖK Katsayıları hesaplanıyor. Sınıflandırma kısmında ise şablonlarla öznitelik çıkarma kısmında elde edilen katsayılar DZB algoritmasıyla karşılaştırılıyor. Giriş ses sinyalinin öznitelik vektörüne en yakın şablon DZB

algoritmasıyla hesaplanıyor. Ses sinyalinin özniteliklerine en yakın şablon özniteliği tanınmış kelime olarak, sistem çıktısı olarak ekrana yazılıyor.

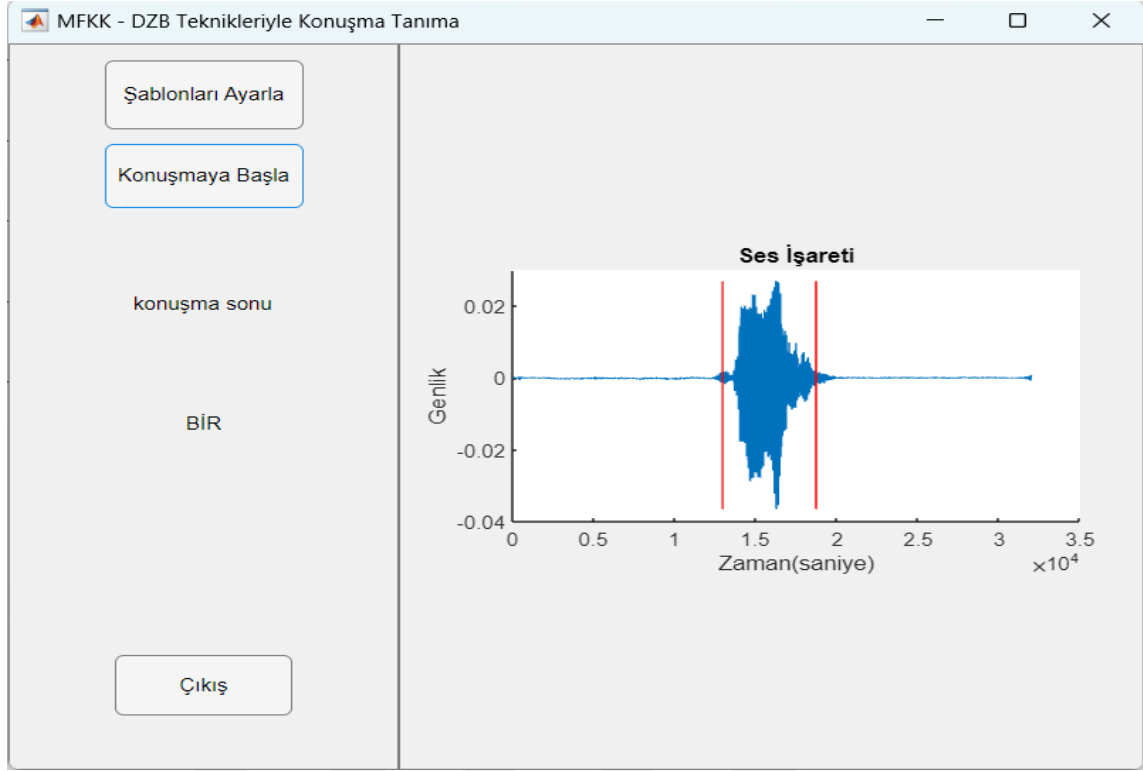
Öznitelik çıkarma kısmında MFKK ve DÖK katsayıları olarak iki farklı yöntem kullanılmıştır. MFKK katsayıları kullanılan sistemin DÖK katsayıları kullanan sisteme göre daha iyi olduğu gözlemlenmiştir.



Şekil 5.1 : Geliştirilen konuşma tanıma sisteminin blok diyagramı

5.1 MFKK –DZB Teknikleriyle Konuşma Tanıma

Konuşma tanıma sisteminin öznitelik çıkarma bölümünde MFKK katsayıları kullanılmıştır. Sınıflandırma için DZB algoritması kullanılmıştır. Programın MATLAB kullanıcı arayüzü Şekil 5.2’ dendir.



Şekil 5.2 : MFKK ve DZB teknikleriyle konuşma tanıma programı kullanıcı arayüzü

Program şu şekilde çalışmaktadır: Şablonları ayarla düğmesine tıklandığında daha önce bilgisayara kaydettiğimiz 33 kelimenin MFKK katsayıları MATLAB'ın mfcc fonksiyonu kullanılarak bulunuyor ve 33 MFKK değeri “sablonlar.mat” dosyasına sonra kullanılmak üzere kaydediliyor.

Konuşmaya başla düğmesine tıklandığında ise program 2 saniye kullanıcıdan kelime söylemesini bekliyor. Kullanıcının söylediği bu kelimedeki sessizlik kısımları ayrılıp (detectSpeech fonksiyonu kullanarak), bu kelimenin MFKK katsayıları hesaplanıyor. Kullanıcının söylediği kelime ile şablonlarımızdaki 1'den 33'üncü kelimeye kadar her bir kelimeye DZB algoritması MATLAB'daki dtw fonksiyonu ile uygulanıyor. Söylenen kelime ile 33 şablonun herbirine DZB algoritması kullanılarak elde edilen sayılardan en küçük olan, tanınmış kelime olarak ekrana yazılıyor.

Ayrıca söylenen her kelimenin grafiği bu kelimenin başlama ve bitme noktaları da gösterilerek form üzerinde çiziliyor.

5.1.1 MFKK –DZB teknikleriyle konuşma tanımanın performansı

Bu sistemin KHO ve KTO parametrelerini bulmak için kelime hazinesindeki her kelime 10 defa okunmuştur ve OKT sisteminin kelimeleri doğru ve yanlış tanıma miktarları aşağıdaki tabloya işlenmiştir.

Çizelge 5.2: MFKK-DZB OKT sisteminin kelimeleri doğru ve yanlış tanıma miktarları

S.N.	Kelime	Doğru	Yanlış	Toplam
1	Sıfır	10	0	10
2	Bir	10	0	10
3	İki	9	1	10
4	Üç	10	0	10
5	Dört	7	3	10
6	Beş	10	0	10
7	Altı	10	0	10
8	Yedi	10	0	10
9	Sekiz	10	0	10
10	Dokuz	10	0	10
11	On	10	0	10
12	Yirmi	10	0	10
13	Otuz	9	1	10
14	Kırk	9	1	10
15	Elli	9	1	10
16	Altmış	9	1	10
17	Yetmiş	10	0	10
18	Seksen	10	0	10
19	Doksan	10	0	10
20	Yüz	9	1	10
21	Bin	7	3	10
22	Milyon	10	0	10
23	Milyar	8	2	10
24	Evet	10	0	10
25	Hayır	8	2	10
26	Başla	10	0	10

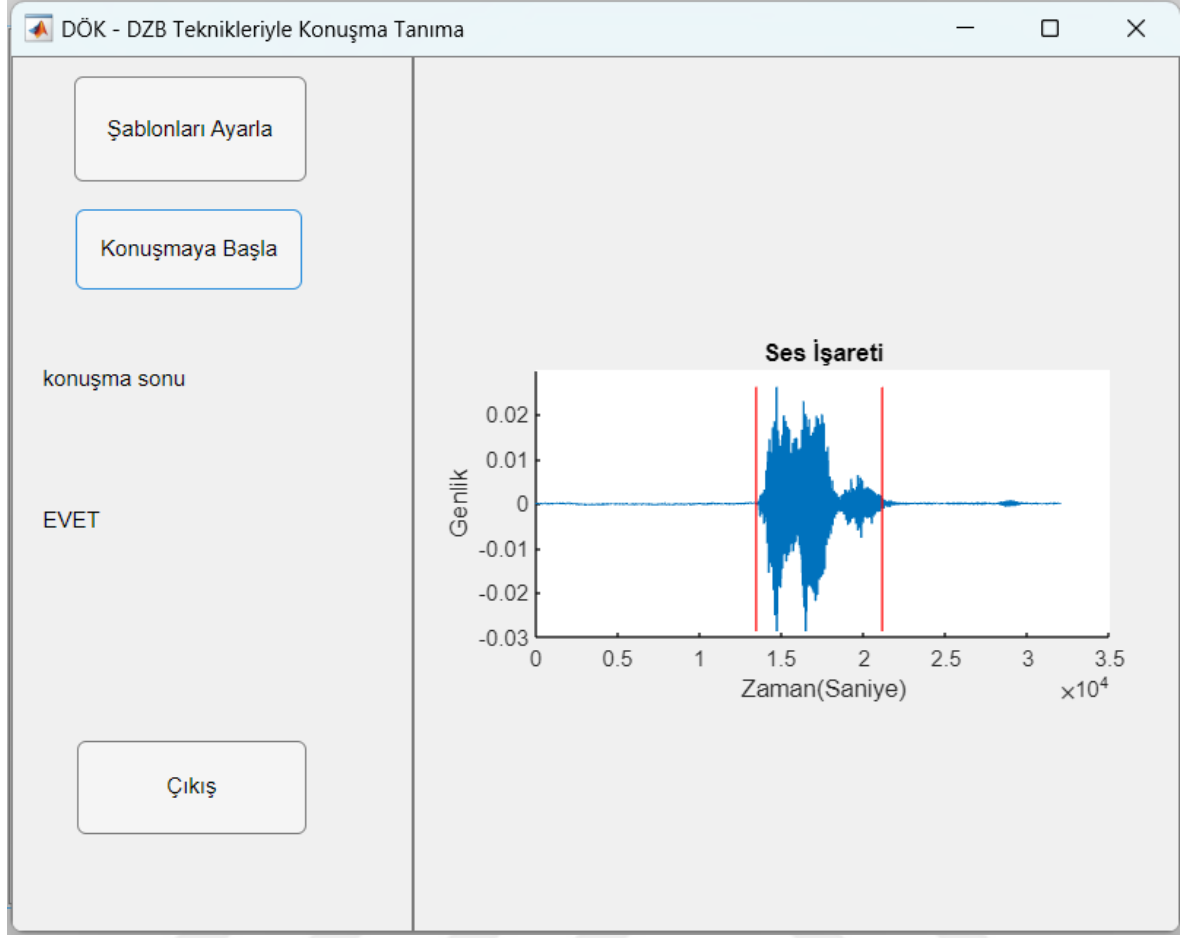
27	Dur	2	8	10
28	İleri	9	1	10
29	Geri	3	7	10
30	Sağ	9	1	10
31	Sol	9	1	10
32	Aç	10	0	10
33	Kapa	9	1	10
Toplam		295	35	330

$$KHO = \frac{\text{Toplam Yanlış}}{\text{Toplam}} * 100 = \frac{35}{330} * 100 = \% 10,6$$

$$KTO = 1 - KHO = 1 - 0,106 = 0,894 \text{ ise } KTO = \% 89,4$$

5.2 DÖK –DZB Teknikleriyle Konuşma Tanıma

Konuşma tanıma sisteminin öznelilik çıkarma bölümünde DÖK katsayıları kullanılmıştır. Sınıflandırma için DZB algoritması kullanılmıştır. Programın MATLAB kullanıcı arayüzü Şekil 5.3' dir.



Şekil 5.3 : DÖK ve DZB teknikleriyle konuşma tanıma programı kullanıcı arayüzü

Program şu şekilde çalışmaktadır: Şablonları ayarla düğmesine tıklandığında daha önce bilgisayara kaydettiğimiz 33 kelimenin DÖK katsayıları MATLAB’ın lpc fonksiyonu kullanılarak bulunuyor ve 33 DÖK değeri “sablonlar.mat” dosyasına sonra kullanılmak üzere kaydediliyor.

Konuşmaya başla düğmesine tıklandığında ise program 2 saniye kullanıcıdan kelime söylemesini bekliyor. Kullanıcının söylediği bu kelimedeki sessizlik kısımları ayrılıp (detectSpeech fonksiyonu kullanarak), bu kelimenin DÖK katsayıları hesaplanıyor. Kullanıcının söylediği kelime ile şablonlarımızdaki 1’den 33’üncü kelimeye kadar her bir kelimeye DZB algoritması MATLAB’daki dtw fonksiyonu ile uygulanıyor. Söylenen kelime ile 33 şablonun herbirine DZB algoritması kullanılarak elde edilen sayılardan en küçük olan, tanınmış kelime olarak ekrana yazılıyor.

Ayrıca söylenen her kelimenin grafiği bu kelimenin başlama ve bitme noktaları da gösterilerek form üzerinde çiziliyor.

5.2.1 DÖK –DZB teknikleriyle konuşma tanımanın performansı

Bu sistemin KHO ve KTO parametrelerini bulmak için kelime hazinesindeki her kelime 10 defa okunmuştur ve OKT sisteminin kelimeleri doğru ve yanlış tanıma miktarları aşağıdaki tabloya işlenmiştir.

Çizelge 5.3: DÖK-DZB OKT sisteminin kelimeleri doğru ve yanlış tanıma miktarları

S.N.	Kelime	Doğru	Yanlış	Toplam
1	Sıfır	10	0	10
2	Bir	9	1	10
3	İki	8	2	10
4	Üç	5	5	10
5	Dört	2	8	10
6	Beş	6	4	10
7	Altı	9	1	10
8	Yedi	10	0	10
9	Sekiz	10	0	10
10	Dokuz	9	1	10
11	On	2	8	10
12	Yirmi	5	5	10
13	Otuz	7	3	10
14	Kırk	9	1	10
15	Elli	10	0	10
16	Altmış	5	5	10
17	Yetmiş	8	2	10
18	Seksen	10	0	10
19	Doksan	5	5	10
20	Yüz	9	1	10
21	Bin	10	0	10
22	Milyon	4	6	10
23	Milyar	6	4	10
24	Evet	8	2	10
25	Hayır	10	0	10
26	Başla	10	0	10

27	Dur	2	8	10
28	İleri	9	1	10
29	Geri	8	2	10
30	Sağ	10	0	10
31	Sol	5	5	10
32	Aç	10	0	10
33	Kapa	7	3	10
Toplam		247	83	330

$$KHO = \frac{\text{Toplam Yanlış}}{\text{Toplam}} * 100 = \frac{83}{330} * 100 = \% 25,15$$

$$KTO = 1 - KHO = 1 - 0,2515 = 0,7485 \text{ ise } KTO = \% 74,85$$

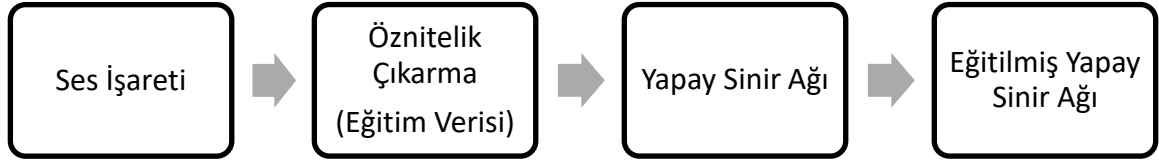
6. YAPAY SİNİR AĞI İLE TÜRKÇE RAKAM TANIMA UYGULAMASI

Bu uygulama konuşmacı bağımlı, küçük kelime hazineli, yalıtılmış kelime konuşma tanıma sistemidir. Kelime hazinesi rakamlardan oluşan 10 kelimedir. Kelime hazinesi aşağıdaki çizelgede verilmiştir.

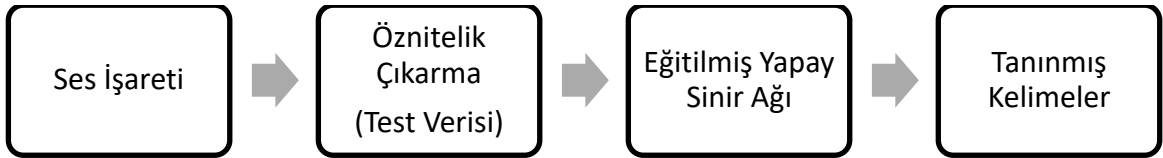
Çizelge 6.1: Konuşma tanıma sisteminin tanıdığı kelimeler.

Sıfır	Bir	İki	Üç	Dört
Beş	Altı	Yedi	Sekiz	Dokuz

Konuşma tanıma sistemi iki aşama ile oluşturulur. İlk aşama yapay sinir ağının eğitim aşaması. Bu aşamada yapay sinir ağı her kelimededen (toplam 10 kelime) 15 adet eğitim verisi, toplamda da 150 adet eğitim verisiyle eğitilir. İkinci aşamada ise eğitilmiş ağa test verisi olan ses işareti uygulanır. Daha önce eğitilmiş olan ağ test için girilen ses işaretini tanır.



Şekil 6.1: Konuşma tanıma sisteminin yapısı (Yapay Sinir Ağının eğitimi aşaması)



Şekil 6.2: Konuşma tanıma sisteminin yapısı (Yapay Sinir Ağının test aşaması)

Konuşma tanıma sisteminin öznitelik çıkarma kısmında ilk uygulamada MFKK ikinci uygulamada ise DÖK katsayıları kullanılmıştır.

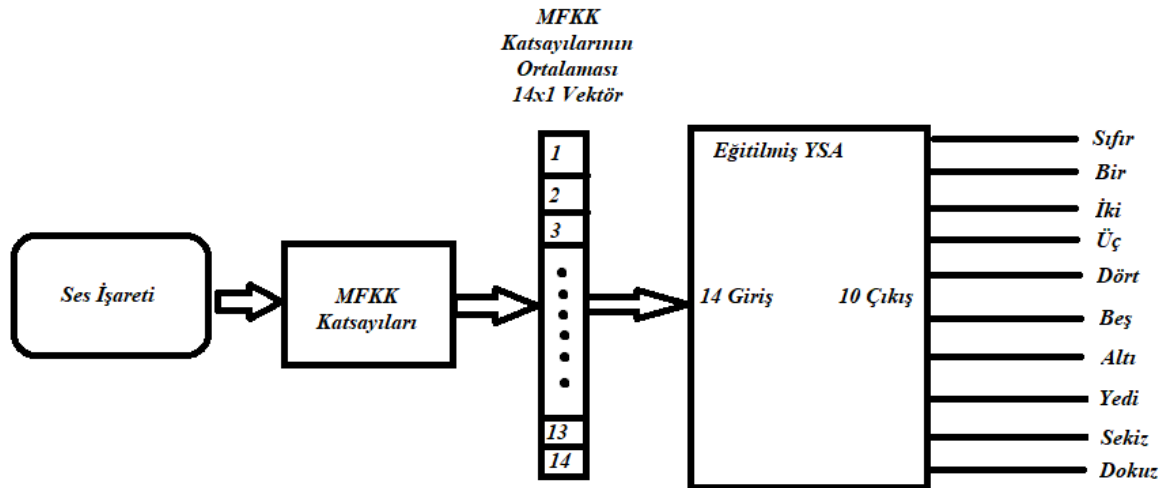
6.1 MFKK –Yapay Sinir Ağı ile Türkçe Rakam Tanıma Uygulaması

Konuşma tanıma sisteminin öznitelik çıkarma kısmında MFKK kullanılır. Ses işaretinin her çerçevesinin MFKK bulunduktan sonra, MFKK'yı gösteren matriste her satırın ortalaması alınır, sonuç olarak 14x1 boyutunda bir vektör ortaya çıkar. 14 elemanlı bu vektör YSA'nın girişine uygulanır. YSA'nın çıkışında ise 10 eleman vardır. YSA 20 tane gizli katman barındırır.

Uygulama iki aşamadan oluşur:

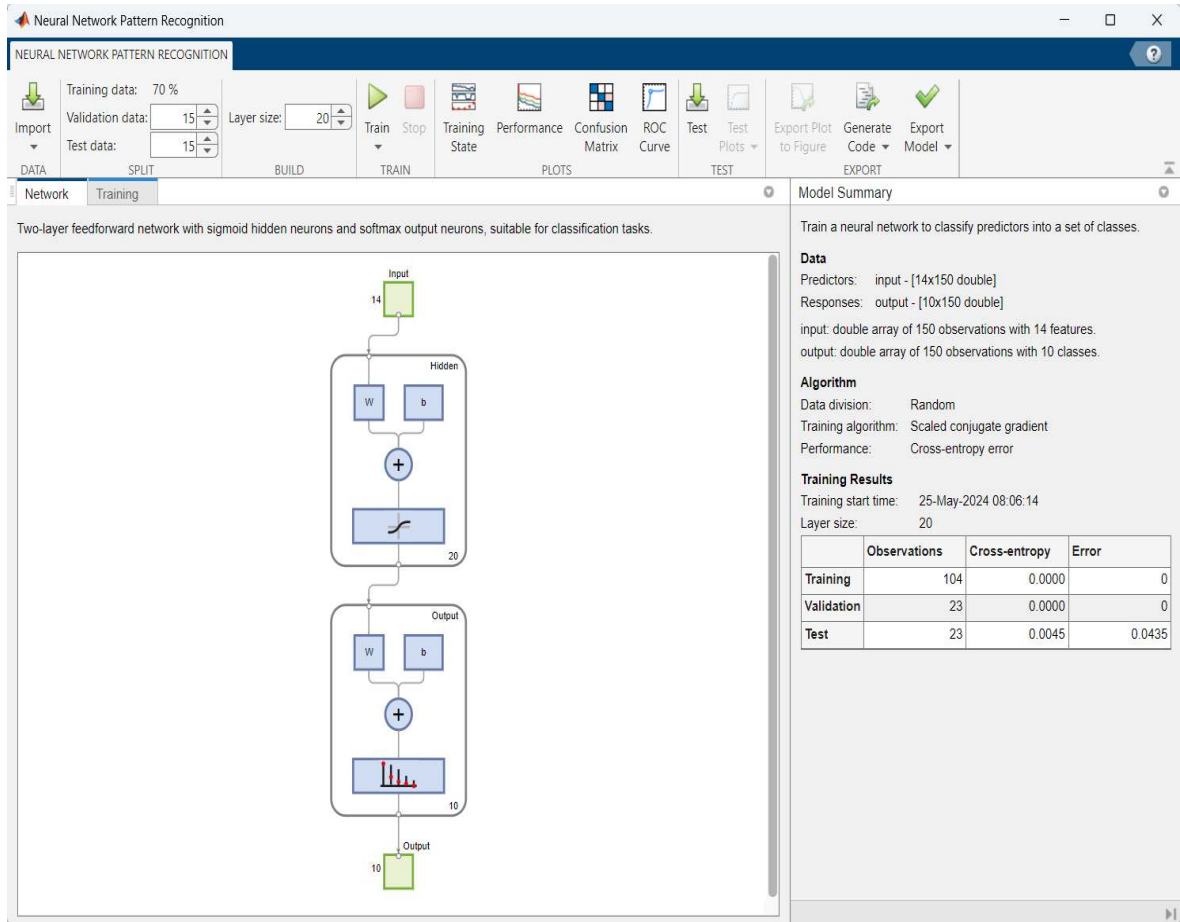
1. Aşama: YSA'nın eğitilmesi aşaması
2. Aşama: YSA'nın test edilmesi. Ses işaretinin tanıma için YSA'ya uygulanması

Şekil 6.3'te Öznitelik çıkarım aşamasında MFKK kullanılan, sınıflandırma aşamasında da YSA'nın kullanıldığı konuşma tanıma sisteminin blok şeması gösterilmiştir.



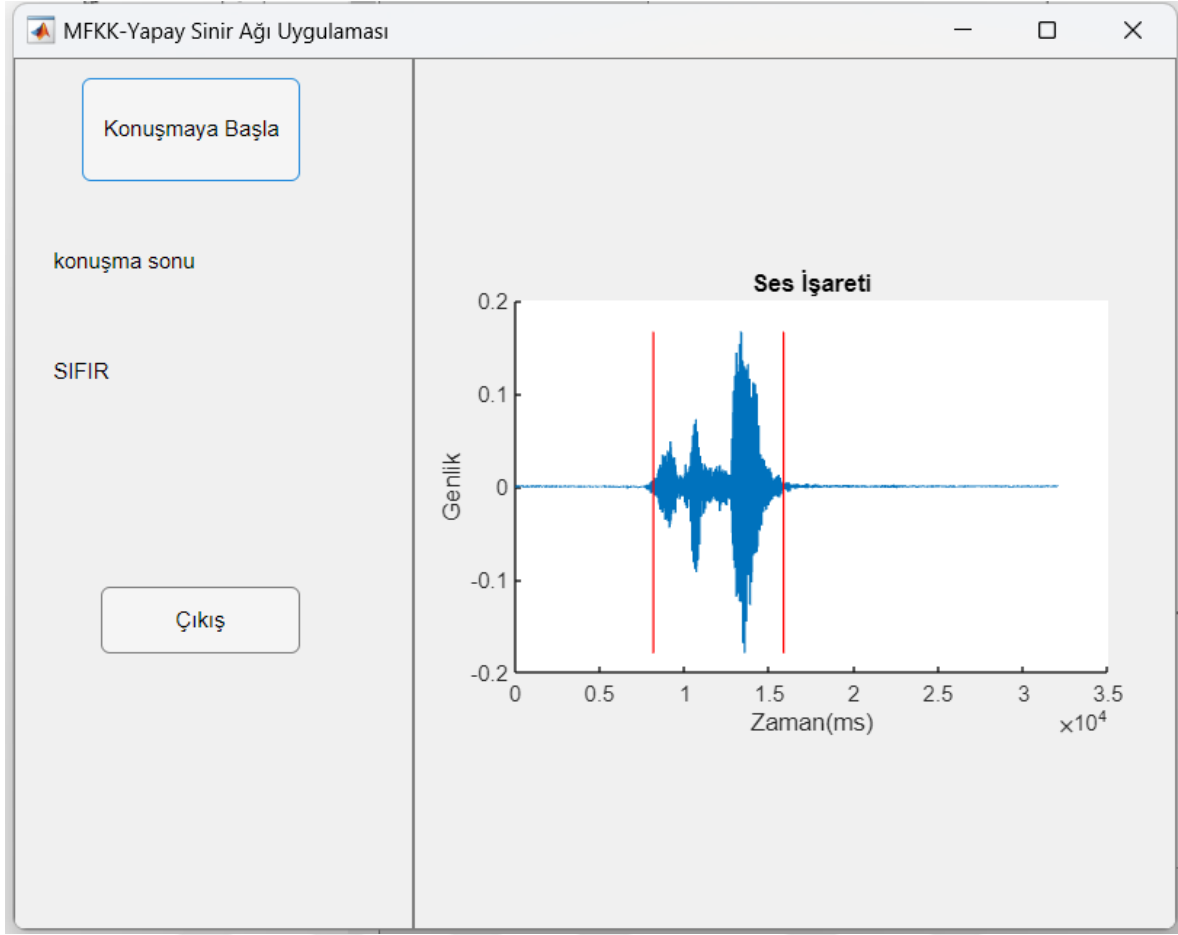
Şekil 6.3: MFKK –Yapay Sinir Ağı Uygulaması

Şekil 6.4'te OKT sisteminin eğitim aşamasında kullanılan MATLAB arayüzü görülmektedir.



Şekil 6.4: MFKK –Yapay Sinir Ağı Uygulamasının Eğitim Aşaması

Şekil 6.5’te MFKK –Yapay sinir ağı uygulamasının MATLAB kullanıcı arayüzü görülmektedir. “Konuşmaya Başla” yazılı düğmeye tıklanır ve 2 saniye içinde bir rakam söylenir. OKT sistemi ses işaretinin grafiğini işaretin başlangıç ve bitiş noktalarını da tespit ederek çizer ve söylenen rakamı tanıyarak ekrana yazar.



Şekil 6.5: MFKK –Yapay Sinir Ağı Uygulamasının MATLAB Kullanıcı Arayüzü

6.1.1 MFKK –Yapay sinir ağı ile Türkçe rakam tanıma uygulamasının performansı

Bu sistemin KHO ve KTO parametrelerini bulmak için kelime hzinesindeki her kelime 10 defa okunmuştur ve OKT sisteminin kelimeleri doğru ve yanlış tanıma miktarları aşağıdaki tabloya işlenmiştir.

Çizelge 6.2: MFKK-YSA OKT sisteminin kelimeleri doğru ve yanlış tanıma miktarları

S.N.	Kelime	Doğru	Yanlış	Toplam
1	Sıfır	10	0	10
2	Bir	10	0	10
3	İki	3	7	10
4	Üç	9	1	10
5	Dört	9	1	10
6	Beş	10	0	10
7	Altı	10	0	10

8	Yedi	10	0	10
9	Sekiz	10	0	10
10	Dokuz	10	0	10
Toplam		91	9	100

$$KHO = \frac{\text{Toplam Yanlış}}{\text{Toplam}} * 100 = \frac{9}{100} * 100 = \%9$$

$$KTO = 1 - KHO = 1 - 0,09 = 0,91 \text{ ise } KTO = \% 91$$

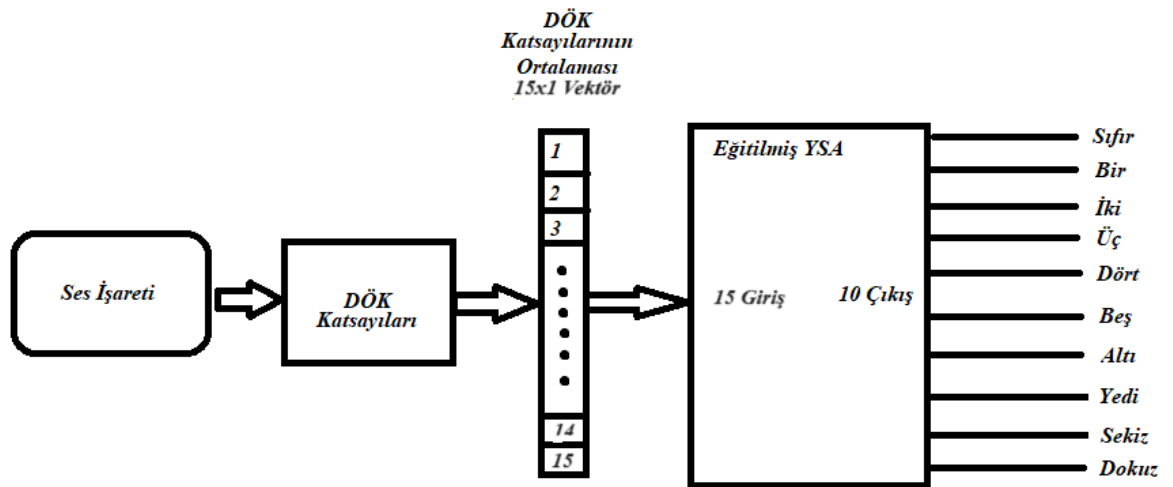
6.2 DÖK –Yapay Sinir Ağı ile Türkçe Rakam Tanıma Uygulaması

Konuşma tanıma sisteminin öznitelik çıkarma kısmında DÖK katsayıları kullanılır. Ses işaretinin her çerçevesinin DÖK katsayısı bulunduğundan sonra, DÖK katsayılarını gösteren matriste her satırın ortalaması alınır, sonuç olarak 15x1 boyutunda bir vektör ortaya çıkar. 15 elemanlı bu vektör YSA'nın girişine uygulanır. YSA'nın çıkışında ise 10 eleman vardır. YSA 20 tane gizli katman elemanı barındırır.

Uygulama iki aşamadan oluşur:

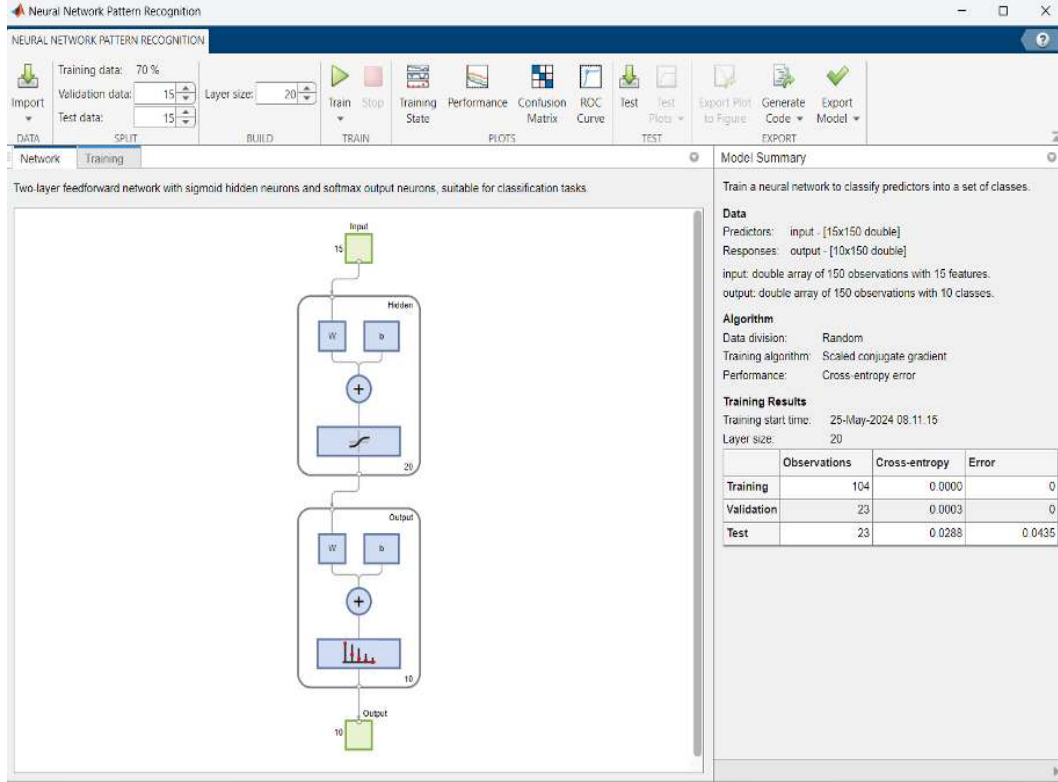
1. Aşama: YSA'nın eğitilmesi aşaması
2. Aşama: YSA'nın test edilmesi. Ses işaretinin tanıma için YSA'ya uygulanması

Şekil 6.6'da Öznitelik çıkarım aşamasında DÖK katsayıları kullanılan, sınıflandırma aşamasında da YSA'nın kullanıldığı konuşma tanıma sisteminin blok şeması gösterilmiştir.



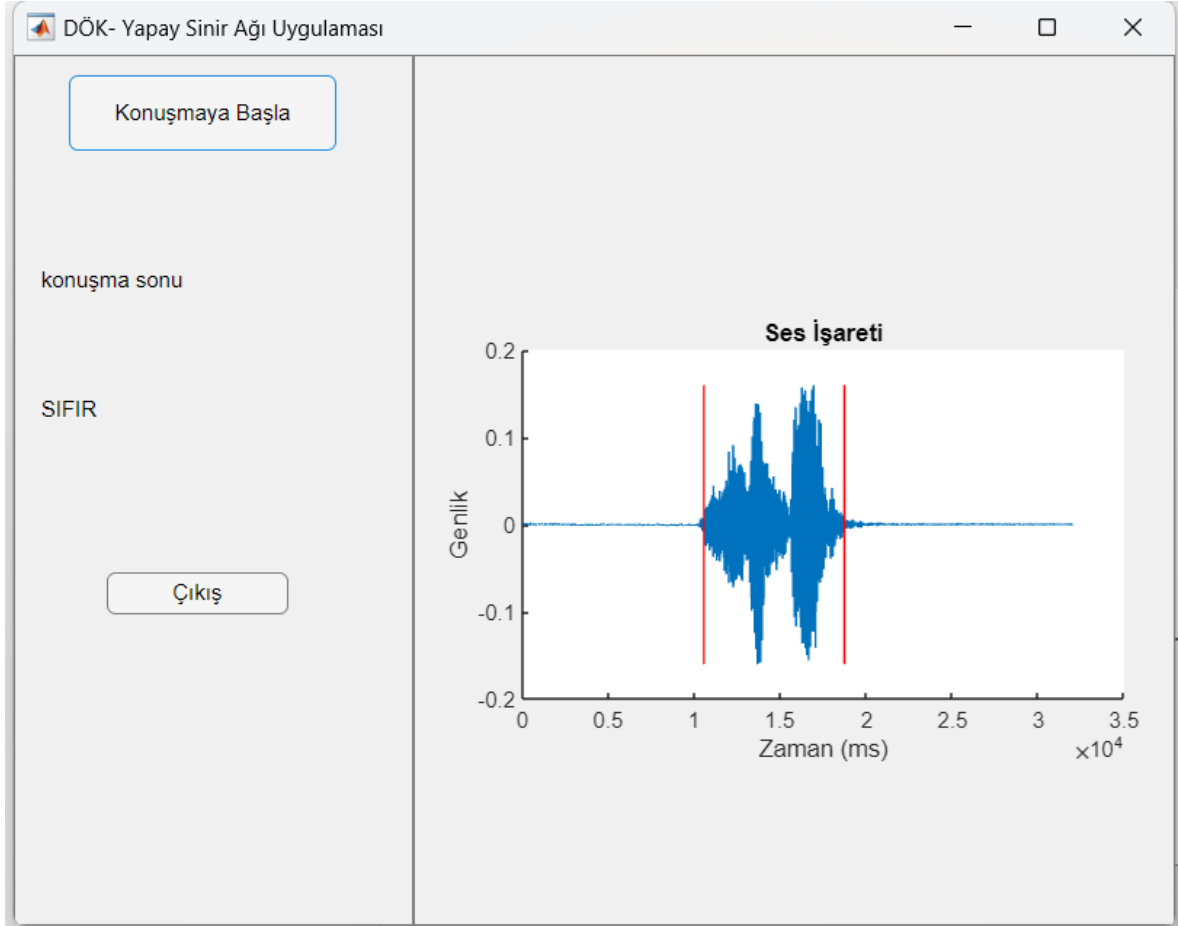
Şekil 6.6: DÖK –Yapay Sinir Ağı Uygulaması

Şekil 6.7’de OKT sisteminin eğitim aşamasında kullanılan MATLAB arayüzü görülmektedir.



Şekil 6.7: DÖK –Yapay Sinir Ağı Uygulamasının Eğitim Aşaması

Şekil 6.8’de DÖK –Yapay sinir ağı uygulamasının MATLAB kullanıcı arayüzü görülmektedir. “Konuşmaya Başla” yazılı düğmeye tıklanır ve 2 saniye içinde bir rakam söylenir. OKT sistemi ses işaretinin grafiğini işaretin başlangıç ve bitiş noktalarını da tespit ederek çizer ve söylenen rakamı tanıyarak ekrana yazar.



Şekil 6.8: DÖK –Yapay Sinir Ağı Uygulamasının MATLAB Kullanıcı Arayüzü

6.2.1 DÖK –Yapay sinir ağı ile Türkçe rakam tanıma uygulamasının performansı

Bu sistemin KHO ve KTO parametrelerini bulmak için kelime hazinesindeki her kelime 10 defa okunmuştur ve OKT sisteminin kelimeleri doğru ve yanlış tanıma miktarları aşağıdaki tabloya işlenmiştir.

Çizelge 6.3: DÖK-YSA OKT sisteminin kelimeleri doğru ve yanlış tanıma miktarları

S.N.	Kelime	Doğru	Yanlış	Toplam
1	Sıfır	10	0	10
2	Bir	9	1	10
3	İki	6	4	10
4	Üç	1	9	10
5	Dört	9	1	10
6	Beş	10	0	10
7	Altı	10	0	10

8	Yedi	9	1	10
9	Sekiz	10	0	10
10	Dokuz	10	0	10
Toplam		84	16	100

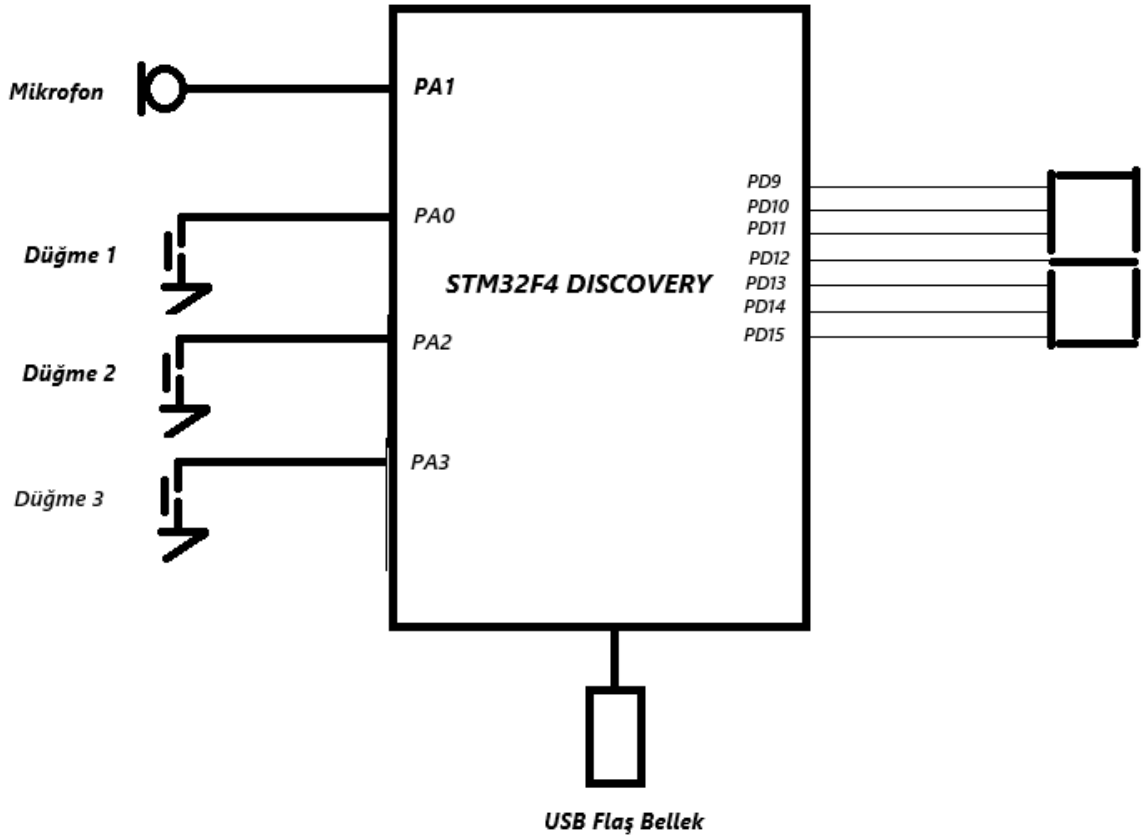
$$KHO = \frac{Toplam\ Yanliş}{Toplam} * 100 = \frac{16}{100} * 100 = \%16$$

$$KTO = 1 - KHO = 1 - 0,16 = 0,84 \text{ ise } KTO = \% 84$$



7. MİKRODENETLEYİCİ UYGULAMASI

STM32F4 DISCOVERY Board kullanarak öznitelik çıkarmada MFKK yöntemini kullanan ve sınıflandırma yöntemi olarak da DZB tekniğini kullanan bir sistem geliştirilmeye çalışıldı. Sistem 10 rakamı tanıyacaktır. Sistemin donanım kısmı aşağıdaki şekilde gösterilmiştir.



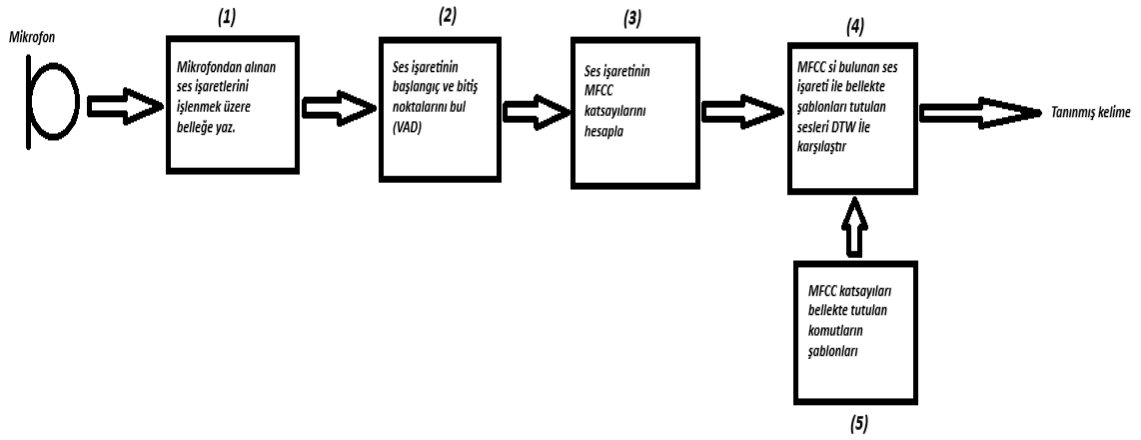
Şekil 7.1: Mikrodenetleyici Uygulamasının Donanım Mimarisi

Sistem şu şekilde çalışacaktır:

- Düğme 1'e basıldıktan sonra 2 saniye içinde 0'dan 9'a kadar bir rakam mikrofona söylenecektir. Söylenen rakam kartta yüklü yazılımla tespit edilip sonuç 7 segment ekranda gösterilecektir.

- Düğme 2 rakam şablonlarının söylenmesi ve bu şablonların USB Flaş Belleğe kaydedilmesi için kullanılacaktır. Düğme 2'ye tıklandığında 7 segment ekranda gösterilen rakam söylenecek bu rakamın MFKK katsayıları hesaplanıp şablon olarak USB Flaş Belleğe kaydedilecek. Bütün 10 rakam için bu işlem gerçekleştirilip her rakam için MFKK sayılarından oluşan şablonlar USB Flaş belleğe kaydedilecektir.
- Düğme 3 çeşitli amaçlar için yedek olarak yerleştirilmiştir.

Sistem donanım olarak gerçekleştirildikten sonra sisteme yüklenecek yazılım tasarımına geçilmiştir. Aşağıdaki şekilde OKT sisteminin yazılım mimarisi yani modülleri gösterilmiştir.



Şekil 7.2: Mikrodenetleyici Uygulamasının Yazılım Mimarisi

Şekil 7.2’den görüleceği üzere OKT sisteminin yazılımı 5 modülden oluşmakta.

- 1. Modülde mikrofondan alınan ses ana belleğe(RAM) yazılacak. Sesin örnekleme frekansı 8 KHz olarak seçilmiştir. Düğme 1’e basıldığında 2 saniyelik yani 16000 örneklik ses işareti RAM’e kaydedilir. Bu ses işareti bellekte $2 \times 16000 = 32000$ bayt yer kaplar.
- 2. Modül Voice Activity Detection işlemi yapacaktır. Kaydedilen 2 saniyelik sesin başlangıç ve bitiş noktaları bulunup sesin bilgi içeren kısmı ayrılacaktır. Başlangıç ve bitiş noktaları arasındaki ses işaretinin bir sonraki modülde MFKK katsayıları hesaplanacaktır.
- 3. Modülde 2. Modülden gelen ses işaretinin MFKK katsayıları hesaplanacaktır. Bu kısım OKT sistemi yazılımının kalbi diyebileceğimiz kısımdır. MFKK katsayılarını

hesaplamak için CMSIS DSP kütüphanesinde yer alan MFCC fonksiyonu kullanılabilir.

- 5. Modülde Düğme 2'ye tıklandığında mikrofona söylenen 10 rakamın şablon olarak MFKK katsayıları bulunup bunlar karşılaştırma için kullanılmak üzere USB Flaş belleğe kaydedilecektir.
- 4. Modül Dinamik Zaman Bükmesi Modülüdür. 3. Modülde MFKK katsayıları bulunan ses işareti ile USB Flaş bellekte tutulan ses şablonlarını karşılaştırıp en yakın şablon seçilecek ve kelime tanınmış olacaktır. CMSIS DSP kütüphanesinde yer alan DTW fonksiyonu kullanılabilir.



8. SONUÇ

Bölüm 5 ve Bölüm 6’da yaptığımız uygulamaların performanslarını gösteren çizelgelerden ve hesaplamalardan elde ettiğimiz bazı sonuçlar aşağıdadır:

DZB tekniği YSA’ya oranla daha fazla kelime hazinesi imkanı sunmaktadır.

MFKK tekniği ile yapılan OKT sistemlerinin KTO, DÖK tekniği ile yapılan OKT sistemlerinden fazladır.

MFKK ile yapılan OKT sistemleri %90 civarı KTO ile ihtiyaçlara cevap verebilecek düzeyde çalışmaktadır.

OKT sistemleri bazı kelimeleri tanımada %50’nin altında performans göstermiştir. MFKK ve DZB tekniklerinde DUR ve GERİ kelimelerinin KTO azdır. DÖK ve DZB tekniklerinde ÜÇ, DÖRT, ON, YİRMİ, ALTMİŞ, DOKSAN, MİLYON, DUR, SOL kelimelerinin KTO azdır. MFKK ve YSA tekniklerinde İKİ kelimesinin KTO azdır. DÖK ve YSA tekniklerinde ÜÇ kelimesinin KTO azdır.

Öznitelik çıkarma kısmında MFKK ve DÖK yöntemlerinden elde edilen sayıları birleştirerek sınıflandırma kısmına uygularsak daha iyi sonuçlar alma ihtimalimiz yüksektir. Çünkü bazı kelimeleri MFKK iyi tanıyamazken diğer bazı kelimeleri de DÖK yöntemi iyi tanıyamıyor. MFKK-YSA uygulamasında, ses işaretinin her çerçevesinin MFKK’nın ortalaması alınarak 14 MFKK, YSA girişine uygulanmıştır. Oysa bir kelimenin MFKK örneğin 28x14 boyutlu bir matristir. 28x14 boyutlu matrisi 1x14 boyutlu bir matrise indirgiyoruz. 28x14 boyutlu matrisi 1x14 boyutlu bir matris temsil ediyor. 28*14 boyutlu matrisi 3x14 veya 4x14 boyutlu matrislere indirebilseydik daha az veri kaybı olacaktı ve daha performanslı OKT sistemleri geliştirmiş olabilirdik. MATLAB’da bir kümeleme fonksiyonu olan kmeans fonksiyonu büyük boyutlu matrisleri, düşük boyutlu matrislere indirgemek için kullanılabilir. DÖK-YSA ile Türkçe rakam tanıma uygulaması için de bu paragrafta anlatılanlar geçerlidir. İleriki çalışmalarda bu paragrafta anlatılanlar uygulanacaktır.

7. Bölümde anlatılan Mikrodenetleyici uygulamasının donanım kısmı tam uygulanmasına rağmen yazılım kısmının sadece ilk iki modülü gerçekleştirilmiştir. Bu sebeple bu uygulama için KTO tespit edilememiştir. Sonraki çalışmalarda mikrodenetleyici uygulaması tamamlanacaktır.

İleriki çalışmalarda öznitelik çıkarma yöntemi olarak MFKK ve DÖK kullanırken sınıflandırma olarak saklı markov modelleri ve derin sinir ağları kullanan sistemler üzerinde çalışmalar hedeflenmektedir.



KAYNAKLAR

- Aggarwal, R. K. & Dave, M.** (2012). Integration of multiple acoustic and language models for improved Hindi speech recognition system. *International Journal of Speech Technology*, 15(2), 165–180. doi:10.1007/s10772-012-9131-y
- Ahad, A., Fayyaz, A., & Mehmood, T.** (2002). Speech recognition using multilayer perceptron. *IEEE Students Conference, ISCON '02. Proceedings*.
- Anusuya, M. A. & Katti, S. K.** (2009). Speech Recognition by Machine: A Review. (IJCSIS) *International Journal of Computer Science and Information Security*, Vol. 6, No. 3
- Başaran, S.** (2007). Yapay Sinir Ağları Kullanarak Konuşmacı Tanıma (Yüksek Lisans Tezi). Uludağ Üniversitesi, Fen Bilimleri Enstitüsü, Bursa.
- Chavan, R. S. & Sable, G. S.** (2013). An Overview of Speech Recognition Using HMM. *International Journal of Computer Science and Mobile Computing* , Vol. 2, Issue. 6, June 2013, pg.233 – 238
- Çakır, M. Y.** (2017). Gerçek Zamanlı Yüksek Kalitede Ses Tanıma (Yüksek Lisans Tezi). İstanbul Sabahattin Zaim Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.
- Das, B. P. & Parekh, R.** (2012). Recognition of Isoleted Words using Features based on LPC, MFCC, ZCR and STE, with Neural Network Classifiers. *International Journal of Modern Engineering Research (IJMER)* Vol.2, Issue.3, May-June 2012 pp-854-858
- Das, S.** (2012). Speech Recognition Technique: A Review. *International Journal of Engineering Research and Applications*. Vol. 2, Issue 3, May-Jun 2012, pp.2071-2087
- Dede, G.** (2008). Yapay Sinir Ağları ile Konuşma Tanıma (Yüksek Lisans Tezi). Ankara Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.
- Dongare, A.D., KHarde, R.R. & Kachare, A.D.** (2012). Introduction to Artificial Neural Network. *International Journal of Engineering and Innovative Technology (IJEIT)* Volume 2, Issue 1, July 2012
- Dynamic Time Warping(DTW) Algorithm in Time Series.** Erişim:06 Haziran 2024, <https://www.theaidream.com/post/dynamic-time-warping-dtw-algorithm-in-time-series>
- Elberri, M. A.** (2020). Tekrarlayan Sinir Ağlarını Kullanarak Konuşma Tanıma (Yüksek Lisans Tezi). Kastamonu Üniversitesi, Fen Bilimleri Enstitüsü, Kastamonu.
- Eray, O.** (2008). Destek Vektör Makineleri İle Ses Tanıma Uygulaması (Yüksek Lisans Tezi). Pamukkale Üniversitesi, Fen Bilimleri Enstitüsü, Denizli.
- Eskidere, Ö.** (2007). İstatiksel Modelleme ile Konuşmacı Tanıma(Doktora Tezi). Uludağ Üniversitesi, Fen Bilimleri Enstitüsü, Bursa.

- Gamit, M. R. & Dhameliya, K.** (2015). Isolated Words Recognition Using MFCC, LPC and Neural Network. *IJRET: International Journal of Research in Engineering and Technology*. Volume: 04 Issue: 06 | June-2015
- Jain, A. K., Jianchang Mao, & Mohiuddin, K. M.** (1996). Artificial neural networks: a tutorial. *Computer*, 29(3), 31–44. doi:10.1109/2.485891
- James, P. E., Kit, M. H., Vaithilingam, C. A., & Chiat, A. T. W.** (2020). Recurrent neural network-based speech recognition using MATLAB. *International Journal of Intelligent Enterprise*, Vol. 7, Nos. 1/2/3, 2020
- Karpagavalli, S. & Chandra, E.** (2016). A Review on Automatic Speech Recognition Architecture and Approaches. *International Journal of Signal Processing, Image Processing and Pattern Recognition*. Vol.9, No.4, (2016), pp.393-404
- Malik, M., Malik, M. K., Mehmood, K. & Makhdoom, I.** (2020). Automatic speech recognition: a survey. *Multimedia Tools and Applications*. doi:10.1007/s11042-020-10073-7
- Oyucu, S., Sever, H. & Polat, H.** (2019). Otomatik Konuşma Tanımaya Genel Bakış, Yaklaşımlar ve Zorluklar: Türkçe Konuşma Tanımanın Gelecekteki Yolu. *Gazi Üniversitesi Fen Bilimleri Dergisi. Part C*, 7(4):834-854
- Öcal, K.** (2005). Otomatik Konuşma Tanıma Algoritmalarının Uygulamaları (Yüksek Lisans Tezi). Ankara Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.
- Özişbakan, T.** (2011). Konuşmacıya Bağımlı Ayrık Sözcük Tanıma (Yüksek Lisans Tezi). Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.
- Öztemel, E.** (2006). Yapay Sinir Ağları. İstanbul : Papatya Yayıncılık
- Pour, M. M., & Farokhi, F.** (2009). A New Approach for Persian Speech Recognition. *2009 IEEE International Advance Computing Conference*. doi:10.1109 / iadcc.2009.4808998
- Saksamudre, Suman K. , Shrishrimal, P. P., & Deshmukh, R. R.** (2015). A Review on Different Approaches for Speech Recognition System. *International Journal of Computer Applications (0975 – 8887) Volume 115 – No. 22*.
- Şaşmaz, M. B.** (2011). Yapay Zeka Tabanlı Konuşma Tanıma Sistemi (Yüksek Lisans Tezi). Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elazığ.
- Tanweer, S., Mobin, A., & Alam, A.** (2014). Analysis of Combined Use of NN and MFCC for Speech Recognition. *World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering* Vol:8, No:9, 2014
- Tunalı, V.** (2005). Türkçe İçin Konuşmacı Bağımlı, Geniş Sözcük Dağarcıklı, Ayrık Sözcüklü Konuşma Tanıma Sistemi (Yüksek Lisans Tezi). Marmara Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.
- Urgun, D.** (2012). Platformdan Bağımsız Bir Otomatik Konuşma Tanıma Sisteminin Tasarlanması ve Uygulanması (Yüksek Lisans Tezi). Atılım Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.
- Vimala, C. & Radha, V.** (2012). A Review on Speech Recognition Challenges and Approaches. *World of Computer Science and Information Technology Journal (WCSIT)*, Vol. 2, No. 1, 1-7, 2012

Yakar, Ö. (2016). Sözcük ve Hece Tabanlı Konuşma Tanıma Sistemlerinin Karşılaştırılması (Yüksek Lisans Tezi). Adnan Menderes Üniversitesi, Fen Bilimleri Enstitüsü, Aydın.



EKLER

EK A: MFKK –DZB Teknikleriyle Konuşma Tanıma Uygulaması Kodları

Şekil 5.2’de yer alan MFKK ve DZB teknikleriyle konuşma tanıma programı kullanıcı arayüzünde yer alan Şablonları Ayarla düğmesine bağlı kodlar aşağıdadır:

```
run("sablon_hazirlama.m");
```

```
app.Label.Text = "şablonlar hazırlandı";
```

Bu komut “sablon_hazirlama.m” MATLAB dosyasını çalıştırmaktadır. “sablon_hazirlama.m” dosyasının kodlarının bir kısmı aşağıdadır.

```
% 1- 2 saniyelik şablon sesi oku ve değişkene ata
```

```
% 2- sesin başlangıç ve bitiş noktalarını belirle(detectSpeech fonksiyonu)
```

```
% 3- sesi başlangıç ve bitiş noktalarından itibaren yeni bir değişkene ata
```

```
% böylece 2 saniyelik sesteki sessizlik kısımları atılır
```

```
% 4- Sessizlik kısımları atılan ses işaretinin MFKK katsayılarını bul
```

```
% MATLAB'daki mfcc fonksiyonu ile MFKK katsayıları hesaplanır
```

```
% 5- MFKK katsayılarının transpozunu al
```

```
% 6- 33 adet ses şablonu için yukarıdaki 1-5 işlemlerini gerçekleştir
```

```
sifir1= audioread("../sesler/sifir1.wav");
```

```
sifir1_ds =detectSpeech(sifir1,16000);
```

```
sifir1_sig = sifir1(sifir1_ds(1):sifir1_ds(2));
```

```
sifir1_mfcc =mfcc(sifir1_sig,16000);
```

```
sifir1_mfcc_t = sifir1_mfcc.';
```

```
bir1= audioread("../sesler/bir1.wav");
```

```
bir1_ds =detectSpeech(bir1,16000);
```

```
bir1_sig = bir1(bir1_ds(1):bir1_ds(2));
```

```

bir1_mfcc =mfcc(bir1_sig,16000);

bir1_mfcc_t = bir1_mfcc.';

iki1= audioread("../sesler/iki1.wav");

iki1_ds =detectSpeech(iki1,16000);

iki1_sig = iki1(iki1_ds(1):iki1_ds(2));

iki1_mfcc =mfcc(iki1_sig,16000);

iki1_mfcc_t = iki1_mfcc.';

uc1= audioread("../sesler/uc1.wav");

uc1_ds =detectSpeech(uc1,16000);

uc1_sig = uc1(uc1_ds(1):uc1_ds(2));

uc1_mfcc =mfcc(uc1_sig,16000);

uc1_mfcc_t = uc1_mfcc.';

% diğ er ses ş ablonları için de aynı iş lemleri gerç ekleřtir

save("sablonlar.mat");

```

ř ekil 5.2’de yer alan MFKK ve DZB teknikleriyle konuř ma tanı ma programı kullanıcı arayüzünde yer alan Konuřmaya Bař la düğ mesine tıkladığ ımızda ise ař ağıdaki kodlar ç alıř acaktır:

```

function KonumayaBalaButtonPushed(app, event)

KonusmaKayit = audiorecorder(16000,16,1);

app.Label.Text = "konuřmaya bař la";

recorblocking(KonusmaKayit,2);

app.Label.Text ="konuřma sonu";

y = getaudiodata(KonusmaKayit);

plot(app.UIAxes,y);

encok = max(y);

enaz = min(y);

s1= detectSpeech(y,16000);

```

```

line(app.UIAxes,[s1(1) s1(1)],[encok enaz],"color","red");
line(app.UIAxes,[s1(2) s1(2)],[encok enaz],"color","red");

y1 = y(s1(1):s1(2));

y1_mfcc = mfcc(y1,16000);

y1_mfcc_t = y1_mfcc.';

myvars
={ 'sifir1_mfcc_t','bir1_mfcc_t','iki1_mfcc_t','uc1_mfcc_t','dort1_mfcc_t','bes1_mfcc_t','alti
1_mfcc_t','yedi1_mfcc_t','sekiz1_mfcc_t','dokuz1_mfcc_t','on1_mfcc_t','yirmi1_mfcc_t','ot
uz1_mfcc_t','kirk1_mfcc_t','elli1_mfcc_t','altmis1_mfcc_t','yetmis1_mfcc_t','seksen1_mfcc
_t','doksan1_mfcc_t','yuz1_mfcc_t','bin1_mfcc_t','milyon1_mfcc_t','milyar1_mfcc_t','evet1
_mfcc_t','hayir1_mfcc_t','basla1_mfcc_t','dur1_mfcc_t','ileri1_mfcc_t','geri1_mfcc_t','sag1
_mfcc_t','soll1_mfcc_t','ac1_mfcc_t','kapa1_mfcc_t'};

S= load("sablonlar.mat",myvars{:});

sifir1_dtw = dtw(y1_mfcc_t,S.sifir1_mfcc_t);

bir1_dtw = dtw(y1_mfcc_t,S.bir1_mfcc_t);

iki1_dtw = dtw(y1_mfcc_t,S.iki1_mfcc_t);

uc1_dtw = dtw(y1_mfcc_t,S.uc1_mfcc_t);

dort1_dtw = dtw(y1_mfcc_t,S.dort1_mfcc_t);

bes1_dtw = dtw(y1_mfcc_t,S.bes1_mfcc_t);

alti1_dtw = dtw(y1_mfcc_t,S.alti1_mfcc_t);

yedi1_dtw = dtw(y1_mfcc_t,S.yedi1_mfcc_t);

sekiz1_dtw = dtw(y1_mfcc_t,S.sekiz1_mfcc_t);

dokuz1_dtw = dtw(y1_mfcc_t,S.dokuz1_mfcc_t);

on1_dtw = dtw(y1_mfcc_t,S.on1_mfcc_t);

yirmi1_dtw = dtw(y1_mfcc_t,S.yirmi1_mfcc_t);

otuz1_dtw = dtw(y1_mfcc_t,S.otuz1_mfcc_t);

kirk1_dtw = dtw(y1_mfcc_t,S.kirk1_mfcc_t);

elli1_dtw = dtw(y1_mfcc_t,S.elli1_mfcc_t);

```

```

altmis1_dtw = dtw(y1_mfcc_t,S.altmis1_mfcc_t);
yetmis1_dtw = dtw(y1_mfcc_t,S.yetmis1_mfcc_t);
seksen1_dtw = dtw(y1_mfcc_t,S.seksen1_mfcc_t);
doksan1_dtw = dtw(y1_mfcc_t,S.doksan1_mfcc_t);
yuz1_dtw = dtw(y1_mfcc_t,S.yuz1_mfcc_t);
bin1_dtw = dtw(y1_mfcc_t,S.bin1_mfcc_t);
milyon1_dtw = dtw(y1_mfcc_t,S.milyon1_mfcc_t);
milyar1_dtw = dtw(y1_mfcc_t,S.milyar1_mfcc_t);
evet1_dtw = dtw(y1_mfcc_t,S.evet1_mfcc_t);
hayir1_dtw = dtw(y1_mfcc_t,S.hayir1_mfcc_t);
basla1_dtw = dtw(y1_mfcc_t,S.basla1_mfcc_t);
dur1_dtw = dtw(y1_mfcc_t,S.dur1_mfcc_t);
ileri1_dtw = dtw(y1_mfcc_t,S.ileri1_mfcc_t);
geri1_dtw = dtw(y1_mfcc_t,S.geri1_mfcc_t);
sag1_dtw = dtw(y1_mfcc_t,S.sag1_mfcc_t);
sol1_dtw = dtw(y1_mfcc_t,S.sol1_mfcc_t);
ac1_dtw = dtw(y1_mfcc_t,S.ac1_mfcc_t);
kapa1_dtw = dtw(y1_mfcc_t,S.kapa1_mfcc_t);

A = [sifir1_dtw bir1_dtw iki1_dtw uc1_dtw dort1_dtw bes1_dtw alti1_dtw yedi1_dtw
sekiz1_dtw dokuz1_dtw on1_dtw yirmi1_dtw otuz1_dtw kirk1_dtw elli1_dtw altmis1_dtw
yetmis1_dtw seksen1_dtw doksan1_dtw yuz1_dtw bin1_dtw milyon1_dtw milyar1_dtw
evet1_dtw hayir1_dtw basla1_dtw dur1_dtw ileri1_dtw geri1_dtw sag1_dtw sol1_dtw
ac1_dtw kapa1_dtw];

minA = min(A);

switch minA

    case sifir1_dtw

        app.Label2.Text ="SIFIR";

```

```
case bir1_dtw
    app.Label2.Text ="BİR";
case iki1_dtw
    app.Label2.Text ="İKİ";
case uc1_dtw
    app.Label2.Text ="ÜÇ";
case dort1_dtw
    app.Label2.Text ="DÖRT";
case bes1_dtw
    app.Label2.Text ="BEŞ";
case alti1_dtw
    app.Label2.Text ="ALTI";
case yedi1_dtw
    app.Label2.Text ="YEDİ";
case sekiz1_dtw
    app.Label2.Text ="SEKİZ";
case dokuz1_dtw
    app.Label2.Text ="DOKUZ";
case on1_dtw
    app.Label2.Text ="ON";
case yirmi1_dtw
    app.Label2.Text ="YİRMİ";
case otuz1_dtw
    app.Label2.Text ="OTUZ";
case kirk1_dtw
    app.Label2.Text ="KIRK";
```

```
case elli1_dtw
    app.Label2.Text ="ELLİ";
case altmis1_dtw
    app.Label2.Text ="ALTMİŞ";
case yetmis1_dtw
    app.Label2.Text ="YETMİŞ";
case seksen1_dtw
    app.Label2.Text ="SEKSEN";
case doksan1_dtw
    app.Label2.Text ="DOKSAN";
case yuz1_dtw
    app.Label2.Text ="YÜZ";
case bin1_dtw
    app.Label2.Text ="BİN";
case milyon1_dtw
    app.Label2.Text ="MİLYON";
case milyar1_dtw
    app.Label2.Text ="MİLYAR";
case evet1_dtw
    app.Label2.Text ="EVET";
case hayir1_dtw
    app.Label2.Text ="HAYIR";
case basla1_dtw
    app.Label2.Text ="BAŞLA";
case dur1_dtw
    app.Label2.Text ="DUR";
```

```
case ileri1_dtw
    app.Label2.Text ="İLERİ";
case geri1_dtw
    app.Label2.Text ="GERİ";
case sag1_dtw
    app.Label2.Text ="SAĞ";
case sol1_dtw
    app.Label2.Text ="SOL";
case ac1_dtw
    app.Label2.Text ="AÇ";
case kapa1_dtw
    app.Label2.Text ="KAPA";
end
end
```

EK B: DÖK –DZB Teknikleriyle Konuşma Tanıma Uygulaması Kodları

Şekil 5.3’de yer alan DÖK ve DZB teknikleriyle konuşma tanıma programı kullanıcı arayüzünde yer alan Şablonları Ayarla düğmesine bağlı kodlar aşağıdadır:

```
run("sablon_hazirlama.m");
```

```
app.Label3.Text = "şablonlar hazırlandı";
```

Bu komut “sablon_hazirlama.m” MATLAB dosyasını çalıştırmaktadır. “sablon_hazirlama.m” dosyasının kodlarının bir kısmı aşağıdadır.

```
clear;
```

```
k=14; % lpc(sinyal,k)
```

```
sifir1= audioread("../sesler/sifir1.wav");
```

```
sifir1_ds =detectSpeech(sifir1,16000);
```

```
sifir1_sig = sifir1(sifir1_ds(1):sifir1_ds(2));
```

```
sifir1_lpc =mylpc(sifir1_sig,k);
```

```
bir1= audioread("../sesler/bir1.wav");
```

```
bir1_ds =detectSpeech(bir1,16000);
```

```
bir1_sig = bir1(bir1_ds(1):bir1_ds(2));
```

```
bir1_lpc =mylpc(bir1_sig,k);
```

```
iki1= audioread("../sesler/iki1.wav");
```

```
iki1_ds =detectSpeech(iki1,16000);
```

```
iki1_sig = iki1(iki1_ds(1):iki1_ds(2));
```

```
iki1_lpc =mylpc(iki1_sig,k);
```

```
uc1= audioread("../sesler/uc1.wav");
```

```
uc1_ds =detectSpeech(uc1,16000);
```

```
uc1_sig = uc1(uc1_ds(1):uc1_ds(2));
```

```
uc1_lpc =mylpc(uc1_sig,k);
```

```
dort1= audioread("../sesler/dort1.wav");
```

```
dort1_ds =detectSpeech(dort1,16000);
```

```
dort1_sig = dort1(dort1_ds(1):dort1_ds(2));
```

```
dort1_lpc =mylpc(dort1_sig,k);
```

```
bes1= audioread("../sesler/bes1.wav");
```

```
bes1_ds =detectSpeech(bes1,16000);
```

```
bes1_sig = bes1(bes1_ds(1):bes1_ds(2));
```

```
bes1_lpc =mylpc(bes1_sig,k);
```

```
•
```

```
•
```

```
•
```

```
save("sablonlar.mat");
```

“sablon_hazirlama.m” MATLAB dosyasınıda kullanılan mylpc fonksiyonunun kodları aşağıdadır. Bu fonksiyon ses işaretinin önvurguluyor, çerçevelere ayırıyor, pencereleme fonksiyonu uyguluyor ve her çerçevenin 14. dereceden lpc katsayılarını MATLAB’ın lpc fonksiyonunu kullanarak hesaplıyor.

```
function LPC = mylpc(signal,k)
```

```
a=0.95;
```

```
len=length(signal);
```

```
s1= zeros(len);
```

```
for i=2:len
```

```
    s1(i)=signal(i)-a*signal(i-1);
```

```
end
```

```
n= 256;
```

```
m = 100;
```

```
frame=floor((len-n)/m)+1;
```

```
z= zeros(n,frame);
```

```
for j=1:frame
```

```
    for i=1:n
```

```

        z(i,j)=s1((j-1)*m+i);
    end
end
h=hamming(n);
z2= zeros(n,frame);
for j=1:frame
    for i=1:n
        z2(i,j)=h(i)*z(i,j);
    end
end
z3 = zeros(k+1,frame);
for f=1:frame
    z3(:,f)=lpc(z2(:,f),k);
end
LPC = z3;
end

```

Şekil 5.3’de yer alan DÖK ve DZB teknikleriyle konuşma tanıma programı kullanıcı arayüzünde yer alan Konuşmaya Başla düğmesine tıkladığımızda ise aşağıdaki kodlar çalışacaktır:

```

function KonuşmayaBaşlaButtonPushed(app, event)
    KonuşmaKayıt = audiorecorder(16000,16,1);
    app.Label3.Text = "konuşmaya başla";
    recordblocking(KonuşmaKayıt,2);
    app.Label3.Text ="konuşma sonu";
    y = getaudiodata(KonuşmaKayıt);
    plot(app.UIAxes,y);
    encok = max(y);

```

```

enaz = min(y);

s1= detectSpeech(y,16000);

line(app.UIAxes,[s1(1) s1(1)],[encok enaz],"color","red");

line(app.UIAxes,[s1(2) s1(2)],[encok enaz],"color","red");

y1 = y(s1(1):s1(2));

k=14; % lpc(sinyal,k)

y1_lpc = mylpc(y1,k);

myvars
={ 'sifir1_lpc','bir1_lpc','iki1_lpc','uc1_lpc','dort1_lpc','bes1_lpc','alti1_lpc','yedi1_lpc','sekiz
1_lpc','dokuz1_lpc','on1_lpc','yirmi1_lpc','otuz1_lpc','kirk1_lpc','elli1_lpc','altmis1_lpc','yet
mis1_lpc','seksen1_lpc','doksan1_lpc','yuz1_lpc','bin1_lpc','milyon1_lpc','milyar1_lpc','evet
1_lpc','hayir1_lpc','basla1_lpc','dur1_lpc','ileri1_lpc','geri1_lpc','sag1_lpc','sol1_lpc','ac1_lp
c','kapa1_lpc'};

S= load("sablonlar.mat",myvars{:});

sifir1_dtw = dtw(y1_lpc,S.sifir1_lpc);

bir1_dtw = dtw(y1_lpc,S.bir1_lpc);

iki1_dtw = dtw(y1_lpc,S.iki1_lpc);

uc1_dtw = dtw(y1_lpc,S.uc1_lpc);

dort1_dtw = dtw(y1_lpc,S.dort1_lpc);

bes1_dtw = dtw(y1_lpc,S.bes1_lpc);

alti1_dtw = dtw(y1_lpc,S.alti1_lpc);

yedi1_dtw = dtw(y1_lpc,S.yedi1_lpc);

sekiz1_dtw = dtw(y1_lpc,S.sekiz1_lpc);

dokuz1_dtw = dtw(y1_lpc,S.dokuz1_lpc);

on1_dtw = dtw(y1_lpc,S.on1_lpc);

yirmi1_dtw = dtw(y1_lpc,S.yirmi1_lpc);

otuz1_dtw = dtw(y1_lpc,S.otuz1_lpc);

```

```

kirk1_dtw = dtw(y1_lpc,S.kirk1_lpc);
elli1_dtw = dtw(y1_lpc,S.elli1_lpc);
altmis1_dtw = dtw(y1_lpc,S.altmis1_lpc);
yetmis1_dtw = dtw(y1_lpc,S.yetmis1_lpc);
seksen1_dtw = dtw(y1_lpc,S.seksen1_lpc);
doksan1_dtw = dtw(y1_lpc,S.doksan1_lpc);
yuz1_dtw = dtw(y1_lpc,S.yuz1_lpc);
bin1_dtw = dtw(y1_lpc,S.bin1_lpc);
milyon1_dtw = dtw(y1_lpc,S.milyon1_lpc);
milyar1_dtw = dtw(y1_lpc,S.milyar1_lpc);
evet1_dtw = dtw(y1_lpc,S.evet1_lpc);
hayir1_dtw = dtw(y1_lpc,S.hayir1_lpc);
basla1_dtw = dtw(y1_lpc,S.basla1_lpc);
dur1_dtw = dtw(y1_lpc,S.dur1_lpc);
ileri1_dtw = dtw(y1_lpc,S.ileri1_lpc);
geri1_dtw = dtw(y1_lpc,S.geri1_lpc);
sag1_dtw = dtw(y1_lpc,S.sag1_lpc);
sol1_dtw = dtw(y1_lpc,S.sol1_lpc);
ac1_dtw = dtw(y1_lpc,S.ac1_lpc);
kapa1_dtw = dtw(y1_lpc,S.kapa1_lpc);

signalDTW=[sifir1_dtw bir1_dtw iki1_dtw uc1_dtw dort1_dtw bes1_dtw alti1_dtw
yedi1_dtw sekiz1_dtw dokuz1_dtw on1_dtw yirmi1_dtw otuz1_dtw kirk1_dtw elli1_dtw
altmis1_dtw yetmis1_dtw seksen1_dtw doksan1_dtw yuz1_dtw bin1_dtw milyon1_dtw
milyar1_dtw evet1_dtw hayir1_dtw basla1_dtw dur1_dtw ileri1_dtw geri1_dtw sag1_dtw
sol1_dtw ac1_dtw kapa1_dtw];

minDTW= min(signalDTW);

switch minDTW

```

```
case sifir1_dtw
    app.Label2.Text ="SİFİR";
case bir1_dtw
    app.Label2.Text ="BİR";
case iki1_dtw
    app.Label2.Text ="İKİ";
case uc1_dtw
    app.Label2.Text ="ÜÇ";
case dort1_dtw
    app.Label2.Text ="DÖRT";
case bes1_dtw
    app.Label2.Text ="BEŞ";
case alti1_dtw
    app.Label2.Text ="ALTI";
case yedi1_dtw
    app.Label2.Text ="YEDİ";
case sekiz1_dtw
    app.Label2.Text ="SEKİZ";
case dokuz1_dtw
    app.Label2.Text ="DOKUZ";
case on1_dtw
    app.Label2.Text ="ON";
case yirmi1_dtw
    app.Label2.Text ="YİRMİ";
case otuz1_dtw
    app.Label2.Text ="OTUZ";
```

```
case kirk1_dtw
    app.Label2.Text ="KIRK";
case elli1_dtw
    app.Label2.Text ="ELLİ";
case altmis1_dtw
    app.Label2.Text ="ALTMİŞ";
case yetmis1_dtw
    app.Label2.Text ="YETMİŞ";
case seksen1_dtw
    app.Label2.Text ="SEKSEN";
case doksan1_dtw
    app.Label2.Text ="DOKSAN";
case yuz1_dtw
    app.Label2.Text ="YÜZ";
case bin1_dtw
    app.Label2.Text ="BİN";
case milyon1_dtw
    app.Label2.Text ="MİLYON";
case milyar1_dtw
    app.Label2.Text ="MİLYAR";
case evet1_dtw
    app.Label2.Text ="EVET";
case hayir1_dtw
    app.Label2.Text ="HAYIR";
case basla1_dtw
    app.Label2.Text ="BAŞLA";
```

```
case dur1_dtw
    app.Label2.Text ="DUR";
case ileri1_dtw
    app.Label2.Text ="İLERİ";
case geri1_dtw
    app.Label2.Text ="GERİ";
case sag1_dtw
    app.Label2.Text ="SAĞ";
case sol1_dtw
    app.Label2.Text ="SOL";
case ac1_dtw
    app.Label2.Text ="AÇ";
case kap1_dtw
    app.Label2.Text ="KAPA";
end
end
```

EK C: MFKK –Yapay Sinir Ağı ile Türkçe Rakam Tanıma Uygulaması Kodları

Şekil 6.5’de MFKK –Yapay sinir ağı ile Türkçe rakam tanıma uygulamasında sistemin eğitilmesi için kullanılan kodların bir kısmı aşağıdadır:

```
sifir1= audioread("../sesler/sifir1.wav");
sifir1_ds =detectSpeech(sifir1,16000);
sifir1_sig = sifir1(sifir1_ds(1):sifir1_ds(2));
sifir1_mfcc =mfcc(sifir1_sig,16000);
sifir1_m = mean(sifir1_mfcc);
sifir1_m_t = sifir1_m.';
sifir2= audioread("../sesler/sifir2.wav");
sifir2_ds =detectSpeech(sifir2,16000);
sifir2_sig = sifir2(sifir2_ds(1):sifir2_ds(2));
sifir2_mfcc =mfcc(sifir2_sig,16000);
sifir2_m = mean(sifir2_mfcc);
sifir2_m_t = sifir2_m.';
sifir3= audioread("../sesler/sifir3.wav");
sifir3_ds =detectSpeech(sifir3,16000);
sifir3_sig = sifir3(sifir3_ds(1):sifir3_ds(2));
sifir3_mfcc =mfcc(sifir3_sig,16000);
sifir3_m = mean(sifir3_mfcc);
sifir3_m_t = sifir3_m.';
bir1= audioread("../sesler/bir1.wav");
bir1_ds =detectSpeech(bir1,16000);
bir1_sig = bir1(bir1_ds(1):bir1_ds(2));
bir1_mfcc =mfcc(bir1_sig,16000);
bir1_m = mean(bir1_mfcc);
```

```

bir1_m_t = bir1_m.';

bir2= audioread("../sesler/bir2.wav");

bir2_ds =detectSpeech(bir2,16000);

bir2_sig = bir2(bir2_ds(1):bir2_ds(2));

bir2_mfcc =mfcc(bir2_sig,16000);

bir2_m = mean(bir2_mfcc);

bir2_m_t = bir2_m.';

bir3= audioread("../sesler/bir3.wav");

bir3_ds =detectSpeech(bir3,16000);

bir3_sig = bir3(bir3_ds(1):bir3_ds(2));

bir3_mfcc =mfcc(bir3_sig,16000);

bir3_m = mean(bir3_mfcc);

bir3_m_t = bir3_m.';

.

.

.

dokuz1= audioread("../sesler/dokuz1.wav");

dokuz1_ds =detectSpeech(dokuz1,16000);

dokuz1_sig = dokuz1(dokuz1_ds(1):dokuz1_ds(2));

dokuz1_mfcc =mfcc(dokuz1_sig,16000);

dokuz1_m = mean(dokuz1_mfcc);

dokuz1_m_t = dokuz1_m.';

dokuz2= audioread("../sesler/dokuz2.wav");

dokuz2_ds =detectSpeech(dokuz2,16000);

dokuz2_sig = dokuz2(dokuz2_ds(1):dokuz2_ds(2));

dokuz2_mfcc =mfcc(dokuz2_sig,16000);

```

```

dokuz2_m = mean(dokuz2_mfcc);
dokuz2_m_t = dokuz2_m.';
dokuz3= audioread("../sesler/dokuz3.wav");
dokuz3_ds =detectSpeech(dokuz3,16000);
dokuz3_sig = dokuz3(dokuz3_ds(1):dokuz3_ds(2));
dokuz3_mfcc =mfcc(dokuz3_sig,16000);
dokuz3_m = mean(dokuz3_mfcc);
dokuz3_m_t = dokuz3_m.';

```

```

input = [sifir1_m_t sifir2_m_t sifir3_m_t . . . ]

```

```

output = [ ..... ];

```

Şekil 6.5’de MFKK –Yapay sinir ağı ile Türkçe rakam tanıma uygulamasının MATLAB kullanıcı arayüzündeki “**Konuşmaya Başla**” yazılı butona tıklandığında çalıştırılan kodlar aşağıdadır:

```

function KonumayaBalaButtonPushed(app, event)
KonusmaKayit = audiorecorder(16000,16,1);
app.Label.Text = "konuşmaya başla";
recordblocking(KonusmaKayit,2);
app.Label.Text ="konuşma sonu";
y = getaudiodata(KonusmaKayit);
plot(app.UIAxes,y);
encok = max(y);
enaz = min(y);
s1= detectSpeech(y,16000);
line(app.UIAxes,[s1(1) s1(1)],[encok enaz],"color","red");
line(app.UIAxes,[s1(2) s1(2)],[encok enaz],"color","red");

```

```

y1 = y(s1(1):s1(2));

y1_mfcc = mfcc(y1,16000);
y1_mfcc_m = mean(y1_mfcc);
y1_mfcc_m_t= y1_mfcc_m.';
[Y,Xf,Af] = myNeuralNetworkFunction(y1_mfcc_m_t);
if round(Y(1,:))==1
    app.Label2.Text ="SİFİR";
elseif round(Y(2,:))==1
    app.Label2.Text ="BİR";
elseif round(Y(3,:))==1
    app.Label2.Text ="İKİ";
elseif round(Y(4,:))==1
    app.Label2.Text ="ÜÇ";
elseif round(Y(5,:))==1
    app.Label2.Text ="DÖRT";
elseif round(Y(6,:))==1
    app.Label2.Text ="BEŞ";
elseif round(Y(7,:))==1
    app.Label2.Text ="ALTI";
elseif round(Y(8,:))==1
    app.Label2.Text ="YEDİ";
elseif round(Y(9,:))==1
    app.Label2.Text ="SEKİZ";
elseif round(Y(10,:))==1
    app.Label2.Text ="DOKUZ";

```

end

end

myNeuralNetworkFunction fonksiyonu eğitilmiş olan YSA'nı çağırarak, girilen ses işaretine göre bir sonuç üretip onu döndürmektedir.



EK D: DÖK –Yapay Sinir Ağı ile Türkçe Rakam Tanıma Uygulaması Kodları

Şekil 6.8’de DÖK –Yapay sinir ağı ile Türkçe rakam tanıma uygulamasında sistemin eğitilmesi için kullanılan kodların bir kısmı aşağıdadır:

```
clear;

k=14; % lpc(sinyal,k)

sifir1= audioread("../sesler/sifir1.wav");

sifir1_ds =detectSpeech(sifir1,16000);

sifir1_sig = sifir1(sifir1_ds(1):sifir1_ds(2));

sifir1_lpc =mylpc(sifir1_sig,k);

sifir1_m = mean(sifir1_lpc,2);

sifir2= audioread("../sesler/sifir2.wav");

sifir2_ds =detectSpeech(sifir2,16000);

sifir2_sig = sifir2(sifir2_ds(1):sifir2_ds(2));

sifir2_lpc =mylpc(sifir2_sig,k);

sifir2_m = mean(sifir2_lpc,2);

sifir3= audioread("../sesler/sifir3.wav");

sifir3_ds =detectSpeech(sifir3,16000);

sifir3_sig = sifir3(sifir3_ds(1):sifir3_ds(2));

sifir3_lpc =mylpc(sifir3_sig,k);

sifir3_m = mean(sifir3_lpc,2);

bir1= audioread("../sesler/bir1.wav");

bir1_ds =detectSpeech(bir1,16000);

bir1_sig = bir1(bir1_ds(1):bir1_ds(2));

bir1_lpc =mylpc(bir1_sig,k);

bir1_m = mean(bir1_lpc,2);

bir2= audioread("../sesler/bir2.wav");
```

```

bir2_ds =detectSpeech(bir2,16000);
bir2_sig = bir2(bir2_ds(1):bir2_ds(2));
bir2_lpc =mylpc(bir2_sig,k);
bir2_m = mean(bir2_lpc,2);
bir3= audioread("../sesler/bir3.wav");
bir3_ds =detectSpeech(bir3,16000);
bir3_sig = bir3(bir3_ds(1):bir3_ds(2));
bir3_lpc =mylpc(bir3_sig,k);
bir3_m = mean(bir3_lpc,2);
.
.
.
dokuz1= audioread("../sesler/dokuz1.wav");
dokuz1_ds =detectSpeech(dokuz1,16000);
dokuz1_sig = dokuz1(dokuz1_ds(1):dokuz1_ds(2));
dokuz1_lpc =mylpc(dokuz1_sig,k);
dokuz1_m = mean(dokuz1_lpc,2);
dokuz2= audioread("../sesler/dokuz2.wav");
dokuz2_ds =detectSpeech(dokuz2,16000);
dokuz2_sig = dokuz2(dokuz2_ds(1):dokuz2_ds(2));
dokuz2_lpc =mylpc(dokuz2_sig,k);
dokuz2_m = mean(dokuz2_lpc,2);
dokuz3= audioread("../sesler/dokuz3.wav");
dokuz3_ds =detectSpeech(dokuz3,16000);
dokuz3_sig = dokuz3(dokuz3_ds(1):dokuz3_ds(2));
dokuz3_lpc =mylpc(dokuz3_sig,k);

```

```
dokuz3_m = mean(dokuz3_lpc,2);
input = [sifir1_m sifir2_m sifir3_m ...];
output = [ ];
```

Şekil 6.8’de DÖK –Yapay sinir ağı ile Türkçe rakam tanıma uygulamasının MATLAB kullanıcı arayüzündeki “Konuşmaya Başla” yazılı butona tıklandığında çalıştırılan kodlar aşağıdadır:

```
function KonumayaBalaButtonPushed(app, event)
KonusmaKayit = audiorecorder(16000,16,1);
app.Label.Text = "konuşmaya başla";
recordingblock(KonusmaKayit,2);
app.Label.Text = "konuşma sonu";
y = getaudiodata(KonusmaKayit);
plot(app.UIAxes,y);
encok = max(y);
enaz = min(y);
s1= detectSpeech(y,16000);
line(app.UIAxes,[s1(1) s1(1)],[encok enaz],"color","red");
line(app.UIAxes,[s1(2) s1(2)],[encok enaz],"color","red");

y1 = y(s1(1):s1(2));
k=14; % lpc(sinyal,k)

y1_lpc = mylpc(y1,k);
y1_lpc_m = mean(y1_lpc,2);

[Y,Xf,Af] = myNeuralNetworkFunction(y1_lpc_m);
```

```

if round(Y(1,:))==1
    app.Label2.Text ="SİFİR";
elseif round(Y(2,:))==1
    app.Label2.Text ="BİR";
elseif round(Y(3,:))==1
    app.Label2.Text ="İKİ";
elseif round(Y(4,:))==1
    app.Label2.Text ="ÜÇ";
elseif round(Y(5,:))==1
    app.Label2.Text ="DÖRT";
elseif round(Y(6,:))==1
    app.Label2.Text ="BEŞ";
elseif round(Y(7,:))==1
    app.Label2.Text ="ALTI";
elseif round(Y(8,:))==1
    app.Label2.Text ="YEDİ";
elseif round(Y(9,:))==1
    app.Label2.Text ="SEKİZ";
elseif round(Y(10,:))==1
    app.Label2.Text ="DOKUZ";
end
end

```

myNeuralNetworkFunction fonksiyonu eğitilmiş olan YSA'nı çağırarak, girilen ses işaretine göre bir sonuç üretip onu döndürmektedir.

ÖZGEÇMİŞ

Ad-Soyad : Mehmet ÖZTÜRK

ÖĞRENİM DURUMU:

- **İlkokul** : 1988, Yavuz İlkokulu, Gölbaşı- Adıyaman
- **Ortaokul** : 1991, Gölbaşı Ortaokulu, Gölbaşı - Adıyaman
- **Lise** : 1994, Cumhuriyet Fen Lisesi, Diyarbakır
- **Lisans** : 2000, Gazi Üniversitesi, Mühendislik-Mimarlık Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü, Ankara
- **Yüksek Lisans** : 2010, Ahmet Yesevi Üniversitesi, Bilgisayar Mühendisliği (Tezsiz)

MESLEKİ DENEYİM:

- 2001-2006 Gaziantep Üniversitesi Gölbaşı MYO'da Öğretim Görevlisi olarak çalıştı.
- 2006-2012 Adıyaman Üniversitesi Gölbaşı MYO'da Öğretim Görevlisi olarak çalıştı.
- 2013-2016 Adıyaman Üniversitesi Gölbaşı MYO'da dışarıdan dersler verdi.
- 2018-2019 Gaziantep Sadestone Ltd. Şti. 'de çalıştı
- 2022- Günümüze. Milli Eğitim Bakanlığı'nda Memur olarak çalışmakta.

YÜKSEK LİSANS TEZİNDEN TÜRETİLEN ÇALIŞMALAR (Makaleler, Bildiriler, Patentler v.b.)

- Öztürk, M.,Özgüven, Ö. (2025). Türkçe Konuşma Tanımada Mel Frekansı Kepstral Katsayıları ve Doğrusal Öngörülü Kodlama Yöntemlerinin Karşılaştırılması. 5. Bilsel International Çatalhöyük Scientific Researches Congress, 19-20 April, 2025, Konya/Türkiye
- Öztürk, M.,Özgüven, Ö. (2025). Yapay Sinir Ağları Kullanarak Türkçe Rakam Tanıma. Cumhuriyet 14th International Conference on Applied Sciences, April 23, 2025, Ankara/Türkiye