



**DERİN ÖĞRENME TABANLI
SESLİ KOMUT SİSTEMİ**

Ayşe Nur ÇAYIR

**YÜKSEK LİSANS TEZİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

AĞUSTOS 2021

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Ayşe Nur ÇAYIR

16/08/2021

DERİN ÖĞRENME TABANLI SESLİ KOMUT SİSTEMİ
(Yüksek Lisans Tezi)

Ayşe Nur ÇAYIR

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Ağustos 2021

ÖZET

Günümüzde yapay zeka uygulamalarında kullanılmak üzere ses tanıma ve sınıflandırmaya dayalı çalışmalara önem verilmektedir. Sesli komutları tanıyan bir sistem tasarlanırken derin öğrenmeden yararlanmak başarılı sonuçlar elde etmeyi sağlamaktadır. Bu tez çalışmasında derin öğrenme mimarilerinden olan evrişimli sinir ağı kullanılarak 12 adet İngilizce sesli komutun tanınmasını sağlayan bir sistem tasarlanmıştır. Tasarlanan sistem test doğruluğu açısından %94,64 başarı sağlamıştır. Ayrıca model eğitilirken daha önce hiç karşılaşmadığı, anadili İngilizce olmayan kişilerden alınan seslerden oluşan bir veri seti oluşturulup modelin gerçek hayattaki başarısı araştırıldığında %63,29 doğruluk oranı elde edilmiştir. Türkçe için sesli verilerin kısıtlı olduğu görülmüş ve bu doğrultuda kısıtlı veriye sahipken veri artırma teknikleri kullanılarak derin öğrenme modelinin başarısının artırılıp artırılamayacağı incelenmiştir. 12 adet sesli Türkçe komut içeren çok küçük bir veri seti oluşturulduktan sonra ses verileri manipüle edilerek veri seti büyütülmüştür. Veri artırma işlemi yapılmadan önce ortalama test doğruluğu %53,947 iken veri artırma sonrasında %85,756'ya ve ortalama gerçek hayat doğruluğu %58,074 iken veri artırmadan sonra %64,761'e çıkmıştır. Deneysel sonuçlar kısıtlı sayıda veriye sahipken dahi veri artırma sayesinde doğruluk oranlarının artırılabilceğini göstermiştir. Ayrıca veri artırımların etkisi sınıf bazlı olarak incelenmiştir. İnceleme sonucunda veri artırma tekniklerinin sınıflara özgü olarak uygulanmasının başarıyı daha da artıracığı ve bu yönde ilerleyen çalışmalara katkı sağlayacağı düşünülmektedir.

Bilim Kodu : 90541
Anahtar Kelimeler : Derin öğrenme, evrişimli sinir ağı, sesli komut, veri artırma
Sayfa Adedi : 76
Danışman : Doç. Dr. Tuğba Selcen NAVRUZ

DEEP LEARNING BASED VOICE COMMAND SYSTEM

(M. Sc. Thesis)

Ayşe Nur ÇAYIR

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

August 2021

ABSTRACT

Today, studies based on voice recognition and classification are important for use in artificial intelligence applications. Taking advantage of deep learning while designing a system that recognizes voice commands provides successful results. In this thesis, a system that enables the recognition of 12 English voice commands is designed using convolutional neural network, which is one of the deep learning architectures. The designed system achieved 94.64% success in terms of test accuracy. In addition, when the model was trained, a data set consisting of sounds taken from non-native English speakers was created and the real-life success of the model was investigated, and an accuracy rate of 63.29% was obtained. It has been seen that the data for Turkish is limited and it has been examined whether the success of the deep learning model can be increased by using data augmentation techniques while having limited data in this direction. After creating a very small data set containing 12 voice Turkish commands, the data set was enlarged by manipulating the voice data. While the average test accuracy was 53.947% before data augmentation, it increased to 85.756% after data augmentation, and while the average real-life accuracy was 58.074%, it increased to 64.761% after data augmentation. Experimental results have shown that even with limited data, accuracy rates can be increased by data augmentation. In addition, the effect of data augmentation was examined on a class basis. As a result of the examination, it is thought that the application of data augmentation techniques specific to classes will increase the success even more and contribute to the studies progressing in this direction.

Science Code : 90541

Key Words : Deep learning, convolutional neural network, voice command, data augmentation

Page Number : 76

Supervisor : Assist. Prof. Dr. Tuğba Selcen NAVRUZ

TEŞEKKÜR

Tez çalışmam boyunca düzenli aralıklarla toplantılar yaparak tezimin planlı bir şekilde ilerlemesini sağlayan, tecrübesiyle ve bilgisiyle bana yol gösteren danışmanım Doç. Dr. Tuğba Selcen NAVRUZ'a, hiç bir zaman desteklerini üzerimden çekmeyen aileme, veri seti oluşturmam için kıymetli seslerini benimle paylaşan yakın çevreme, deneysel çalışmalar esnasında donanıma ihtiyaç duyduğum anda bilgisayarını benimle paylaşan ve yardımlarını esirgemeyen Arş. Gör. Serhat KARAÇAM'a, tezin başlığından kaynakçasına kadar her aşamasında yanımda olan, cümlelerimi okuyup yorumlayan, değerlendiren, fikirlerini ve bilgisini benimle paylaşan, her daim motive kalmamı sağlayan Bilgisayar Mühendisi arkadaşım Hümeysra Eslem GÜLER'e, farklı alanlara olan ilgisi sayesinde hayal gücümü zorlamamı sağlayan ve fikir alışverişleriyle tez konusunu belirlememde yardımcı olan, psikolojik olarak yorulduğumda neşesiyle beni ayakta tutmayı başaran ve desteğini her an hissettiğim değerli arkadaşım Pınar KAYA'ya, başta Merve KARASU ve Ebru ÇAKMAK olmak üzere adını burada yazamadığım ancak yanımda olan tüm arkadaşlarıma, bilgimin yetmediği yerlerde yanına koşup sorularımı ilettiğim ve cevabını da aldığım tüm hocalarıma teşekkürlerimi iletmeyi bir borç bilirim.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ŞEKİLLERİN LİSTESİ	ix
ÇİZELGELERİN LİSTESİ.....	xi
SİMGELER VE KISALTMALAR.....	xii
1. GİRİŞ	1
2. LİTERATÜR TARAMASI	5
3. DERİN ÖĞRENME MİMARİLERİ VE ARAÇLARI	11
3.1. Evrişimli Sinir Ağlar (ESA).....	19
3.1.1. Evrişim katmanı.....	20
3.1.2. Doğrusal olmayan katman.....	21
3.1.3. Havuzlama katmanı.....	22
3.1.4. Düzleme ve tam bağlantı katmanı.....	23
3.2. Özyinelemeli Sinir Ağları (ÖSA).....	24
3.3. Uzun-Kısa Süreli Bellek (UKSB)	25
3.4. Diğer Mimariler	27
3.5. Derin Öğrenme Araçları.....	27
4. SES SINIFLANDIRMA VE ÖZNİTELİK ÇIKARMA	31
4.1. Mel Frekans Kepstral Katsayıları Yöntemi	31
5. SES VERİSİ ARTIRMA VE KULLANILAN YÖNTEMLER	37
6. DENEYLER.....	41

	Sayfa
6.1. Veri Setlerinin Oluřtuurulması	42
6.2. Mimari Seçimi	45
6.2.1. ESA deneyleri.....	45
6.2.2. UKSB deneyleri.....	47
6.2.3. ESA ve UKSB karşılařtırması.....	49
6.3. Veri Artırma Deneyleri	52
6.4. Sınıfsal Doğruluk Deneyleri	57
7. SONUÇ VE ÖNERİLER	63
KAYNAKLAR	67
EKLER.....	71
EK-1. İngilizce komutların MFKK grafikleri.....	72
EK-2. Türkçe komutların MFKK grafikleri	74
ÖZGEÇMİř	77

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 3.1. Yapay sinir hücresinin yapısı.....	11
Şekil 3.2. Derin sinir ağının yapısı.....	13
Şekil 3.3. Gizli katman içerisindeki bir nöron	15
Şekil 3.4. Evrişimli sinir ağının katmanları	20
Şekil 3.5. Evrişim işleminde kullanılan giriş matrisi, filtre ve çıkış matrisi örneği	21
Şekil 3.6. a) Sigmoid b) Tanh c) ReLU	22
Şekil 3.7. Maksimum havuzlama işlemi	23
Şekil 3.8. Düzleme katmanından tam bağlantı katmanına geçiş örneği	23
Şekil 3.9. ÖSA çalışma diyagramı	25
Şekil 3.10. UKSB' nin çalışma yapısı	26
Şekil 4.1. Sesin MFKK özniteliğinin çıkartılmasına ait blok diyagram	32
Şekil 4.2. Ses sinyalinin çerçeveleme işlemi	32
Şekil 4.3. Ses sinyaline ait pencereleme işlemi	33
Şekil 4.4. Mel ölçeği uygulaması.....	34
Şekil 4.5. MFKK katsayıları	35
Şekil 4.6. a) Yes için MFKK b) No için MFKK.....	36
Şekil 6.1. Deneysel çalışma planı	41
Şekil 6.2. ESA mimarisinin katmanları.....	47
Şekil 6.3. UKSB mimarisinin katmanları.....	49
Şekil 6.4. a) ESA doğruluk ve kayıp grafiği b) UKSB doğruluk ve kayıp grafiği	50
Şekil 6.5. Küçük İngilizce veri seti ile oluşturulan modelin doğruluk ve kayıp grafiği .	52
Şekil 6.6. Artırılmış daraltılmış İngilizce veri seti ile oluşturulan modelin doğruluk ve kayıp grafiği	53

	Sayfa
Şekil 6.7. Türkçe orijinal veri seti ile oluşturulan modelin doğruluk ve kayıp grafiği .	55
Şekil 6.8. Artırılmış Türkçe veri seti ile oluşturulan modelin doğruluk ve kayıp grafiği	55
Şekil 6.9. Çok artırılmış Türkçe veri seti ile oluşturulan modelin doğruluk ve kayıp grafiği	56

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 6.1. Veri setindeki mevcut komutlar ve ses örneği sayıları.....	42
Çizelge 6.2. İngilizce gerçek hayat doğruluğu testleri için kullanılan veri seti içeriği...	43
Çizelge 6.3. Türkçe veri seti içeriği	43
Çizelge 6.4. Çok Artırılmış Türkçe veri seti içeriği	44
Çizelge 6.5. ESA mimarisi deneyleri	44
Çizelge 6.6. UKSB mimarisi deneyleri	46
Çizelge 6.7. Aktivasyon fonksiyonları karşılaştırması.....	48
Çizelge 6.8. UKSB ve ESA sonuçlarının karşılaştırması.....	51
Çizelge 6.9. Küçük veri seti ile UKSB ve ESA sonuçlarının karşılaştırması	51
Çizelge 6.10. İngilizce veri seti ile veri artırma kıyaslaması	53
Çizelge 6.11. Daha önceki çalışmayla kıyaslama.....	54
Çizelge 6.12. Orijinal ve artırılmış veri seti kıyaslaması	55
Çizelge 6.13. Gerçek hayattaki doğruluk oranları	56
Çizelge 6.14. İngilizce orijinal veri seti ile eğitimin sınıfsal doğruluğu	57
Çizelge 6.15. Daraltılmış İngilizce veri seti ile eğitimin sınıfsal doğruluğu	58
Çizelge 6.16. Artırılmış daraltılmış İngilizce veri seti ile eğitimin sınıfsal doğruluğu ..	58
Çizelge 6.17. Daraltılmış İngilizce- artırılmış daraltılmış İngilizce veri seti değişimi ..	59
Çizelge 6.18. Türkçe orijinal veri seti ile yapılan eğitimin sınıfsal doğrulukları	60
Çizelge 6.19. Artırılmış Türkçe veri seti ile yapılan eğitimin sınıfsal doğrulukları	60
Çizelge 6.20. Türkçe orijinal- artırılmış Türkçe veri seti değişimi	61

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklamalar
ACD	Ayrık cosinüs dönüşümü
AFD	Ayrık Fourier dönüşümü
BG	Background noise
CYG	Çapraz yineleme grafiği
DBM	Derin Boltzmann makinesi
DİA	Derin inanç ağları
DRC	Dynamic range compression
DSA	Derin sinir ağları
DOK	Derin oto-kodlayıcı
DÖKK	Doğrusal öngörü kepsral katsayıları
ESA	Evrişimli sinir ağı
FFT	Fast Fourier transform
GAN	Generative adversial networks
GRU	Gated recurrent units
GTÜ	Geçitli tekrarlayan üniteli
IDE	Integrated Devolopment Environment
MFKK	Mel-frekans kepsral katsayıları
ÖSA	Özyinelemeli sinir ağı
PS	Pitch shifting
TS	Time stretch
UKSB	Uzun kısa süreli bellek

1. GİRİŞ

Teknolojinin üst düzeyde kullanıldığı günümüz dünyasında en ufak bir ürünü bile akıllı hale getirmek mümkündür. Teknolojinin hızla gelişmesiyle bu teknolojiye ayak uyduracak insan gücüne de duyulan ihtiyaç artmıştır. Ancak bu ihtiyacı yalnızca insanlar üzerinden karşılamak yeterli değildir. İnsanlara destek olmak için yapay bir zekaya sahip akıllı makineler geliştirilmiştir. Böylece “yapay zeka” kavramı birçok alanda karşımıza çıkmaya başlamıştır.

Hayatımıza giren makinelerin insanlara fayda sağladığı en önemli yapılardan birisi yapay zekanın en temel bileşenlerinden olan yapay sinir ağlarıdır. İnsan sinir sisteminden ilham alınarak geliştirilen yapay sinir ağları eğitilebilen ve kendi kendine değerlendirme yapıp duruma göre çıkış oluşturabilen yapay bir sinir sistemidir. Sistemin çalışma mantığı nöronlara (sinir hücresi) gelen bilgilerin (giriş) belirli parametrelerle (ağırlık, bias) çarpılıp toplanmasıyla ve gerekli fonksiyonlara (aktivasyon fonksiyonları) tâbi tutulmasıyla giriş bilgisinin işlenerek ve gerektiği gibi farklılaşarak çıkışa ulaşması süreci olarak tanımlanabilir. Yapay sinir ağları kullanan ve bu tez çalışmasının da temelini oluşturan derin öğrenme, çok sayıda giriş bilgisi içeren veri setlerinden yararlanarak bir uygulamanın amacına göre istenilen çıkışları tahmin etmeye yarayan algoritmadır. Bu algoritmanın kullanılması gelişen teknoloji ile birlikte ortaya çıkan birçok problemin hızla çözülmesini mümkün hale getirmiştir.

Bu tez çalışmasında özellikle yapay zekayla üretilmiş robotlara entegre edilmesi amacıyla derin öğrenme kullanılarak sesli komut tanıma sistemi tasarlanmıştır. Tasarlanan derin öğrenme sistemi insanlar tarafından söylenen 1 saniyelik “down, forward, follow, go, left, no, off, on, right, stop, up, yes” olmak üzere 12 farklı kısa komutu algılar ve hangi komutun verildiğini tespit eder. Bu sistem verilen komutu yerine getirme görevlerini yapmak için hazırlanmış başka bir sisteme entegre edilerek geliştirilebilir.

Ses tanıma ile ilgili bir sistem tasarlamadaki başlıca sebeplerden birisi, en bilindik örneklerden olan Siri, Google Asistan gibi oldukça gelişmiş insanla konuşan sistemlerde bile kişilerin konuşma tarzından kaynaklı yanlış anlaşılma veya hiç anlaşılama durumunun meydana gelmesidir.

İnsanların bazıları çok hızlı konuşurken bazıları oldukça yavaş konuşabilir, kimisi düzgün bir Türkçe kullanırken kimisi bölgesel farklılıklardan dolayı farklı ağızlar ile konuşabilir ya da kelime telaffuzlarında farklılıklar olabilir. Ayrıca kişinin bulunduğu ortamdan kaynaklı olarak sessiz konuşması veya gürültüden dolayı bağırarak konuşması ve daha birçok sebepten dolayı sesle çalışan sistemlerde algılama sorunu oluşabilmektedir. İnsanlardan ve ortamdan kaynaklı bu gibi sorunların başarılı bir derin öğrenme mimarisi yardımıyla üstesinden gelinebileceği düşünülmüştür ve bu tez çalışmasında teorik olarak bir sonuç elde etmek yerine mümkün olduğunca insan ve ortam kaynaklı sorunları içeren bir veri seti oluşturularak modelin gerçek hayattaki başarısı artırılmaya çalışılmıştır.

Derin öğrenme ile eğitim yapılırken veri seti ne kadar büyükse eğitimin başarılı olma şansı o denli artmaktadır. Ne yazık ki literatürde sesli komutlara ait veri setleri çok fazla değildir. Bu tez çalışması için yeterli olacak büyüklüğe sahip İngilizce veri seti kullanılmıştır. Ancak Türkçe gibi dünya dili olmayan diğer dillerde veri seti bulmak çok daha zor olmaktadır. Bu olumsuzluğa bir çözüm olarak veri setlerine uygulanan farklı deformasyonlar sayesinde veri artırma işlemi yapılmaktadır.

Bu tez çalışmasının amacı kısıtlı veri seti ile eğitim yapıldığında başarının ne derecede düştüğünü gözlemlemek ve bu az veriyle yapılan eğitimlerin çeşitli veri artırma yöntemleri kullanarak yeniden yapılmasıyla daha başarılı sonuçlar elde etmeye çalışmaktır. Bu amaca ulaşmak için öncelikle ses ile ilgili uygulamalarda en çok kullanılan derin öğrenme mimarileri ile deneyler yapılmış ve bu tez çalışması için en başarılı olan mimari tespit edilmiştir. En başarılı olan mimari seçilirken tez çalışmasının yalnızca teoride kalmaması hedefiyle gerçek hayattaki başarı oranı baz alınmıştır. Gerçek hayattaki başarı oranı, eğitim ve test başarılarından ziyade anadili Türkçe olan insanlardan alınan ve toplam 651 sesten oluşan çok daha küçük, gürültülü ve yanlış tanımlanmaya müsait sesler kullanılarak test edilen doğruluk değeridir. Mimari seçimi tamamlandıktan sonra orijinal veri setinden her bir komut için ellişer adet ses alınıp küçük bir veri seti oluşturulmuştur. Bu veri seti ile eğitim yapılmış ve gerçek hayattaki başarı oranının küçük veri seti kullanımından dolayı düşük olduğu gösterilmiştir.

Küçük veri setinin düşük başarı oranına sebep olduğu gösterildikten sonra tezin esas amacı olan veri artırma kullanarak başarıyı artırmak için deneylere başlanmıştır. İlk olarak ellişer adet ses içeren İngilizce komutlar için veri artırma yapılmıştır. Ardından “aç, aşağı, dur,

evet, git, hayır, ileri, kapat, sağ, sol, takip et, yukarı” komutlarının seslendiriliği küçük bir Türkçe veri seti oluşturulmuştur. Oluşturulan veri seti ile düşük doğruluk oranına sahip bir eğitim gözlemlenmiştir ve ardından bu düşük doğruluk oranı seslerin manipüle edilerek veri setinin artırılması sayesinde artırılmıştır. Böylece veri seti kısıtlıyken dahi belirli bir seviyede başarıya ulaşılabileceği ve veri artırmanın önemi gösterilmiştir.

Ayrıca veri artırma deneylerinin gerçek hayattaki doğruluk oranları sınıfsal olarak belirlenmiştir. Veri artırmanın toplam her komuta ait sınıfın doğruluğunu farklı şekilde etkilediği görülmüştür. Dolayısıyla veri artırma işlemlerinin her bir komut özel olarak yapılması gerektiği çıkarımında bulunulmuştur.

Bu tez çalışmasında 2. Bölümde literatürdeki benzer çalışmalara değinilmiştir. 3. Bölümde derin öğrenme mimarileri tanıtılmış, tez boyunca kullanılan Evrişimli Sinir Ağı (ESA) mimarisi detaylı olarak incelenmiştir ve derin öğrenmeyi kullanabilmek için gerekli olan araçlar verilmiştir. 4. Bölümde ise sesin Mel Frekans Kepstral Katsayıları (MFKK) özniteliğinin ne olduğu ve nasıl çıkartıldığı açıklanmıştır. 5. Bölümde veri artırma teknikleri üzerinde durulmuş ve ses için kullanılabilecek artırmalar anlatılmıştır. 6. Bölümde tez süresince gerçekleştirilen deneysel çalışmalar ve sonuçları gösterilmiştir ve 7. Bölümde tez çalışması boyunca elde edilen sonuçlar ve çıkarımlar belirtilerek tez çalışması tamamlanmıştır.

2. LİTERATÜR TARAMASI

Literatür incelendiğinde çeşitli ses verilerinin kullanıldığı derin öğrenme tabanlı birçok çalışma olduğu görülmüştür. Bazı çalışmalar sesin farklı temsillerinin kullanılması temelinde ilerlerken bazıları konuya derin öğrenme mimarileri açısından bakmıştır. Ayrıca bu tezde de üzerinde durulan, ses için kısıtlı veri olması durumuna değinen çalışmalar olmuştur ve bu kısıtlı veri sorununa çözüm olarak çeşitli veri artırma tekniklerinin incelendiği çalışmalar da mevcuttur. Bu başlık altında tez konusuna yakın, tezin daha iyi açıklanmasına ve sonuca varmasına katkı sağlaması açısından önemli olduğu düşünülen, şimdiye kadar üzerinde çalışılmış konular anlatılmıştır.

Konuşma tanıma çalışmalarında kullanılan teknikleri ve aşamaları incelendiği makalede [1], MATLAB yardımıyla kadınlara ve erkeklere ait seslerin MFKK öznitelikleri kullanılarak cinsiyet tahmini gerçekleştirilmiştir. Derin sinir ağları kullanılan bu çalışmada istenilen durumun verimli bir şekilde yapıldığı sonucuna varılmıştır.

Sesin duygulara göre farklılık gösteren özelliklerinin kullanılmasıyla otomatik duygu tanıma sistemi geliştiren Demir ve diğerleri [2] MFKK özniteliklerinin yanı sıra Lineer Öngörülü Kodlama, spektral alt-bant ağırlık merkezi, Doğrusal Öngörü Kepstral Katsayıları (DÖKK), yansıma katsayıları ve log enerji özellikleri kullanılıp tüm özelliklerin ayrı ayrı ortalamaları alınarak elde edilen özellik matrisi kullanılmıştır. Uzun-Kısa Süreli Bellek (UKSB) algoritmasının kullanıldığı önerilen yöntem, önceki çalışmalarla karşılaştırılmış ve RAVDESS veri seti ile % 68,8; SAVEE veri seti ile %72,13; RML veri seti ile %70,35 test doğruluğu elde edilmiştir. RAVDESS ve SAVEE veri setleri ile önerilen yöntem eski çalışmalardan daha yüksek doğruluk sağlamıştır. Çalışmada en başarılı olan veri setinin SAVEE olması, içerdiği ses dosyalarının anadili İngilizce olan kişilerce seslendirilmesinin bir sonucudur.

Derin öğrenme algoritmalarını akustik sahne sınıflandırma için kullanan ve başarılarını analiz eden çalışmada [3] öznitelik çıkarımında mel-ölçeği özniteliklerini seçmişlerdir. 4 ve 8 katmanlı ESA yöntemine ek olarak UKSB-yapay sinir ağı ve Geçitli Tekrarlayan Üniteli (GTÜ)- yapay sinir ağı algoritmalarının gerekli katmanlarının bu yöntemine eklenmesi önerilmiştir. Sonuçta GTÜ-yapay sinir ağının sınıflandırma başarısının UKSB-yapay sinir

ağı %1,6 daha yüksek olduğu ve GTÜ-yapay sinir ağının 4 katmanlı ESA'ya eklenmesi durumunda 8 katmanlı ESA'ya göre daha yüksek başarı sağladığı sonucuna varılmıştır. Daha önceki bir çalışmaya kıyasla 4 katmanlı ESA+ GTÜ önerisi %2,1 daha başarılı sonuç vermiştir.

Seslerin farklı temsilleri kullanılarak en iyi performansı elde etmek için farklı ESA mimarilerinin kullanıldığı makalede [4] sesli komutların sınıflandırılması üzerine çalışılmıştır. Ses komutları MFKK öznitelikleri ile temsil edilmiştir. Ayrıca makalede konuşma tanımda veri artırma yöntemleri üzerinde durulmuş ve oynatma hızı oranının 0,7-1,4 aralığında rastgele değişimi, -0,1 ila 0,1 saniye aralığında rastgele bir değerle zaman kaydırma ve mevcut dalga formunun maksimum değerinin 0-0,05'i arasında arkaplan gürültüsünden rastgele gürültü ekleme işlemleri ile veri artırma uygulanmıştır. Artırma işlemlerinin modelleri genelleştirmeye ve hız, hacim, perde, arka plan gürültüsü gibi değişikliklere karşı dayanıklı hale getirdiği kanısına varılmıştır.

Türkçe numara (0'dan 10'a kadar) tanıma sistemi öneren çalışmada [5] özellik çıkarma olarak MFKK yöntemi kullanılmıştır. MFKK ile elde edilen katsayılar Hidden Markov Modeli için giriş olarak kullanılmıştır ve sonuçta sistemin test doğruluğu %83,3 olarak gözlemlenmiştir.

MFKK ve derin sinir ağlarına dayalı olarak gerçekleştirilen çalışmada [6] Endonezce açık (buka) ve kapalı (tutup) sesli komutlarının sınıflandırılıp otomatik olarak bir kanal kapısı kontrolü sağlanmıştır. Sonuçta %99,9 eğitim başarısı ve "buka" komutu için %75, "tutup" komutu için %55 test başarısı elde edilmiştir.

Bir iletişim arayüzünün tasarlandığı çalışmada [7] yapay zeka ve derin öğrenme ile mobil robot hareketleri "go, go forward, turn left" gibi sesli komutları kullanarak kontrol edilmiştir. Programlama dili olarak Python seçilmiş ve "SpeechRecognition", "PyAudio" gibi ses tanımda standartlaşmış kütüphaneler kullanılmıştır.

İnsanların sesleri tanımlama ve bir şeylerle ilişkilendirmesi yeteneği makine öğrenmesi için yeni bir araştırma konusu olmuştur ve farklı veri setlerine olan ihtiyaç artmıştır. Bu ihtiyacı karşılamak üzere oluşturulmuş "Audio Set" veri seti literatürde kullanıma sunulmuştur. Bu veri seti human voice, vehicle, noise gibi 632 farklı sesli olay kategorisinden oluşur ve

toplam 4 971 saatlik ses içermektedir [8]. Audio set veri setinin, akustik olayların algılanması üzerine yapılan çalışmalara katkı sağlayacağı düşünülmektedir.

Literatürde anahtar kelime belirleme sistemlerini eğitmeye ve değerlendirmeye yardımcı olmak amacıyla tasarlanmış “Speech commands dataset version 2” ses veri kümesinin tanıtıldığı bir çalışma mevcuttur [9]. Veri setinde toplam 35 farklı kelimedenden oluşan 105 829 ses verisi bulunmaktadır. Bu tez çalışmasında kullanılacak olan ses dosyaları belirtilen veri setinden alınmıştır. Sonuçta tek kelimelik seslerden oluşan bu veri setinin birçok modeli eğitmek ve değerlendirmek için faydalı olacağı kanısına varılmıştır.

Çevresel ses sınıflandırması çalışmalarında veri kıtlığından dolayı bir ESA mimarisi kullanmayı öneren çalışmada [10] ses verisi artırma yöntemleri kullanılmıştır. Farklı artırma yöntemlerinin mimari üzerindeki etkisi incelenmiştir. Veri artırma yöntemlerinden; zaman germe (TS), perde değiştirme (PS), dinamik aralık sıkıştırma (DRC) ve arka plan gürültüsü (BG) ekleme kullanılmıştır. Perde değiştirmenin faydalı bir artırma olduğu görüldüğünden farklı değerler ile bir kez daha aynı işlem uygulanmıştır. Veri artırma işlemi uygulanmadan önce en fazla %74 test doğruluğu elde edilirken veri artırma ile bu oran %79’a yükselmiştir.

Çevresel ses sınıflandırmasında etiketli verilerin kıtlığından dolayı birçok veri artırma yönteminin incelendiği ancak mevcut artırımların olumsuz yönlerinin de olduğuna değinen bir makalede [11] metrik öğrenme tabanlı veri artırma önerilmiştir. Bu çalışmada [10] makalesinde kullanılan aynı deformasyonlar uygulanmıştır ve kabakuvvet artırması (brute-force augmentation), sınıf koşullu artırma (class-conditional augmentation) ve metrik tabanlı artırma (metric-based augmentation) olmak üzere 3 tür artırma deneyi yapılmıştır. Önerilen yöntem ile diğer yöntemler arasında karşılaştırılabilir bir performans elde edilirken eğitim için gereken veri miktarının büyük ölçüde azaltıldığı görülmüştür.

Çevresel ses sınıflandırması için bir ESA mimarisi önerilen çalışmada [12] veri artırmayı kullanarak sentetik veri üretme olasılığı araştırılmıştır. [10] makalesinde kullanılan 4 temel artırma yöntemi ve Üretken Çekişmeli Ağ (GAN) yöntemi kullanılarak artırımlar yapıp sonuçlar karşılaştırılmıştır. Temel deformasyonlardan TS, PS1, PS2’nin temel modelin başarısını (%94,84) düşürdüğü (sırasıyla %80,08; %76,69; %77,16), DRC ve BG’nin karşılaştırılabilir sonuçlar verdiği (sırasıyla %95,31; %93) ve önerilen GAN yönteminin test doğruluğunu %97,03 değerine çıkardığı tespit edilmiştir. Buradan hareketle artırımların

olumsuz etkileri olabileceği unutulmamalı ve en uygun artırma yöntemlerinin seçilmesinin doğruluğu artırmak için önemli bir karar olduğu bilinmelidir.

Ses öz niteliklerini çıkarmak için MFKK kullanılan ve bir ESA mimarisi öneren çalışmada [13], çevresel ses sınıflandırmasında veri artırma konulu diğer çalışmalara benzer olarak TS, PS1, PS2, DRC ve BG yöntemlerine ek olarak DÖKK yöntemi de kullanılmıştır. Yapılan deneyler sonucunda artırmasız veri seti ile %83,5 test doğruluğu elde edilmiştir. Artırma yöntemleri kullanıldığında her bir sınıf için farklı bir artırma metodu sınıfsal olarak sonucu olumlu etkilemiş olsada toplamda doğruluk oranı azalmıştır. TS, PS1, PS2, DRC, BG ve DÖKK yöntemleri için test doğrulukları sırasıyla %67,8; %70; %65; %78,5; %79,7 ve %80'dir. Diğer yöntemlerle karşılaştırıldığında DÖKK'nin en iyi sonucu verdiği görülmüştür. Çalışmanın sonuç kısmında sınıf koşullu veri artırmanın doğruluğu önemli ölçüde artırabileceği önerisinde bulunmuştur.

İncelenen diğer bir çalışmada [14] derin öğrenme mimarileri için kullanılan 10 adet veri seti ve derin öğrenmede sıklıkla tercih edilen Theano, Caffe, Torch, TensorFlow, Keras ve MatConvNet yazılım kütüphaneleri incelenmiştir. Keras'ın hem konvolüsyonel hem yinelenebilir hem de ikisinin kombinasyonu olan ağırları desteklediği belirtilmiştir.

Çevresel seslerin zaman içerisinde sabit olmayan bir yapısı olduğundan uzun süreli bağımlılıkları öğrenme yeteneğine sahip UKSB yapay sinir ağını kullanarak çevresel ses sınıflandırması üzerinde çalışılan makalede [15] %84,25 test doğruluğu elde edildiği gözlemlenmiştir. Ayrıca ESA ile sonuçlar karşılaştırılmıştır ve ESA mimarisinin test doğruluğu %81,67 olarak gözlemlenmiştir. Karşılaştırma sonucunda UKSB sinir ağına performans ESA mimarisine göre biraz daha artırdığı ve daha tutarlı olduğu söylenmiştir.

Ses sınıflandırma konulu çalışmalarda ESA mimarisi çoğunlukla tercih edilmektedir [16]. Çünkü seslerin görüntü ile ifade edildiği farklı temsilleri mevcuttur ve bu temsiller aracılığıyla görüntü işleme sürecinde kullanılan ağlardan (ESA gibi) yararlanılabilmektedir. Bu sebepten dolayı çevresel seslerin spektrogram, MFKK ve Çapraz Yineleme Grafiği (CYG) gibi farklı temsilleri kullanılarak sınıflandırma yapılmıştır ve bu işlem için ESA tercih edilmiştir. Hem nesne hem de ses tanıma ve sınıflandırma uygulamalarında aynı teknolojinin yani ESA'nın kullanılabileceği sonucuna varılmıştır.

ESA tabanlı tekniklerin ve seslerin görüntüye dönüştürülmüş temsillerinin kullanıldığı görüntü işleme görevlerinde başarılı olan çalışmalardan etkilenerek ses olaylarını tanımlamak için sesin görüntü temsillerini kullanırken ESA tercih edilmesi önerilmiştir. ESA kullanmayı tercih eden çalışmada [17] literatürdeki benzer çalışmalarla benzer sonuçlara ulaşılmış ve ses tanıma görevi için ESA kullanımının uygun olduğu çıkarımı yapılmıştır.

Kısa çevresel ses kliplerinin sınıflandırılmasında ESA'nın potansiyeli değerlendirilmiş ve oluşturulan ESA modelinin MFKK'ye dayalı temel uygulamalarda daha iyi başarı gösterdiği görülmüştür ve diğer son teknoloji çalışmalarla kıyaslanabilir sonuçlar vermiştir [18]. Ayrıca veri setlerinin kısıtlı olması konusuna değinip kısıtlı veriyle ESA'nın başarılı olup olmayacağı da değerlendirilmiştir. Sonuçta ESA'nın kısıtlı veri seti ve basit veri artırmaları ile bile çevresel ses sınıflandırma görevlerinde etkili bir şekilde kullanılabileceği çıkarımı yapılmıştır.

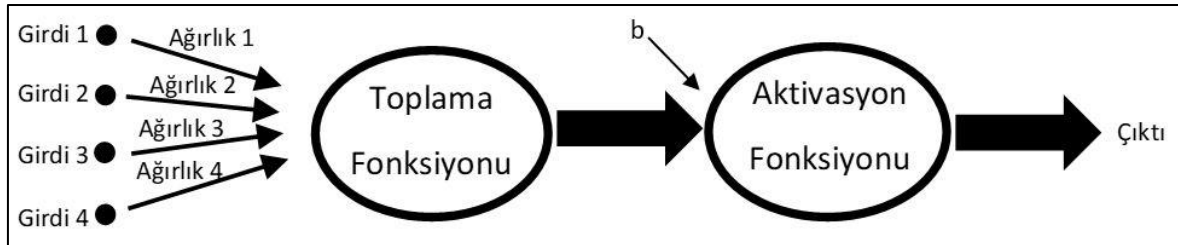
Akciğer seslerinin sınıflandırılması için ses sinyallerinin görüntü temsili olan spektrograma dönüştürüldüğü çalışmada [19] mimari olarak ESA kullanılmıştır. Çalışma görüntü işleme üzerinden yürüdüğü için veri setine görüntüye yönelik veri artırma yöntemlerinden; rastgele döndürme, kesme, renk kaydırma, Gauss filtresi ve Tuz ve Biber Gürültüsü uygulanmıştır. Artırma işlemi olmadan önce %60,14 başarı elde edilirken artırma sonrası bu oran %64,5'e ulaşmıştır.

Bu tez çalışmasına benzer isimli, 2019 yılına ait tez çalışmasında [20] uçtan-uca ses tanıma modeli tasarlanmış ve özyinelemeli sinir ağı kullanılarak Speech Commands Dataset içerisindeki temel komut kelimelerinin sınıflandırması yapılmıştır ve bu teze benzer olarak 18 adet komut için sınıflandırma yapılan deneyde elde edilen başarı ortalama %94,76 olmuştur. Ayrıca çalışmada Türkçe bir veri seti olan TURTEL kullanılmış ve yetersiz veri sebebiyle eğitimin iyi olmadığı (%16,67 sınıflandırma başarısı) görülmüştür. Tez çalışmasına benzer olarak bu tez çalışmasında yapılan 12 adet komutu tanıma işlemi %94,64 test doğruluğu elde edilmiştir. Farklı olarak eğitimin başarısız olduğu durumlarda veri artırma yöntemleri kullanılarak başarı artırılmıştır ve Türkçe eğitimi esnasında hazır bir veri seti kullanılmamış tez çalışması için özel olarak bir veri seti oluşturulmuştur.

3. DERİN ÖĞRENME MİMARİLERİ VE ARAÇLARI

Bu bölüm altında öncelikle yapay sinir ağı kavramı açıklanmış, ardından yapay sinir ağının yetersiz kaldığı alanlarda uygulama ihtiyacına göre geliştirilen ESA, Özyinelemeli Sinir Ağı (ÖSA), UKSB ve diğer derin öğrenme mimarileri anlatılmıştır. Ayrıca derin öğrenme yönteminin kullanılabilmesi için ihtiyaç duyulan temel araçlar hakkında bilgi verilmiştir.

İnsan beyninin çalışmasını sağlayan en önemli yapı biyolojik sinir ağlarıdır. Sinir ağına beş duyu organı aracılığıyla bilgiler (sinyaller) gelir. Beyin sinir sistemi sayesinde bu sinyaller beyne taşınır ve beyin verdiği kararlar (Beynin farklı bölgeleri farklı fonksiyonları gerçekleştirir ve farklı kararlar verir.) yine sinir sistemi aracılığıyla vücudun diğer organlarına eylem olarak aktarılır [21]. Duyu organları aracılığıyla, alınan sinyalleri girdi; organların gerçekleştirdiği eylemleri çıktı olarak düşündüğümüzde, girdi ile çıktı boyunca gerçekleşen tüm süreç, matematiksel benzetimler ile yapay bir sinir ağına dönüştürülebilir. Şekil 3.1’de yapay sinir ağını oluşturan sinir hücresinde gerçekleşen işlemler gösterilmiştir. Yapay sinir ağları, giriş katmanını, gizli katmanlar ve çıkış katmanından oluşur. Gizli katmanların sayısı bir veya daha fazla olabilir ve bilgilerin işlenmesi bu katmanlarda meydana gelir.



Şekil 3.1. Yapay sinir hücresinin yapısı

Şekil 3.1. incelendiğinde yapay sinir hücresine gelen bilgilerin girdi olarak ifade edildiği görülmektedir. Gelen bilgilerin önemini ve ağa olan etkisinin bir değeri olarak ağırlık kavramı kullanılmaktadır. Ağırlıklar sabit ya da değişken olabilirler. Toplama fonksiyonu ile bir hücreye gelen net girdi hesaplanır. Bu kısımda tasarımcıya bağlı olarak farklı fonksiyonlar kullanılabilir ancak en yaygın olan uygulama ağırlıklı toplamı hesaplamaktır [21]. NET ifadesi ağırlıklı toplamın çıktısını belirtmektedir ve “Eş.3.1” ile elde edilmektedir.

$$NET = \sum_i^n x_i * w_i \quad (3.1)$$

Burada x girdileri, w ağırlıkları ve n hücreye gelen toplam girdi sayısını temsil etmektedir. Burada her girdi değeri kendisine ait ağırlık değeriyle çarpılır ve tüm çarpımlar toplanır. Sonuçta ağa gelen net girdi hesaplanmış olur.

Nörona gelen net girdi hesaplandıktan sonra, bu net girdi nöronların giriş sinyallerinin sıfırdan farklı olmasını sağlayan bir b (bias-yanlılık) değeri ile toplanır ve elde edilen sonuç bir aktivasyon fonksiyonuna sokulur. Aktivasyon fonksiyonları kullanmaktaki amaç girişe karşılık çıkış değerlerini belirli bir aralıkta tutmayı veya belirli bir kurala göre çıkış üretmeyi sağlamaktır. Böylece çıkış değerlerinin kolay ve belirli kurallar dahilinde sınıflandırılması sağlanmış olur. Aktivasyon fonksiyonu net girdiyi işleyerek bir çıktı üretir ve bu fonksiyonlar da tasarımcıya bağlı olarak, deneme-yanılma yolu ile seçilir. “Eş 3.1” ile elde edilen NET değeri, bir aktivasyon fonksiyonu çeşiti olan sigmoid fonksiyonunda giriş olarak kullanıldığında oluşan çıktı değeri “Eş.3.2” ile hesaplanır.

$$S(NET) = \frac{1}{1+e^{-NET}} \quad (3.2)$$

Son aşama olarak aktivasyon fonksiyonu aracılığıyla belirlenen değer hücrenin çıktısı olmaktadır ve farklı hücrelere girdi olarak ya da dış dünyaya aktarılarak kullanılmaktadır.

Yapay sinir hücresi için verilen eşitliklerin nasıl kullanıldığının anlaşılması amacıyla 4 adet girdisi olan bir yapay sinir hücresi örneği verilecek olursa; girdi değerleri sırasıyla 0,5; 0,7; 0,2; 0,9 ve ağırlık değerleri sırasıyla -0,2; 0,5; -0,1; 0,6 ve b değerinin 0 olduğu varsayılırsa hücreye giren net bilgi “Eş.3.1” kullanılarak hesaplandığında;

$$NET = 0,5 * (-0,2) + 0,7 * 0,5 + 0,2 * (-0,1) + 0,9 * 0,6 \quad (3.3)$$

$$NET = -0,1 + 0,35 - 0,02 + 0,54 \quad (3.4)$$

$$NET = 0,77 \quad (3.5)$$

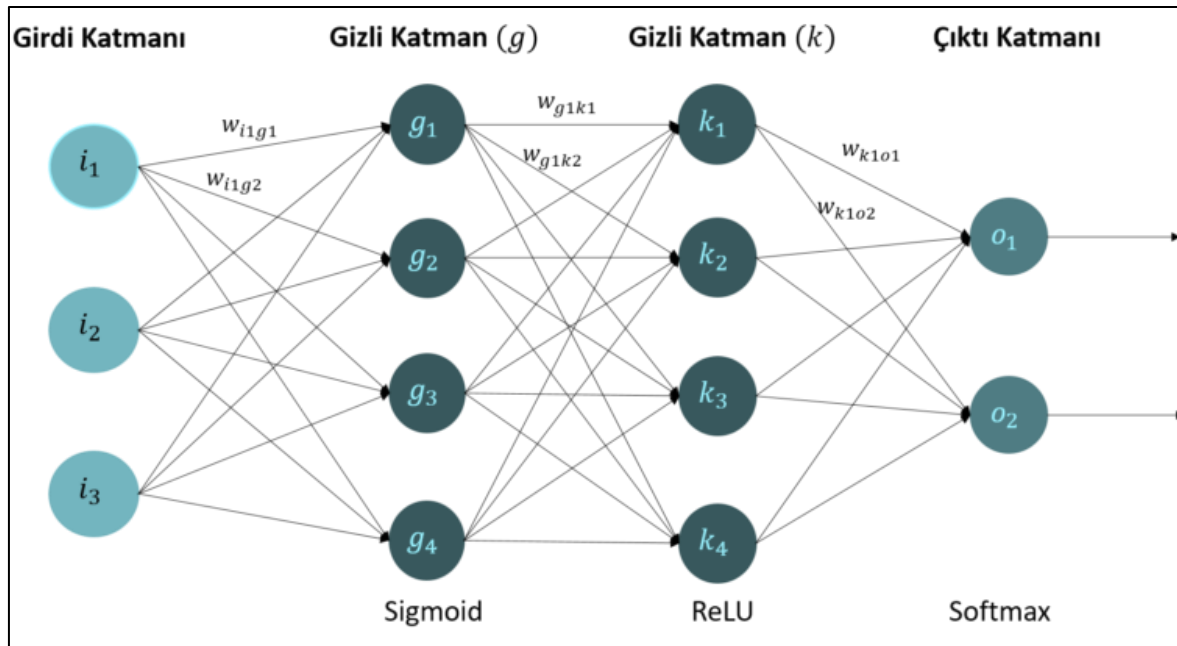
sonucu elde edilir ve elde edilen NET değeri “Eş.3.2” ile gösterilen sigmoid fonksiyonunda kullanılırsa;

$$S(0,77) = \frac{1}{1+e^{-0,77}} \quad (3.6)$$

$$S(0,77) = 0,68 \quad (3.7)$$

sonucu elde edilir. Bir sinir ağındaki bütün elemanların çıktıları örnekteki gibi hesaplandığında ağı girdilere karşılık çıktıları nasıl ürettiği görülmüş olur [21].

Yapay sinir ağlarında bulunan gizli katman sayısının ikiden fazla olması ve bu katmanlarda bulunan nöron sayısının da fazla olması durumu derin sinir ağları kavramı olarak karşımıza çıkmaktadır [22]. Şekil 3.2 iki gizli katman içeren bir yapay sinir ağının yani derin sinir ağının yapısını göstermektedir.



Şekil 3.2. Derin sinir ağının yapısı [23]

Sinir ağının çalışması ileri yayılım algoritması ve geri yayılım algoritması olmak üzere iki aşamadan oluşur [23]. Bu aşamalarda gerçekleşen olaylar Şekil.3.2. ile verilen derin sinir ağındaki gizli katman sayısı, nöron sayısı, çıktı sayısı, nöronların ait olduğu katmanı temsil eden harfler (i, g, k, o), katmanlarda kullanılan aktivasyon fonksiyonları (Sigmoid, ReLU, Softmax) ve işlemlerin hangi katmanın hangi nöronu için yapıldığını gösteren alt indisler kullanılarak anlatılmıştır. Şekil üzerindeki parametreler sırası geldikçe daha detaylı bir şekilde açıklanmıştır.

İleri yayılım algoritması:

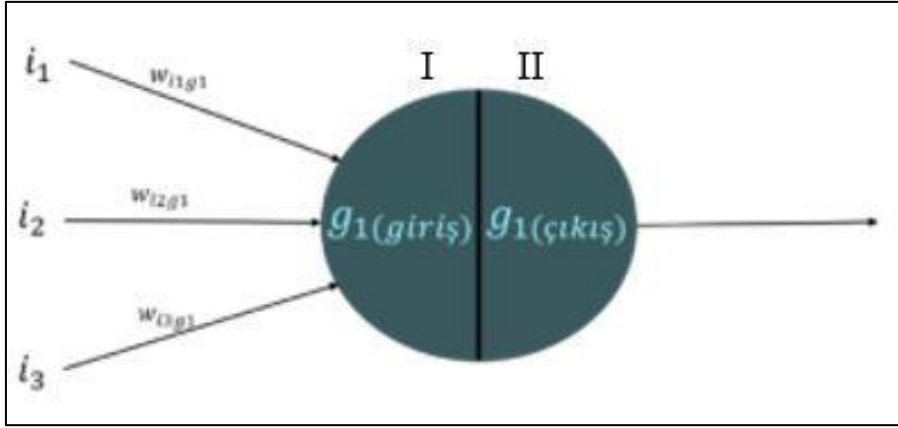
Verilerin sinir ağına gönderildiği katmana girdi katmanı denilir ve “Eş 3.8” ile verilen matris ile gösterilir.

$$[i_1 \quad i_2 \quad i_3] \quad (3.8)$$

Buradaki i değerleri girdi katmanına aitliği temsil etmektedir ve alt indisler girdi katmanının kaçınıcı nöronu olduğunu göstermektedir. Örneğin i_1 , girdi katmanının birinci nöronunu temsil etmektedir. Sinir ağındaki her bir katmanın nöronlarını birbirine bağlayan sayısal değerler mevcuttur ve bu değerlere ağırlık denir. Ağırlık değerleri, bağlı olduğu nöronun, eğitim sonunda ulaşılacak olan çıkış değerini ne denli etkileyeceğini gösterir. Birinci gizli katmana gelen ağırlıkların matris gösterimi “Eş 3.9” ile verilmiştir.

$$w_{ig} = \begin{bmatrix} w_{i_1g_1} & w_{i_1g_2} & w_{i_1g_3} & w_{i_1g_4} \\ w_{i_2g_1} & w_{i_2g_2} & w_{i_2g_3} & w_{i_2g_4} \\ w_{i_3g_1} & w_{i_3g_2} & w_{i_3g_3} & w_{i_3g_4} \end{bmatrix} \quad (3.9)$$

Eşitlikte kullanılan w , ağırlık değeridir ve alt indisler sırasıyla mevcut katman ve bir sonraki katmanı temsil eden harflerdir ve harflerin altındaki indisler de katmandaki kaçınıcı nörondan bahsedildiğini belirten sayılardır. Örneğin $w_{i_2g_3}$ ifadesi, girdi katmanının ikinci nöronundan ilk gizli katmanın üçüncü nöronuna bağlantılı ağırlık değeridir. Girdi katmanındaki nöronlara ait değerler, kendisine ait ağırlık değerleriyle çarpılır ve bu çarpımların hepsi toplanır. Gizli katmanda tasarımcı tarafından seçilmiş olan aktivasyon fonksiyonuna göre farklılık gösteren hesaplamalar yapılır ve her bir nöron için elde edilen değer bir sonraki katmanda o katmana ait ağırlık değeriyle çarpılarak ileriye aktarılır. Gizli katman içerisindeki bir nöronu Şekil 3.3 ile gösterildiği gibi iki bölüme ayırmak mümkündür.



Şekil 3.3. Gizli katman içerisindeki bir nöron [23]

Nöronun I bölümünde girdi katmanından gelen i girişleri kendilerine ait g ağırlıklarıyla çarpılır ve daha sonra tüm çarpımlar toplanır. Ayrıca toplama işlemi anında girişin sıfır olmasını engellemek amacıyla yanlılık (b) değeri de toplama eklenmektedir. II bölümünde ise $g_{1(giriş)}$ değeri aktivasyon fonksiyonuna sokulur. Elde edilen yeni değerler “Eş 3.10” ve sigmoid aktivasyon fonksiyonu kullanılan “Eş 3.11” ile gösterilmiştir.

$$g_{1(giriş)} = i_1 w_{i_1 g_1} + i_2 w_{i_2 g_1} + i_3 w_{i_3 g_1} + b_{g_1} \quad (3.10)$$

$$g_{1(çıkış)} = \frac{1}{1 + e^{-g_{1(giriş)}}} \quad (3.11)$$

Her bir katmandaki tüm nöronlar için aynı işlemler tekrarlanır. Dolayısıyla tek bir nöron için yapılan hesaplamalar matrisler kullanılarak tüm nöronlar için yapılmış olur. Bu hesaplamalar “Eş 3.12” ve “Eş 3.13” ile gösterilmiştir. Burada aktivasyon fonksiyonu olarak sigmoid kullanılmıştır.

$$\begin{bmatrix} i_1 & i_2 & i_3 \end{bmatrix} * \begin{bmatrix} w_{i_1 g_1} & w_{i_1 g_2} & w_{i_1 g_3} & w_{i_1 g_4} \\ w_{i_2 g_1} & w_{i_2 g_2} & w_{i_2 g_3} & w_{i_2 g_4} \\ w_{i_3 g_1} & w_{i_3 g_2} & w_{i_3 g_3} & w_{i_3 g_4} \end{bmatrix} + \begin{bmatrix} b_{g_1} & b_{g_2} & b_{g_3} & b_{g_4} \end{bmatrix} = \begin{bmatrix} g_{1(giriş)} & g_{2(giriş)} & g_{3(giriş)} & g_{4(giriş)} \end{bmatrix} \quad (3.12)$$

$$\begin{bmatrix} g_{1(çıkış)} & g_{2(çıkış)} & g_{3(çıkış)} & g_{4(çıkış)} \end{bmatrix} = \begin{bmatrix} \sigma_{1(giriş)} & \sigma_{2(giriş)} & \sigma_{3(giriş)} & \sigma_{4(giriş)} \end{bmatrix} \quad (3.13)$$

Verilen eşitlikte $g_{1-4(giriş)}$, I bölümünün çıkışını; $g_{1-4(çıkış)}$, II bölümünün çıkışını temsil eder ve altlarındaki sayılar katmanın kaçınıcı nöronu olduğunu göstermektedir. Ayrıca σ ,

sigmoid fonksiyonunu ve b , yanlılık değerlerini temsil etmektedir. Yanlılık değeri için de alt indisler diğer değerlerdeki ile aynı şekilde düşünülmelidir (b_{g_2} ; birinci gizli katmanın ikinci nöronuna giren yanlılık değeri).

Birinci gizli katmanda elde edilen $g_{1-4(\text{çıkış})}$ değerleri ikinci gizli katmana giriş olarak aktarılır. Birinci gizli katman için yapılan hesaplamalar indislerde ve aktivasyon fonksiyonundaki farklılık haricinde aynı şekilde gerçekleştirilir. Hesaplamalar sonucunda ikinci gizli katmandaki nöronların çıkışları “Eş 3.14” ve “Eş 3.15” ile gösterilmiştir. Burada aktivasyon fonksiyonu olarak ReLU kullanılmıştır.

$$[g_{1(\text{çıkış})} \quad g_{2(\text{çıkış})} \quad g_{3(\text{çıkış})}] * \begin{bmatrix} w_{g_1 k_1} & w_{g_1 k_2} & w_{g_1 k_3} & w_{g_1 k_4} \\ w_{g_2 k_1} & w_{g_2 k_2} & w_{g_2 k_3} & w_{g_2 k_4} \\ w_{g_3 k_1} & w_{g_3 k_2} & w_{g_3 k_3} & w_{g_3 k_4} \end{bmatrix} + [b_{k_1} \quad b_{k_2} \quad b_{k_3} \quad b_{k_4}] = [k_{1(\text{giriş})} \quad k_{2(\text{giriş})} \quad k_{3(\text{giriş})} \quad k_{4(\text{giriş})}] \quad (3.14)$$

$$[k_{1(\text{çıkış})} \quad k_{2(\text{çıkış})} \quad k_{3(\text{çıkış})} \quad k_{4(\text{çıkış})}] = [ReLU_{1(\text{giriş})} \quad ReLU_{2(\text{giriş})} \quad ReLU_{3(\text{giriş})} \quad ReLU_{4(\text{giriş})}] \quad (3.15)$$

İkinci gizli katmanda elde edilen $k_{1-4(\text{çıkış})}$ değerleri çıktı katmanına giriş olarak aktarılır ve yine aynı işlemler gerçekleşir. Çıktı katmanına ait nöronların çıkışları “Eş 3.16” ve “Eş 3.17” ile gösterilmiştir.

$$[k_{1(\text{çıkış})} \quad k_{2(\text{çıkış})} \quad k_{3(\text{çıkış})} \quad k_{4(\text{çıkış})}] * \begin{bmatrix} w_{k_1 o_1} & w_{k_1 o_2} & w_{k_1 o_3} & w_{k_1 o_4} \\ w_{k_2 o_1} & w_{k_2 o_2} & w_{k_2 o_3} & w_{k_2 o_4} \\ w_{k_3 o_1} & w_{k_3 o_2} & w_{k_3 o_3} & w_{k_3 o_4} \end{bmatrix} + [b_{o_1} \quad b_{o_2} \quad b_{o_3} \quad b_{o_4}] = [o_{1(\text{giriş})} \quad o_{2(\text{giriş})}] \quad (3.16)$$

$$[o_{1(\text{çıkış})} \quad o_{2(\text{çıkış})}] = [Softmax_{1(\text{giriş})} \quad Softmax_{2(\text{giriş})}] \quad (3.17)$$

Sınıflandırmaya yönelik çalışmalarda çıktı katmanında aktivasyon fonksiyonu olarak softmax kullanılır. Bu katmanda softmax fonksiyonu seçmenin amacı, softmax fonksiyonunun giriş değerlerini 0 ile 1 arasına çekmesi ve sonuçta bu değerlerin toplamının 1 olacak şekilde ayarlamasıdır. Bu sayede çıktı katmanında her bir sınıf için olasılık değeri elde edilmiş olur. Böylece softmax fonksiyonu girdilere karşılık en son çıktının ne olacağı tahmin ederek doğru olma olasılığını sunmuş olur.

Geri yayılım algoritması:

İleri yayılım algoritması ile derin sinir ağı tahmin değerleri oluşturmuş olmaktadır. Bu tahmin değerleri kesin olarak doğru olmaz ve gerçek değer ile tahmini değer arasında bir hata değeri ortaya çıkar. Bu hata değerinin mümkün olduğunca küçültülmesi gerekir. Bu küçültmeyi sağlamak için geri yayılım algoritması kullanılır [23].

Geri yayılım algoritmasının başlayabilmesi için gerçek ve tahmini değer arasındaki hata değerinin hesaplanması gerekir. Çıktı katmanından girdi katmanına doğru ilerleyen türev işlemleri ile geriye doğru gidilir. Kareler ortalaması hatası olarak bilinen eşitlik kullanılarak hata değerini hesaplamak için “Eş 3.18” kullanılır.

$$E = \frac{1}{2n} \sum_{k=1}^K (y_k - o_{k(\text{çıkış})})^2 \quad (3.18)$$

“Eş 3.18” de E , hata değerini; n , veri setinin eğitim için ayrılan kısmının içerdiği veri sayısını; k çıktı katmanındaki nöron sayılarını; y_k , gerçek çıkış değerini göstermektedir ve Şekil.3.2 ile verilen derin sinir ağına göre hesaplama yapıldığından burada $K=2$ olur. Ayrıca çıkışta da iki sınıf olduğu için E_1 ve E_2 olarak iki farklı hata değeri hesaplanması gerekir.

Hata değeri $o_{k(\text{çıkış})}$ değerine bağlıdır ve $o_{k(\text{çıkış})}$ değerleri de nöronlara ait ağırlıklara bağlıdır. Dolayısıyla geriye yayılım algoritmasının amacına ulaşmak için hatayı küçültmeye çalışırken hangi ağırlık değerlerinin hatayı en küçük hale getirdiği tespit edilir. Ağırlık değerlerinin değişim miktarı “Eş 3.19” ile hesaplanır.

$$\Delta W = -n * \frac{\partial E}{\partial W} \quad (3.19)$$

Değişim miktarının hesaplanabilmesi için öncelikle $\frac{\partial E}{\partial W}$ değerinin hesaplanmasına ihtiyaç vardır. Bu hesap için hata değerinin katmanlardan geriye doğru gidecek şekilde kısmi türevi alınır. Çıktı katmanından ikinci gizli katmana doğru olan hata değeri bulma işlemi “Eş 3.20” ile verilmiştir.

$$\begin{bmatrix} \frac{\partial E_1}{\partial o_{1(\text{çıkış})}} \\ \frac{\partial E_2}{\partial o_{2(\text{çıkış})}} \end{bmatrix} = \begin{bmatrix} o_{1(\text{çıkış})} - y_1 \\ o_{2(\text{çıkış})} - y_2 \end{bmatrix} \quad (3.20)$$

Çıktı katmanındaki $o_{k(çıkış)}$ değerleri $o_{k(giriş)}$ değerlerine göre kısmi türevinin alınması “Eş 3.21” ile gösterilmiştir.

$$\begin{bmatrix} \frac{\partial o_{1(çıkış)}}{\partial o_{1(giriş)}} \\ \frac{\partial o_{2(çıkış)}}{\partial o_{2(giriş)}} \end{bmatrix} = \begin{bmatrix} \frac{e^{o_{1(giriş)}} * [e^{o_{1(giriş)}} + e^{o_{2(giriş)}}]}{[e^{o_{1(giriş)}} + e^{o_{1(giriş)}}]^2} \\ \frac{e^{o_{2(giriş)}} * [e^{o_{1(giriş)}} + e^{o_{2(giriş)}}]}{[e^{o_{2(giriş)}} + e^{o_{2(giriş)}}]^2} \end{bmatrix} \quad (3.21)$$

Çıktı katmanındaki $o_{k(çıkış)}$ değerlerinin bir önceki katmanla arasındaki ağırlık değerleriyle (W_{ko}) kısmi türevinin alınması birinci çıkış için “Eş 3.22” ile ikinci çıkış için “Eş 3.23” ile gösterilmiştir.

$$\begin{bmatrix} \frac{\partial o_{1(giriş)}}{\partial W_{k_1 o_1}} \\ \vdots \\ \frac{\partial o_{1(giriş)}}{\partial W_{k_4 o_1}} \end{bmatrix} = \begin{bmatrix} k_{1(çıkış)} \\ \vdots \\ k_{4(çıkış)} \end{bmatrix} \quad (3.22)$$

$$\begin{bmatrix} \frac{\partial o_{2(giriş)}}{\partial W_{k_1 o_2}} \\ \vdots \\ \frac{\partial o_{2(giriş)}}{\partial W_{k_4 o_2}} \end{bmatrix} = \begin{bmatrix} k_{1(çıkış)} \\ \vdots \\ k_{4(çıkış)} \end{bmatrix} \quad (3.23)$$

Buraya kadar anlatılan kısmi türev işlemleri zincir kuralına sokulduğunda katmanlardaki ağırlıkları güncellenmiş hali (W^{yeni}) “Eş 3.24” ile gösterilmiştir.

$$W_{ko}^{yeni} = \begin{bmatrix} W_{k_1 o_1} - n * \frac{\partial E_1}{\partial W_{k_1 o_1}} & W_{k_1 o_2} - n * \frac{\partial E_2}{\partial W_{k_1 o_2}} \\ \vdots & \vdots \\ W_{k_4 o_1} - n * \frac{\partial E_1}{\partial W_{k_4 o_1}} & W_{k_4 o_2} - n * \frac{\partial E_2}{\partial W_{k_4 o_2}} \end{bmatrix} \quad (3.24)$$

“Eş 3.24” deki güncelleme değerleri çıktı katmanı ile ikinci gizli katman arasındaki ağırlıklar için yapılmıştır ve alt indisler önceki eşitliklerde kullanıldığı gibi kullanılmıştır. Yani buradaki W_{ko}^{yeni} değeri çıktı katmanından ikinci gizli katmana giden güncellenmiş ağırlığı ifade etmektedir.

Şimdiye kadar yapılan hesaplamalar diğer katmanlar için de yapılır ve her ağırlık değeri güncellenerek girdi katmanına kadar ulaşılır. Ağırlıkların güncellenmesi işlemi hatanın en küçük olduğu karara varana dek devam eder ve böylece geri yayılım süreci sona erer.

Yapay sinir ağı çalışmaları, gelişen teknolojiyle birlikte sistemlerin öğrenme, karar verme, akıl yürütme gibi işlevleri gerçekleştirmesi için mevcut olan yapay/derin sinir ağı modellerinin yetersiz kalmasıyla farklı bakış açılarının ortaya çıkmasını sağlamıştır. 1998’de Yann LeCun ve arkadaşları tarafından LeNet mimarisi ile evrişimsel sinir ağlarının kullanılması ve 2006 yılında Hinton ve Salakhutdinov’ın sınırlı Boltzman makinesi yaklaşımını kullanması, mevcut yapay sinir ağlarının yetersiz kaldığı durumlar için derin öğrenme konusunun popüler hale gelmesini sağlamıştır [24]. Derin öğrenme; özellik çıkarma, dönüştürme, model analizi ve sınıflandırması için çok sayıda doğrusal olmayan bilgi işleme katmanı kullanan makine öğrenme sınıfıdır [25]. Derin öğrenme yöntemleri kullanılarak ham verilerden verimli bir üst seviye soyutlama yapılır, otomatik olarak öznitelik dizileri oluşturulur ve böylece genelde insanlar tarafından ortaya çıkarılan öznitelikler otomatik olarak çıkartılıp kullanıma hazır hale gelir [26, 27].

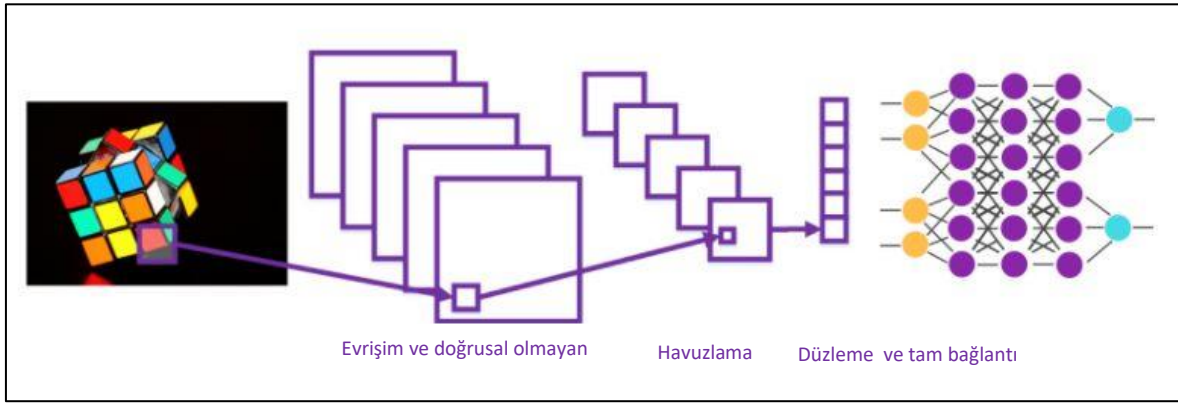
Derin öğrenme algoritmaları çok sayıda katmana sahip olduklarından dolayı işlem yükleri de fazladır ve artan iş yükü eğitim sürecini olumsuz etkilemektedir. Bundan dolayı farklı uygulama alanlarında farklı mimariler tercih edilmektedir. Doğru bir mimari tercihi sonucunda yapılacak uygulamaya göre en iyi performansı elde etmek mümkün hale gelir. Bölüm 3.1-3.3 başlıkları altında literatürde yaygın olarak kullanılan ESA, ÖSA ve UKSB mimarileri anlatılmıştır ve Bölüm 3.4 başlığı altında da diğer mimarilere kısaca değinilmiştir. Bu tez çalışmasında ESA mimarisi tercih edilmiştir ve bu tercihin sebebi 6. Bölümde anlatılmıştır.

3.1. Evrişimli Sinir Ağı (ESA)

Bir nesneyi diğerlerinden farklı kılan özellikleri tespit etmek konusunda yapay sinir ağlarının yetersiz kaldığı görülmüştür ve bu yetersizliğe çözüm olarak ESA ortaya çıkmıştır. Dolayısıyla ESA, herhangi bir nesneyi diğerlerinden farklı kılan özelliklerinin tespit edilmesi prensibiyle çalışan yapay sinir ağı şeklinde tanımlanabilir. Bir görüntüde kenar bulmak gibi temel bir özellik veya daha soyut özelliklerin yardımıyla sınıflandırma yapar. Ses verilerinden elde edilen MFKK özneliği bir görüntü olarak işlenebilmektedir. MFKK

yöntemi Bölüm 4.1’de anlatılmıştır ve seslerden elde edilen MFKK değerleri tasarlanan sinir ağlarında giriş verisi olarak kullanılmıştır.

ESA farklı görevler için farklı katmanlara sahiptir. Bu katmanlar evrişim katmanı, doğrusal olmayan katman, havuzlama (pooling) katmanı, düzleme (flatten) katmanı, tam bağlantı katmanıdır ve Şekil 3.4 ile görselleştirilmiştir [28].



Şekil 3.4. Evrişimli sinir ağının katmanları [28]

3.1.1. Evrişim katmanı

Evrişim katmanında giriş verisine farklı filtreler uygulanır ve girdinin özelliklerinin yakalanması sağlanır. Kullanılan her bir filtre farklı bir özneliği yakalar. Şekil 3.5’de 5x5 boyutlu giriş verisi, 3x3 boyutlu girişe uygulanacak filtre ve sonuçta oluşan 3x3 boyutlu çıkış matrisi örnek olarak gösterilmiştir. Buradaki evrişim işleminde filtre matrisi girişin sol üst köşesine yerleştirilir ve çakışan sayılar birbirleriyle çarpılır, ardından çarpımların hepsi toplanır ve çıkış matrisinin birinci kısmına yazılır. Bu işlem filtre matrisi birer adım (sağa ve satır bitince aşağıya olacak şekilde) kaydırılarak yinelenir, sonuçların hepsi çıkışta ilgili kısma yazılır. Bu evrişim için bir yöntemdir ancak farklı yöntemlerde kullanılabilir. Örnekten anlaşılacağı üzere evrişim katmanında girişe bir filtre uygulanarak ve matris çarpımı yapılarak bir öznelik belirlenmiş olur. Birden fazla katman dolayısıyla aynı sayıda filtre kullanılarak birden fazla öznelik çıkarmak mümkündür [28].

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

1	0	1
0	1	0
1	0	1

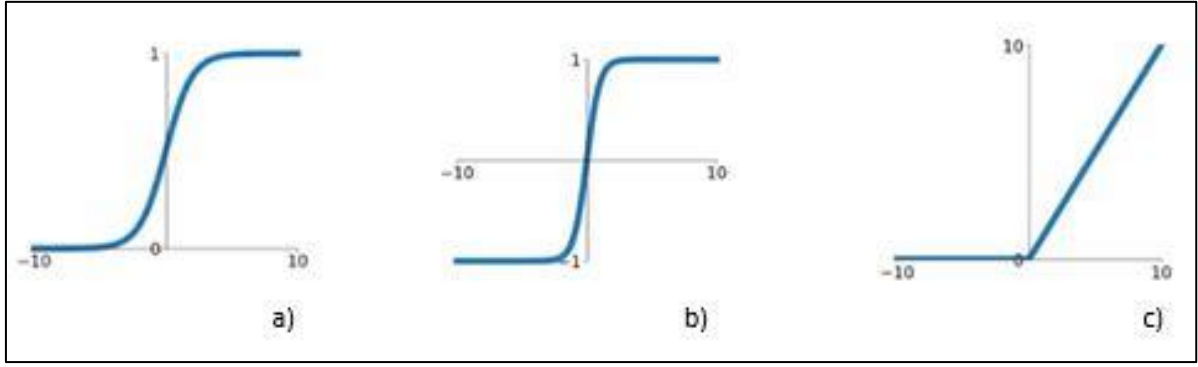
4	3	4
2	4	3
2	3	4

Şekil 3.5. Evrişim işleminde kullanılan giriş matrisi, filtre ve çıkış matrisi örneği

Şekil 3.5’de giriş matrisi 5x5 iken çıkışta 3x3’lük bir matris elde edildiği görülmektedir. Çıkış matrisinin boyutu $çıkış = giriş - filtre + 1$ şeklinde hesaplanabilir. Eğer giriş ile aynı boyutta bir çıkış elde edilmek istenirse piksel (ESA mimarisinde giriş olarak görüntü kullanıldığı için adım yerine genellikle piksel ifadesi tercih edilmektedir.) ekleme (padding) yöntemine başvurulabilir. Bu yöntemle giriş matrisinin çevresine hayali sıfırlar veya girişteki değerler eklenebilir. Girişteki değerlerden ekleme yapıldığında çıkış matrisinin ilk haliyle arasındaki fark çok büyük olmaz ancak her adımda işlem yapılacağı için hesaplama yavaş olur. Evrişim sürecinde kullanılan bir nitelik de kaydırmadır (stride). Kaydırma, filtrenin kaç adım aralıklarla kaydırılacağı bilgisidir. Kaydırma miktarını değiştirerek de çıkış boyutunda değişiklik elde etmek mümkündür. Ancak adım aralığı arttıkça işlemler hızlanırken hassasiyet azalır ve bilgi kaybı artar [28].

3.1.2. Doğrusal olmayan katman

Evrişim katmanının ardından doğrusal olmayan katman gelmektedir. Bu katman aktivasyon fonksiyonlarının kullanıldığı kısım olduğundan dolayı aktivasyon katmanı olarak da isimlendirilmektedir. Doğrusal olmayan sigmoid, hiperbolik tanjant (tanh) ve Rectified Linear Unit (ReLU) en çok kullanılan aktivasyon fonksiyonlarıdır. Problemlere uygun olan doğru aktivasyon fonksiyonu seçimi ile başarı doğru orantılıdır. Eğitim hızı bakımından en iyi sonuçları verdiği için ReLU en çok tercih edilen fonksiyondur [28]. Şekil 3.6.a sigmoid, Şekil 3.6.b tanh ve Şekil 3.6.c ReLU fonksiyonlarına ait grafikleri göstermektedir.



Şekil.3.6. a) Sigmoid b) tanh c) ReLU [29]

Verilen grafiklerde yatay eksen giriş değerlerini dikey eksen çıkış değerlerini göstermektedir. Buna göre sigmoid fonksiyonu her giriş elemanına karşılık sıfır ile bir arasında çıkış verirken tanh fonksiyonu -1 ile 1 arasında çıkış verir ve ReLU fonksiyonu sıfırdan küçük girişlerde sıfır çıkışını verirken sıfırdan büyük girişler için çıkış, giriş ile aynı değere sahiptir. Eş.3.8 Sigmoid fonksiyonun matematiksel eşitliğini, Eş.3.9 tanh fonksiyonun matematiksel eşitliğini ve Eş.3.10 ReLU fonksiyonun matematiksel eşitliğini göstermektedir.

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (3.23)$$

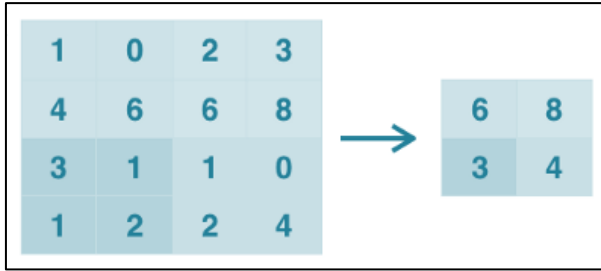
$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.24)$$

$$\text{relu}(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (3.25)$$

3.1.3. Havuzlama katmanı

Peş peşe gelen evrişim katmanları arasında sıklıkla havuzlama katmanı kullanılır [28]. Burada amaç matris boyutlarını küçültmektir ve böylece ağıdaki hesaplama yükünü azaltmaktır. Maksimum havuzlama, ortalama havuzlama, minimum havuzlama ve L2 havuzlama gibi çok sayıda yöntem kullanılabilir. En çok tercih edilen yöntem maksimum havuzlamadır.

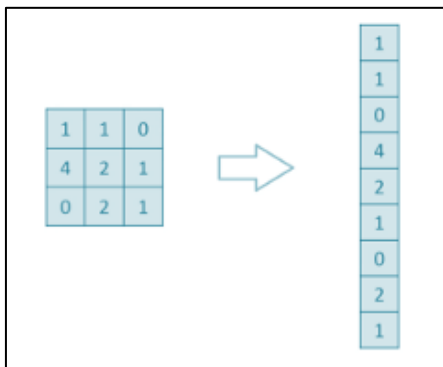
Şekil 3.7’de verilmiş olan maksimum havuzlama örneğinde 4x4 boyutunda bir giriş için 2x2’lik bir filtre uygulanmıştır. Bu filtre giriş matrisi üzerine yerleştirildiğinde girişte çakıştığı değerlerden en büyük olanı seçer ve çıkış için o değeri alır. Yani girişteki her dört değerden en büyük olan bir değeri alıp çıkışa iletir. Sonuçta çıkış matrisi 2x2 boyutunda olur. Bölüm 6.1’de ESA mimarisi oluşturulurken de maksimum havuzlama kullanılmıştır.



Şekil 3.7. Maksimum havuzlama işlemi [28]

3.1.4. Düzleme ve tam bağlantı katmanı

Evrişim ve havuzlama katmanlarından gelen çok boyutlu dizilerin tam bağlantı katmanına giriş olarak verilebilmesi için tek boyuta dönüştürülmesi gerekir ve bu işlem düzleme (flatten) katmanında gerçekleşir. Şekil 3.8’de düzleme katmanı için bir örnek verilmiştir. Düzleme katmanı aracılığıyla tek boyuta indirgenmiş dizi, son katman olan tam bağlantı katmanı için giriş verisi olmaya hazır hale gelmiştir. Tam bağlantı katmanına gelen veriler artık işlenmeye hazırdır ve bu tam bağlantı katmanı sonucunda uygulamanın sonunda kaç adet sınıfa ihtiyaç varsa o kadar sayıda çıkış elde edilir. Böylece bütün giriş verileri sınıflandırılarak öğrenme süreci tamamlanır.



Şekil 3.8. Düzleme katmanından tam bağlantı katmanına geçiş örneği [28]

Bir ESA mimarisi oluşturmak için giriş verisi, birkaç evrişim katmanı, aktivasyon (özellikle ReLu) katmanı, havuzlama katmanı, düzleme katmanı ve aktivasyon katmanı miktarında tam bağlantı katmanına ihtiyaç duyulmaktadır [28]. ESA mimarisinin oluşturulması için kullanılan aktivasyon fonksiyonu, katman sayısı ve çeşitleri gibi tasarımcıya bağlı özellikler Bölüm 6.2’de mimari seçimi deneyleri anlatılırken verilmiştir.

3.2. Özyinelenebilir Sinir Ağı (ÖSA)

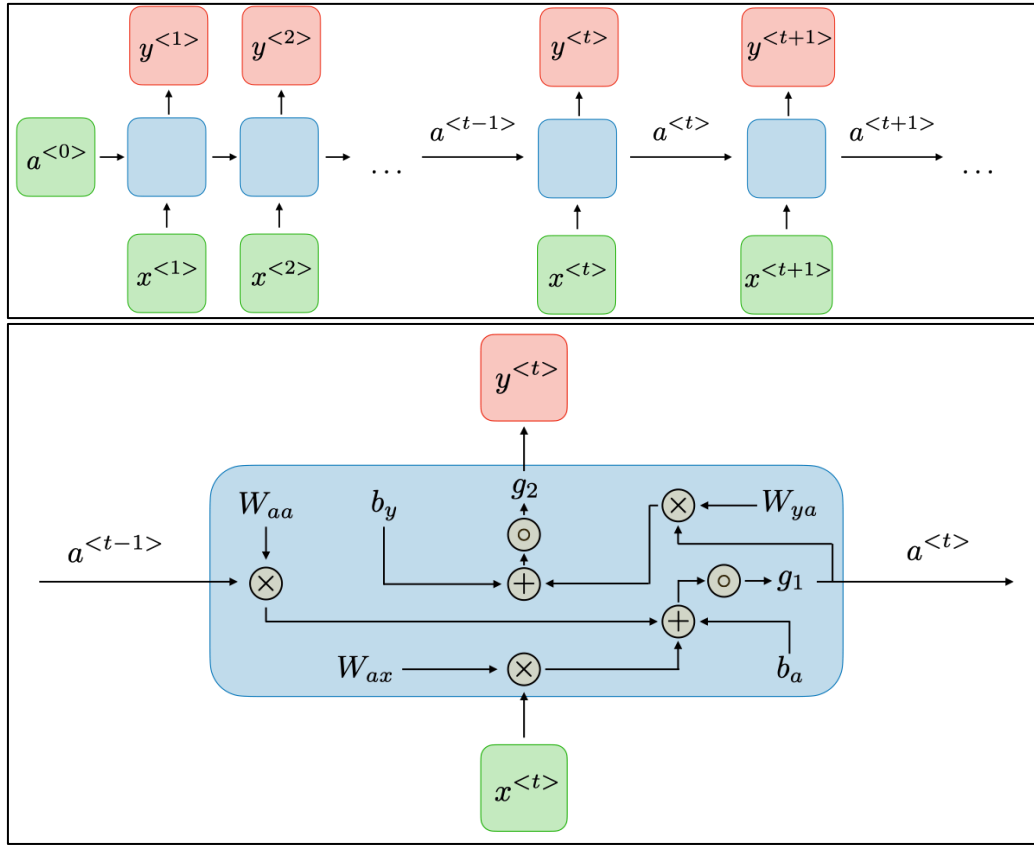
Tahmin yapmak için önceki bilgiye hatta bazı durumlarda sonraki bilgiye de ihtiyaç duyulan zaman serisi şeklinde ifade edilen problemlerde ESA kullanılamamaktadır. Dolayısıyla bu tarz problemlerin çözümünde ÖSA tercih edilir [30]. ÖSA mimarisinde öncelikle tahmin yapmak için Bölüm 3’de anlatılan ileri yayılım algoritması kullanılır. Ardından tahminin gerçek değerle arasındaki farkın hesaplandığı kayıp (loss) fonksiyonu yardımıyla bir hata değeri elde edilir. Son olarak elde edilen hata değeri ile her düğüm için bir gradyan değerinin (en uygun ağırlık değeri) hesaplandığı geri yayılım algoritması kullanılır.

Her bir t anında aktivasyon fonksiyonu çıktısı $a^{<t>}$ “Eş. 3.26” ve nöron çıktısı $y^{<t>}$ “Eş. 3.27” ile verilmiştir.

$$a^{<t>} = g_1(W_{aa} * a^{<t-1>} + W_{ax} * x^{<t>} + b_a) \quad (3.26)$$

$$y^{<t>} = g_2(W_{ya} * a^{<t>} + b_y) \quad (3.27)$$

W_{aa} , W_{ax} ve W_{ya} nöronlara ait ağırlık değerlerini; b_a , b_y yanlılık değerlerini ve g_1 ile g_2 aktivasyon fonksiyonlarını temsil eder. Şekil 3.9.a ve Şekil 3.9.b ÖSA mimarisinin çalışmasını gösteren blok diyagramlardır [30].



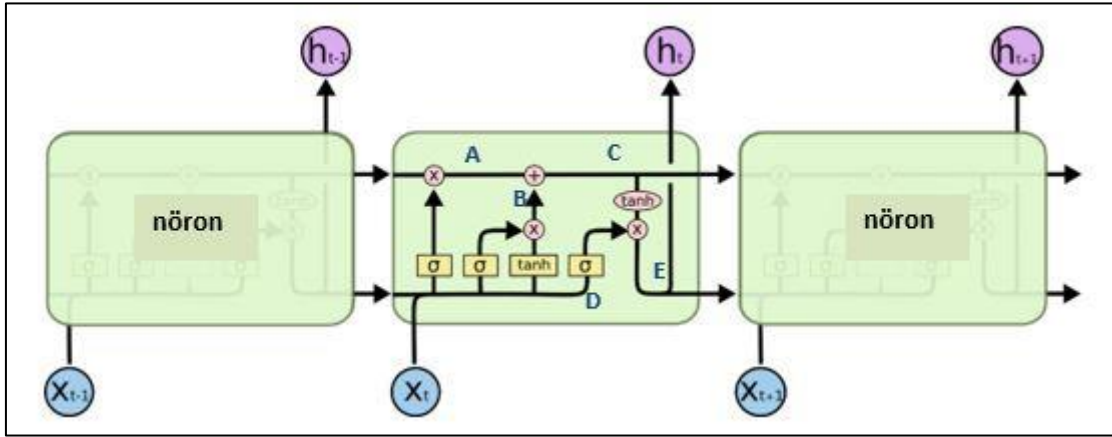
Şekil 3.9. ÖSA çalışma diyagramı [30]

İleri yayılım sürecinde yapılan çarpma ve toplama işlemleri geriye yayılımda türev olarak işlenir. Kullanılan aktivasyon fonksiyonun türevinin sıfır olduğu noktalarda öğrenme gerçekleşmez. Öğrenmenin gerçekleşmediği kısımlarda “gradyanların yok olması” problemi meydana gelir. Bu problemi çözmek için Geçitlenmiş Özyinelemeli Birimler (GRU), Çift Yönlü ÖSA, Deep-ÖSA ve UKSB mimarileri kullanılabilmektedir. Bu tez çalışmasında ÖSA mimarisi değil; güçlü, genellenebilir olması ve gradyanların yok olması problemi yaşamamasından dolayı özel bir ÖSA türü olan UKSB de mimarisi araştırılmıştır.

3.3. Uzun-Kısa Süreli Bellek (UKSB)

Ses verilerinde sinyalin sabit bir değeri yoktur. Sinyalin sabit olmayan bir yapısı olduğundan sinyaldeki farklılıklar hem zaman hem de frekans alanında yakalanmalıdır. Zamanın etkisi, bir sonraki oluşacak bilginin bir önceki bilgiye ihtiyaç duyması yani birbirine bağımlı bilgilerin varolması şeklinde ortaya çıkmaktadır. UKSB ağları çıkışı bulmak için uzun zaman aralıklarında giriş verisi yardımıyla bağımlılıkları öğrenebilen özel bir ÖSA türüdür. UKSB hafıza blokları denilen alt ağ dizilerini içerir.

UKSB ağındaki işlemler unutmaya kapısı, girdi kapısı, hücre geçiti ve çıkış kapısı aracılığıyla gerçekleşmektedir. Şekil 3.10 UKSB' nin çalışma mantığının anlaşılması için ileride anlatılanlar ile eş zamanlı olarak takip edilmelidir. Şekil 3.10'da X_t , t anındaki girişi, h_t , t anındaki gizli katmandaki mevcut durumu ifade eder. Nöron içerisindeki unutmaya kapısı, önceki adımlarda hangi bilginin unutulmasının eğitime olumlu etki edeceğine karar verir. Unutmaya kapısında sigmoid fonksiyonu kullanılır ve önceki katmandan gelen ağırlıklar 0'a ne kadar yakınsa o kadar unutulur ve 1'e ne kadar yakınsa o kadar hafızada tutulur (Hafızada tutulan bilgiler A ile gösterilmiştir.). Hücre geçitini güncellemek için girdi kapısı kullanılır. Burada da ilgili değerler sigmoid fonksiyonuna sokulur ve yine girişler tanh fonksiyonuna sokulur. Aktivasyon fonksiyonlarına sokulan bu iki değerın çıkışları çarpılır (Çarpım sonucu B ile gösterilmiştir.). Hücre geçiti, bilgiyi taşıma görevini üstlenir. Taşınması gereken veriyi alır ve hücre sonuna ardından da diğer hücrelere aktarır ve bu esnada $A+B$ işlemi gerçekleşir (İşlem sonucu C ile gösterilmiştir.). Çıktı kapısı bir sonraki katmana gönderilecek değere karar verir ve bu değer tahmin için kullanılır. Önceki değer ile şimdiki girdi sigmoide girer (D ile gösterilmiştir.) ve tanh dan geçen C değeri ile çarpılır (Çarpım sonucu E ile gösterilmiştir.). Bu toplamda bir önceki değer olarak sonraki katmanın nöronuna iletilir.



Şekil 3.10. UKSB'nin çalışma yapısı

UKSB'nin yapısı ve özellikleri incelendiğinde önceki bilgilere ihtiyaç duyulan; konuşulanı yazıya dökme, müzik üretme, duygu analizi ve doğal dil işleme gibi özellikle sesli verilerin kullanıldığı uygulamalarda tercih edildiği görülmektedir. UKSB'nin mevcut özelliklerinden dolayı sesli komutları tanıma işleminde başarılı olabileceği öngörülmüştür ve bu sebeple tez kapsamında UKSB mimarisinin başarısı değerlendirilmiştir.

3.4. Diğer Mimariler

Derin Sinir Ağları (DSA):

DSA, gizli katman sayısı ikiden fazla olan ve doğrusal olmayan karmaşık bağlantıların belirlenmesini sağlayan sinir ağlarıdır [31]. Özellikle sınıflandırma ve regresyon işlemlerinde tercih edilirler ve başarılı sonuçlar verirler. Öğrenme süreçlerinin yavaş olması gibi bir dezavantaja sahiptirler [26, 27].

Derin Oto-kodlayıcılar (DOK):

Oto-kodlayıcılar (OK), giriş verisinin sıkıştırılmış olarak gösteriminden en uygun özelliklerin belirlenmesini sağlayan ileri beslemeli bir sinir ağıdır [32,33]. DOK ise her katman çıkışının sonraki katmana giriş olarak bağlandığı OK'lerin çok katmanlı halidir [32, 34]. Öğrenme sürecinde etiketli veriye ihtiyaç duymaz [26].

Derin İnanç Ağları (DİA):

DİA, hem gözetimli hem de gözetimsiz olarak kullanılabilen ama eğitim sürecinin çok yavaş olduğu bir sinir ağıdır [26]. Görüntü tanıma, hareket yakalama ve konuşma tanıma gibi alanlarda kullanılabilmektedir.

Derin Boltzmann Makinesi (DBM):

DBM'de DİA'dan farklı olarak ilk 2 katman haricinde bütün katmanlarda yönsüz bağlantılar mevcuttur. Bundan dolayı hesaplama açısından DİA'ya göre daha maliyetlidir [26].

3.5. Derin Öğrenme Araçları

Derin öğrenme sürecinin uygulanması için ihtiyaç duyulan birçok araç vardır. Bu ihtiyaçlardan ilki yazılımların kolay bir şekilde yapılabilmesi ve kütüphaneleri rahatça kullanarak uygulamaların geliştirilmesi için hazırlanmış “Integrated Development Environment (IDE)” yazılım ortamıdır [35]. Bu çalışma boyunca IDE olarak “Spyder” tercih edilmiştir. Spyder'ın tercih edilmesinin sebepleri arasında kullanımının kolay olması, hata yapılan yerlerde hatayı sebebiyle beraber ve renkli olarak gösteriyor olması, yazılan fonksiyonları otomatik olarak tamamlıyor olması gösterilebilir. Spyder, Python ile yazılmış olan ve Python geliştirmek için kullanılan ücretsiz ve açık kaynaklı bir IDE'dir [36]. Spyder'ı kullanmak için veri bilimi, yapay zeka ve buna benzer diğer bilimsel çalışmalarda

kullanılmak üzere hazırlanmış tümleşik bir Python dağıtımı olan “Anaconda Navigator” gereklidir. Anaconda içerisinde Spyder gibi araçların yanı sıra birçok kütüphane de mevcuttur [37]. Anaconda’nın bilgisayara kurulumu ile beraber Spyder da kurulmuş olacaktır ve gerekli kütüphanelerin de yüklenmesiyle çalışmak için uygun ortam hazır hale gelmiş olur. TensorFlow, keras, librosa, numpy ve matplotlib kütüphaneleri tez çalışması sürecinde önemli görevleri yerine getirmek için kullanılmıştır.

TensorFlow:

Google tarafından geliştirilen TensorFlow, veri akış grafikleri yardımıyla sayısal hesaplamalar için kullanılan bir derin öğrenme kütüphanesidir [38]. Bu kütüphane yapay sinir ağları oluşturmak için kullanılabilmektedir. Esnek bir yapıya sahip olduğu için birçok farklı platformda kullanım için uygundur [38]. Ses ve görsel tanıma, video algılama, yazı tabanlı uygulamalar ve zaman serisi içeren uygulamalarda TensorFlow kullanılabilmektedir.

Keras:

Keras, TensorFlow üzerine inşa edilmiş bir pakettir [39]. Girişten sonuca kadar mümkün olduğunca az gecikme yaşanması hedeflenerek geliştirilmiş bir kütüphanedir. Basit ve hızlı bir şekilde modellenme imkanı sağlaması, evrimsel ve yinelenebilir ağları hatta ikisinin kombinasyonu şeklinde olan ağları desteklemesi, CPU ve GPU üzerinde kusursuz çalışması Keras’ın avantajlarından [14].

Bu tez çalışmasının en önemli kısımlarından birisi olan sinir ağı modelleri inşa edilirken katmanlar Keras kullanılarak oluşturulmuştur. Model oluşturulurken evrişim katmanı, normalizasyon katmanı, seyreltme katmanı, erken durdurma katmanı, çıkış katmanı ve ağ için gerekli olan diğer katmanlar Keras aracılığıyla kolayca eklenmiştir.

Librosa:

Müzik ve ses analizi yapmak için kullanılan Librosa; seslerin kaydedilmesi, görüntülenmesi, dalga formunun çıkartılması, Fourier analizi aracılığıyla spektrogramlarının elde edilmesi ve özniteliklerinin çıkartılması gibi önemli birçok işlemi gerçekleştiren bir kütüphanedir [40]. Bu tez çalışmasında "librosa.load" ile veri setinin bulunduğu konum belirtilerek ses dosyalarının programa aktarılması sağlanmıştır ve “librosa.feature.mfcc” ile seslerin MFKK öznitelikleri çıkartılmıştır.

Numpy:

MFKK öznitelikleri çok sayıda katsayı ile ifade edilir. Bu katsayılar matematiksel işlemlere tâbi tutulduğu için belirli bir düzen içerisinde olmalıdır. Bu düzen katsayıların dizi şeklinde ifade edilmesi ile sağlanabilmektedir. Ayrıca bir sese ait MFKK'nin kendisine karşılık gelen bir etikete de ait olması gerekir. Numpy kütüphanesi ile MFKK öznitelikleri ve ses komutlarına ait etiketleri diziler halinde ifade edilmiştir. Bu durum yazılımda birden fazla dizi oluşturulması ihtiyacı doğurmuştur. Diziler arasında boyut uyumu sağlanması gerektiği durumlarda da numpy kütüphanesi kullanılmıştır.

Matplotlib:

Eğitim sonrasında verilerin görselleştirilmesini ve böylece daha anlaşılır sonuçlar elde edilmesi matplotlib kütüphanesiyle sağlanmıştır. Bu tez çalışmasında oluşturulan mimarilerin sonuçlarının karşılaştırılmalarının rahatça yapılması için eğitim, doğrulama ve test için epoch sayılarına göre doğruluk oranları ve hata oranlarına ait grafikler “matplotlib.pyplot” kütüphanesiyle oluşturulmuştur.

4. SES SINIFLANDIRMA VE ÖZNİTELİK ÇIKARMA

Bu başlık altında sesli komutların eğitime giriş verisi olarak gönderilebilmesi için uygun hale getirilmesi amacıyla sesin bir öz niteliği olan MFKK'sinin elde edilme işleminin nasıl gerçekleştirildiği anlatılmıştır.

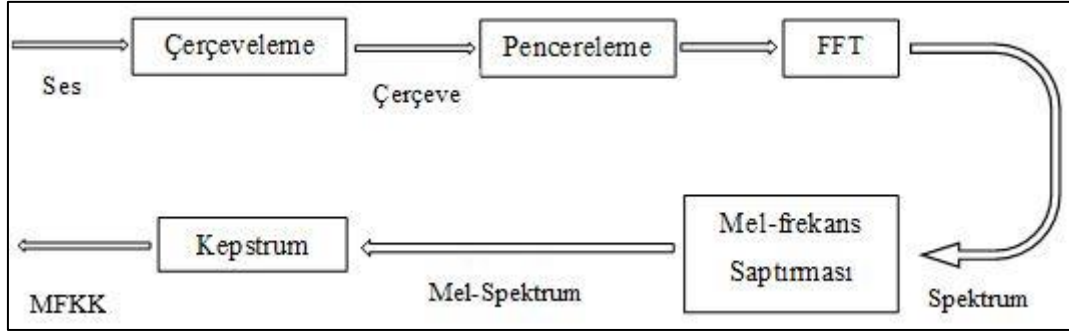
Ses tanıma; herhangi bir ses kaynağından gelen ses sinyallerinin karakteristik özelliklerinin çözümlenmesi işlemidir. Ses, dalgalar halinde yayılır ve ses dalgalarının genlik, frekans gibi ayırt edici parametreleri vardır. Sesin genlik, frekans veya diğer özellikleri arasındaki farklılıklar sayesinde sesler tanımlanır ve sınıflandırılır. Örneğin belirli bir frekansın altındaki sesler erkek sesi, üstündeki sesler kadın sesi olarak etiketlenip sınıflanlandırılabilir.

Ses tanıma sürecine, seslerin dijital ortamda kaydedilmesi veya hazır olarak kaydedilmiş, etiketlenmiş ve literatürde kullanıma sunulmuş veri setlerinin temin edilmesiyle başlanır. Yapılacak çalışmaya uygun olan ses öz nitelikleri çıkarılarak ham ses verilerinin kullanıma hazır hale getirilmesi sağlanır. Çıkarılan öz niteliklerin benzerliklerine göre sesler sınıflara ayrıştırılır.

Ses tabanlı uygulamalarda sesin öz niteliklerini çıkarmak amacıyla kullanılan Spektrogram, CYG, MFKK gibi birçok yöntem vardır. Bu çalışmada literatürde en çok tercih edilen yöntem olan MFKK kullanılmıştır [41]. MFKK ile ses sinyalleri belirli katsayılarla ifade edilir. Bu katsayılar ses sinyalinin öz niteliklerini temsil eder ve böylece ses sayısal değerlere dönüşmüş olur ve giriş verisi olarak kullanıma uygun hale gelir.

4.1. Mel Frekans Kepstral Katsayıları Yöntemi

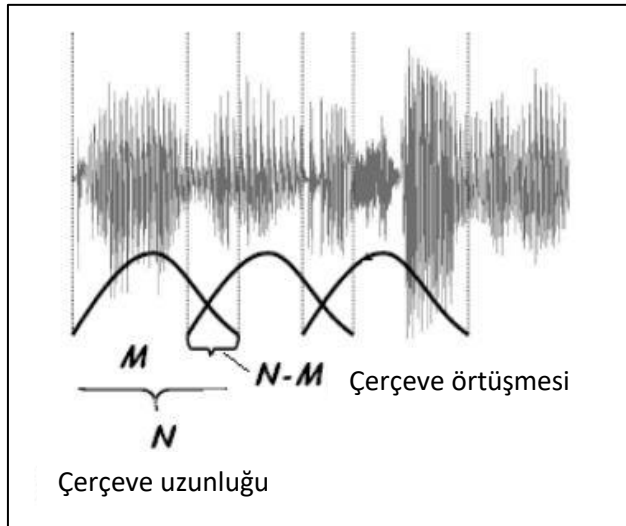
MFKK, insan kulağındaki kritik bant genişliğinin frekansa göre değişimi temeline dayanarak geliştirilmiş ve düşük frekansta doğrusal, yüksek frekansta logaritmik olarak çalışan bir filtredir. Bu filtre ses sinyalinin önemli karakteristik fonetiğini yakalamak için kullanılır. İnsan kulağını taklit etmek için mel-frekans ölçeği kullanılır. Mel-frekans ölçeği 1 000 Hz'in altında doğrusal frekans, üzerinde ise logaritmik frekans olarak ifade edilir. Şekil 4.1 ile MFKK'nin blok diyagramı verilmiştir.



Şekil 4.1. Sesin MFKK özneliliğinin çıkartılmasına ait blok diyagram

Çerçeveleme:

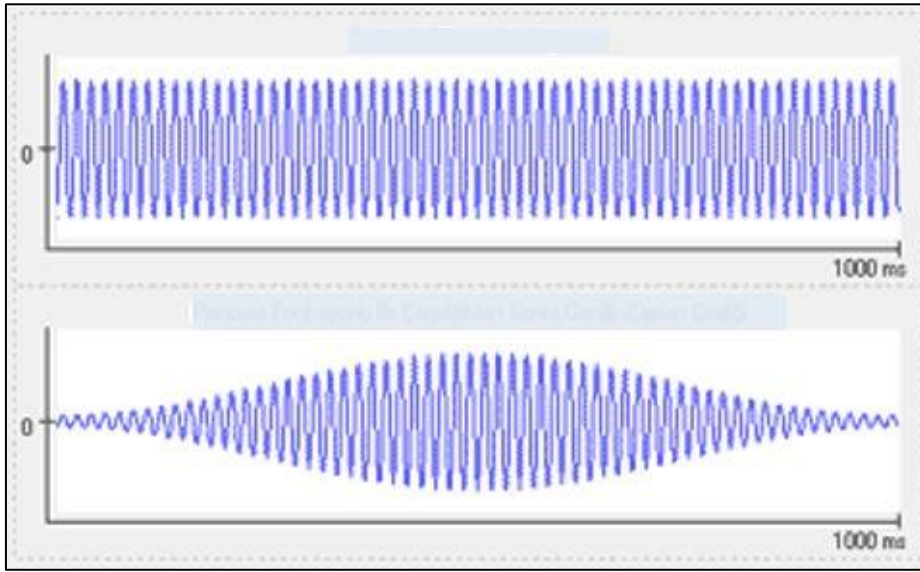
Ses sinyalleri uzun zaman aralıklarında kararsız bir yapıya sahiptir. Ancak kısa zaman aralıklarında kararlı bir karakteristik gösterirler. Ses sinyalleri bir işleme tâbi tutulacağı zaman bu kararlı yapısı göz önünde bulundurulur ve sinyaller küçük parçalara ayrılarak işlenir. Çerçeveleme, ses sinyalinin tek parça halinde analiz edilmesi yerine sinyalin 0-20 ms boyutunda bir kaç çerçeveye bölünmesi işlemidir [6]. Her bir çerçeve önündeki ve sonundaki çerçeveye üst üste gelir ve bu duruma örtüşme denir. Örtüşme pencereler arası geçişin yumuşak olması amacıyla kullanılır. Şekil 4.2 bir konuşma sinyalinin çerçevelere ayrılma işlemini göstermektedir.



Şekil 4.2. Ses sinyalinin çerçeveleme işlemi [42]

Pencereleme:

Pencereleme işlemi ile örtüşme sebebiyle oluşabilecek süreksizlikler yok edilir. Pencereleme işlemi sayesinde çerçevelerin başındaki ve sonundaki kısımlarından atılan bilgilerin yeniden kullanılmasına imkan verilmiş olur. Konuşma tanıma sistemlerinde en çok tercih edilen pencereleme Hamming penceresidir. Bir sinüs sinyali için Hamming pencerelemesinden önce ve sonra sinyalin durumları Şekil 4.3’de verildiği gibidir.



Şekil 4.3. Ses sinyaline ait pencereleme işlemi

Hızlı Fourier dönüşümü (FFT):

Ayrık Fourier dönüşümü (AFD), ayrık zamanlı sinyallerin analizinde kullanılan bir algoritmadır. FFT ise zaman uzayında gösterilmiş bir sinyalin frekans uzayındaki gösterimine dönüşmesini sağlayan AFD tabanlı bir algoritmadır. Her bir çerçeve FFT’ye uğrar ve böylece giriş sinyalleri zaman uzayından frekans uzayına geçmiş olur. Sonuçta sinyaller zamana bağlı olarak değil frekansa bağlı olarak değerlendirilebilir hale gelmiştir. Bu işlemin sonucunda spektrumlar elde edilir. N adet örnek içeren bir sinyal için X girişlerine için FFT “Eş 4.1” ile hesaplanır.

$$X_n = \sum_{k=0}^{N-1} X_k e^{\frac{-2\pi kn}{N}}, \quad n = 0, 1, 2, \dots, N-1 \quad (4.1)$$

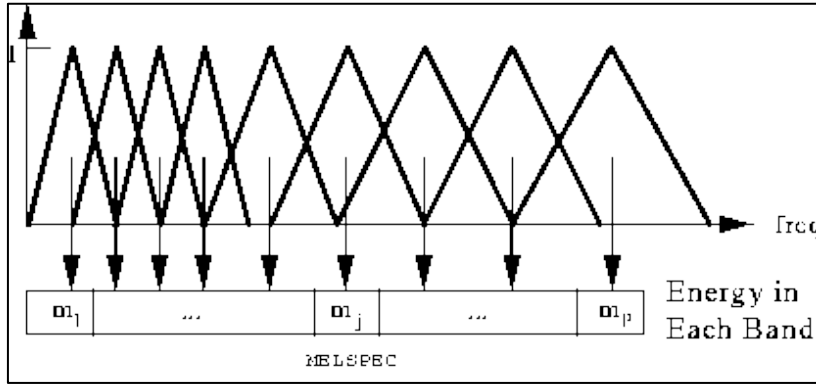
Mel-frekans saptırması:

İnsan kulağı ses sinyallerini doğrusal bir skalada algılamaz. Bu sebeble gerçek frekansı Hertz cinsinden “f” olan her bir ton için “mel skalası” denilen bir skalada öznel bir perde

ölçülür [6]. Hertz cinsinden verilmiş bir frekansın mel ölçeğindeki karşılığını bulmak için “Eş. 4.2” kullanılır.

$$mel(f) = 2595 * \log_{10}\left(1 + \frac{f}{700}\right) \quad (4.2)$$

Mel-frekans saptırmasında sinyal üçgen bir filtrede gruplandırılır. Gruplandırma, her bir FFT değerinin uygun kazanç filtresi ile çarpılarak sonuca eklenmesini sağlar. Her grup $m_1, \dots, m_p$ ile gösterilen sinyal enerji aralıklarını içerir [6]. Şekil 4.4 mel ölçeğinin bir uygulaması olarak filtre bankasını göstermektedir.



Şekil 4.4. Mel ölçeği uygulaması [6]

Kepstrum:

Spektrumun tersi olarak tanımlanan kepstrum, genellikle insan konuşma sinyallerinden bilgi almak amacıyla kullanılır. Her bir çerçeveye ters Fourier dönüşümü uygulanır ve ses sinyalleri tekrar frekans uzayından zaman uzayına geçirilmiş olur. Bu adımda log-mel spektrumu ayrık cosinüs dönüşümü (ACD) kullanılarak kepstruma dönüştürülür ve işlem sonucunda MFKK elde edilir. MFKK elde etme işlemi “Eş 4.3” ile gösterilmiştir.

$$MFKK_i = \sum_{k=1}^K (X_k) \cos \left[i \left(k - \frac{1}{2} \right) \frac{\pi}{2} \right] \quad i = 0, 1, 2, \dots, K \quad (4.3)$$

K , kepstrum sayısını, X_k , $i = 0, 1, 2, \dots, K$ ike k . Filtrenin log-enerji çıktısını temsil etmektedir.

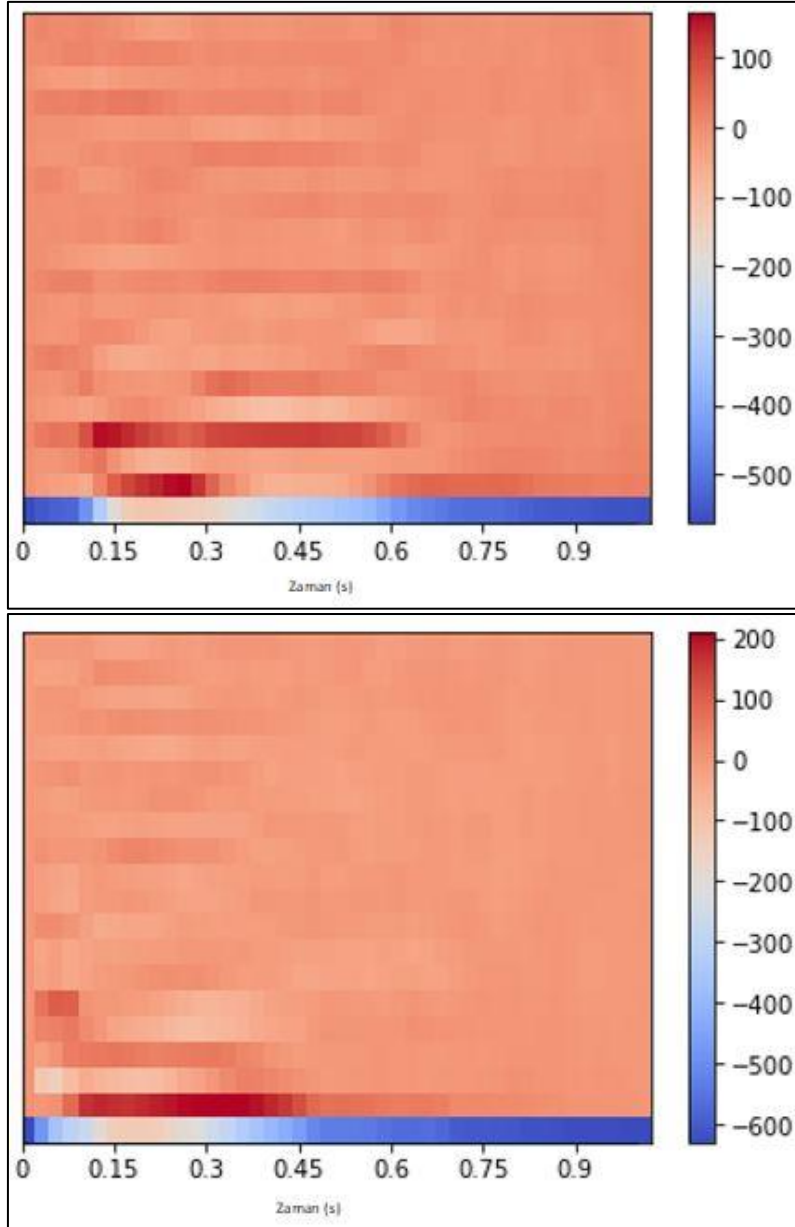
Tez için tasarlanan sinir ağlarında kullanılmak üzere hazırlanan veri setlerindeki tüm ses sinyallerinin MFKK özniteliği çıkarılmış ve mimarinin giriş verileri ses olarak değil MFKK

özniteliği olarak alınmıştır. Bölüm 6’da gerçekleştirilen tüm deneylerde ses dosyaları MFKK’ye dönüştürüldükten sonra mimariye giriş verisi olarak verilmiştir.

Veri seti içinden yes etiketli “0a2b400e_nohash_0.wav” adlı ses örneği için (20,44) boyutlu bir katsayı matrisi elde edilmiştir. Elde edilen MFKK katsayılarının bir kısmı Şekil 4.5’ de verilmiştir. Ayrıca bu sayıların ifade ettikleri görüntü temsilleri tüm komutlar için grafikler ile gösterilmiştir. Şekil 4.6.a “0a2b400e_nohash_0.wav” adlı yes sesi için MFKK grafiğini ve Şekil 4.6.b “0a2b400e_nohash_0.wav” adlı no sesi için MFKK grafiğini göstermektedir. Geriye kalan diğer 10 komuta ait MFKK grafikleri EK-1’de verilmiştir. Ayrıca Türkçe komutlar için de MFKK grafikleri EK-2 ile verilmiştir.

mfkk - NumPy object array								
	0	1	2	3	4	5	6	
0	-569.288	-548.4	-534.567	-525.32	-448.893	-306.245	-191.288	-
1	-5.10103	-21.1138	-32.908	-34.6588	-52.7079	14.6448	69.6399	1
2	-3.04304	-11.3135	-12.9311	-0.93026	18.5381	39.2426	-5.91488	-
3	8.02547	30.0139	40.8571	39.7573	90.885	155.663	151.088	1
4	-5.76666	-17.4463	-24.6701	-25.1728	-38.4124	-27.7837	-6.72522	3
5	-1.05028	-7.17092	-8.54579	0.98035	28.8865	-0.761892	-8.53885	-
6	5.37019	19.2664	25.2089	15.3799	4.92745	-27.2783	-46.8223	-
7	-4.3462	-9.18723	-11.5064	-8.14532	4.22287	9.13503	6.24681	-
8	-0.0057513	-7.29078	-12.0632	-14.8452	-6.66142	-21.8295	-20.4059	-
9	2.63871	11.5839	18.7908	19.753	18.7026	-2.66813	-3.8333	1
10	-1.39006	-4.05219	-9.77772	-20.409	-27.5075	-31.8027	-34.8789	-
11	-0.633336	-3.11337	-1.41965	-2.36669	2.85896	2.22345	-3.04307	0

Şekil 4.5. MFKK katsayıları



Şekil 4.6. a) Yes için MFKK b) No için MFKK

5. SES VERİSİ ARTIRMA VE KULLANILAN YÖNTEMLER

Derin öğrenme uygulamalarında ne kadar çok veri kullanılırsa eğitim başarısı o denli yüksek olmaktadır. Ancak bazı durumlarda çok sayıda veri bulmak kolay değildir. Veri miktarının yetersiz olduğu durumlarda mevcut verilere farklı deformasyonlar uygulanarak yeni veriler elde etmek mümkündür. Bu sayede verinin kısıtlı olduğu koşullarda dahi başarı oranı yüksek derin öğrenme eğitimleri yapmak mümkün hale gelmektedir. Bu bölümde ses verilerine uygulanabilecek veri artırma yöntemleri hakkında bilgi verilmiştir. Ayrıca tez kapsamında gerçekleştirilen deneylerde kullanılan Türkçe veri setine ait seslere hangi artırımların uygulandığı belirtilmiştir.

Veri kümesi küçük olduğunda eğitilen model veriler aracılığıyla öğrenmek yerine ezber yapar (aşırı uydurma-overfitting). Bu durumda model mevcut veri setiyle test edildiğinde yüksek doğruluk oranı sağlar ancak daha önce görmediği farklı bir veri kümesiyle test edilirse çok daha düşük doğruluk oranı gözlemlenir. Çünkü model az veri ile eğitildiğinde arka planda kullandığı hesaplamalar, tıpkı hassas bir denklemlerle çözülmesi gereken bir problemin yalnızca bir toplama işlemiyle ve hatta yuvarlanarak yapılması gibi basit olacaktır. Oysa ki daha fazla veri, daha kapsamlı bir hesaplama böylece daha doğru bir sonuç verecektir. Veri artırma sayesinde veri sayısı artırıldığından aşırı uydurma sorunu çözülmektedir. Ayrıca veri artırma modelin eğitim verisi içerisinde en ayırt edici özellikleri öğrenmesini sağlar [12]. Böylece gerçek hayatta karşılaşılabilecek farklılıklara karşı model daha güçlü bir hale gelmiş olur [12].

Veri artırmada kullanılan her bir yöntem doğruluğu farklı oranlarda etkilemektedir [12]. Hatta her bir komut (sınıf) için uygulanan veri artırma yöntemi farklı sonuçlar verebilmektedir. Örneğin “yes” komutu için başarılı olan bir veri artırma metodu “forward” komutu için başarılı olmayabilir. Bu sebeple uygulanan yöntemlerin her bir sınıftaki etkisini de gözlemlemek uygulama başarısını artırmak için önemli bir ölçüttür. Çünkü bir sınıfta başarılı olan yöntem farklı bir sınıfa hiç bir katkı sağlamayabilir veya olumsuz bir etkiye sebep olabilir [13]. Artırma yapılırken belirli bir deformasyonun uygulanmasından önce ve sonraki doğruluk değerleri karşılaştırılarak en faydalı olan artırma yöntemine karar verilip uygulanması başarıyı olumlu anlamda etkileyebilmektedir. Sınıfsal doğruluğa dikkat edilerek yapılan bu artırımlar sınıf koşullu artırma olarak bilinir. Sınıf koşullu veri artırma doğruluk oranını önemli ölçüde yükseltebilir [13].

Ses için veri artırmada en çok kullanılan yöntemler; zaman germe (time stretching), perde değiştirme (pitch shifting), dinamik aralık sıkıştırma (dynamic range compression) ve arkaplan gürültüsü (background noise) eklemek olarak sayılabilir. Bu temel yöntemlerin yanı sıra GAN, DÖKK gibi farklı yöntemlerde kullanılabilir [12, 13]. Ayrıca rastgele döndürme, kesme, renk kaydırma, Gauss filtresi ve Tuz ve Biber Gürültüsü gibi görüntü için kullanılan veri artırma teknikleri de ses artırma için kullanılabilir [19]. Bu yöntemlere ek olarak kaba kuvvet artırması (brute-force augmentation), sınıf koşullu artırma (class-conditional augmentation) ve metrik tabanlı artırma (metric-based augmentation) deneyleriyle de artırmaların doğruluk oranlarına etkisi gözlemlenebilmektedir [13].

Ses için geliştirilmiş artırma teknikleri doğrudan ses sinyaline uygulanırken, temelde görüntü artırma tekniği olan yöntemler sesin görselleştirilmiş temsiline uygulanır. Bu tez kapsamında seslerin manipüle edilmesiyle sentetik ses verileri oluşturmak hedeflenmiştir ve direkt olarak ses sinyalleri üzerinden artırma yapılmıştır. Doğrudan ses sinyaline uygulanan bazı deformasyonlar; zaman germe, ses perdesi değiştirme, dinamik aralık sıkıştırma ve arka plan gürültüsü ekleme olarak örneklendirilebilir. Bu deformasyonların daha iyi algılanması için genel tanımlamalar şöyledir:

- TS: Zaman germe, ses örneklerinin hızının azaltılması veya artırılması şeklinde uygulanan bir deformasyon türüdür.
- PS: Ses perdesi, sesi oluşturan dalgaların frekansı ile ilişkilendirilen bir değerdir. Ses örneğinin süresinde herhangi bir değişiklik yapılmadan python ile librosa.effect kütüphanesi kullanılarak ses örneğinin perdesi yükseltip alçaltılabilir.
- DRC: Bir seste duyulabilir olan en yumuşak sese göre üretilen en yüksek bozulmamış ses oranına dinamik aralık denir [43]. Her bir ses örneğinin dinamik aralığı yüksek seslerin sınırlandırılması ya da alçak seslerin yükseltilmesiyle sıkıştırılabilir [12].
- BG: Orijinal ses örneklerinin farklı ortam sesleriyle karıştırılmasına arka plan gürültüsü ekleme denir. Sokak trafiği, korna sesi, parktaki çocukların sesi, hayvan sesleri gibi sahneler arka plan gürültüsüne örnek gösterilebilir. Arka plan gürültüsü ekleme matematiksel olarak x , orijinal ses örneği; y , arka plan sahne örneği ve w , ağırlık parametresi (0.9) olmak üzere $z = x + w.y$ eşitliği kullanılarak gösterilebilmektedir [12].

Bu tez kapsamında öncelikle Speech commands dataset version 2 veri seti içerisindeki her bir sınıfın içinde 50 adet ses olacak şekilde seçim yapıp küçük bir veri seti oluşturulmuştur

ve bu veri setine anlatılan teknikleri de kapsayan birçok parametre değeri ile farklı deformasyonlar uygulanarak veri seti iki katına çıkartılmıştır. Ardından oluşturulan Türkçe veri setindeki ses verileri aynı deformasyonlara uğratarak iki katı büyüklüğüne çıkartılmıştır. Daha sonra farklı deformasyon teknikleri sınıflara rastgele olarak uygulanmıştır ve daha da büyük bir veri seti elde edilmiştir. Oluşturulan veri setlerine ait bilgiler Bölüm 6.1’de verilmiştir. Deformasyonlar, bir ses yazılımı olan “Audacity” uygulaması kullanılarak gerçekleştirilmiştir.

Sese uygulanan manipülasyonlar her sınıf için ayrı ayrı yapılmıştır. Seslerin dinlenerek farklılaşmanın olduğu ama aynı zamanda anlaşılır olduğu manipülasyonlar tercih edilmiştir. Çünkü zaten küçük olan veri setinin kötü yönde manipüle edilmesi kalitesiz veriler ortaya çıkaracaktır ve bu durumda eğitimi olumsuz etkileyecektir. Oluşturulan Türkçe veri setinin çok küçük olmasından dolayı, öncelikle seslerde iyileştirme yapılarak artırmanın daha faydalı olması amacıyla, 12 sesli komut için de tüm seslere gürültü azaltma uygulanmıştır. Türkçe veri setine gürültü azaltmaya ek olarak uygulanan diğer deformasyonlar şunlardır:

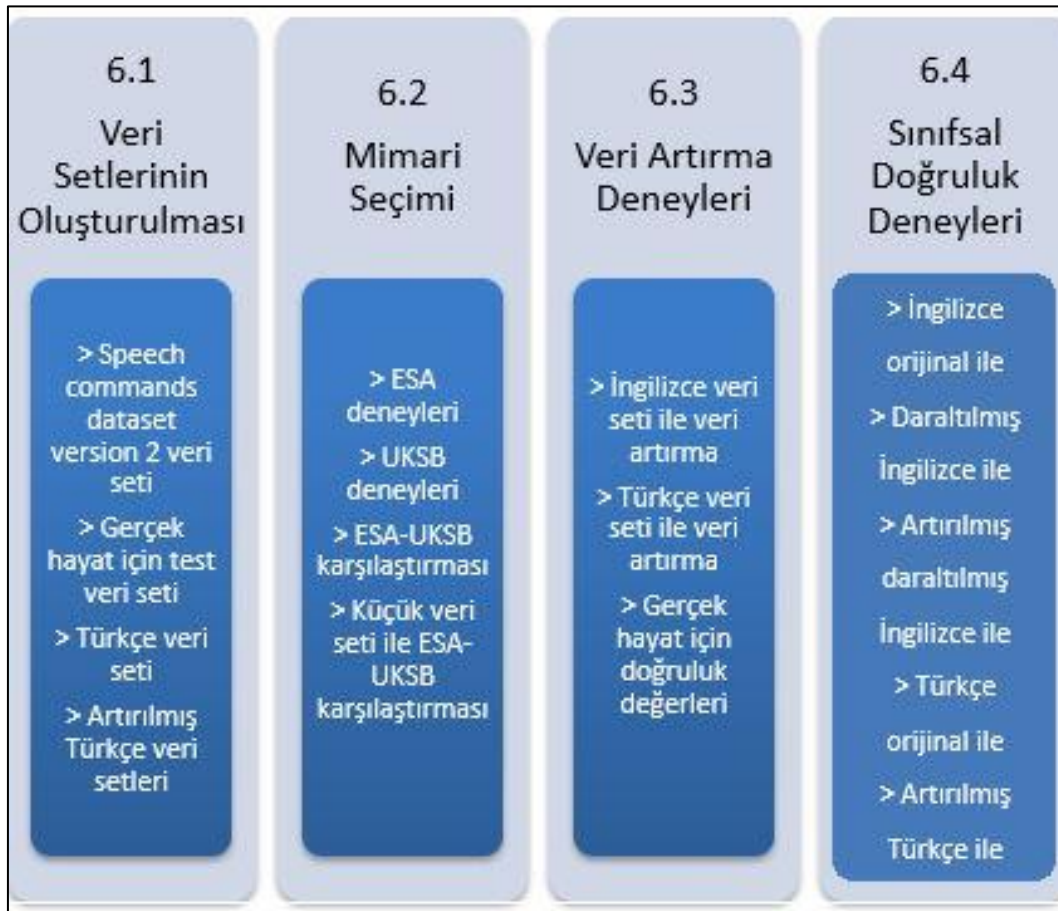
- “Aç” komutu için; sesler 0,75 oranında TS’ye uğratılmıştır.
- “Aşağı” komutu için; sesler 1,25 oranında TS’ye uğratılmıştır.
- “Dur” komutu için; seslerin perdesi değiştirilmiştir (B’den E’ye).
- “Evet” komutu için; seslerin perdesi değiştirilmiştir (A’dan C’ye).
- “Git” komutu için; seslere 0,2 saniye gecikmeli olarak yankı eklenmiştir.
- “Hayır” komutu için; tepe genliği -1dB’de sabitlenerek seslere normalleştirme uygulanmıştır.
- “İleri” komutu için; seslerin genliği artırılmıştır.
- “Kapat” komutu için; sesler 0,8 oranında TS’ye uğratılmıştır.
- “Sağ” komutu için; seslerin bas ve tiz değerleri 30 dB’e ayarlanmıştır.
- “Sol” komutu için; seslere yankı eklenmiştir.
- “Takip et” komutu için; seslere 0,2 saniye gecikmeli olarak BG eklenmiştir ve ses genliği azaltılmıştır.
- “Yukarı” komutu için; seslere 0,2 saniye gecikmeli olarak BG eklenmiştir.

Seslerin anlatılan tekniklerle manipüle edilmesi sayesinde veri seti boyutu artırılmıştır. Böylece artırma olmadan önceki veri seti kullanıldığında gerçekleşen öğrenme sonucundaki

başarının artırılacağı düşünülmektedir. Bölüm 6.3’de veri artırma işleminden önce ve sonraki veri setleri ile oluşturulan mimarilerin başarıları karşılaştırılarak veri artırmanın öğrenmeye olan etkisi açıklanmıştır.

6. DENEYLER

Tez çalışması boyunca kullanılacak olan veri setleri Bölüm 6.1 başlığı altında gösterilmiştir. Uygulama için en uygun olan mimariyi seçebilmek adına Bölüm 6.2 ‘de ESA ve UKSB ağı için en uygun olan katman sayısı, aktivasyon fonksiyonu gibi parametrelerin belirlenmesi için deneyler yapılmıştır. Bu amaç doğrultusunda yapılan deneyler ile ESA ve UKSB ağlarından hangisinin tercih edilmesi gerektiği kararına varılmıştır. Seçilen mimari ile büyük ve küçük veri seti kullanılması farketmeksizin benzer sonuçlara ulaşılabilineceği gösterilmiştir. Bölüm 6.3’de ise Speech commands dataset version 2 veri setinin küçültülüp ardından veri artırma ile artırılması ve Türkçe veri setinin artırılması sağlanmıştır. Ardından veri artırma öncesi-sonrası gerçek hayattaki doğruluk değerleri karşılaştırılmıştır. Son olarak veri artırmanın her bir komut sınıfını nasıl etkilediğini görmek amacıyla Bölüm 6.4’de sınıfsal doğruluk deneyleri yapılmıştır. Deneyler başlığı altında takip edilen çalışma planı Şekil 6.1 ile gösterilmiştir.



Şekil 6.1. Deneysel çalışma planı

6.1. Veri Setlerinin Oluşturulması

Bu tez çalışması boyunca anahtar kelime belirleme sistemlerini eğitmeye yardımcı olmak amacıyla tasarlanmış sözlü kelimelerden oluşan Creative Commons BY 4.0 [44] lisansı altında yayınlanan Speech commands dataset version 2 isimli veri seti [45] kullanılmıştır. Veri seti 35 kelimeden oluşan ve her biri 1 saniye veya daha kısa süreli sesler içeren toplam 105 829 adet ses verisi içermektedir [9]. Bu tez kapsamında 35 farklı kelimeye ihtiyaç duyulmadığından, komut niteliğinde olabilecek 12 adet kelime seçilmiştir. Oluşturulan bu veri seti artık İngilizce orijinal veri seti olarak anılacaktır. Bu kelimeler ve içerdikleri ses örneği sayıları Çizelge 6.1’de verilmiştir.

Çizelge 6.1. İngilizce orijinal veri setindeki mevcut komutlar ve ses örneği sayıları

<i>Komut</i>	<i>Örnek Sayısı</i>
Down	3 917
Follow	1 579
Forward	1 557
Go	3 880
Left	3 801
No	3 941
Off	3 745
On	3 845
Right	3 778
Stop	3 872
Up	3 723
Yes	4 044

Deneylerin ilk aşamasında sesli komut tanıma işlemi için en uygun mimariyi seçmek amacıyla eğitimler ve testler mevcut veri seti ile yapılmıştır. Daha sonraki deney aşamalarında kullanılmak üzere ise mevcut veri setinin içerisinden her bir komut için rastgele olarak 50 adet ses örneği alınmış ve toplamda 600 adet ses içeren daraltılmış İngilizce isimli küçük bir veri seti oluşturulmuştur. Ardından veri artırma deneylerinde kullanılmak üzere native_50 veri setine veri artırma uygulanarak iki katı büyüklüğündeki artırılmış daraltılmış İngilizce veri seti oluşturulmuştur.

Çalışmanın gerçek hayattaki başarısını test edebilmek amacıyla anadili Türkçe olan kişilerden 12 komutun seslendirildiği 651 adet ses kaydı toplanmıştır. Toplanan sesler genellikle arka plan gürültüsü içermektedir ve telaffuzlar hatalıdır. Bu durum çalışmanın

gerçek hayattaki başarısını göstermek için bilerek uygulanmıştır. Oluşturulan yeni veri seti İngilizce test ismi ile anılacaktır ve bu veri setine ait bilgiler Çizelge 6.2’de verilmiştir.

Çizelge 6.2. İngilizce gerçek hayat doğruluğu testleri için kullanılan veri seti içeriği

<i>Komut</i>	<i>Örnek Sayısı</i>
Down	52
Follow	55
Forward	54
Go	54
Left	60
No	56
Off	51
On	55
Right	53
Stop	55
Up	52
Yes	54

Deneylein son aşaması ise 12 komutun Türkçe hali ile gerçekleştirilmiştir. Bu aşamada ortalama 35 farklı kişiden alınan toplam 412 adet ses ile oluşturulan Türkçe orijinal veri seti kullanılmıştır. Veri setinin boyutu başarılı bir eğitim için ihtiyaç duyulandan ciddi manada küçük seçilmiştir. Çünkü bu tezin esas amaçlarından birisi veri setinin kısıtlı olduğu durumlarda verilerin manipüle edilmesi sayesinde veri seti boyutunun artırılması ve bu sayede de başarının artırılmasını sağlamaktır. Oluşturulan Türkçe veri setine ait komutlar ve komutlara ait ses dosyalarının sayısı Çizelge 6.3’de verilmiştir.

Çizelge 6.3. Türkçe orijinal veri seti içeriği

<i>Komut</i>	<i>Örnek Sayısı</i>
Aç	37
Aşağı	33
Dur	34
Evet	34
Git	34
Hayır	34
İleri	34
Kapat	32
Sağ	34
Sol	36
Takip et	36
Yukarı	34

Türkçe veri seti içerisindeki her bir ses için gürültü azaltma uygulanmıştır ve böylece sesler gürültüden arındırılmıştır. Ardından her bir sınıfa Bölüm 5’de anlatılan deformasyonlar uygulanmıştır. Böylece her bir ses için manipülasyona uğramış farklı bir ses üretilmiştir. Sonuçta toplamda 824 ses içeren ilk veri setinin iki katı büyüklüğünde artırılmış Türkçe veri seti elde edilmiştir. Ardından veri seti bir kez daha artırılarak daha büyük bir veri seti oluşturulmuştur ve bu da çok artırılmış Türkçe veri seti olarak isimlendirilmiştir. Çok artırılmış Türkçe veri setine ait bilgiler Çizelge 6.4’ de verilmiştir.

Çizelge 6.4. Çok artırılmış Türkçe veri seti içeriği

<i>Komut</i>	<i>Örnek Sayısı</i>
Aç	222
Aşağı	198
Dur	202
Evet	204
Git	204
Hayır	204
İleri	204
Kapat	192
Sağ	206
Sol	214
Takip et	216
Yukarı	204

Son olarak Türkçe ile yapılan eğitimlerin gerçek hayattaki başarı oranını görmek için öğrenme sürecinde hiç karşılaşılmayan toplamda 83 adet ses içeren Türkçe test veri seti oluşturulmuştur. Oluşturulan tüm veri setlerinin isimleri ve içerdikleri toplam örnek sayısı Çizelge 6.5 ile verilmiştir. Veri setlerine ait isimler Bölüm 6.3’de kullanılmıştır.

Çizelge 6.5. Kullanılan veri setleri

<i>Veri setinin ismi</i>	<i>Toplam örnek sayısı</i>
İngilizce orijinal	41 682
Daraltılmış İngilizce	600
Artırılmış daraltılmış İngilizce	1200
İngilizce test	651
Türkçe orijinal	412
Artırılmış Türkçe	824
Çok artırılmış Türkçe	2 470
Türkçe test	83

6.2. Mimari Seçimi

Literatür incelendiğinde ses sinyallerinin kullanıldığı derin öğrenme çalışmalarında gösterdikleri başarıdan dolayı çoğu kez ESA ve ÖSA'nın özel bir türü olan UKSB mimarisi tercih edildiği görülmüştür. Bu bölümde literatür dikkate alınarak hem ESA hem de UKSB için deneyler yapılmıştır ve sonuçlar karşılaştırılmıştır. Böylece sesli komut tanıma uygulaması için optimum olduğu düşünülen mimari seçilmiş ve sonraki deneyler bu mimari kullanılarak devam etmiştir.

6.2.1. ESA deneyleri

Derin öğrenmede eğitilen modelin veriler aracılığıyla öğrenmek yerine ezber yapması sıklıkla karşılaşılan bir durumdur. Bu sorunun çözülmesi için ESA mimarisinde normalde kullanılması gereken katmanlara ek olarak normalizasyon, seyreltme veya erken durdurma gibi yöntemlerden de faydalanılmaktadır.

Normalizasyon işlemi yapay sinir ağını daha hızlı ve daha kararlı hale getirmeye yarar. Deneyde normalizasyon işlemi modele “batch normalization” katmanı eklenmesiyle sağlanmıştır. Bu normalizasyon yöntemi ile girdilerin normalleştirilmesi sağlanır ve böylece daha başarılı sonuçlar alındığı görülmüştür.

Dropout olarak bilinen seyreltme işlemi sinir ağındaki rastgele olarak bazı nöronların seyreltilmesi yani o nöronların unutturulması ve eğitim esnasında kullanılmaması olarak açıklanabilir. Unutturma işlemiyle modelin test esnasında karşısına çıkacak olan yabancı verilere hazırlıklı olması sağlanır ve böylece test doğruluğunu artırılır. Seyreltme değeri 0 ile 1 arasında olmalıdır ve çok yüksek seçilmesi çok sayıda nöronun unutulması demektir ve unutilan nöronların çok olması eğitimi olumsuz etkiler. En çok tercih edilen seyreltme değerleri 0,3 ve 0,5'dir.

Derin öğrenme sürecinde veri seti eğitim, doğrulama ve test aşamalarında kullanılmak için üç parçaya ayrılır. Deneylerde kullanılan veri seti, eğitim için %80, doğrulama için %10 ve test için %10 olacak şekilde ayrılmıştır. Eğitim verileri, eğitim yapılırken kullanılır ve her eğitim sonunda bir eğitim doğruluğu oranı elde edilir. Bütün eğitim verileri kullanılarak bir kez eğitim yapılmasıyla bir epoch tamamlanmış olur ve öğrenme boyunca eğitimin kaç kez

yaptırılacağı epoch sayısı ile belirtilir. Doğrulama verileri ile bir eğitim sonucundaki başarı test edilir ve öğrenme için yeterli olup olmadığına eğitim ve doğrulama arasındaki farka bakılarak karar verilir. Early stopping olarak bilinen erken durdurma yöntemi ile model eğitilirken doğrulama ile test arasındaki hata oranına bakılır ve hata oranı düştüğü sürece eğitim devam eder, arttığında ise epoch sayısına ulaşılmaya dahi eğitim durdurulur. Böylece programın gereksiz yere çalışması engellenmiş ve doğruluk hemen hemen en iyi durumda olduğunda eğitim durdurulmuş olur. Her bir eğitim işleminin sonunda doğrulama için de doğrulama doğruluğu oranı elde edilir. Test verileri ise bütün eğitim tekrarları bittikten sonra yani öğrenme işlemi gerçekleştiğinde doğruluğun test edilmesi için kullanılır. Öğrenme sonucunda test doğruluğu değeri elde edilir.

ESA mimarisi için tanh, sigmoid ve ReLU aktivasyon fonksiyonlarının başarısının gözlemlendiği, epoch sayısının 50 seçildiği, normalizasyon, seyreltme ve erken durdurma yöntemlerinin tasarlanan model içerisinde varlık yokluk durumundaki etkisinin gözlemlendiği deneylere ait eğitim ve test doğruluğu yüzdeleri Çizelge 6.6'da verilmiştir.

Çizelge 6.6. ESA mimarisi deneyleri

Mimari	Akt. Fonk.	Epoch sayısı	Normalizasyon	Seyreltme	Erken durdurma	Eğitim doğruluğu (%)	Test doğruluğu (%)
ESA	Tanh	50	yok	yok	yok	97,4	92,8
ESA	Sigmoid	50	yok	yok	yok	84,43	82,13
ESA	ReLu	50	yok	yok	yok	96,84	92
ESA	ReLu	50	var	yok	yok	98,14	94,32
ESA	ReLu	50	var	var	yok	97,91	94
ESA	ReLu	48	var	var	var	97,97	92,82
ESA	ReLu	46	yok	var	var	96,13	91,79

ESA mimarisinde aktivasyon fonksiyonlarının etkisi gözlemlenmiş ve en iyi sonuçları veren fonksiyon tanh olmuştur. Ancak ağır kolay ve hızlı öğrenmesi açısından işlem yükü hafifletilmek istenmiştir ve model oluşturulurken işlem yükü daha az olmasından ve tanh kullanıldığında sonuçlar ile çok uzak olmadığından dolayı ReLU fonksiyonu tercih edilmiştir. Modelde normalizasyon kullanıldığında en yüksek doğruluk elde edilmiştir ve bu sebeple tasarlanan mimaride 1 adet normalizasyon katmanı kullanılmıştır. Seyreltmenin etkisi öğrenmeyi eğitim doğruluğu açısından hem olumlu hem de olumsuz olarak, test doğruluğu açısından ise olumsuz olarak etkilemiştir. Ancak seyreltmenin unutturma işlemi olduğu düşünülürse modelin gerçek hayatta karşısına çıkacak seslere daha hazırlıklı olduğu söylenir. Bu tezin amacı gerçek hayattaki başarıları yükseltmek olduğu için seyreltme

işleminin olumsuz etkisi göz ardı edilerek kullanılmaya karar verilmiştir. Erken durdurma eğitimin gereksiz uzun süre çalışmasını engellemiştir aynı zamanda modelin ezber yapmamasına da katkı sağladığı için deneylerde kullanılmaya devam edilmiştir. Tasarlanan mimaride seyreltme değeri olarak 0,3 seçilmiştir. Sonuçta tasarlanan mimari, 3 adet evrişim katmanı ve her evrişim katmanından sonra maksimum havuzlama katmanı içermektedir. 1 adet normalizasyon katmanı ve 0,3 değerlikli seyreltme ile erken durdurma içermektedir.

Optimum ESA mimarisinin katman bilgileri Şekil 6.2’de gösterilmiştir. Evrişim katmanlarında 256 adet, çıktı katmanında 12 adet nöron bulunmaktadır. Son katmanda 12 komut olduğu için nöron sayısı da 12 seçilmiştir. Sınıflandırmada olasılıksal hesap yaptığı için çıktı katmanında aktivasyon fonksiyonu olarak softmax ile kullanılmıştır.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 42, 11, 256)	2560
batch_normalization (Batch Normalization)	(None, 42, 11, 256)	1024
max_pooling2d (MaxPooling2D)	(None, 21, 6, 256)	0
conv2d_1 (Conv2D)	(None, 19, 4, 256)	590080
max_pooling2d_1 (MaxPooling2D)	(None, 10, 2, 256)	0
conv2d_2 (Conv2D)	(None, 9, 1, 256)	262400
max_pooling2d_2 (MaxPooling2D)	(None, 5, 1, 256)	0
flatten (Flatten)	(None, 1280)	0
dense (Dense)	(None, 256)	327936
dense_1 (Dense)	(None, 12)	3084
Total params: 1,187,084		
Trainable params: 1,186,572		
Non-trainable params: 512		

Şekil 6.2. ESA mimarisinin katmanları

6.2.2. UKSB deneyleri

Tasarlanan ESA ve UKSB mimarilerinin karşılaştırılabilmesi için katman ve nöron sayıları mümkün olduğunca benzer seçilmeye çalışılmıştır. Bu sebeple UKSB modeli oluşturulurken ESA’da tercih edilen normalizasyon, seyreltme ve erken durdurma burada da tercih edilmiştir. UKSB mimarisinde tez çalışmasına en uygun modeli belirlemek amacıyla katman sayılarında ve aktivasyon fonksiyonlarında değişiklikler yapılmıştır. ESA’daki başarısından dolayı ilk olarak UKSB katmanı içerisinde ReLU fonksiyonu tercih edilmiştir ancak burada

aşırı düşük bir doğruluk elde edilmiştir. Bunun sebebi geri yayılım algoritmasında türev alma işlemi gerçekleştirildiğinden ve ReLU fonksiyonunda sıfırdan küçük girişler için çıktının sıfır olmasından dolayıdır. Bu durum verilerin türevinin sıfıra gitmesine sebep olur ve bu da verilerin eğitimde kullanılmaması anlamına gelir. Daha sonra UKSB için standartlaşmış olan tanh fonksiyonu kullanılarak daha yüksek sonuçlar elde edilmesi sağlanmıştır. Çünkü tanh fonksiyonunda her giriş için -1 ile 1 aralığında bir çıktı oluşturulur ve bu da daha az sayıda verinin türevinin sıfıra gitmesi anlamına gelir. Çizelge 6.7 ile verilen deneylerde katmanlar sütununda belirtilen uksb ifadeleri tanh aktivasyon fonksiyonuna sahip UKSB katmanını ifade etmektedir (birinci satır hariç). Katmanlar sütunundaki rakamlar bitişiğinde yazılı olan katmanın kat adet olduğunu ifade eder. İkinci olarak ifade edilen katmanlar ekstra olarak kullanılan aktivasyon katmanlarıdır. Sonraki bilgiler ise normalizasyona ve seyreltmeye ait kısaltmalardır. Epoch sayıları sabit 100 verilmiştir ancak bir deney hariç diğerlerinde erken durdurma olduğundan eğitimler farklı değerlerde durdurulmuştur. Bu bilgilerden yararlanarak farklı modeller için eğitim ve test doğruluğunun gösterildiği Çizelge 6.7 test doğrulukları küçükten büyüğe doğru sıralanacak şekilde düzenlenmiştir.

Çizelge 6.7. UKSB mimarisi deneyleri

Mimari	Katmanlar	Epoch	Eğitim Doğruluğu (%)	Test Doğruluğu (%)
UKSB	3uksb(relu)+2relu+1norm+seyr.(0,3)	38	11,7	11,1
UKSB	3uksb+2relu+2norm+seyr.(0,3)	35	100	50
UKSB	4uksb+4tanh+1norm+seyr.(0,3)	28	99,48	55,97
UKSB	3uksb+1tanh+1norm+seyr.(0,3)	30	100	57,46
UKSB	3uksb+2tanh+1norm+seyr.(0,5)	32	99,37	58,96
UKSB	3uksb+2tanh+1norm+seyr.(0,3)	29	100	58,97
UKSB	3uksb+2tanh+1norm+seyr.(0,3)	29	100	60,45
UKSB	3uksb+1relu+2norm+seyr.(0,3)	100	100	61,2
UKSB	3uksb+4relu+seyr.(0,3)	46	99,3	61,2
UKSB	3uksb+2relu+1norm+seyr.(0,3)	30	100	63,43
UKSB	3uksb+2relu+seyr.(0,3)	37	99,3	63,43
UKSB	3uksb+2tanh+1norm+seyr.(0,5)	39	98,56	63,43
UKSB	3uksb+tanh+1norm+seyr.(0,3)	29	100	64,18
UKSB	8uksb+1relu+1norm+seyr.(0,3)	70	100	64,93

Çizelge 6.7’de gösterilen deneyler çalışma süresinin azaltılması amacıyla daraltılmış İngilizce veri seti ile yapıldığından test doğrulukları ESA deneylerine göre (ESA deneylerinde İngilizce orijinal veri seti kullanılmıştır.) düşük elde edilmiştir. Ancak aynı mimarinin katmanları ile ilgili bir karşılaştırma söz konusu olduğundan küçük ya da büyük veri setini kullanmak kıyaslamayı etkilemez. Gerçekleştirilen deneyler sonrasında, UKSB mimarisi oluşturulurken ReLU ve tanh aktivasyon fonksiyonları denenmiştir ve en iyi sonucu veren fonksiyon tanh olmuştur. Modelde aktivasyon fonksiyonu tanh olan 3 adet UKSB katmanı kullanılmıştır. En yüksek test doğruluğu veren mimari 8 adet UKSB katmanı içermektedir ancak 3 adet içeren mimarinin doğruluğu da ikinci en yüksek test doğruluğuna sahiptir. Dolayısıyla eğitim için 3 katman tercih edilmiştir. Ayrıca gerçek hayattaki doğruluğun artırılması hedeflendiği için sinir ağına 1 adet 0,3 değerinde seyreltme katmanı eklenmiştir. Seyreltme ile sinir ağındaki bazı düğümler seyreltilir yani o düğümler unutturulur ve eğitim esnasında kullanılmaz. Böylece gerçek hayatta karşılaşılabilecek olası hatalar eğitim yapılırken de dikkate alınmış olur. Dolayısıyla seyreltme uygulandığında kayıp (loss) değeri azalır. Oluşturulan UKSB mimarisi çıkış katmanında 12, diğer katmanlarda 256 nöron olacak şekilde seçilmiştir. Seçimlerde ESA’ya benzer tercihler yapılmaya çalışılmıştır. Optimum UKSB mimarisinin katman bilgileri Şekil 6.3’de gösterilmiştir.

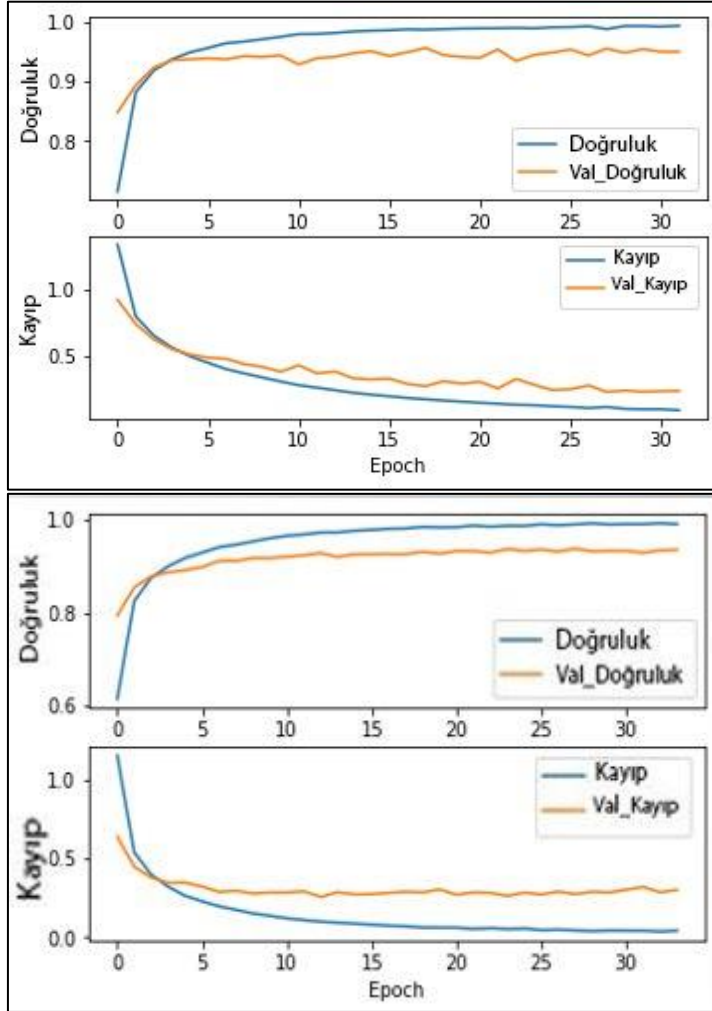
Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 44, 256)	276480
dense_12 (Dense)	(None, 44, 256)	65792
batch_normalization_6 (Batch Normalization)	(None, 44, 256)	1024
dropout_6 (Dropout)	(None, 44, 256)	0
lstm_1 (LSTM)	(None, 44, 256)	525312
dense_13 (Dense)	(None, 44, 256)	65792
lstm_2 (LSTM)	(None, 256)	525312
dense_14 (Dense)	(None, 12)	3084
Total params: 1,462,796		
Trainable params: 1,462,284		
Non-trainable params: 512		

Şekil 6.3. UKSB mimarisinin katmanları

6.2.3. ESA ve UKSB karşılaştırması

ESA ve UKSB için optimum modeller belirlendikten sonra aynı veri setleri kullanılarak program yeniden çalıştırılmıştır. Her iki model için de epoch sayısı 100 seçilmiştir. Ancak

modellere erken durdurma eklendiği için ESA 32. epochta ve UKSB 34. epochta eğitimi sonlandırmıştır. Eğitim ve doğrulama için doğruluk ile hata değerlerinin gösterildiği grafikler Şekil 6.4.a ve Şekil 6.4.b’de verilmiştir.



Şekil 6.4. a) ESA doğruluk ve kayıp grafiği, b) UKSB doğruluk ve kayıp grafiği

Grafiklerde yatay eksen epoch sayısını, dikey eksen ise doğruluk (üstteki) ve kayıp (alttaki) oranlarının yüze bölünmüş halini göstermektedir. Mavi renkli grafik eğitim doğruluğu oranlarını, turuncu renkli grafik ise doğrulama doğruluğu oranları göstermektedir.

İngilizce orijinal veri seti ile birden çok kez eğitim yapılmıştır. ESA mimarisi için gerçek hayattaki başarı oranının en iyi hali İngilizce test veri setindeki 651 sesli komuttan 412 tanesi doğru tahmin edilerek %63,29; UKSB mimarisi için gerçek hayattaki başarı oranının en iyi hali yine İngilizce test veri setindeki 651 sesli komuttan 354 tanesi doğru tahmin edilerek %54,38 olmuştur. Bu iki modele ait sonuçlar Çizelge 6.8’de verilmiştir.

Çizelge 6.8. UKSB ve ESA sonuçlarının karşılaştırması

Mimari	Epoch	Eğitim (%)	Test (%)	Gerçek hayat doğruluğu(%) (Doğru)
ESA	32	99,38	94,64	63,29 (412)
UKSB	34	98,96	92,94	54,38 (354)

Çizelgedeki eğitim, test ve gerçek hayat doğruluğu yüzdeleri ESA mimarisi kullanıldığında daha yüksek çıkmıştır. UKSB mimarisinin doğrulukları çok kötü olmamakla birlikte ses uygulamalarında başarılı olduğu için daha yüksek doğruluk elde etmek beklenmiştir. Ancak ses verileri mimariye ham ses olarak değil Bölüm 4’de anlatıldığı gibi MFKK’ye dönüştürülmüş şekliyle gönderilmiştir. Ayrıca UKSB’nin genel olarak uzun süreli sesleri içeren doğal dil işleme gibi uygulamalarda kullanıldığı bilinmektedir. Örneğin “Bugün hava çok soğuk olduğu için kazak giyeceğim.” cümlesinin tanımlanacağı bir mimaride havanın soğuk olduğu bilgisi hatırlanmalıdır ki sonrasında gelen kelimenin kazak ya da gömlek olma ihtimali daha doğru bir oranda hesaplanabilsin. Bu tez çalışmasında kullanılan sesler 1 saniyedir ve örnekteki gibi birbirine çok bağımlı kelimeler içermemektedir. Bu çalışmadaki uygulama için UKSB’nin ESA’dan daha başarısız olma sebebi arasında gösterilebilir.

ESA’nın başarısının küçük veri seti kullanılması halinde de geçerli olup olmadığını test etmek amacıyla aynı mimariler farklı veri seti ile bir karşılaştırma deneyi daha yapılmıştır. Aynı mimariler her sınıfta ellişer adet ses içeren daraltılmış İngilizce veri seti ile eğitildiğinde ESA için 651 sesli komuttan 195 tanesi doğru tahmin edilerek %29,95 UKSB için 651 sesli komuttan 135 tanesi doğru tahmin edilerek %20,73 ve sonuçları gözlemlenmiştir. Her iki modele ait sonuçlar Çizelge 6.9’da verilmiştir.

Çizelge 6.9. Küçük veri seti ile UKSB ve ESA sonuçlarının karşılaştırması

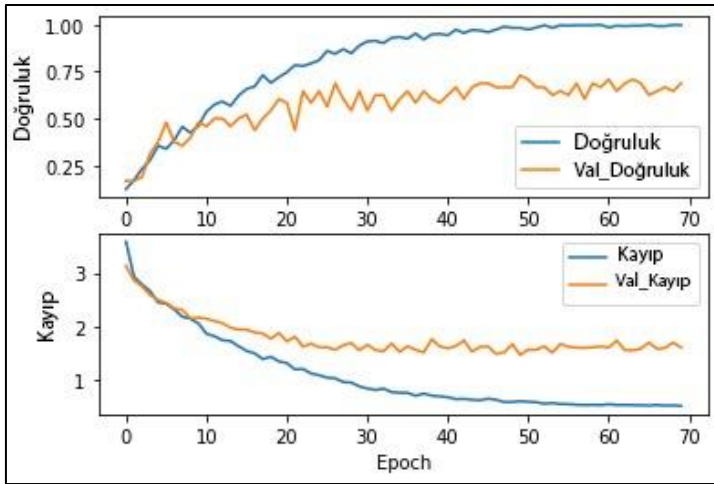
Mimari	Epoch	Eğitim (%)	Test (%)	Gerçek hayat doğruluğu (%) (Doğru)
ESA	32	100	79,63	29,95(195)
UKSB	28	100	64,2	20,73(135)

Çizelge 6.8 ve Çizelge 6.9 incelendiğinde veri setinin büyük ya da küçük olması farketmeksizin sesli komut tanıma sistemi için ESA mimarisinin daha başarılı olduğu görülmüştür. Bu sebeple ilerleyen deneylerde artık yalnızca ESA mimarisi üzerinden veri artırma yapılmıştır ve çıkarımlar ESA mimarisine göre değerlendirilmiştir.

6.3. Veri Artırma Deneyleri

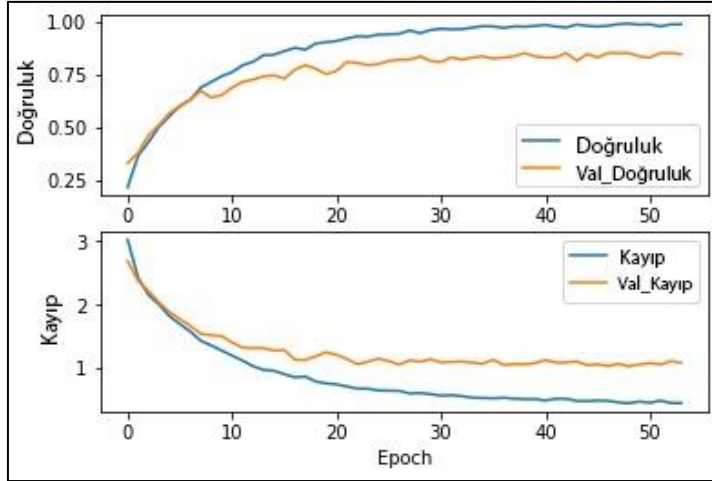
Önceki deneylerde bu tez çalışması için derin öğrenme mimarisi olarak ESA kullanıldığında daha başarılı sonuçlar elde edildiği sonucuna varılmıştır. Dolayısıyla artırma deneyleri Bölüm 6.2’de optimum kabul edilen ESA mimarisi kullanılarak ve Çizelge 6.5 ile verilen veri setleri kullanılarak gerçekleştirilmiştir.

ESA mimarisi ile daraltılmış İngilizce veri seti kullanıldığında en iyi gerçek hayat doğruluğunun %29,95 olarak elde edildiği önceki deneylerde görülmüştür. Aynı şartlar altında ortalama sonuçlar bulmak adına model 5 kez daha eğitilmiş ve sonuçta ortalama olarak %99,85 eğitim doğruluğu, %69,63 test doğruluğu ve %29,832 gerçek hayat doğruluğu elde edilmiştir. 5 deneyden test doğruluğu en büyük olan deneye ait sonuç Şekil 6.5’de gösterilmiştir.



Şekil 6.5. Daraltılmış İngilizce veri seti ile oluşturulan modelin doğruluk ve kayıp grafiği

Daraltılmış İngilizce veri seti ile elde edilen bu düşük doğruluk oranının artırılması amacıyla veri setindeki seslere rastgele olarak gürültü azaltma, hız değiştirme, perde değiştirme işlemleri uygulanarak veri seti boyutu iki katına çıkartılmıştır. Artırılmış daraltılmış İngilizce veri seti ile 5 kez model eğitilmiştir ve ortalama olarak %99,2 eğitim doğruluğu, %74,904 test doğruluğu ve %30,813 gerçek hayat doğruluğu elde edilmiştir. 5 deneyden test doğruluğu en büyük olan deneye ait sonuç Şekil 6.6’da gösterilmiştir.



Şekil 6.6. Artırılmış daraltılmış İngilizce veri seti ile oluşturulan modelin doğruluk ve kayıp grafiği

Şekil 6.5 ve Şekil 6.6 karşılaştırıldığında artırma sonucunda dalgalanmaların bir miktar azaldığı ve doğruluğun artıp kayıpların azaldığı görülebilmektedir. Veri artırma işlemi olmadan ve artırma işlemi sonrasında elde edilen doğruluk yüzdeleri Çizelge 6.10 ile verilmiştir.

Çizelge 6.10. İngilizce veri seti ile veri artırma kıyaslaması

Veri Sayısı	Eğitim Doğruluğu (%)	Test Doğruluğu (%)	Gerçek Hayat Doğruluğu (%)
600	99,85	69,63	29,832
1200	99,2	74,904	30,813

Elde edilen sonuçlara göre yalnızca iki katına çıkan veri seti ile bile test doğruluğu %5,274 artış sağlamıştır. Veri setinin daha fazla büyütülmesi ile bu oran daha da artırılabilir.

Ayrıca gerçek hayat doğruluğunda da yaklaşık %1 oranında artış olmuştur. Gerçek hayat doğruluğunun çok artmamasının sebebi telafuzdan ve İngilizce komutların fonemlerindeki (ses birimlerindeki) benzerlikten kaynaklıdır. Test veri setindeki çok zor tespit edilen bazı seslerin küçük bir veri artırma sonucunda anlaşılmasını sağlamak mümkün değildir.

Modelin büyük bir veri seti ile başarılı bir şekilde eğitilmesi sonucunda gerçek hayat doğrulukları da daha fazla artacağı düşünülmektedir. Ancak yine de azda olsa görülen bu artış veri artırmanın faydalı olduğunu göstermektedir.

İngilizce veri seti için herhangi bir kısıt yokken komutların Türkçe olması durumunda ne yazık ki veri seti kısıtlamasıyla karşılaşmaktadır. Bu sebeple veri artırma deneyleri Türkçe veri seti kullanılarak devam etmiştir.

Oluşturulan Türkçe veri seti ile sistem 10 kez eğitilmiş ve sonuçları kaydedilmiştir. 10 eğitim sonucunda ortalama eğitim doğruluğu %96,056, ortalama test doğruluğu %53,947 olmuştur. Bu %53,947'lik test doğruluğu, 600 adet ses içeren native_50 isimli, kaliteli bir veri seti ile yapılan eğitim sonucunda %64,81 test doğruluğu elde eden çalışmayla [46] karşılaştırıldığında 412 adet ve ortam gürültüsü içeren seslerden oluşan Türkçe orijinal veri seti ile yapılan eğitim için makul düzeydedir. Bu karşılaştırmaya yönelik sonuçlar Çizelge 6.11'de verilmiştir.

Çizelge 6.11. Daha önceki çalışmayla kıyaslama

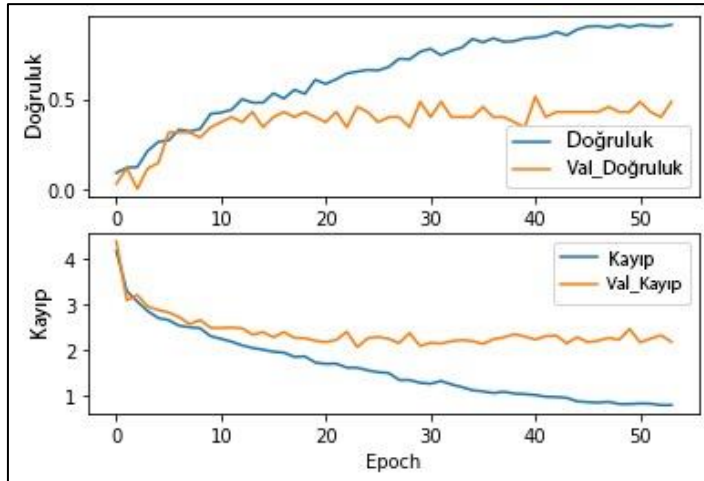
<i>Veri Seti</i>	<i>Toplam Veri Sayısı</i>	<i>Eğitim Hatası</i>	<i>Eğitim Doğruluğu (%)</i>	<i>Test Hatası</i>	<i>Test Doğruluğu (%)</i>
native_50	600	0,5315	98,87	2,1789	64,815
Türkçe orijinal	412	0,63846	96,056	2,5876	53,947

Sese uygulanan manipülasyonlar ile artırılmış veri seti kullanılarak deneyler yapıldığında ise ortalama olarak eğitim doğruluğunun %97,826'ya ve test doğruluğunun %70,133'e çıktığı gözlemlenmiştir. Veri seti yalnızca iki katına çıkarılmasına rağmen başarıdaki artışın kayda değer bir miktarda olması veri setini daha büyük hale getirmek ve böylece derin öğrenmeye daha uygun hale getirmek fikrini akla getirmiştir. Bu fikirle veri setine yeniden manipülasyonlar uygulanmış ve toplamda 2470 adet ses verisi elde edilmiştir. Önceki artırılmış veri setine göre daha çok artırılmış bu veri seti ile yapılan eğitim sonucunda eğitim doğruluğu %98,274'e ve test doğruluğu %85,756'ya çıkmıştır. Deneylerde gözlemlenen kayıp değeri ve doğruluk yüzdelerini içeren bilgiler Çizelge 6.12'de verilmiştir. Elde edilen sonuçlar incelendiğinde veri artırma sayesinde %53,947 olan test doğruluğunun %85,756'ya ulaşmasıyla %31,809'luk bir artış elde edilmiştir. Buradan anlaşılabacağı üzere komutların sınıflandırılması az veri ile neredeyse yarı yarıya hatalı olurken veri artırma sayesinde hata oranının %15 civarına düşmesi sağlanmıştır. Gözlemlenen bu artış miktarı, veri artırmanın başarıyı önemli ölçüde etkilediğinin kanıtıdır.

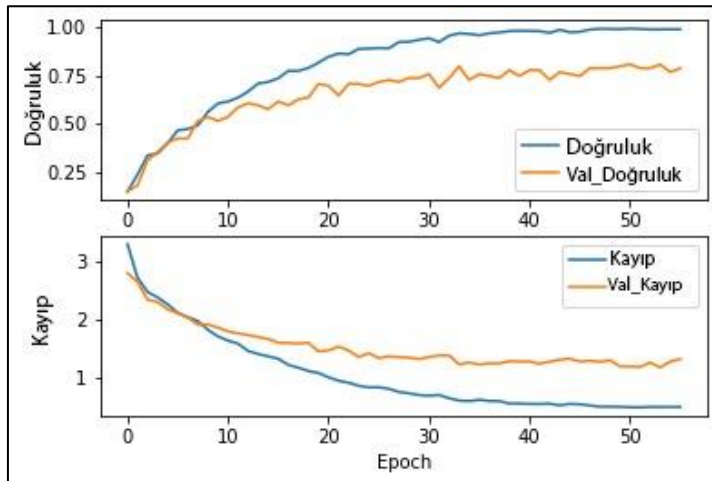
Çizelge 6.12. Orijinal ve artırılmış veri seti kıyaslaması

Veri Seti	Toplam Veri Sayısı	Eğitim Hatası	Eğitim Doğruluğu (%)	Test Hatası	Test Doğruluğu (%)
Türkçe orijinal	412	0,63846	96,056	2,58767	53,947
Artırılmış Türkçe	824	0,54993	97,826	1,62973	70,133
Çok Artırılmış Türkçe	2470	0,47678	98,274	0,94652	85,756

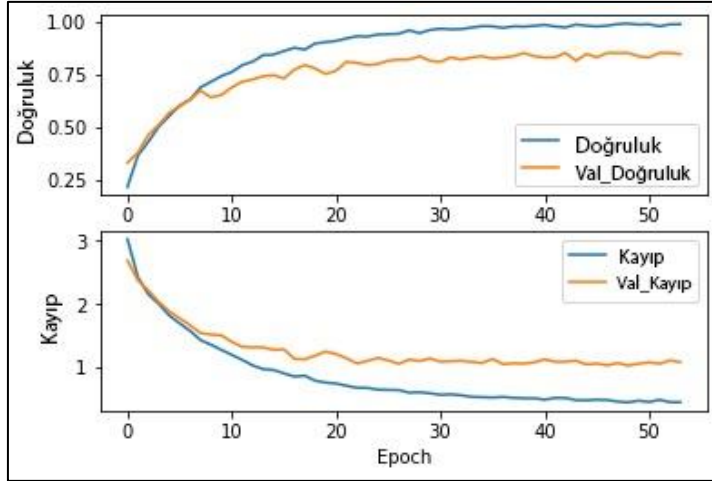
Veri artırma işleminden önceki ve sonraki doğruluk ve kayıp değerlerini gösteren grafikler Şekil 6.7, Şekil 6.8, Şekil 6.9 ile gösterilmiştir. Grafikler ortalamaya en yakın sonucu veren deneye ait grafik olacak şekilde seçilmiştir. Grafikler epoch sayısına karşılık doğruluk, epoch sayısına karşılık kayıp olmak üzere eğitim ve geçerleme değerleri için çizdirilmiştir. Veri artırma yokken doğruluğun daha düşük ve eğitim ile geçerleme doğrulukları arasındaki farkın daha çok olduğu görülmektedir. Veri artırma ile bu olumsuzlukların bir miktar giderildiği ve doğruluk oranlarının arttığı grafiklerden anlaşılmaktadır.



Şekil 6.7. Türkçe orijinal veri seti ile oluşturulan modelin doğruluk ve kayıp grafiği



Şekil 6.8. Artırılmış Türkçe veri seti ile oluşturulan modelin doğruluk ve kayıp grafiği



Şekil 6.9. Çok artırılmış Türkçe veri seti ile oluşturulan modelin doğruluk ve kayıp grafiği

Türkçe için gerçekleştirilen artırma deneylerine ait çizelgeler ve grafikler veri sayısı arttıkça eğitim ve tes doğruluklarının da arttığını göstermektedir. Bu tez kapsamında eğitim esnasında hiç duyulmamış seslerin de başarılı bir şekilde tanımlanıp tanımlanamayacağını anlamamızı sağlayacak olan gerçek hayat testleri de yapılması amaçlandığı için gerçek hayat test doğruluk oranları da gözlemlenmiştir. Bu deneyler için 83 adet ses içeren Türkçe test veri seti kullanılmıştır. Sonuçta Türkçe orijinal veri seti ile eğitilen model 83 sesin yaklaşık 48 tanesinin doğru tahmin ederek ortalama %58,074, artırılmış veri seti ile eğitilen model 83 sesin yaklaşık 54 tanesini doğru tahmin ederek ortalama %64,582 oranında doğruluk sağlamıştır. Çizelge 6.13’de verilen sonuçlar incelendiğinde veri artırma sayesinde gerçek hayattaki test doğruluğunun %6,508 oranında arttığı görülmektedir.

Çizelge 6.13. Gerçek hayattaki doğruluk oranları

Veri Seti	Toplam Veri Sayısı	Gerçek Hayat Doğruluğu (%)	Doğru Sayısı
Türkçe orijinal	412	58,074	48
Çok artırılmış Türkçe	2470	64,761	54

Yapılan deneyler sonucunda test doğruluğunda %31,809 oranında artış gözlemlenmesine rağmen gerçek hayat testinde %6,508 oranında bir artış gözlemlenmiştir. Çünkü gerçek hayat doğruluğu araştırılırken kullanılan sesler mümkün olduğunca insan ve ortam kaynaklı sorunları içeren ve eğitim esnasında kullanılmamış ses dosyalarıdır. Dolayısıyla gerçek hayattaki başarı oranının bilgisayar üzerindeki test başarı oranından daha az artış göstermesi giriş kısmında da belirtildiği gibi beklenildik bir durumdur. Ayrıca İngilizce için yapılan

artırmanın doğruluğa etkisi Türkçeye göre daha az olmuştur. Bunun sebebi İngilizce olarak seslendirilen komutların fonemlerinin birbirine Türkçeden daha çok benzerlik göstermesidir.

6.4. Sınıfsal Doğruluk Deneyleri

Bölüm 6.3’de komutlara ait fonemlerin birbirine benzer olması durumunda başarının daha zor arttığı görülmüştür. Bu tez çalışmasında kullanılan 12 komuta ait gerçek hayat doğruluklarını sınıfsal olarak değerlendirmek amacıyla optimum ESA mimarisi kullanılarak eğitimler yeniden yapılmıştır ve bu eğitim sonuçları sınıfsal olarak incelenmiştir. Önceki deney sonuçlarında ortalama veya en iyi sonuçlar verilmiştir. Bu bölümde verilen sonuçların önceki deneyler ile farklı ama yakın yüzdelerle değerlere sahip olabileceği bilinmelidir. Farklı doğruluk değerleri elde edilmesinin sebebi her eğitim sırasında veri setinin rastgele olarak kullanılmasından dolayı farklı doğruluk değerlerinin elde edilmesidir.

İngilizce orijinal veri seti ile yapılan eğitim sonucunda her bir komutun gerçek hayat doğruluklarına ait yüzdeler Çizelge 6.14’de verilmiştir.

Çizelge 6.14. İngilizce orijinal veri seti ile eğitimin sınıfsal doğruluğu

Komut	Gerçek hayat doğruluğu (%)
down	53,85
follow	52,73
forward	62,97
go	57,4
left	71,67
no	53,57
off	66,67
on	70,91
right	69,81
stop	63,64
up	61,54
yes	74,07

Çizelge 6.14 incelendiğinde her bir sesli komutun farklı oranlarda doğru olarak tahmin edildiği görülmektedir. En yüksek oranda doğru tahmin edilen komut “yes” iken en az oranda anlaşılan komut “follow” olmuştur.

Sınıfsal olarak veri artırmanın etkisini görmek amacıyla daraltılmış İngilizce veri seti kullanılarak yeniden yapılan eğitimin sonuçları Çizelge 6.15 ile verilmiştir.

Çizelge 6.15. Daraltılmış İngilizce veri seti ile eğitimin sınıfsal doğruluğu

Komut	Gerçek hayat doğruluğu (%)
down	19,23
follow	32,72
forward	42,59
go	14,81
left	40
no	23,21
off	23,53
on	29
right	22,64
stop	32,73
up	36,54
yes	40,74

Çizelge 6.15 incelendiğinde “*forward*” komutunun en yüksek oranda “*go*” komutunun en düşük oranda doğru tahmin edildiği görülmektedir. Kullanılan veri setine uygulanan veri artırma sonrasında elde edilen artırılmış daraltılmış İngilizce veri seti ile eğitim yapıldığında elde edilen sonuçlar Çizelge 6.16’da verilmiştir.

Çizelge 6.16. Artırılmış daraltılmış İngilizce veri seti ile eğitimin sınıfsal doğrulukları

Komut	Gerçek hayat doğruluğu (%)
down	34,62
follow	16,36
forward	24,07
go	29,63
left	45
no	16,07
off	37,25
on	25,46
right	11,32
stop	47,27
up	38,46
yes	50

Çizelge 6.16. incelendiğinde “yes” komutu en yüksek oranda doğru tahmin edilirken “right” komutu en düşük oranda doğru tahmin edilmiştir. Veri artırmanın sınıflara olan etkisinin net bir şekilde görülmesi için Çizelge 6.17 oluşturulmuştur.

Çizelge 6.17. Daraltılmış İngilizce- artırılmış daraltılmış İngilizce veri seti değişimi

Komut	Daraltılmış İngilizce (%)	Artırılmış daraltılmış İngilizce (%)	Fark (%)
down	19,23	34,62	15,39
follow	32,72	16,36	-16,36
forward	42,59	24,07	-18,52
go	14,81	29,63	14,82
left	40	45	5
no	23,21	16,07	-7,14
off	23,53	37,25	13,72
on	29	25,46	-3,54
right	22,64	11,32	-11,32
stop	32,73	47,27	14,54
up	36,54	38,46	1,92
yes	40,74	50	9,26

Çizelge 6.17 incelendiğinde veri artırmanın her bir sınıfa farklı etki ettiği görülmektedir. “Down” komutu için veri artırma işlemi gerçek hayattaki test doğruluklarında %15,39 oranında artış sağlarken “forward” komutu için %18,52 oranında bir düşüşe sebep olmuştur. Diğer komutlar için de benzer farklılıklar söz konusudur.

Sınıfsal gerçek hayat doğruluklarının tespiti Türkçe orijinal veri seti için yapıldığında elde edilen sonuçlar Çizelge 6.18’de verilmiştir.

Çizelge 6.18. Türkçe orijinal veri seti ile yapılan eğitimin sınıfsal doğrulukları

Komut	Gerçek hayat doğruluğu (%)
aç	50
aşağı	14,29
dur	28,57
evet	57,14
git	71,43
hayır	71,43
ileri	71,43
kapat	66,67
sağ	83,34
sol	42,86
takip et	57,14
yukarı	57,14

Çizelge 6.18 incelendiğinde en çok doğru tahmin %83,34 ile “sağ” komutu için elde edilirken “aşağı” komutu için %14,29 gibi oldukça düşük oranda doğru tahmin yapılmıştır. Aynı eğitim sonucunda oluşan bu büyük farkın sebebi az sayıda veri ile test yapılmasıdır. Çünkü 1 doğru tahmin bile doğruluk oranlarını fazla etkilemektedir. Burada odaklanılan nokta artırımların sınıfsal olarak farklı etkilere sebep olup olmayacağı görmek olduğundan bu yüksek fark göz ardı edilmektedir. Daha sonra artırılmış Türkçe veri seti ile de eğitim yapılmıştır ve sonuçları Çizelge 6.19’da verilmiştir.

Çizelge 6.19. Artırılmış Türkçe veri seti ile yapılan eğitimin sınıfsal doğrulukları

Komut	Gerçek hayat doğruluğu (%)
aç	75
aşağı	42,86
dur	57,14
evet	28,57
git	57,14
hayır	42,86
ileri	85,71
kapat	33,33
sağ	83,33
sol	42,86
takip et	57,14
yukarı	71,43

Çizelge 6.19 incelendiğinde doğrulukların önceki veri setine kıyasla genel olarak arttığı görülmektedir. Veri artırma öncesi ve sonrası sonuçların daha net görülmesi için Çizelge 6.20 verilmiştir.

Çizelge 6.20. Türkçe orijinal- artırılmış Türkçe veri seti değişimi

Türkçe orijinal- artırılmış Türkçe veri seti değişimi			
Komut	Türkçe orijinal (%)	Artırılmış Türkçe (%)	Fark (%)
aç	50	75	25
aşağı	14,29	42,86	28,57
dur	28,57	57,14	28,57
evet	57,14	28,57	-28,57
git	71,43	57,14	-14,29
hayır	71,43	42,86	-28,57
ileri	71,43	85,71	14,28
kapat	66,67	33,33	-33,34
sağ	83,34	83,33	-0,01
sol	42,86	42,86	0
takip et	57,14	57,14	0
yukarı	57,14	71,43	14,29

Çizelge 6.20 incelendiğinde “aşağı” ve “dur” komutları %28,57 oranında artırmadan olumlu etkilenirken “kapat” komutu %33,34 oranında olumsuz etkilenmiştir. Ayrıca “sol” ve “takip et” komutları için veri artırmanın olumlu ya da olumsuz bir etkisinin olmadığı görülmüştür.

Verilen deney sonuçlarına göre veri artırmalar ortalama olarak doğruluğu artırmıştır. Ancak sınıfsal olarak incelendiğinde artırmanın olumlu etkilerinin yanı sıra farklı sınıflar (komutlar) için etkisiz olduğu ya da olumsuz etkilerinin de olduğu görülmüştür. Bu deneyler sonucunda birçok veri artırma yönteminin incelendiği ancak mevcut artırmaların olumsuz yönlerinin de olduğuna değinen [11, 12] çalışmaları destekleyici sonuçlar görülmüştür. Her bir komutun fonemlerindeki benzerlik ve farklılıkların daha çok dikkate alındığı ve veri artırma tekniklerinin sınıf bazlı gerçekleştirildiği artırmalar sonucunda daha olumlu etkiler görülebilir.

7. SONUÇ VE ÖNERİLER

Bu tez çalışmasında ilk olarak insanlar tarafından seslendirilen “down, forward, follow, go, left, no, off, on, right, stop, up, yes” olmak üzere 12 farklı kısa komutu algılayan ve hangi komutun verildiğini tespit eden bir sistem derin öğrenme kullanılarak tasarlanmıştır.

Tasarlanan sistem için öncelikle literatürde çok tercih edilen ESA ve UKSB mimarileri incelenmiş ve her ikisi için de optimum modeller belirlendikten sonra çalışmaya uygun olan mimari seçilerek deneylere devam edilmiştir. ESA mimarisi oluşturulurken, sinir ağları için önemli bir parametre olan aktivasyon fonksiyonu seçimi için de deneyler yapılmıştır. ESA mimarisinde aktivasyon fonksiyonu olarak sigmoid kullanıldığında %82,13 test doğruluk oranı elde edilirken ReLU fonksiyonu kullanıldığında %92 test doğruluk oranı ve tanh kullanıldığında %93,1 doğruluk oranı elde edilmiştir. Bu sonuçlara göre ESA mimarisi kullanılırken tanh fonksiyonun seçilmesi doğruluğu olumlu yönde etkileyeceği söylenir. Ancak tanh kullanıldığında işlem yükü çok artmış ve çalışma süresi uzamıştır. Bundan dolayı ESA mimarisi için tanh ile yakın sonuçlar verdiğinden ve işlem yükünün daha az olmasından dolayı ReLU fonksiyonu kullanılmıştır. UKSB mimarisi oluşturulurken de benzer deneyler yapılmıştır ve sonrasında iki mimarinin tez çalışmasına uygunluğu karşılaştırılmıştır.

İngilizce orijinal veri seti kullanıldığında gerçek hayat için yapılan testler sonucunda ESA mimarisi ile %63,29, UKSB mimarisi ile %54,38 test doğruluğu elde edilmiştir. Ayrıca aynı test 600 adet ses içeren daraltılmış İngilizce veri seti ile yapıldığında ESA için %29,95, UKSB mimarisi için %20,73 test doğruluğu elde edilmiştir. Deneysel sonuçlara göre bu çalışma için ESA mimarisinin uygun olduğu anlaşılmıştır. Ayrıca daraltılmış İngilizce veri seti ile yapılan deney sonuçları, küçük boyuttaki veri setleri ile gerçekleştirilen derin öğrenme tabanlı uygulamaların başarısız sonuçlar verdiğini göstermektedir. Bu da derin öğrenme için veri miktarının oldukça önemli bir parametre olduğunun göstergesidir. Buradan yola çıkarak derin öğrenme tabanlı çalışmalarda az sayıda verinin olması halinde veri artırma teknikleri kullanılarak veri sayısının artırılması önerilmektedir.

Derin öğrenme kullanmanın en önemli şartlarından birisi çok sayıda veriye sahip olmaktır. Ancak her zaman istenildiği kadar çok veriye ulaşılamayabilir. Kısıtlı veriye sahip olunan durumlarda derin öğrenme kullanılmak isteniliyorsa çeşitli veri artırma teknikleri

kullanılarak veri sayısı artırılabilir ve bu sayede başarılı bir uygulama yapma şansı artmış olur. Bu doğrultuda öncelikle daraltılmış İngilizce veri seti ile yapılan deneyler sonucunda ortalama olarak %69,63 test doğruluğu ve %29,832 gerçek hayat doğruluğu elde edilirken veri seti boyutu iki katına çıkacak şekilde veri artırma uygulandıktan sonra bu oranlar sırasıyla %74,904 ve %30,813'e yükselmiştir. Artırma olmadan önce ve artırma sonrası sonuçlar kıyaslandığında veri artırmanın doğruluk oranlarına olumlu bir etkisi olduğu görülmüştür.

Türkçe veri seti bulmanın zor olduğu bu çalışma alanında kullanmak üzere Türkçe “aç, aşağı, dur, evet, git, hayır, ileri, kapat, sağ, sol, takip et, yukarı” komutlarını içeren küçük bir veri seti oluşturulmuştur. Türkçe olarak oluşturulan kısıtlı veri seti ile eğitim yapılmıştır ve test doğruluğu ortalama olarak %53,947 iken gerçek hayattaki doğruluk oranı %58,074 olduğu görülmüştür. Veri setine çeşitli deformasyonlar uygulanarak çok artırılmış Türkçe veri seti elde edilmiştir. Çok artırılmış Türkçe veri seti ile yapılan eğitim sonucunda test doğruluğu %85,756'ya ve gerçek hayattaki doğruluk oranı %64,761'e kadar çıkmıştır. Veri artırma sayesinde test doğruluğunda %31,809, gerçek hayat doğruluğunda %6,508 oranında bir artış sağlanmıştır. Bu durum derin öğrenme tabanlı çalışmalarda veri artırmanın çok önemli bir unsur olduğunu ispatlamaktadır.

Ayrıca İngilizce veri seti için yapılan veri artırma işleminden sonraki artış oranı Türkçe'ninkine göre daha az oranda olmuştur. Bunun sebebi İngilizce komutların fonemlerinin Türkçe komutların fonemlerine göre birbirine daha benzer olmasıdır. Yani İngilizce komutlarda söylenişi birbirine benzeyen seslerin daha fazla olması doğru tahmin edilme olasılığı düşürmektedir ve bu durum veri artırmanın etkisini azaltmaktadır. Buradan hareketle veri artırmanın her bir sesli komut sınıfına olan etkisini gözlemlemek amacıyla öğrenme sonrasında gerçek hayattaki doğruluk oranları sınıfsal olarak incelenmiştir. Deneyler sonucunda veri artırmanın veri setinin tamamı hesaba katıldığında olumlu bir etki yarattığı görülürken sınıflar ayrı ayrı incelendiğinde olumlu, olumsuz etkilediği ya da nötr bıraktığı görülmüştür. Tüm bunlar hesaba katıldığında her bir komuta özel iyileştirmeler yapılabilir ve doğruluk oranları sınıf bazlı da artırılabilir. Gelecekteki çalışmalar için veri artırma teknikleri rastgele olarak uygulanmamalı her bir yöntemin etkisinin sınıfa olan etkisi incelenerek en doğru artırma yöntemi seçilmelidir. Böylece veri artırma teknikleri daha verimli bir şekilde kullanılmış ve doğruluk oranları daha fazla olumlu yönde etkilenmiş olur.

Bu tez çalışması boyunca yapılan araştırmalarda ve deneysel çalışmalarda kısıtlı veriye sahip olunan durumlara bir çözüm olarak veri artırmanın etkisi gösterilmiştir. Literatür incelendiğinde Türkçe sesli komutlara ait veri kümelerinin çok az sayıda ve sınırlı düzeyde olduğu görülmüştür. Bu kısıtlılık sebebiyle tez çalışması esnasında “aç, aşağı, dur, evet, git, hayır, ileri, kapat, sağ, sol, takip et, yukarı” komutlarının seslendirildiği bir veri seti oluşturulmuştur. Oluşturulan veri seti seslere uygulanan manipülasyonlar sayesinde artırılmıştır ve derin öğrenme ile eğitim için daha uygun hale getirilmiştir. Bu sayede tez kapsamında sesli komut tanıma sisteminin Türkçe versiyonu da tasarlanmıştır. Oluşturulan veri setinin daha büyük ve kapsamlı olması sağlandığı ve veri setine artırma uygulandığı takdirde başarı oranının çok daha yüksek olacağı düşünülmektedir.

Tez çalışması boyunca mimariler, katman sayıları, aktivasyon fonksiyonları ve diğer birçok parametre kullanıcıya bağlı olarak farklı olarak seçilmektedir. Bu tez boyunca yapılan seçimler, literatürde önerilen ve tercih edilen ayrıca karşılaştırmalı deneyler yapılarak, hem işlem yükünün az olduğu hem de yüksek test doğruluğu veren seçimler olacak şekilde kullanılmıştır. Dolayısıyla farklı uygulama alanları için aynı değişkenleri kullanmak aynı başarıyı vermeyecektir. Bu çalışmada tezin amacına uygun olan ve mevcut veri setleri ile en başarılı sonuçları veren değişkenler seçilmeye çalışılmıştır. Ancak çok sayıda deney yapılmasına rağmen yapılabilecek çok daha fazla deney halen mevcuttur ve yapılan uygulama için gerçekten en iyi tercihleri yapmak tecrübe ile doğru orantılıdır. Yani çalışmaya birbirleriyle entegre edilerek kullanılan farklı mimariler, farklı veri setleri, görüntü için kullanılan yöntemlerin uygulandığı farklı veri artırma teknikleri gibi farklılıklar ile devam edilirse doğruluk oranlarının daha da artırılabilineceği düşünülmektedir. Gelecekteki çalışmalarda alternatiflerin seçilmesinde deneysel çalışmalara ek olarak tecrübenin de kullanılmasının başarıyı destekleyeceği düşünülmektedir.

KAYNAKLAR

1. Kiani, F., Kutlugün, M. A., ve Çakır, M. Y. (2017). Derin sinir ağları ile konuşma tespiti ve cinsiyet tahmini. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 8(3), 2208-2228.
2. Demir, A., Atila, O., and Şengür, A. (2019). *Deep learning and audio based emotion recognition*. International Artificial Intelligence and Data Processing Symposium, 1-6.
3. Başbuğ, A. M., ve Sert, M. (2019). *Analysis of deep neural network models for acoustic scene classification*. 27th Signal Processing and Communications Applications Conference, 1-4.
4. Solovyev, R. A., Vakhrushev, M., Radionov, A., Romanova, I. I., Amerikanov, A. A., Aliev, V., and Shvets, A. A. (2020). *Deep learning approaches for understanding simple speech commands*. IEEE 40th International Conference on Electronics and Nanotechnology, 688-693.
5. Kocer, H. E., ve Ahmed, M. C. Turkish speech recognition using mel-frequency cepstral coefficients and hidden markov model. *Veri Bilimi*, 2(2), 39-44.
6. Mantoro, T., Fadli, I., Ayu, M. A., and Abdillah, R. (2018). *Speech command for automatic sluice gate based on MFCC and deep neural networks*. Third International Conference on Informatics and Computing, 1-6.
7. Pleshkova-Bekiarska, S., and Bekiarski, A. (2019). *Building human mobile robot audio communication interface with artificial intelligence and deep learning*. International Conference on Creative Business for Smart and Sustainable Growth, 1-5.
8. Gemmeke, J. F., Ellis, D. P., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., and Ritter, M. (2017). *Audio set: An ontology and human-labeled dataset for audio events*. IEEE International Conference on Acoustics, Speech and Signal Processing, 776-780.
9. Warden, P. (2018). Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv* in press.
10. Salamon, J., and Bello, J. P. (2017). Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24(3), 279-283.
11. Lu, R., Duan, Z., and Zhang, C. (2017). *Metric learning based data augmentation for environmental sound classification*. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 1-5.
12. Madhu, A., and Kumaraswamy, S. (2019). *Data augmentation using generative adversarial network for environmental sound classification*. 27th European Signal Processing Conference, 1-5.
13. Davis, N., and Suresh, K. (2018). Environmental sound classification using deep convolutional neural networks and data augmentation. *IEEE Recent Advances in Intelligent Computational Systems*, 41-45.

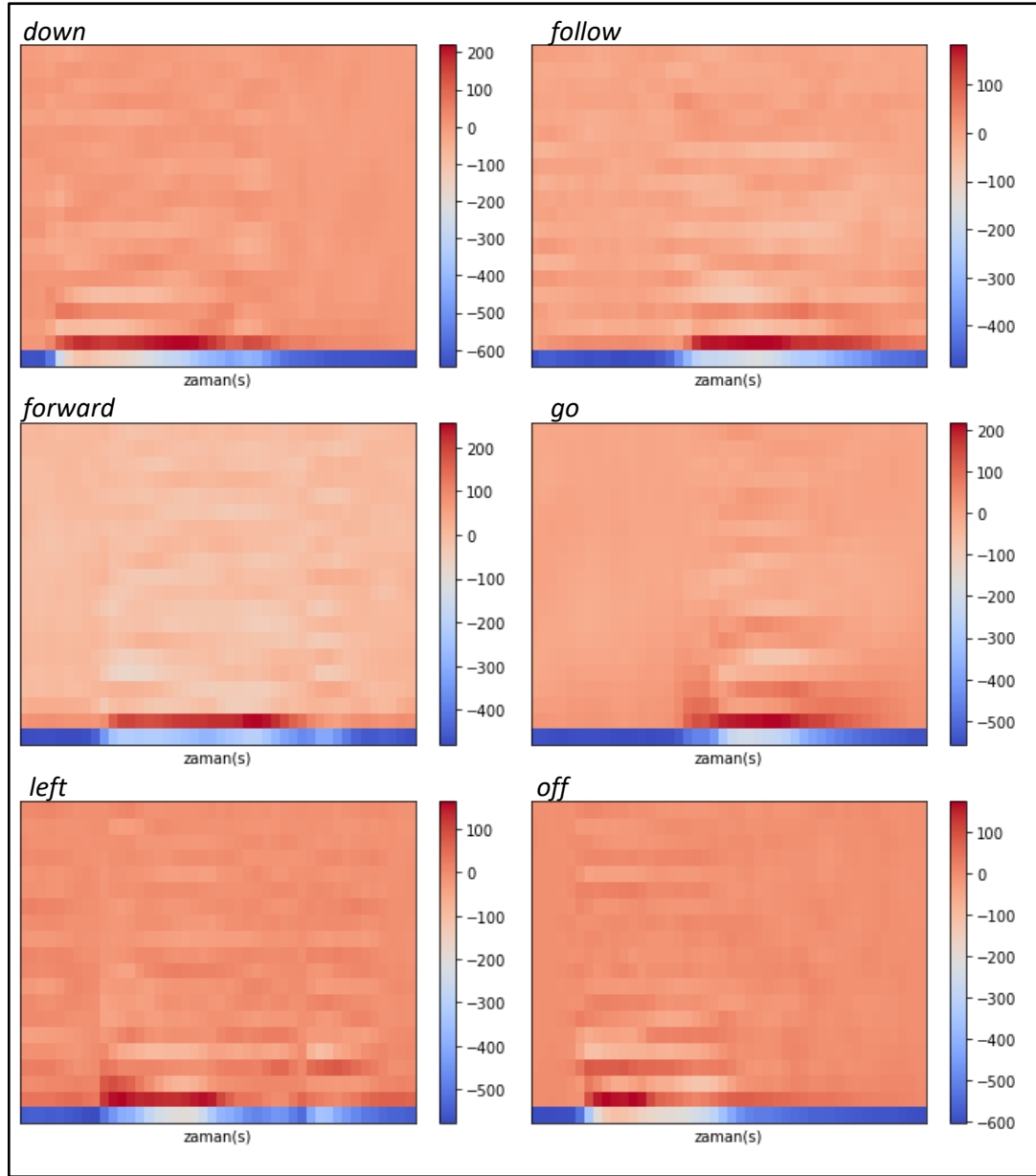
14. İnik, Ö., ve Ülker, E. (2017). *Derin öğrenmede kullanılan veri setleri ve yazılım kütüphaneleri*. 1st International Symposium on Multidisciplinary Studies and Innovative Technologies Proceedings Book, 72-77.
15. Lezhenin, I., Bogach, N., and Pyshkin, E. (2019). Urban sound classification using long short-term memory neural network. *Federated Conference on Computer Science and Information Systems*, 57-60.
16. Boddapati, V., Petef, A., Rasmusson, J., and Lundberg, L. (2017). Classifying environmental sounds using image recognition networks. *Procedia Computer Science*, 112, 2048-2056.
17. Zhang, H., McLoughlin, I., and Song, Y. (2015). *Robust sound event recognition using convolutional neural networks*. IEEE International Conference on Acoustics, Speech and Signal Processing, 559-563.
18. Piczak, K. J. (2015). *Environmental sound classification with convolutional neural networks*. IEEE 25th International Workshop on Machine Learning for Signal Processing, 1-6.
19. Er, M. B. Akciğer seslerinin derin öğrenme ile sınıflandırılması. *Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji*, 8(4), 830-844.
20. Ateş, E. (2019). *Derin öğrenme ile sesli komut tanıma*, Yüksek Lisans Tezi, Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü, Eskişehir, 1-45
21. Öztemel, E. (Editörler). (2006). *Yapay sinir ağları*, İstanbul: Papatya Yayıncılık, 45-53.
22. Fernández, C., Soria, E., Martin, J. D., and Serrano, A. J. (2006). Neural networks for animal science applications: Two case studies. *Expert Systems with Applications*, 31(2), 444-450.
23. İnternet: Jay, P. (April, 2017). Back-propagation is very simple. Web: <https://medium.com/@14prakash/back-propagation-is-very-simple-who-made-it-complicated-97b794c97e5c> adresinden 7 Ağustos 2021'de alınmıştır.
24. İnternet: Kızrak, A. Yapay sinir ağları. Web: <https://ayyucekizrak.medium.com/%C5%9Fu-kara-kutuyu-a%C3%A7alim-yapay-sinir-a%C4%9Flar%C4%B1-7b65c6a5264a> adresinden 1 Ağustos 2021'de alınmıştır.
25. Deng, L., and Yu, D. (2014). Deep learning: Methods and applications. *Foundations and Trends in Signal Processing*, 7(3-4), 197-387.
26. Küçük, D., ve Arıcı, N. (2018). Doğal dil işlemede derin öğrenme uygulamaları üzerine bir literatür çalışması. *Uluslararası Yönetim Bilişim Sistemleri ve Bilgisayar Bilimleri Dergisi*, 2(2), 76-86.
27. Ravi, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, B., and Yang, G. Z. (2016). Deep learning for health informatics. *IEEE Journal of Biomedical and Health Informatics*, 21(1), 4-21.

28. İnternet: Code, R. (February, 2018). Konvolüsyonel sinir ağlarına giriş. Web: <https://rubikscodene.net/2018/02/26/introduction-to-convolutional-neural-networks/> adresinden 7 Nisan 2021'de alınmıştır.
29. İnternet: Pythondunyasi. (Aralık, 2020). Derin öğrenme aktivasyon fonksiyonları. Web: <https://pythondunyasi.com/derin-ogrenme-aktivasyon-fonksiyonlari/> adresinden 7 Nisan 2021'de alınmıştır.
30. İnternet: Amidi, A., and Amidi, S. Recurrent neural networks cheatsheet. Web: <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks> adresinden 1 Ağustos 2021'de alınmıştır.
31. Ser, G., ve Bati, C. T. (2019). Derin sinir ağları ile en iyi modelin belirlenmesi: Mantar verileri üzerine keras uygulaması. *Yüzüncü Yıl Üniversitesi Tarım Bilimleri Dergisi*, 29(3), 406-417.
32. Şeker, A., Diri, B., ve Balık, H. H. (2017). Derin öğrenme yöntemleri ve uygulamaları hakkında bir inceleme. *Gazi Mühendislik Bilimleri Dergisi*, 3(3), 47-64.
33. Krizhevsky, A., and Hinton, G. E. (2011). *Using very deep autoencoders for content-based image retrieval*. 29th European Symposium on Artificial Neural Networks, 1(2).
34. Şeker, A., ve Yüksek, A. G. (2017). Stacked autoencoder method for fabric defect detection. *Cumhuriyet Üniversitesi Fen-Edebiyat Fakültesi Fen Bilimleri Dergisi*, 38(2), 342-354.
35. İnternet: Baydur, B. (Nisan, 2019). Python ideleri ve Anaconda kurulumu. Web: <https://aridoshika.com/python-ideleri-ve-anaconda-kurulumu/> adresinden 17 Nisan 2021'de alınmıştır.
36. İnternet: Yazılım geliştirme programı sayfasıdır. URL: <https://www.spyder-ide.org/> Son Erişim Tarihi: 15.07.2021.
37. İnternet: Kaplan, K. Python ve Anaconda Kurulumu. Web: <https://medium.com/kodcular/python-ve-anaconda-kurulumu-b8931bd80e64> adresinden 17 Nisan 2021'de alınmıştır.
38. İnternet: Kızıldaş, E. TensorFlow nedir? Nasıl kullanılır?. Web: <https://devnot.com/2019/tensorflow-nedir-nasil-kullanilir/> adresinden 17 Nisan 2021'de alınmıştır.
39. İnternet: Yazılım kütüphanesi sayfasıdır. URL: <https://keras.io/> Son Erişim Tarihi 18.04.2021.
40. İnternet: Pandey, P. Python ile müzik türü sınıflandırması. Web: <https://towardsdatascience.com/music-genre-classification-with-python-c714d032f0d8> adresinden 19 Nisan 2021'de alınmıştır.
41. Salamon, J., and Bello, J. P. (2015). *Feature learning with deep scattering for urban sound analysis*. 23rd European Signal Processing Conference, 724-728.

42. Eskidere, Ö., ve Ertaş, D. (2009). Mel frekansı kepsrum katsayılarındaki değişimlerin konuşmacı tanımayaya etkisi. *Uludağ University Journal of The Faculty of Engineering*, 14(2), 93-110.
43. İnternet: Silva, R. Dinamik aralık, sıkıştırma ve headroom ses performansını nasıl etkiler?. Web: <https://tr.eyewated.com/dinamik-aralik-sikistirma-ve-headroom-ses-performansini-nasil-etkiler/> adresinden 11 Nisan 2021’de alınmıştır.
44. İnternet: Creative commons international attribution international 4.0 license. URL: <https://creativecommons.org/licenses/by/4.0/> Son Erişim Tarihi: 09.06.2021.
45. İnternet: Speech commands dataset version 2. (2018). URL: http://download.tensorflow.org/data/speech_commands_v0.02.tar.gz Son Erişim Tarihi: 13.07.2021
46. Çayır, A. N., and Navruz, T. S. (2021, June 11-13). *Effect of dataset size on deep learning in voice recognition*. Presented at 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications, 1-5.

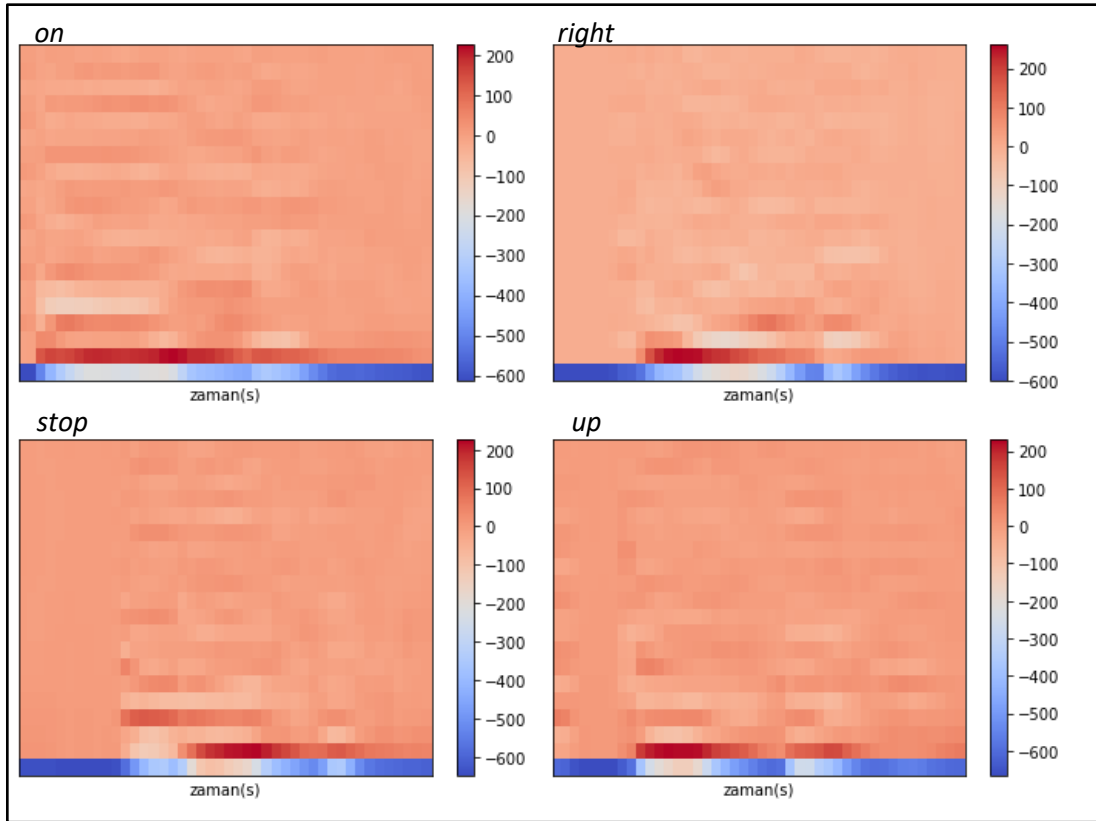
EKLER

EK-1. İngilizce komutların MFKK grafikleri

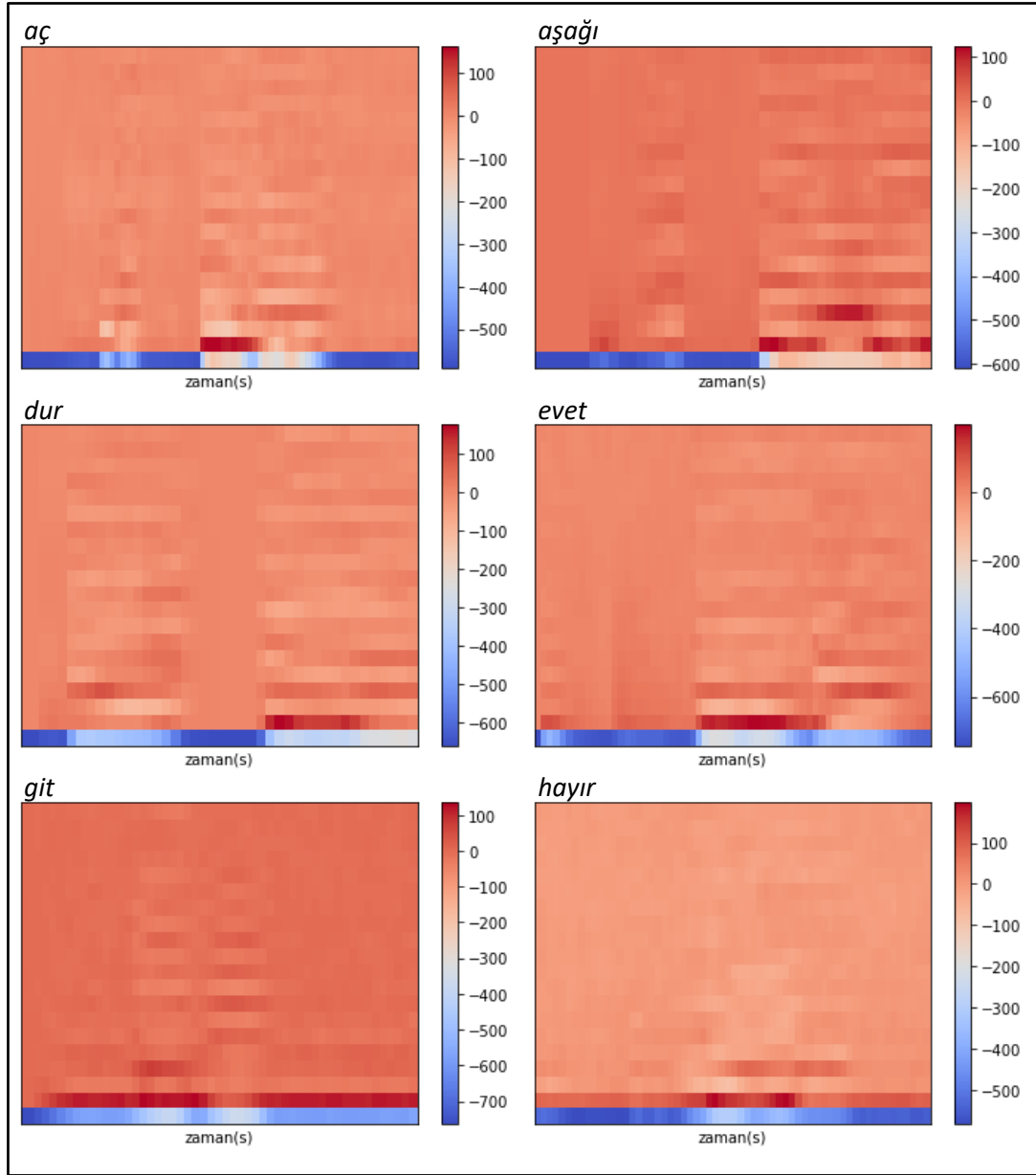


Şekil 1.1. İngilizce komutların MFKK grafikleri

EK-1. (devam) İngilizce komutların MFKK grafikleri

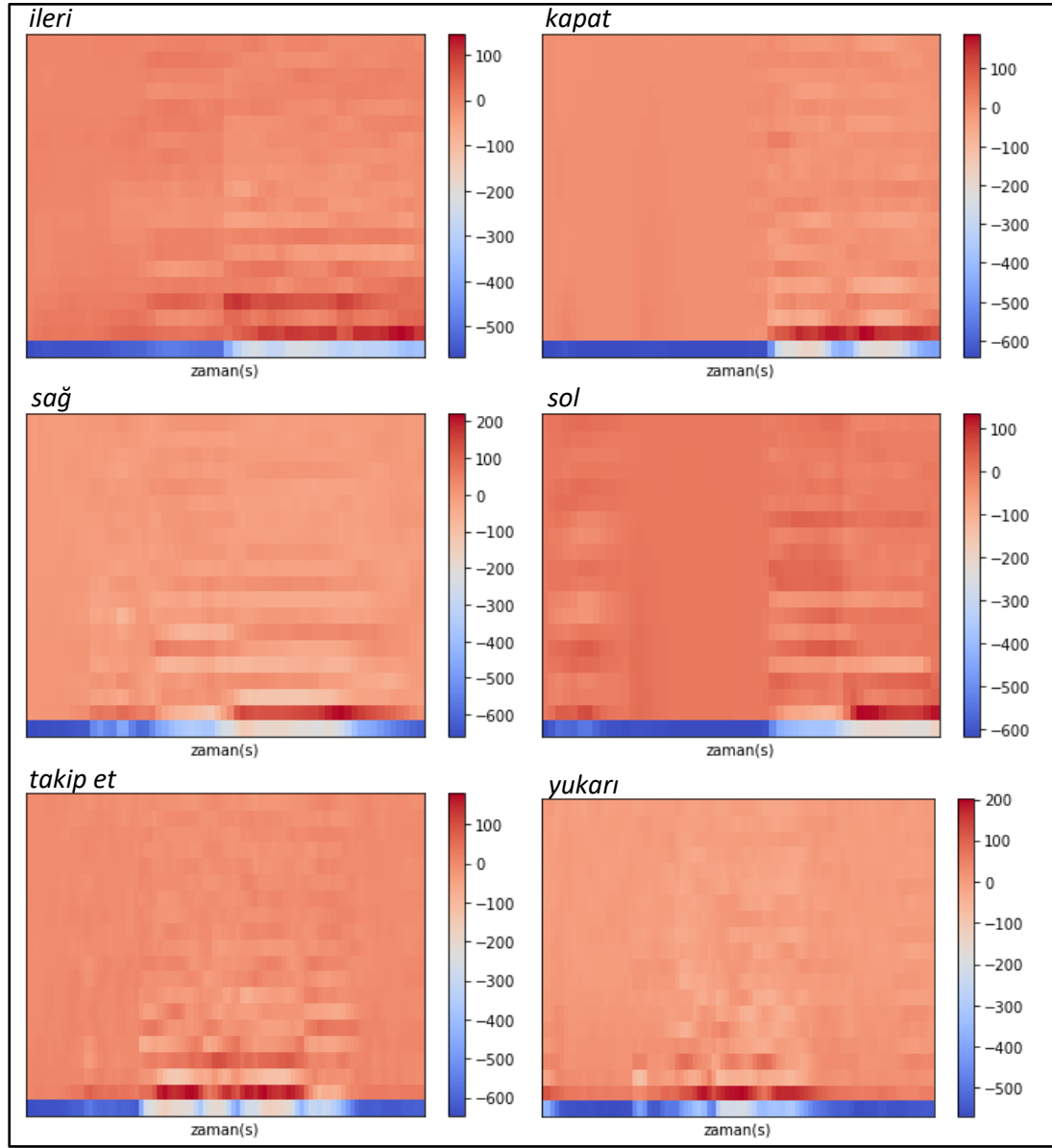


EK-2. Türkçe komutların MFKK grafikleri



Şekil 2.1. Türkçe komutların MFKK grafikleri

EK-2. (devam) Türkçe komutların MFKK grafikleri





GAZİ GELECEKTİR...